WEB-BASED ROUTE OPTIMIZATION SYSTEM FOR LOGISTIC USING AGGLOMERATIVE CLUSTERING AND GENETIC ALGORITHM

TAN CHEE KUAN

UNIVERSITI TUNKU ABDUL RAHMAN

WEB-BASED ROUTE OPTIMIZATION SYSTEM FOR LOGISTIC USING AGGLOMERATIVE CLUSTERING AND GENETIC ALGORITHM

TAN CHEE KUAN

A project report submitted in partial fulfilment of the requirements for the award of Bachelor of Science (Hons.) Software Engineering

Lee Kong Chian Faculty of Engineering and Science
Universiti Tunku Abdul Rahman

April 2020

DECLARATION

I hereby declare that this project report is based on my original work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at UTAR or other institutions.

Signature : 

Name : Tan Chee Kuan

ID No. : 1602812

Date : 20 April 2020

APPROVAL FOR SUBMISSION

I certify that this project report entitled **"WEB-BASED ROUTE OPTIMIZATION SYSTEM FOR LOGISTIC USING AGGLOMERATIVE CLUSTERING AND GENETIC ALGORITHM"** was prepared by **TAN CHEE KUAN** has met the required standard for submission in partial fulfilment of the requirements for the award of Bachelor of Science (Hons) Software Engineering at Universiti Tunku Abdul Rahman.

Approved by,

Signature : _____

Supervisor : _____

Date : _____

ACKNOWLEDGEMENTS

I would like to thank everyone who had contributed to the successful completion of this project. I would like to express my gratitude to my research supervisor, Dr. Wong Whee Yen for her invaluable advice, guidance and her enormous patience throughout the development of the research.

ABSTRACT

Route planning is always a difficult task that challenges all the logistics companies. The difficulties of planning a route is because people may not familiar with those addresses, therefore without any help from other existing applications like Google Map or Waze people are not able to find the most suitable route for them to deliver, especially those companies whose have many trucks involved in their daily operation. Besides, there may be a waste of time if the user keeps using Google Maps to check on the location when planning the route. How if there are hundreds of addresses needed to deliver on that particular day, and it is not wise to plan the route manually. Thus, a web-based route optimization is proposed with the objectives of developing a system that is able to find the shortest route to deliver.

In this project, evolutionary prototype methodology was selected to smooth the software development according to the scopes and also the requirements. There are three iterations in this project. For the first iteration, use case diagrams and also the preliminary prototype had been developed. In the second iteration, activity diagram, ERD diagram and data flow diagram had been developed. The second iteration continues by the development of the backend logic that communicates with several APIs which are firebase authentication, firestore and google cloud service. The third iteration focuses on finding those missing requirements and enhance the system. To achieve the project's objectives, several researches on the existing algorithm and also existing similar systems had been done to collect relevant information.

In conclusion, this system had been developed with a total period of six months and all of the objectives had been fulfilled. This system enables the logistic company to arrange its route to deliver based on the number of trucks selected by the user. This system will first cluster those addresses based on the number of trucks involved and then perform the route optimization algorithm to suggest the shortest route to the user. Although all of the objectives had been achieved the current system restricted user to make changes on the calculated route directly, the clustering algorithm and the route optimization algorithm applied are not wise enough and several enhancements can be made for this system. To solve the system's current limitations, several improvements like enable user to set priority on the D/O, enable user to add/remove a particular D/O on the planned route, review on others clustering and optimization algorithm can be performed in the future.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS / ABBREVIATIONS

D/O	Delivery order
SMART D/O	Transport schedule that consist of the order of customer to be sent
API	Application Programming Interface
TSP	Travelling Salesman Problem
GA	Genetic Algorithm
CVRP	Capacitated Vehicle Routing Problem
VFJ	Variant Fisher and Jaikuamr
CLP	Container Loading Problem
FILO	First In Last Out
SDLC	Software Development Life Cycle

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Introduction

This chapter provides an overview of this project. In this chapter, the background of the project, problem statements, objectives, the proposed solution, proposed approach and also the project scope was discussed and implemented in the final product.

Like other industries, the logistic company is dealing with several change as time goes by. For example, new technology, new customer expectations, new business model and also new market entrants (Tipping and Kauschke, 2016). These may bring opportunities if the company is able to adapt to those changes. One of the trends which can be obviously stated here is customer expectation. As a customer (both individuals and other business), they may seek for transportation which has a low delivery cost, fast and flexible (Tipping. and Kauschke, 2016). Therefore it is a hard time for the logistic company to adapt to this changing environment by providing a competitive price for delivery. To overcome this situation, the logistic company is preferred to maximize the use of technology. This project developed a system which is able to make improvement on the arrangement of transportation and hope that this will bring certain benefits to those logistic company.

1.2 Project Background

To be honest, Malaysia is still in a state of development. Therefore, logistics will become one of the factors that pushing the economics of Malaysia (Abdul Aziz, et al., 2012). As time goes by, the impact of logistic may become greater as it also plays the role of attracting investor from other countries (Abdul Aziz et al., 2012). This may due to the continual development of logistic infrastructure in Malaysia. In the year 2017, transport minister Datuk Seri Liow Tiong Lai said that Malaysia had invested nearly RM 15 billion in four sectors of logistic infrastructure which are rail, road, maritime and aviation (Farah, 2017). For example, expansion of the airport, development of the port at Carey Island, Melaka, Kota Kinabalu and also Kuantan, an increase of railway operation and more which are able to smooth the process of the logistics' business (Farah, 2017). As consequence investors are more likely to invest in those countries

which has the ability to retain their business. Besides, due to the improvement of transportation facilities in the country, customers' and retailers' expectations may be getting higher toward those Logistics Companies. Customers will try to seek for a company who offer lower delivery cost, fast and flexible (Tipping, and Kauschke, 2016). Therefore, it is hard for a manager to satisfy those customers and at the same time maintain and monitor the logistic and operating costs of the company. For Logistics Company, operating cost not only including those monthly expenses and utilities; travelling cost, fuel cost and labor cost may also be needed to be considered. Hence, the purpose of this project is to help the logistic company to maximize the use of the company's resource like labor, time, money and so on.

There may be certain issues that always arise within those logistic company. The first issue, which also the most common one is the planning of route for delivery. Based on the experience of manager from Hondail Trading Sdn Bhd, he stated that the well-planned route is the one which used the shortest distance, minimum cost and minimum time to deliver all those inventories to their customers. But somehow there may have certain areas or particular location that the manager may not know. Therefore, manager may need to spend some time searching those particular destinations on the map and re-arrange the route for delivery. This may waste time and not efficient. For example, a new customer who wishes the company to deliver his goods for him, but the manager may not know the exact location of this customer based on the given address. Hence, it is hard for manager to plan the route. Besides, there may be other alternative routes that used minimum time and shorter distance compares to the one that was planned by the manager. This project aims to implement a system that is able to reduce the total travel distance compare with the route that had planned by the manager. As a result, company's resources are able to utilize well and maintainable.

Another issue will be how to optimize/maximize the space of the trucks. It is hard to measure the length, width, and height of the inventories by using naked eyes. So, it is hard for the workers to reserve space for that particular item. Normally workers will only fill up the truck based on the order of the route to deliver. For example, inventories for the last customer will be placed in the deep corner of the truck, the biggest and heaviest one will at the bottom and the smallest one will be placed at the

top. Logically this is the right thing to do but there can still have some improvement in terms of optimizing the space in the truck. I believe that there may have certain algorithms or sequences that can be adapted in order to improve the volume capacity within the truck. Hence, this project aims to research and implement certain rules that can actually help to utilize the space in the truck by generating the best order of placement for inventories based on the sequence of the inventories drop-off. There may also have the possibility that too many inventories are unable to fit into the truck. As such, the system shall let the manager know the "best-fit" of inventories placement into the truck before workers start to load inventories into the truck.

1.3 Problem Statement

1. Fail to fully utilize company's resources

As fuel cost and travelling cost may be one of the challenges that face by logistic company. To reduce the cost the idle way to do is finding the shortest route to deliver from one location to another location then return back to the starting location. This is also known as travelling salesman problem or vehicle routing problem. The challenge of travelling salesman problem is there may have many combination of routes that made up of different orders of locations and we do not know which route is the most suitable one. While the difference between vehicle routing problem and travelling salesman problem is vehicle routing problem tend to find the optimal set of routes for vehicles to travel in order to deliver toward the customers. Therefore, an optimize search algorithm is needed to calculate and suggest the shortest route to the user within a specific range time.

In addition, the direction to travel must also be considered during the planning of the route to delivery. For example, one of the trucks may responsible to deliver inventories to their customers which located in the east of the city while other trucks may be responsible for a different direction. However, this is hard to achieve due to manager may not know the exact distance from one location to another location and also the direction of the particular location without the help of any existing system. Manager may spend some of the time to find out where is the location and may need to re-arrange the route for delivery. This is not effective and may lead to delays in the inventories. All of the arrangement is based on the heuristic of the manager without the help of any system. Maule (2010) stated that human limitations like limited

processing power and low memory capacity make them impossible to make a rational decision. These limitations make humans more likely to select an alternative which is acceptable instead of the best one.

To solve this problem, this project had implemented a system that is able to suggest the shortest routes to deliver based on a particular direction and with the help of Google Map API. Google Map API may be used to find out the exact distance between two locations. Then calculation may be performed to find out the shortest route to deliver.

2. Cargoes are unable to fit into the truck due to the order of placement and orientation of the inventories.

In order to not waste any available space in the truck and minimize the overall trip to deliver, manager and workers may try to place the inventories as tight as possible in order to utilize all the available space. As a result, the chances for all inventories to fit into a particular truck will be increased and at the same time the overall number of trip can also be minimized. However, humans are not able to plan like a computer as humans may not know the actual size of all inventories and also humans are not able to perform complex calculations and visualization in their mind. Therefore, it is impossible to know there may have how many different order and orientation of the placement of inventories that can exist and which of them is the best.

In reality, there may be chances that one or two invoices are not able to fit into the truck and it is going to meet the deadline if those inventories are not sent to the customers by that day. There may have several solutions that can be performed to address this problem but most of them are not that effective. For example, partially unload the inventories and re-arrange the orientation of the inventories. Ask others to deliver those inventories to the customers, as a deal company may need to pay them some fees in order for them to cover the fuel and travelling cost.

In response to this problem, this project aims to utilize the capacity of a particular truck by developing an algorithm which performs calculations on suggested route, quantity, volume, weight of the inventories. Generally, loading and unloading of cargoes is the daily affair which is inevitable for a logistic company as drivers need

to load and sent cargoes from place to place therefore there is a necessity of container loading/bin packing algorithm to be developed in order to optimize the available space of truck capacity. Optimizing the solution of BEST-FIT truck can therefore not be overlooked. This study will expose the application of container loading algorithms in solving the container loading problem.

1.4 Project Objectives

1. To review literature on the algorithm for travelling salesman problem, vehicle routing problem and container loading problem.

2. To use an algorithm which is able to optimize the route to deliver compare with the previous route that manually planned by manager.

3. To develop a system that is able to show the calculated route on the SMART D/O based on defined rules above.

1.5 Proposed Solution

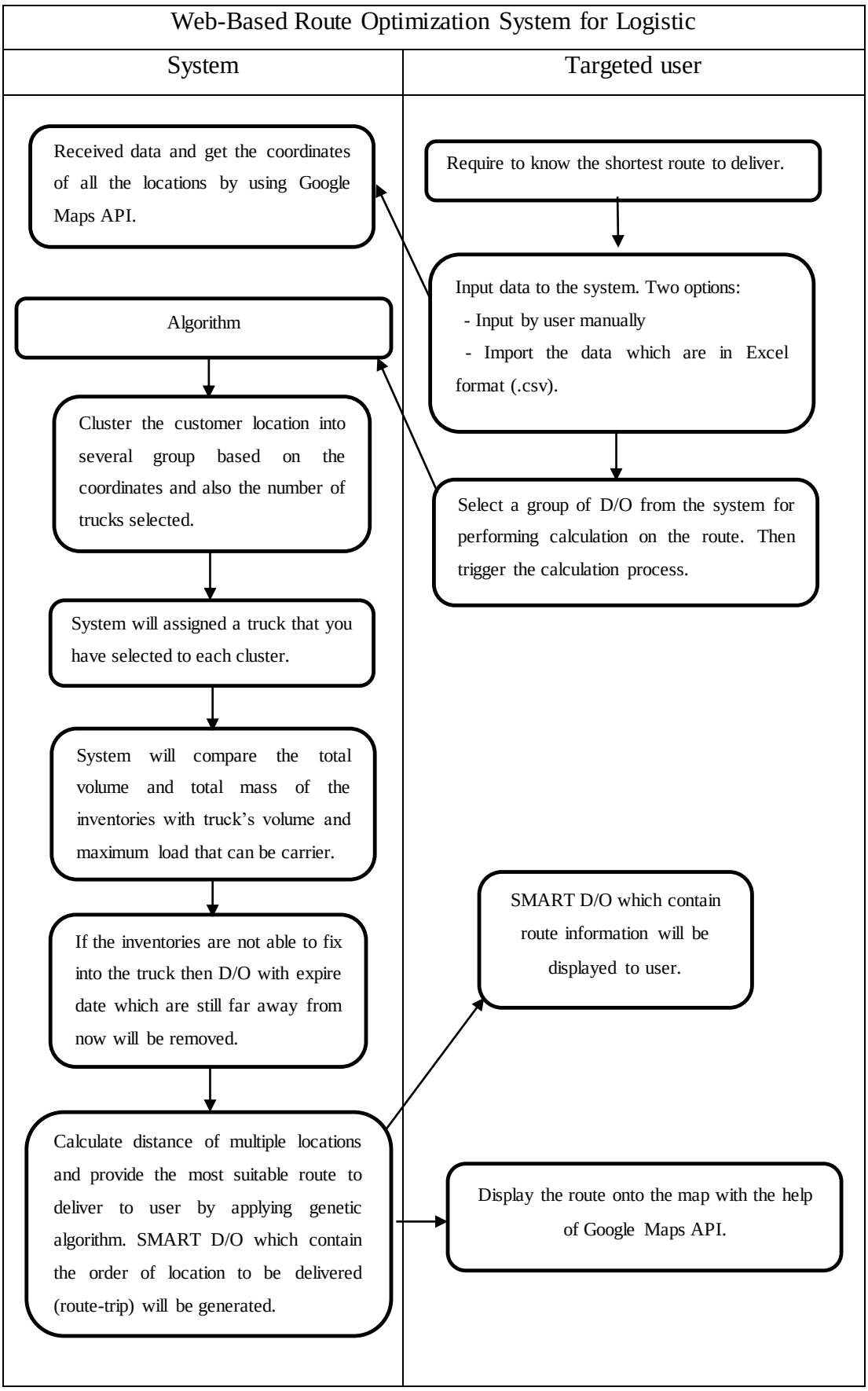


Figure 1.1: System High-Level Architecture (Project Compilation)

To solve the problem faced by most of the logistic company, a web-based route optimization system is developed. The target user of this system will be logistic company's manager and their worker who responsible for planning the route-trip. Users can easily access this system by entering the specific url.

Users may need to register an account before access to the system. The system should prompt the user to enter their data. The data must consist of customer's address, height of the inventory, length of the inventory, width of the inventory, weight of the inventory, number of inventories in the particular D/O, type of inventories in the particular D/O, unique id of the D/O, and deadline of the D/O. There may have two options for user to input their data. One of the options requires users to key in data manually, while another option may only require the user to import file which is in Excel format.

After user import those data into the system, users may be asked to select the number of truck and also select the type of trucks that their company own. User is required to press the "Start Calculation" button to trigger the system to auto calculate the shortest route to deliver. A SMART D/O which contains the transport schedule of customer location will be generated. From the SMART D/O users may know the order of customers to deliver. User is able to save the SMART D/O. After getting all the required data the system must display the route on the map for visualization purposes.

1.6 Proposed Approach

1.6.1 Research Approach

In this project, the research methodology that is going to adapt is qualitative methodology. An interview was conducted (one of the tools under the qualitative methodology) with the manager of Hondail Trading Sdn Bhd. The reason qualitative methodology is going to adapt in this project is because of unfamiliar on the process of route planning and cargoes loading. Langkos (2014) mentioned that qualitative methodology is mostly carried out by those researchers who only know roughly in advance for what he is looking for, therefore this methodology is able to let him gain more understanding by getting a detailed description. The features for this system which are going to develop are mainly supported by an optimize search algorithm. Thus, understanding the process of how the company plans their route and load their cargoes into the truck will bring better reliability on the system when implementing the algorithm. In this part, quantitative approach is not enough to understand the detail of the road planning and cargoes loading process. Besides, Langkos (2014) also point out that quantitative approach may miss some detail when asking for contextual data. Which mean that quantitative approach is not suitable if the aim is to understand the detail of a particular topic just like this project which asks for the detail of how they actually planned the route to deliver and how they load the cargoes into the truck.

Besides, perform analysis on the root cause which will affect the route planning and cargoes loading process is important if a quality algorithm is a plan to be developed. Factors which are needed to consider during the implementation process for the system's algorithm can be determined at the early stage by understanding those root cause and also the workflow. Qualitative approach is able to perform well as interviewer are able to ask for the explanation when they have certain doubt and unclear from the feedbacks that provided from the interviewee. In the process of clarifying, interviewer is able to sense those important factors then implement those factors into the algorithm for example the maximum load weight of the truck, the volume of the truck and the orientation of the cargoes in the truck. Quantitative approach may not perform well due to the data collected is quite general and not specific to a particular issue and it is hard to ask user for extra explanation hence interviewer may miss certain important information. In conclusion, the information which is going to collect is basically how they operate they daily business and this is

hard to convert into a statistical model. Therefore, the only research approach which is suitable for me is qualitative methodology.

1.6.2 Development Approach

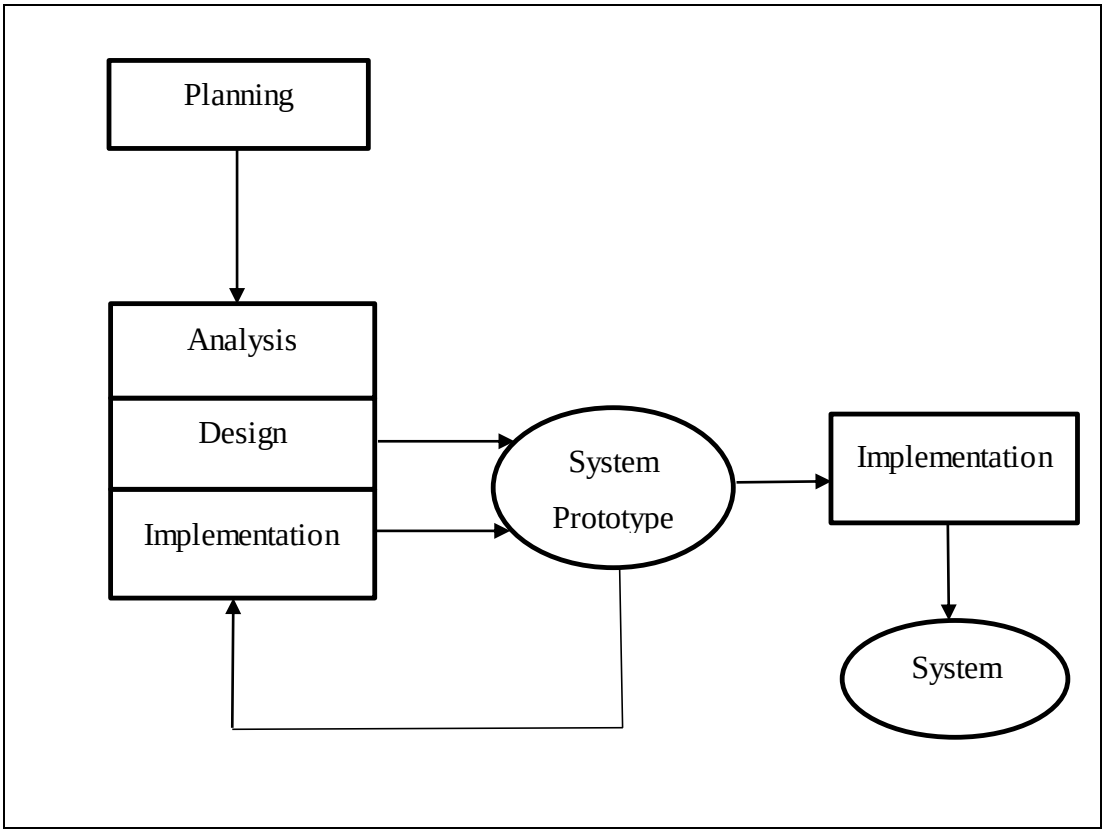


Figure 1.2 Evolutionary Prototyping Methodology (Mak, et al., 2017)

The development methodology used in this project is Evolutionary Prototyping methodology due to the characteristics of prototyping model is more suitable for this project. As developer may not get the full detail of requirements at the beginning of this project due to inexperience in the process of route planning and cargoes loading. By adapting evolutionary prototyping in this project developer are allows and prefers to develop a prototype to clarify for the requirements and the overall flow process of the route planning and cargoes loading process from the user feedback. The difference between prototyping and other traditional methodology is prototyping allows several iterations while traditional methodology may not have this phase (Carter, et al., 2001). By using this project as an example, there may be certain requirements which will never be found out until some research on other's literature review, existing application had been done or when implementation of the system is in the halfway.

For example, after reviewing on the existing application, some cargoes loading applications actually provide option in setting like respect the maximum load weight of the truck which I never consider until I review those existing systems. Some companies may try to ignore the maximum load weight due to they may want to increase the profit per truck. As mentioned, evolutionary prototyping allows several iterations, therefore suggestion and opinion can get from those targeted users whether the setting mentioned above needed to implement in the system to increase user flexibility in respecting the maximum load weight of the particular truck.

Besides, the system that is going to develop in this project is mainly supported by an algorithm, in order for the system to work perfectly without any error, all factors that will affect the result of this system must be clarified. Unclear requirements and misunderstanding of requirements are mostly caused by insufficient knowledge and experience in that particular business domain. Therefore, prototyping methodology allows developer to clarify all the requirements by undergoing several iterations when getting feedback from the user on a different version of prototype (Carter, et al., 2001). New version of the prototype indicates that the vulnerabilities or unclear requirements had been removed hence developer are able to develop better quality and better user-interface system. As long as all the requirements have been clarified and user is satisfied with it, developer can proceed to the next step which are implementation of the real system.

In conclusion, evolutionary prototyping methodology is used in this project and a partially workable system is developed to ask for user satisfaction on the requirements. This partially workable prototype will continue to finalize until the final product had been developed.

1.7 Scope of the Project

1.7.1 Target Users

The target users of this project will be logistic company's manager/staff who responsible for planning route to be delivered to all their customers in a particular day and workers who responded to load the inventories into a particular truck. By using this system, the shortest route to deliver will be generated and manager may not need to arrange the route manually. In the future, this system may also able to suggest the best placement order of inventories to the workers. Therefore, all inventories are able to fix into the truck and workers may not need to re-arrange inventories again and again.

1.7.2 Features Covered

The features stated below must be provided by this system in order for this project to fulfill its objectives.

i) Provide user interface for user to import their data.

This system should provide user interface to let the user to input their data manually. Besides, this system must also provide data import options in its user interface. For example, pull in data which is in Excel format(.csv) for computational calculation and analysis purpose. This feature is aimed to increase the flexibility of the system. User has the choice to input their data which is in excel format as their data may be very huge and it is an inconvenience to let user key in hundreds or thousands of data manually. The data that will be handled by this system is the address of the customer, distance from one location to another location, coordinate of the location, D/O (which contain deadline, number of inventories, and type of inventories), packaging requirements of inventory and capacity of a particular truck. This feature is important due to it is able to reduce the workload of the staff due to required data can straight import into the system instead of the key in one by one. Besides, this feature is able to help the user to reduce their error, there is a high possibility to make a mistake when the user key in manually. Last, by having this feature users are able to save their time on the one by one key in process.

ii) Calculate the shortest route to deliver.

This system must have an algorithm that is able to capture the address given by the user and find out the distance between multiple locations with the help of Google Map API. After all distance between each location had been known, system may need to have an algorithm to handle those distances and calculate the shortest route to deliver. A SMART D/O which contains the order of the address to be delivered will be generated. The driver may only need to follow the route which planned by the system to complete their task.

iii) Calculate the volume of the spaces which occupied by those inventories.

The algorithm implement must be able to calculate the total volume of the truck that occupied by inventories and also the total duration needed to travel from starting location to the final destination. After finish all the calculation process, the total time to travel and also the total space occupied by those inventories will be display on the SMART D/O.

iv) Manage D/O information.

The system must allow user to create a new D/O record by filling in the required data. Besides, the system must be able to retrieve those D/Os' detail from the database and display them on the screen. Filters must be implemented on those data fields therefore user is able to narrow down the data after user enter a specific keyword. In this page user is enabled to edit the data of those D/O by clicking on the "pencil" icon, view the detail information of a D/O by clicking on the "eye" icon and also delete a particular D/O by clicking on the "trash" icon which are located at the action column of the table.

v) Manage Customer information.

The system must allow user to create a new customer record by filling in the required data. Besides, the system must able to retrieve customers' detail from the firestore and display them on the screen. Filters must be implemented on those data fields therefore user is able to narrow down the data after user enter a specific keyword. In this page user is enabled to edit the data of those customer by clicking on the "pencil" icon and delete a particular customer by clicking on the "trash" icon which are located at the action column of the table.

iv) Manage Item information.

The system must allow user to create a new item record by filling in the required data. Besides, the system must able to retrieve items' detail from the firestore and display them on the screen. Filters must be implemented on those data fields therefore user is able to narrow down the data after user enter a specific keyword. In this page user is enabled to edit the data of those item by clicking on the "pencil" icon and delete a particular item by clicking on the "trash" icon which are located at the action column of the table.

v) Manage Cargo Space Information

The system must be able to retrieve cargo space's information from the firestore and display a list of cargo space detail on the screen. Besides, filters will also be implemented on those data fields therefore user can narrow down the after user enter a specific keyword. From this page, user can select a particular cargo space he needed by clicking on the checkbox located at the "Status" column. The selected cargo space will automatically display on the "Vehicle Selection" section which located under the "Settings" tab. Other than that, the system must allow user to edit a particular cargo space record by clicking on the "pencil icon" and delete a particular cargo space by clicking on the "trash" icon.

vi) Route Record page which used to display the information of the previous calculated route.

The system must be able to retrieve previous route information from the firestore and display those data retrieved on the screen. In this page, user is allowed to view the detailed information of those routes by clicking on the "eye" icon and delete a particular route record by clicking on the "trash" icon which is located at the action column of the table.

1.7.3 Out of Scope (Future implementation)

i) Machine Learning on daily traffic

Apply machine learning to keep track of daily traffic therefore system is able to suggest the best route to deliver at a specific day and time based on the data collected. For example, there may be traffic congestion in a particular area at a certain time due to Chinese New Year, therefore the system will consider this as one of the rules to define the best route.

ii) Optimizing available space in the truck

This system must be able to capture those packaging requirements like length, width, height, weight of the inventories, placement pattern of an inventory and order of inventories to be sent as an input to the container loading algorithm. The system may develop a set of rules and algorithms to generate the most suitable order of placement of inventories for user based on the data collected.

iii) Visualization of the placement of inventories in truck

This system must be able to visualize the placement of the inventories in a particular truck which is in 2D/3D format based on the output generated from container loading algorithm. The aim of this feature is to make the loading process more smoothly. Workers may only need to follow the diagram when arranging the inventories in the truck.

1.7.4 Assumption of Scope

i) Truck restricted streets

There may have several streets which do not allow truck to be travel this may due to the fact that not all of the road have the same foundation of design. Some of the roads were not designed to support the weight of the truck and it may cause damage to that particular road. It is hard to know which street is not designed to support the truck and it may be difficult to calculate the route to deliver if relevant data is not provided. Therefore, this project assumes that all streets are able to travel by truck.

ii) Product packaging layer

In reality, there may have three layers of packaging for most of the items. They are outer packaging, inner packaging, and product packaging. Outer packaging will be the first thing that a customer is going to see, also known as a carton. Inner packaging will be a box that contains of several product packaging based on the stated quantity. Product packaging is what people will be seen when walking around the shopping center. For example, the box the toys come in, wrapper of candy bar and more. Most of the logistics will provide delivery services up to layer two of packaging while delivery for items with product packaging will not be provided, unless the company provides special service like re-packing of different items which is in product packaging layer into an appropriate size of box. In order to clarify the requirement when performing literature review on algorithm, an assumption regarding the packaging layer had been made. The data collected for the packaging requirement of inventory will only up to the second layer, which means that only the outer packaging layer and the inner packaging layer will be considered when developing the abovementioned algorithm.

1.8 Work Breakdown Structure of System

Refer to APPENDIX A: Work Breakdown Structure for System

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In this section, several fields that are relevant and helpful to this web-based route optimization system were evaluated and synthesized. The fields that had been reviewed are biased towards the problems that this project will face during the software development life cycle like what features and functionalities are more appropriate for this system, which algorithm can help this route optimization system to find the optimal route and maximize the truck capacity, which methodology is more suitable for this project, which research method can be used to collect relevant data and what tools will be used to develop this system. Therefore research and analysis had been done on those areas which are existing online related systems, existing research problems and its algorithm, software development methodology, research methodology and development tools.

2.2 Existing Online Related System

In this section, an online route planner systems and two online cargo loading systems had been studied, evaluated and analysed on their features and functionalities.

2.2.1 My Route Online (My Route Online, 2010).

My Route Online is a web-based service system which helps user to find the optimal route. Route planning is a difficult task for those company or industry where their works include delivering inventories, provide service to every customer who live in a different location and also for tourist who are not familiar with the particular travel sight. Therefore a system like My Route Online which is able to plan the route may helping them to save fuel, time and money by providing a fast and efficient route. Below are the features that help this system to achieve its goals.

i) Provide input list for user to enter addresses manually

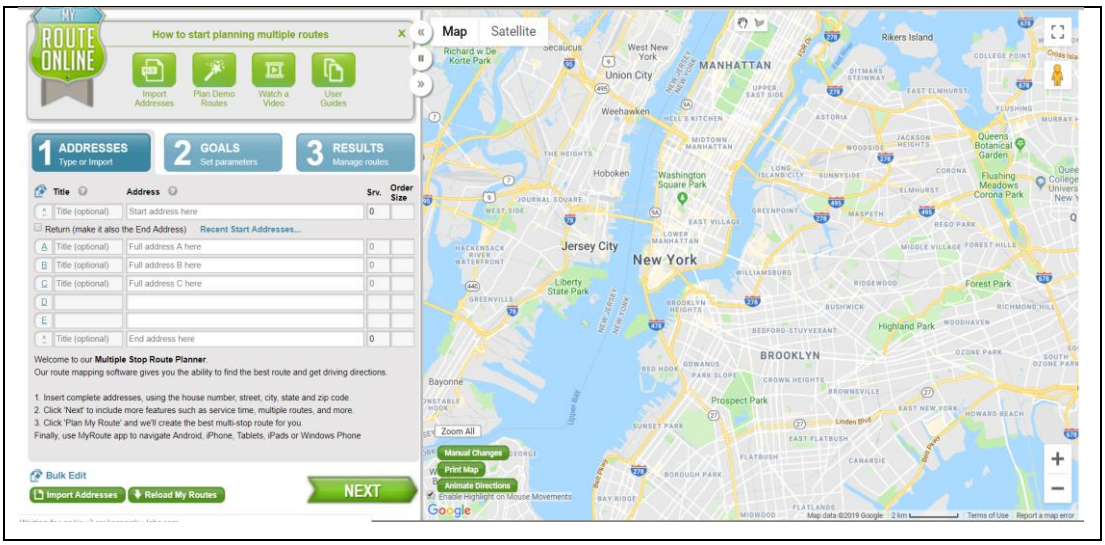


Figure 2.1 Input list to enter the address manually (My Route Online, 2010).

This route planner available user to enter their locations manually by filling up the address. The maximum locations that can be filled in are 350. Besides this system also provide a text input for user to input the service time at a particular location. The default value for service time will be 0 and this means that the driver will just pass through and will not stay at that particular location. The starting address may need to put in the first row of the form and if user wishes to return back to the original starting address after go through several location user may need to input the address of starting location in the last column of the form.

ii) Provide Map for user to visualize

This system provide a map for user hence user can have a better visualization on the route and user can also make sure that the planned route is performing in the right way.

iii) Alternative input option for user

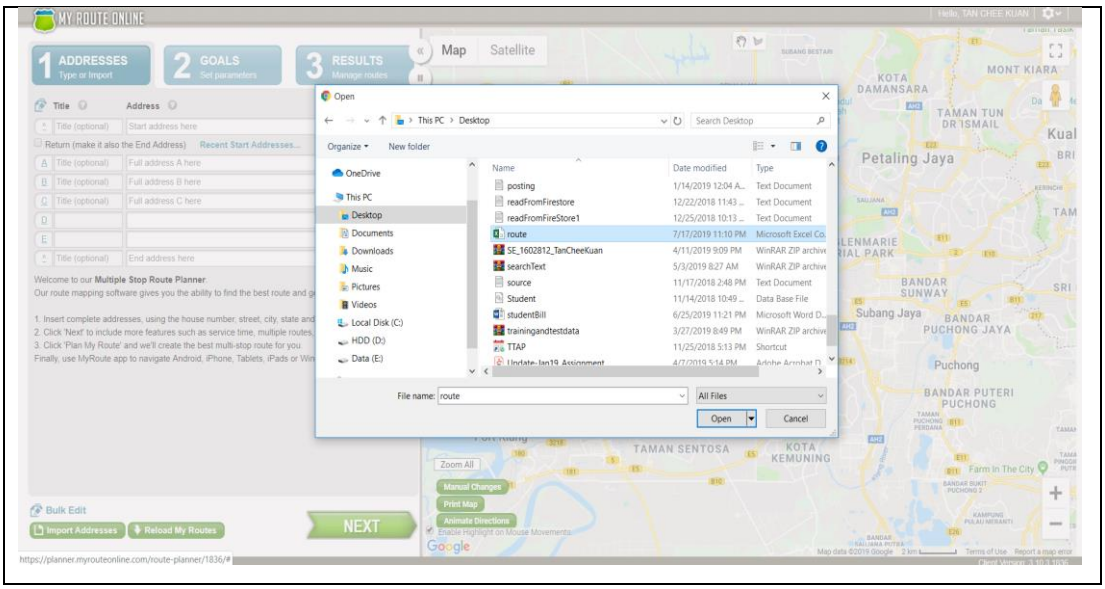


Figure 2.2 Import excel spreadsheet by browsing through the folder (My Route Online, 2010).

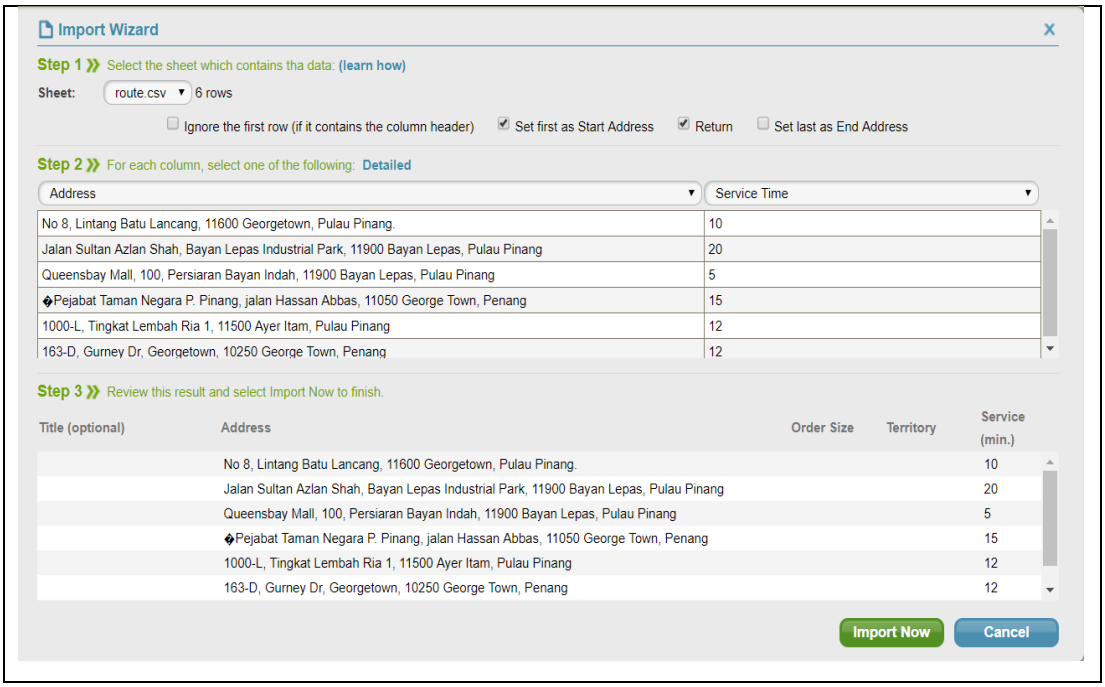


Figure 2.3 Review of the result after the excel had been imported (My Route Online, 2010).

This system provides alternative ways for user to input their data which are written in the .csv or .xsl format. After selecting an appropriate file the system will prompt user for certain set up. User may need to configure certain requirements based on their

needs. For example in the step 1 field user is able to ignore the first row of the data if the .csv file contains column header else the user can select the first data as the starting address. Besides user can also select whether he wants to return to the starting address or not by selecting Return or Set last as End Address. In the step 2 field system is able to detect how many columns are exist in the file and show those data to user. User is able to label those columns by clicking the selection list. There may have several options for user to label their data which are Ignore, Title, Address, Service Time, Territory, Filter-In, Filter-Out, Comments and Order Size. The details of user's data after several settings will show on step 3 field.

iv) Provide advance settings for the route

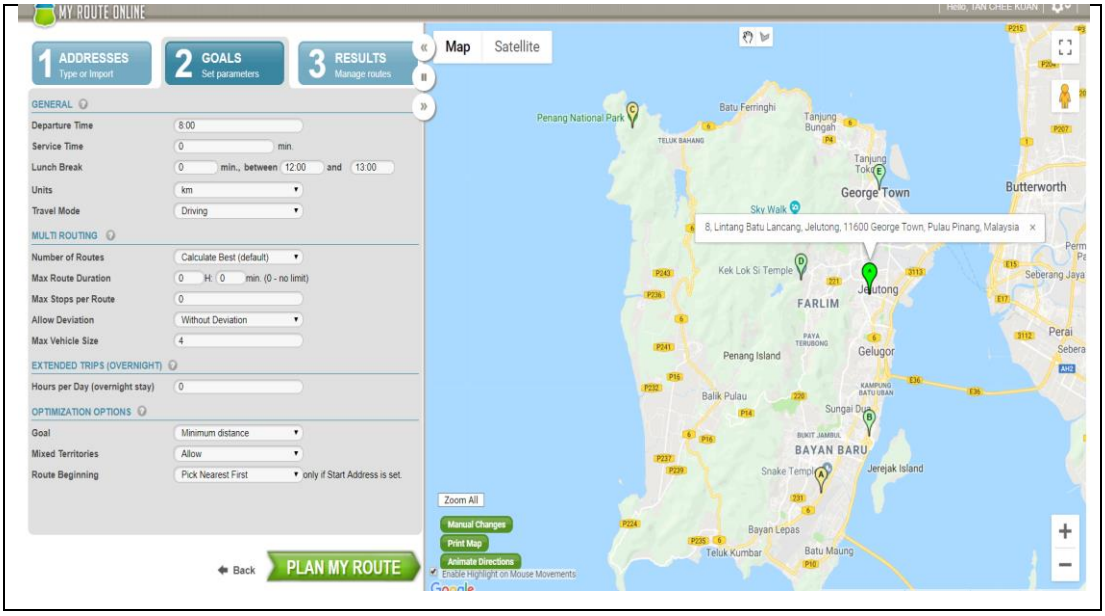


Figure 2.4 Input list for advance settings (My Route Online, 2010).

After clicking the Next button user is prompt to configure extra information that needs to plan the route. In the "General" field, user can set the departure time, default service time for each location, lunch break time for driver, unit of distance (KM/Miles) and travel mode(driving/walking/bicycling). In "Multi Routing" field user is able to set the number of routes he wants, max route duration, max stops per route, allow deviation option, max vehicle size. There may also have an "Extended Trip" field for user to plan the routes a few days ahead by input the value of how many working hours per day. System will plan the route based on the working hour, if the locations going to be visited is too much and unable to visit all of them within the working hour then those

unvisited locations will be planned into another route trip on the next day. Last, there may have the "Optimization Options" for user to select how user want to optimize their route. In this page, user is able to select the optimization goal which is minimum time, minimum distance, balance time and distance and keep order as is. In this field, user can also select whether he/she wishes to mixed territory and user can also select the route beginning by pick nearest first or pick farthest first.

v) Provide dashboard to display the order of the route

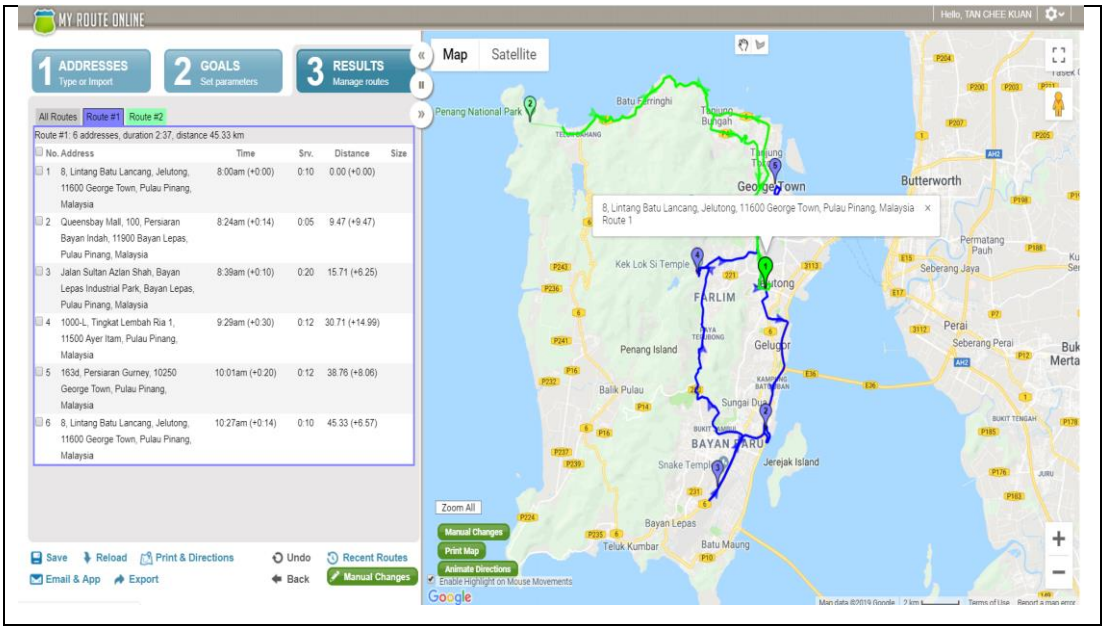


Figure 2.5 Result on the dashboard (My Route Online, 2010).

After the route has been generated, the system provides a dashboard for user to display the outcome of the route. From figure 2.2.1.5 user are able to see that the system had generated two routes based on user's previous configurations. The first row in the dashboard presenting the starting location followed by the next location to be met till the last row in the dashboard which is also the address of the starting location.

vi) Provide Save option after the route has been generated

This system is able to save the route that had been generated, therefore user is able to look back on the previously planned route.

vii) Provide Print & Direction option after the route has been generated

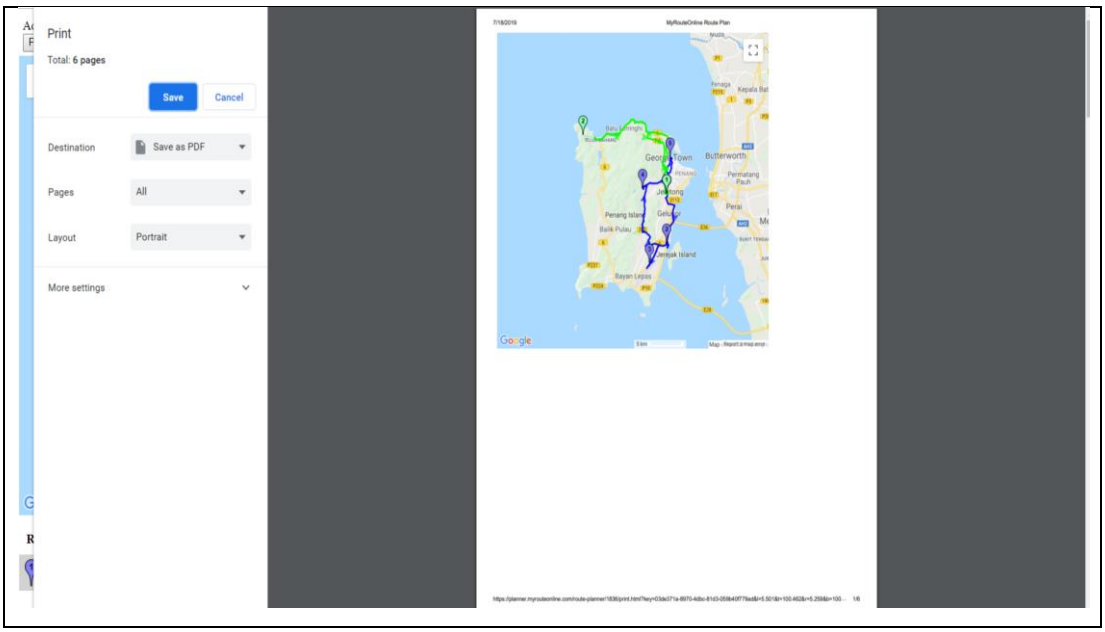


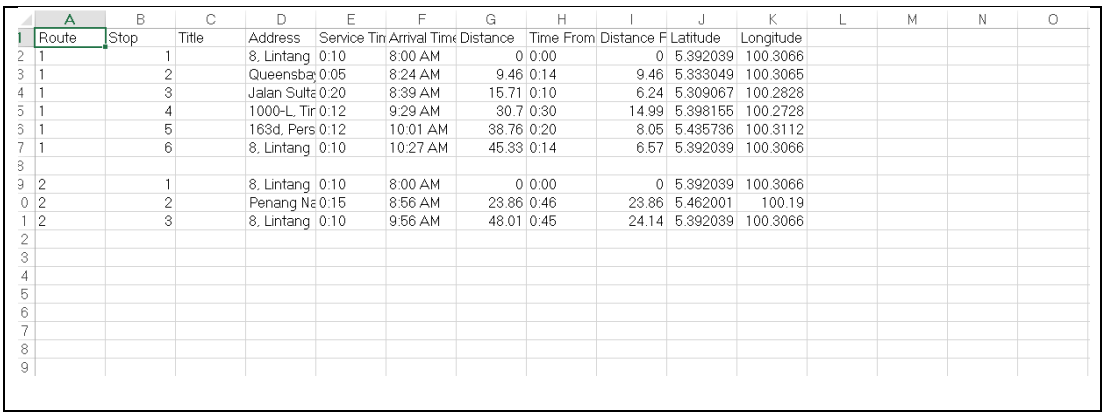
Figure 2.6 Preview before printing (My Route Online, 2010).

8, Lintang Batu Lancang, Jelutong, 11600 George Town, Pulau Pinang, Malaysia			
	1.	Head **south** on **Lintang Batu Lancang** toward **Solok Batu Lancang**	0.28 km	1 min
	2.	Turn **right** onto **Lebuhraya Batu Lancang**	0.37 km	1 min
	3.	Keep **right** to continue toward **Jalan Masjid Negeri/P19**	0.05 km	1 min
	4.	Slight **left** onto **Jalan Masjid Negeri/P19**	0.52 km	1 min
	5.	Slight **right** onto **Jalan Masjid Negeri/Udini Underpass/P19**	0.68 km	1 min
	6.	Continue onto **Jalan Tunku Kudin**	1.03 km	1 min
	7.	Merge onto **Lebuhraya Tun Dr. Lim Chong Eu/Route 3113**	1.27 km	1 min
	8.	Slight **left** to stay on **Lebuhraya Tun Dr. Lim Chong Eu/Route 3113**	3.96 km	4 mins
	9.	Take the exit toward **Queensbay/Eastin Hotel**	0.33 km	1 min
	10.	Slight **right** onto **Persiaran Bayan Indah**	0.23 km	1 min
	11.	At the roundabout, take the **1st** exit and stay on **Persiaran Bayan Indah**	0.48 km	1 min
	12.	Keep **left** to stay on **Persiaran Bayan Indah**	0.05 km	1 min
	13.	Slight **left** Destination will be on the left	0.21 km	2 mins
Queensbay Mall, 100, Persiaran Bayan Indah, 11900 Bayan Lepas, Pulau Pinang, Malaysia			
	1.	Head **north** toward **Persiaran Bayan Baru**	0.23 km	1 min
	2.	Slight **left** onto **Persiaran Bayan Baru**	0.06 km	1 min
	3.	Turn **left** at **Persiaran Bayan Indah**	0.94 km	1 min
	4.	Continue onto **Lebuhraya Sungai Nibong**	0.54 km	1 min
	5.	Slight **left** (signs for **Kompleks Tabung Haji/Route 6/Bayan Baru/Lapangan Terbang Bayan Lepas/Teluk Kumbar**)	1.10 km	2 mins
	6.	Continue straight onto **Jalan Sultan Azlan Shah/Route 6**	2.60 km	3 mins
	7.	Sharp **right** to stay on **Jalan Sultan Azlan Shah/Route 6**	0.78 km	1 min
Jalan Sultan Azlan Shah, Bayan Lepas Industrial Park, Bayan Lepas, Pulau Pinang, Malaysia			
	1.	Head **northeast** on **Jalan Sultan Azlan Shah/Route 6** toward **Jalan Tokong Ular**	1.82 km	3 mins
	2.	Keep **left**	0.93 km	3 mins
	3.	Slight **left** toward **Jalan Tun Dr Awang** (signs for **P6/Bukit Jambul/Relau/P14/Balik Pulau**)	0.18 km	1 min
	4.	Continue onto **Jalan Tun Dr Awang**	1.21 km	2 mins
	5.	Turn **right** onto **Persiaran Bukit Jambul**	1.25 km	3 mins

Figure 2.7 Direction of those particular route (My Route Online, 2010).

This system enables the user to print out the route map and the direction of that particular route, therefore it is easy for the driver to view when driving their car.

viii) Provide Export option after the route has been generated



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Route	Stop	Title	Address	Service Tin	Arrival Time	Distance	Time From	Distance F	Latitude	Longitude				
2	1	1		8, Lintang	0:10	8:00 AM	0	0:00	0	5.392039	100.3066				
3	1	2		Queensbay	0:05	8:24 AM	9.46	0:14	9.46	5.333049	100.3065				
4	1	3		Jalan Sulta	0:20	8:39 AM	15.71	0:10	6.24	5.309067	100.2828				
5	1	4		1000-L, Tir	0:12	9:29 AM	30.7	0:30	14.99	5.398155	100.2728				
6	1	5		163d, Pers	0:12	10:01 AM	38.76	0:20	8.05	5.435736	100.3112				
7	1	6		8, Lintang	0:10	10:27 AM	45.33	0:14	6.57	5.392039	100.3066				
8															
9	2	1		8, Lintang	0:10	8:00 AM	0	0:00	0	5.392039	100.3066				
10	2	2		Penang Na	0:15	8:56 AM	23.86	0:46	23.86	5.462001	100.19				
11	2	3		8, Lintang	0:10	9:56 AM	48.01	0:45	24.14	5.392039	100.3066				
12															
13															
14															
15															
16															
17															
18															
19															

Figure 2.8 Exported result (My Route Online, 2010).

This system enables user to export the generated route to the .csv file.

viiii) Provide Email & App option after the route has been generated

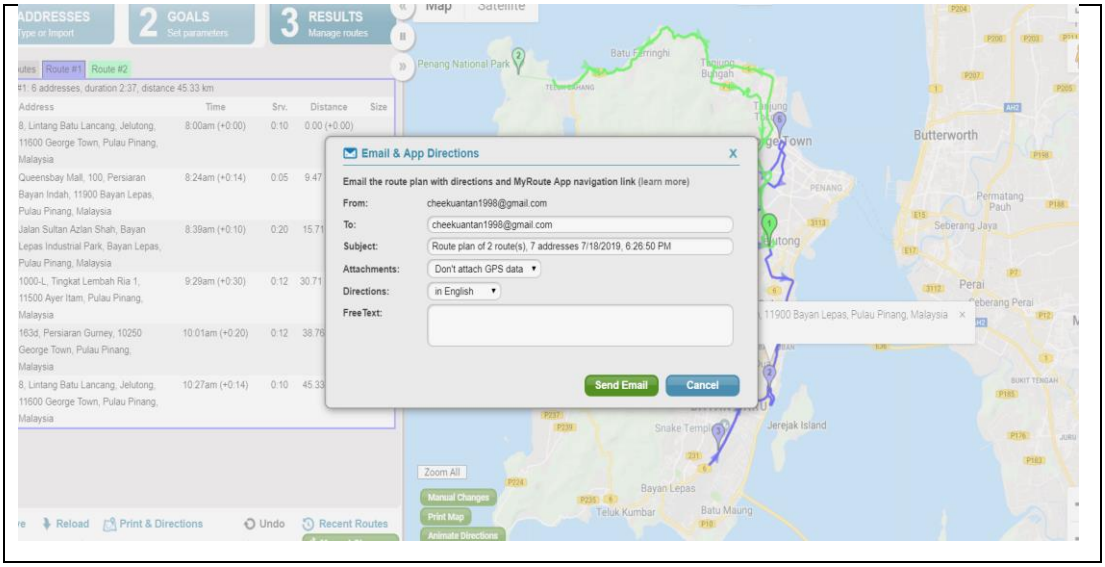


Figure 2.9 Input list for Email & App option (My Route Online, 2010).

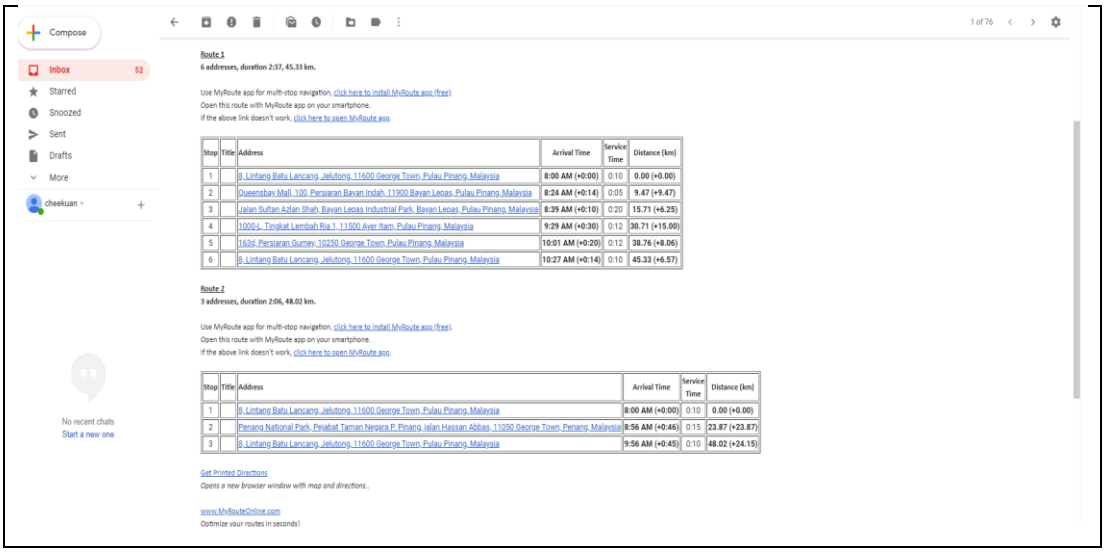


Figure 2.10 View on email (My Route Online, 2010).

This system enables user to send the generated route through email.

2.2.2 Justification on My Route Online's route planner system

"Available" Feature that will be consider during the implementation of this project.

"Not Available" Feature which will not be considered during the implementation of this project.

Table 2.1 Conclusion on the features of My Route Online

	My Route Online (My Route Online, 2010).
Provide input list for user to enter address manually	**Available.** This is the common feature that will be developed in every software system (including system that this project is going to develop) in order to prompt for the user input. This feature is practical if data going to input is small, if the data is large then an alternative option must be provided by the system. This project is developed to find the optimal route to deliver based on the due date of D/O. There may have possibility that a new D/O is generated and this particular D/O and may need to be

	sent at that particular day, therefore user can input the relevant data of this D/O into the system by using this feature.
Provide map for visualization	**Available.** By having a map on this project's system, user are easier to visualize the planned route hence driver can have a better understanding on the overview of the route. Besides, user is able to check weather this system is performing well.
Alternative input option (excel spreadsheet)	**Available.** There may have hundreds of D/O that will be generated on a particular day. Therefore it is not practical and effective for user to key in manually if the data is large. The easier way to do is extract those related data from the company's existing system and store them in an excel spreadsheet. The excel spreadsheet will be used as the input for this feature. Therefore user may not need to key in manually only several setting may need to configure like the label of each column of data, option to ignore the first row of the data in the excel spreadsheet due to this row may be the original label from the company's system and option to return the route to the starting address.
Advance settings for route	
Departure time	**Available.** This setting may be considered in this project, due to it is able to estimate the approximate time to complete the particular planned route. Therefore manager is able to keep track of their drivers.
Default service time	**Available.** This setting will also be develop in this project. User is prompt to enter the service time for their customers based on their previous experience therefore a more reliable estimated time to complete a particular route will be generated.

Lunch break time for driver	**Available.** Lunch break time for driver is quite reasonable. By configuring this setting the reliability of the estimation time to complete a particular route will be increased.
Unit of distance (KM/ Miles)	**Available.** Some people are familiar with KM while some may familiar with Miles hence this setting will be provided in this project to increase user friendly. User may not need to re-calculate the value of distance into the particular unit he wants.
Travel Mode (Driving/ Walking/ Bicycling)	**Not Available.** This setting is not necessary therefore it can be ignored due to all the inventories/cargoes are sent to customer by using company's truck. Driving option will be set as default and is not able to change in this project.
Multiple routes option	**Available.** There may have several numbers of trucks in the company to deliver cargoes to their customers on that particular day therefore system which only able to calculate one route is not sufficient for this project. An algorithm which is able to handle multiple routes will be developed in this project and those routes will be generated based on the number of trucks sent to deliver cargoes.
Max route duration	**Available.** Based on the assumption of this project all route to deliver will be complete on that particular day. Therefore this feature/option can be included in this project in order to make sure the planned route can be complete on that particular day.

Max stop per route	**Not Available.** This setting is not needed in this project because the location for the truck to stop will be the customer's location.
Allow Deviation	**Not Available.** This setting only work well when "Max stop per route" has been configure. Due to "max stop per route" will not implement in this project therefore this feature will be ignore as well.
Max Vehicle Size	**Not Available.** This feature will not be implemented in this project due to this feature does not help in planning the route to deliver.
Extended Trips	**Not Available.** Based on the assumption of this project, all planned route will be complete at that particular day. Therefore, there may have no extended trips for this project.
Option for optimization goal	**Not Available.** Based on the scope of this project, the optimization goal for the particular route will only base on the shortest distance instead of providing extra option likes shortest duration or based on the order of the address that user key in.
Provide dashboard to display the order of the address and direction of the particular planned route	**Available.** User is able to view the outcome of the route easily when a well-formatted dashboard is provided. Based on the scope of this project a SMART D/O will be generated on the dashboard. After researching on this Route Planner system, direction for a particular route may be important to let the driver know the exact direction to archive the distance goal. Therefore, this

	project will adapt the ideas of this feature during the implementation.
Save the recent planned route	**Available.** This feature may be necessary in this project, due to user may wish to view back the previous planned route on a particular day. This project may not only store the information of each route but will also consisting information for the cargoes.
Print out the route and its direction	**Available.** This feature will be consider in this project due to it is easy for driver to view when they are driving but the constraints is this may use up a lot of papers.
Export the route to excel spreadsheet	**Not Available.** This feature is not necessary due to those information will be save in the system which will be developed in this project.
Sent the recent planned route to user's email	**Not Available.** This feature will not be implement because this feature does not provide any benefit to the overall process.

2.2.3 Conclusion for My Route Online's route planner system

My Route Online actually provide a simple layout which put the input list of several configuration and result of the route together with the map hence user may not need to switch between different tabs or pages. This will provide user to have a better experience and "ease of use" feeling when using this online system. Besides, the features and settings of this system are adequate an appropriate compare to other online route planner system like AA's and Mapquest's route planner system which only provide user to enter the input manually and provide several options like avoid the toll and avoid motorway which are not appropriate for this project due to those options may limit the calculation of the system which are going to develop in this project. Last comparing to AA's and Mapquest's route planner system My Route Online's route

planner system enables user to find the optimal route while AA and Mapquest are not able to do that. The route for AA and Mapquest is planned based on the order of the addresses user enter into the system and user may need to re-adjust the sequence manually and may need to re-calculate the route each time. Therefore in above section, only MyRouteOnline's feature and setting is cover due to all the feature in the other two system can also be found on MyRouteOnline. In conclusion, the design and implementation of this project's system is going reference from MyRouteOnline.

2.2.4 **EasyCargo (EasyCargo, n.d.).**

EasyCargo is a truck and container load planning software which are currently used by 500+ company around the world. EasyCargo provides a simple layout that put add item feature and visualization of the result at one screen hence user no need to switch between different tabs. Besides by using EasyCargo user is able to save the space and time due to their team had spent a great deal of time in developing their load planning engine which can help user to plan their cargo quickly. By using EasyCargo user is able to place their cargo effectively into the trucks and containers. Last EasyCargo also provides better visualization for user to have a better understanding of how to place their cargoes. Below are the features that exist in this online software.

i**) Provide input list to add in new cargo.**

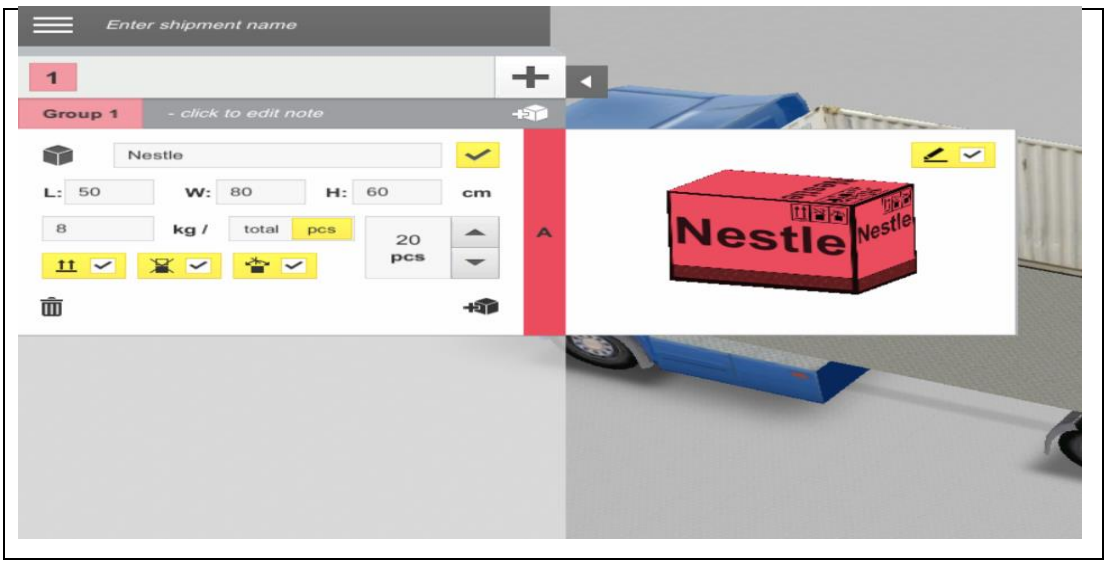


Figure 2.11 Input list for EasyCargo (EasyCargo, n.d.).

EasyCargo enables user to add in the new item manually by clicking the add button which can be view on Figure 2.2.3.1. User is able to enter the item's description, size of the item, weight of the item and quantity of the item. Besides, user can also activate the constraint that applies to the particular item. Those constraints are do not till this cargo item, do not stack on this cargo item and do not rotate this cargo item. Those constraints are set as false for the default value. The input list also consists of delete function and also add a new item. After all items have been included and configured user can click on the load button to start for the calculation.

ii) Alternative input option from Microsoft Excel.

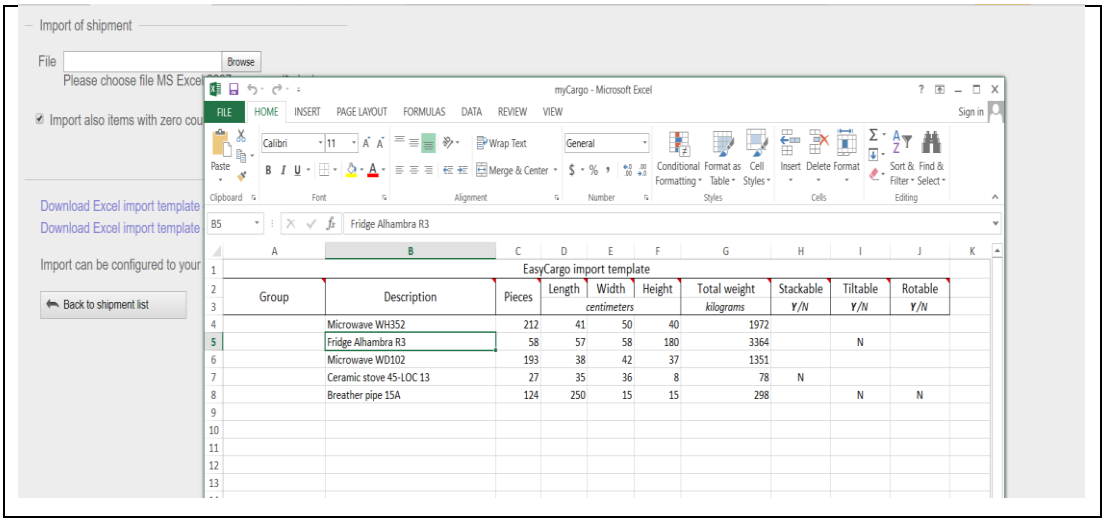


Figure 2.12 Excel Template for EasyCargo (EasyCargo, n.d.).

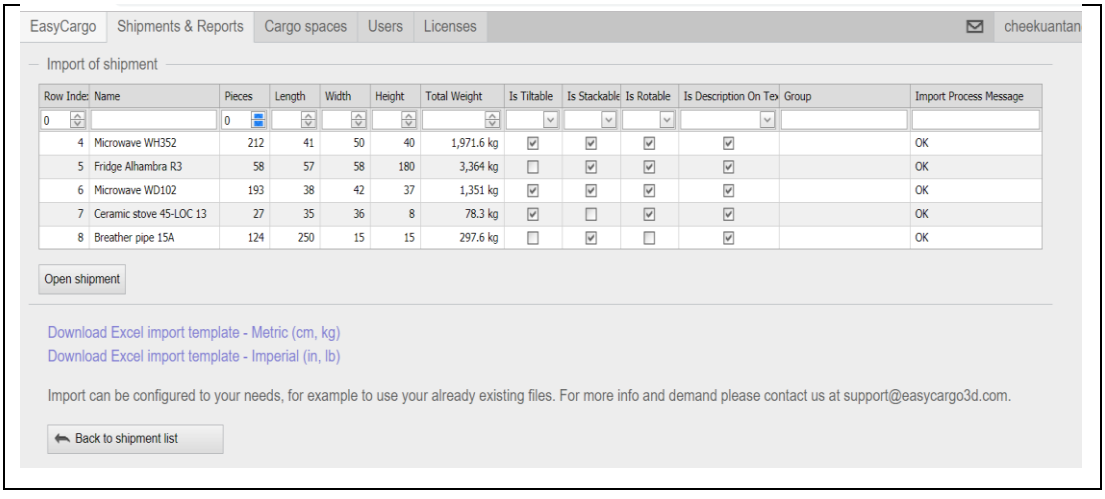


Figure 2.13 Review the result (EasyCargo, n.d.).

EasyCargo enables user to input their cargos' information by importing the Microsoft Excel file. In order to let user have a better understanding of the format on the Excel file, EasyCargo provides a template for user to download. User may only need to modify and the value to fit with his own cargos' information. There may have several attributes needed to fill up by the user which are description, pieces, length, width, height, total weight, stackable, tillable and rotable. Besides, EasyCargo available user to enter 250 items for one shipment. After the data has been imported EsayCargo available user to check for the result which is shown in figure 2.2.3.3. After all, configuration has been done user may prompt to open shipment item and press the load button to start the calculation.

iii) 3D cargo load plan visualization

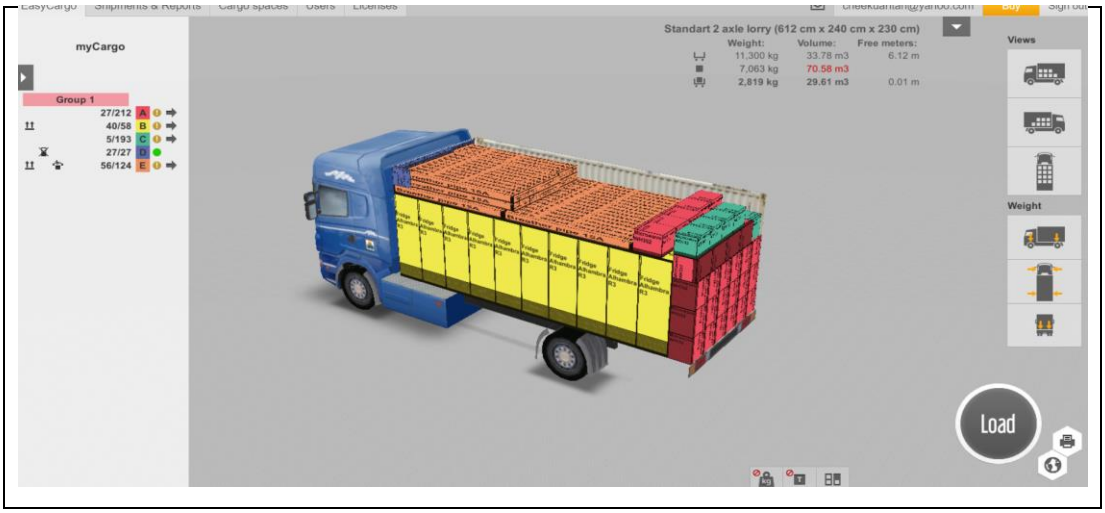


Figure 2.14 3D visualization for loaded items (EasyCargo, n.d.).

Figure 2.15 3D visualization for particular loaded items (EasyCargo, n.d.).

EasyCargo provides a 3D visualization after calculation has been complete. EasyCargo also provides several angle option for user to view on a different angle, by pressing and holding left button of mouse user are able to view the cargo space from 360 degree by moving the mouse. By selecting a particular item from the input list, user is able to view the placement of the particular items in the truck.

iv) Result of loaded cargoes and truck capacity

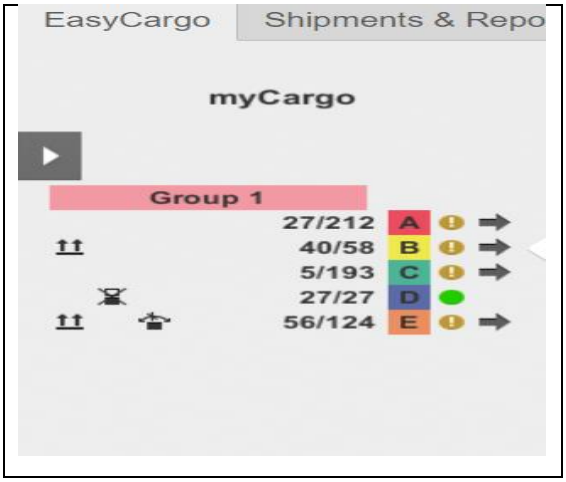


Figure 2.16 Number of items has been loaded (EasyCargo, n.d.).

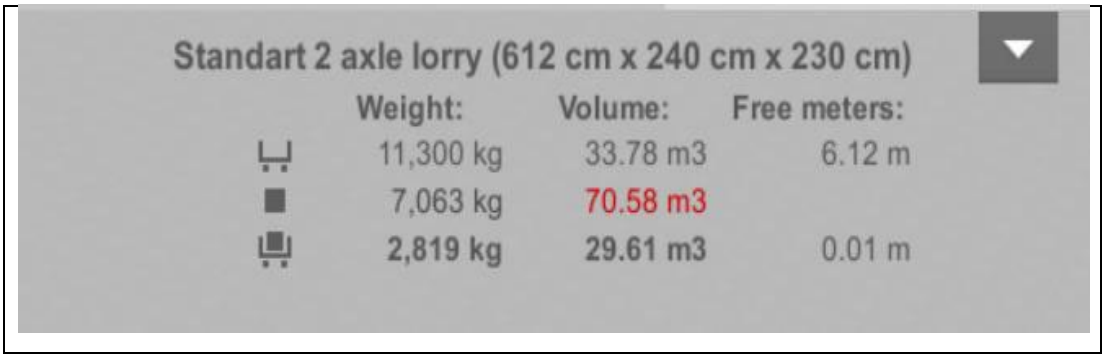


Figure 2.17 Capacity of the truck that had been used up (EasyCargo, n.d.).

EasyCargo enables user to view on how many items had been loaded into the particular truck or container after the calculation had been completed which can be viewed at figure 2.2.3.5. Besides, user is able to view on how well this system optimizes the space of the truck. From figure 2.2.3.6 users are able to see that the maximum weight that the truck can deliver and the volume of the truck in the first row. The second row represents the total weight of all the cargoes and the total volume of the cargoes. If the values of total weight and total weight of the cargoes exceeded the value in the first row then those values will be represented in red colour. The third row shows the total weight and total volume of cargoes that have been loaded into the truck.

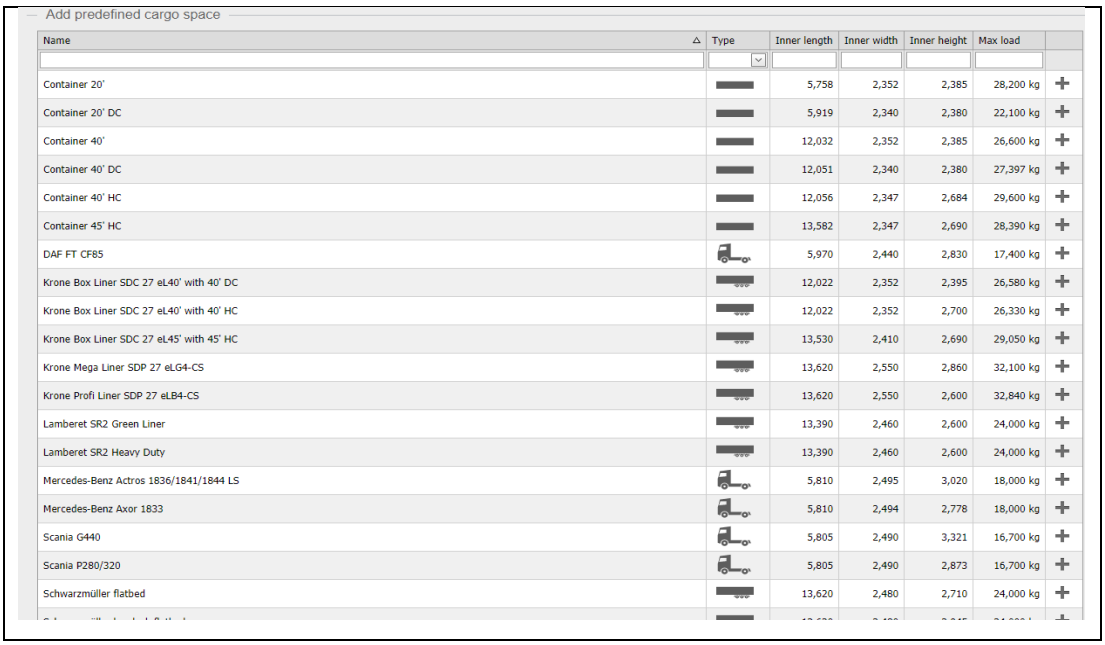
v) Provide option to ignore the weight limit

User can have the option whether he wish to respect the weight limit of the particular truck or ignore the weight limit. If the weight limit is ignore then the system will load the cargoes into the truck until the volume of the truck has been exceeded without considering weight of the cargoes.

vi) Print option

EasyCargo enable the user to print out the 3D visualization of the loaded cargoes and the number and detail of the items which has been loaded into the truck.

vii) Add predefined cargo space



Name	Type	Inner length	Inner width	Inner height	Max load
Container 20'		5,758	2,352	2,385	28,200 kg
Container 20' DC		5,919	2,340	2,380	22,100 kg
Container 40'		12,032	2,352	2,385	26,600 kg
Container 40' DC		12,051	2,340	2,380	27,397 kg
Container 40' HC		12,056	2,347	2,684	29,600 kg
Container 45' HC		13,582	2,347	2,690	28,390 kg
DAF FT CF85		5,970	2,440	2,830	17,400 kg
Krone Box Liner SDC 27 eL40' with 40' DC		12,022	2,352	2,395	26,580 kg
Krone Box Liner SDC 27 eL40' with 40' HC		12,022	2,352	2,700	26,330 kg
Krone Box Liner SDC 27 eL45' with 45' HC		13,530	2,410	2,690	29,050 kg
Krone Mega Liner SDP 27 eLG4-CS		13,620	2,550	2,860	32,100 kg
Krone Profi Liner SDP 27 eLB4-CS		13,620	2,550	2,600	32,840 kg
Lamberet SR2 Green Liner		13,390	2,460	2,600	24,000 kg
Lamberet SR2 Heavy Duty		13,390	2,460	2,600	24,000 kg
Mercedes-Benz Actros 1836/1841/1844 LS		5,810	2,495	3,020	18,000 kg
Mercedes-Benz Axor 1833		5,810	2,494	2,778	18,000 kg
Scania G440		5,805	2,490	3,321	16,700 kg
Scania P280/320		5,805	2,490	2,873	16,700 kg
Schwarzmüller flatbed		13,620	2,480	2,710	24,000 kg

Figure 2.18 Predefined Cargo Space (EasyCargo, n.d.).

EasyCargo provided this feature for user to add in the cargo space which their company own.

viii) Save option for the particular shipment

User is able to save the result of the particular shipment.

2.2.5 **LoadCalculator (SEARATES, 2005).**

LoadCalculator is an application that provides by SEARATES which help the industry to solve the cargoes packing problem. LoadCalculator provides an alternative for those goods deliverable related company to quickly and effectively placing the inventories inside the container and determines their optimal positioning in the truck. Below is the demo that provided for user to have a better understanding of the feature and how its work, the real product may needed to subscribe by paying $25 per month.

i) Select equipment

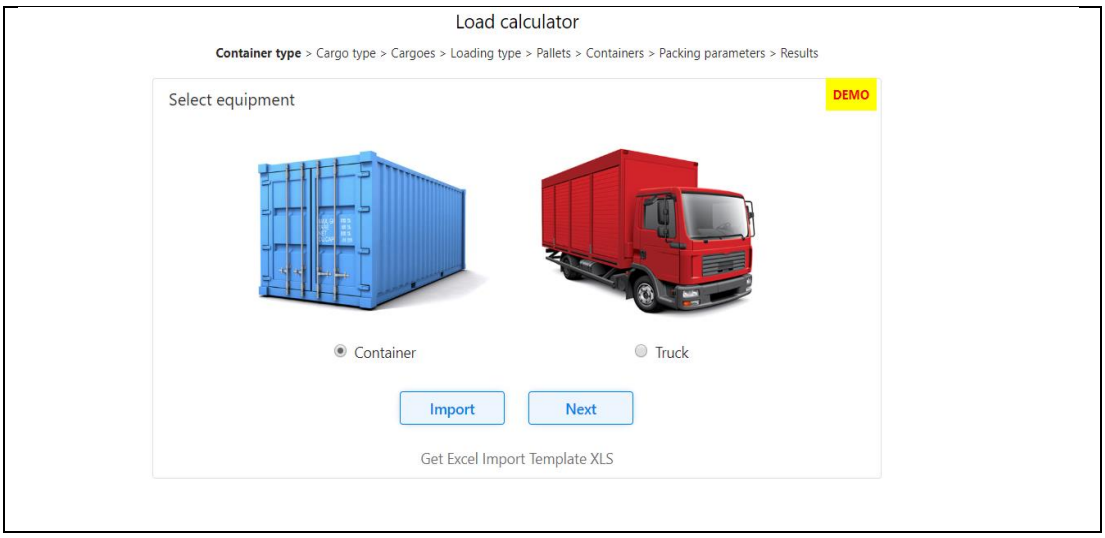


Figure 2.19 Option to select different equipment (SEARATES, 2005).

LoadCalculator enable user to select different equipment for cargo loading.

ii) Cargo types selection, add items and configuration of cargo's parameters

Figure 2.20 Packing List (SEARATES, 2005).

There may have different cargo types that can be selected in LoadCalculator they are Boxes, BigBags, Barrels and Sacks. After selecting the cargo type user can add the particular type of cargo into the packing list by pressing the "Add cargo" button. After cargo is adding into the list user can set the parameters for each cargo.

iii) Alternative input option from Microsoft Excel

LoadCalculator enables user to input their cargos' information by importing the Microsoft Excel file. Same with EasyCargo LoadCalculator provide a template for user to have a better understanding and easy for user to make modification. There may have several attributes needed to fill up by the user which are cargo type, name, length, width/diameter of barrels, height, weight and quantity.

iv) Select equipment for cargo loading

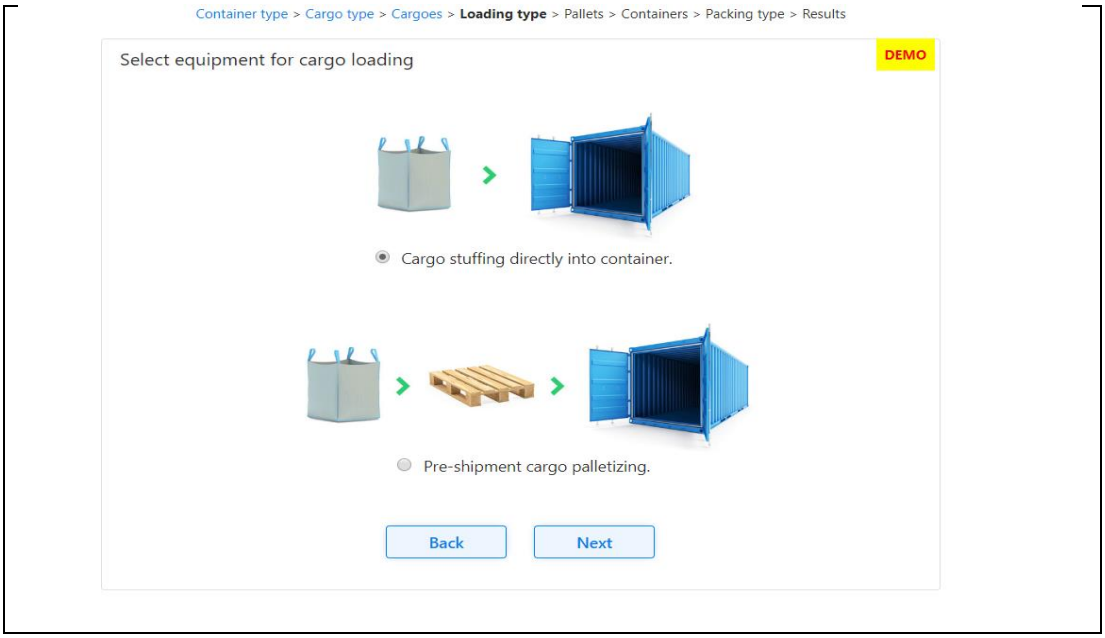


Figure 2.21 Type of loading (SEARATES, 2005).

LoadCalculator provides different type of loading which is directly place the cargo into the truck or pre-shipment cargo palletizing.

v) Container Selection

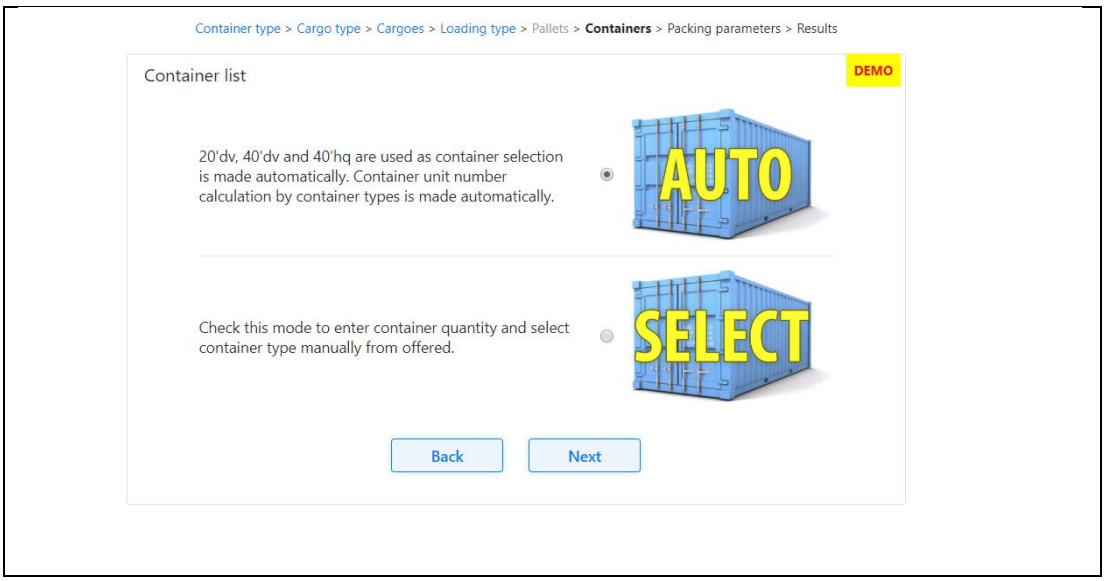


Figure 2.22 Container List (SEARATES, 2005).

Different size of containers which related to the equipment that selected by the user in the beginning will display and user may need to select the container which he own.

vi) Provide selection for cargo rotation

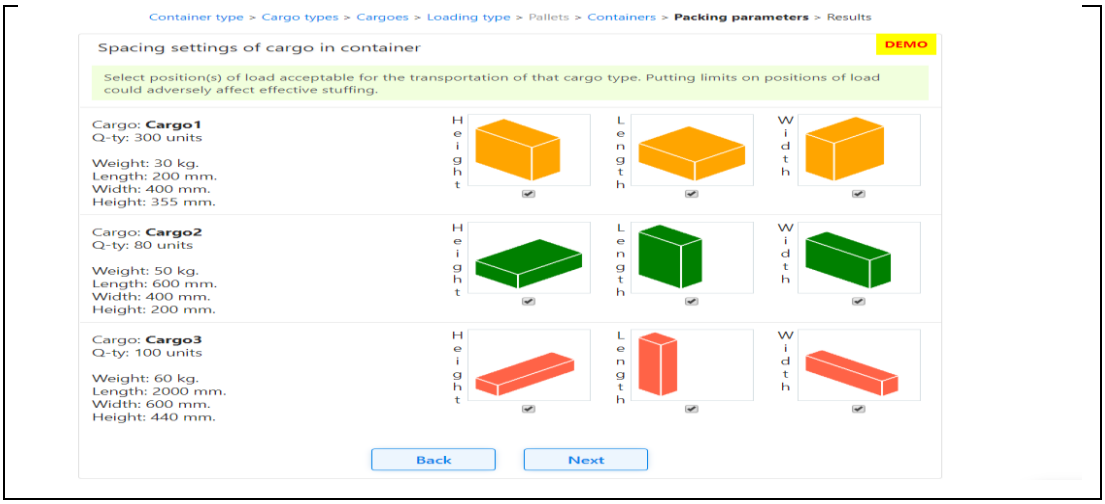


Figure 2.23 Selection of cargo placement (SEARATES, 2005).

LoadCalculator provides option for user to select the placement of the cargo therefore user can have a better control on those cargoes when loading into the truck or container.

v) Result in 3D visualization

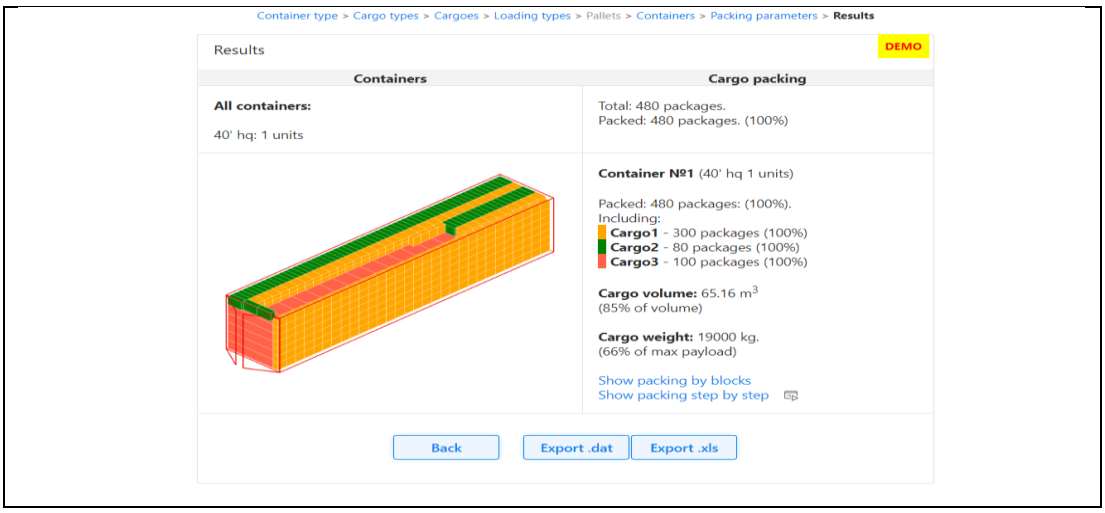


Figure 2.24 View of the result (SEARATES, 2005).

LoadCalculator provides a 3D visualization after calculation has been completed. Besides the 3D visualization of the container user is able to see how many number of that particular cargoes had been packed into the container. Other than that, user is also able to know the percentage of the volume of the truck that being utilize.

vi) Show packing order step-by-step

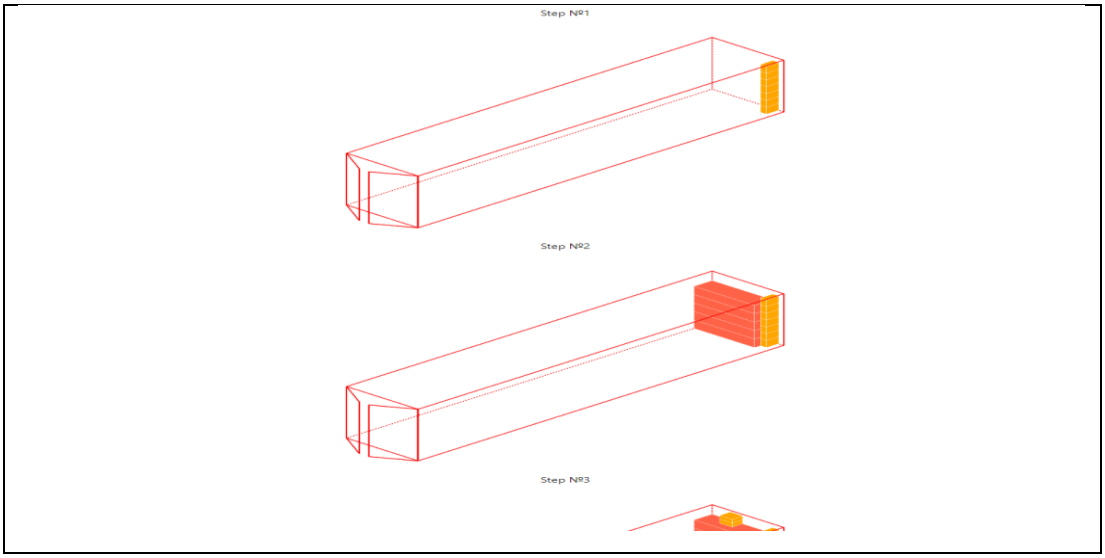


Figure 2.25 View of packing's step (SEARATES, 2005).

LoadCalculator provides visualization of the packing steps for user to have a better understanding on how to load cargoes into a particular container.

2.2.6 Comparison and Justification on online cargo loading systems

"-" Feature does not exist in this system.

"Available" Feature that will be consider during the implementation of this project.

"Not Available" Feature which will not be considered during the implementation of this project.

Table 2.2 Comparison on the features between EasyCargo and LoadCalculator

	EasyCargo (EasyCargo, n.d.).	**LoadCalculator (SEARATES, 2005).**
Input List to add in new cargo	Available	Available
Alternative input option	Available	Available
Selection of cargo types	-	Not Available
3D visualization of the result	Available	Available
Option to ignore weight limit	Available	-
Cargo space selection	Available	Available
Option to activate the rotation for a particular cargo	Not Available	Not Available
Show packing order step-by-step	-	Available
Print option	Available	-
Save option	Available	-

Input List to add in new cargo

Feature that will be developed in every software systems (including system that this project is going to develop) in order to prompt for the user input. By having this feature user is able to manually add in the detail of the cargo when there is a new item needed to be delivered. This provide flexibility for the user due to user may not need to key in the data into the excel file then import all those relevant data again and again.

Alternative input option

This feature will be implemented in this project due to it provides another option for user to input the detail of the cargo when the data is large. It is not convenient and effective when user needs to key in those data line by line, hence providing an alternative input option with excel format will be the best way to increase the efficiency of the overall process and at the same time increase the flexibility of the system.

Selection of cargo types

Based on the assumption of this project, only box cargoes will be considered into the implementation.

3D Visualization of the result

This feature will be the most important and the hardest part during the implementation of this project, but this feature play an important role in order to let user to have a better understanding on how the cargoes were loaded into the container or truck. EasyCargo provides a dynamic 3D visualization for user to view their result while LoadCalculator only provides the image of the arranged cargo in the space but it is enough for user to understand. Although EasyCargo has a better visualization on the result comparing to the LoadCalculator but due to the time constraint and inexperience in construct 3D object in web development the 3D visualization for this project will be referencing on LoadCalculator.

Option to ignore weight limit

Each truck or container has its own maximum weight loaded limit but if user select to just respect to the weight limit then the truck or container may have possibility not to

be fully utilize. Hence this feature will be implement in this project and user can have more option to control on their cargo load plan.

Cargo space selection

This feature may also be implemented in this project due to each company have different type of vehicle. Some companies may be using truck some companies may using a container to deliver their inventories. System may need to has a database to store each type of cargo space and their detail therefore user can select different cargoes' space which own by them.

Option to activate the rotation for a particular car

Based on the assumption of the project scope, all cargoes can be rotated in order to optimize the capacity of the truck. Therefore this option will be set as default in this project.

Show packing order step-by-step

This project will be going to implement similar feature into the route optimization system due to if objects is displayed in the 3D format then there may sure have some objects be blocked by other objects during truck visualization hence user is not able to know how was the placement of the particular object which has been blocked. LoadCalculator provided this feature which lets the user to know how to place their inventories into the truck from the beginning to the end of the loading process. This feature is useful and may help workers in those companies to speed up their loading process.

Print option

This feature will be considered in this project due to it is easy for drivers to view when they are loading the inventories into the truck but the constraints are this may use up a lot of papers.

Save option

This feature may be necessary for this project, due to user may wish to view back the previously loaded cargoes on a particular day. This feature can also act as an evidence

to prove that those cargoes had be loaded to the truck and sent to the customer on that particular date.

2.2.7 Conclusion on existing cargo loading system

EasyCargo and LoadCalculator provided features and functionalities which are appropriate for this web-based route optimization system. Both of them are able to maximize the cargo space with their own algorithm, but when comparing toward the 3D visualization part EasyCargo provide a better design and interaction comparing to LoadCalculator. In EasyCargo the cargo space is able to rotate 360 degrees in order for user to have a view on a different angle, by selecting a particular items on the input list user is able to view only those particular items which had loaded into the cargo space. Therefore user can have the view on how those particular items are placed in the cargo without blocked by other items but LoadCalculator solve this problem by showing the packing order step-by-step, this is also helpful but may use up of many papers. Either one of the solutions will be considered to develop in this system based on constraint time and skills. Last, the algorithm and 3D visualization of the existing system can be further modified to suit this route optimization system like showing the items that had loaded into the cargoes space by clicking on a particular D/O instead of a particular items and the Last-In-First-Out (LIFO) strategy may be involved in the algorithm instead of just loading those existing items without specified the priorities of the items.

2.3 Related research problem and its algorithm

2.3.1 Travelling Salesman Problem (TSP)

TSP is not a currently found problem and it has a long history and always remains a source of challenge for those researchers and mathematicians. TSP is to find out the minimum cost from all the combinations of routes when travel from place to place without duplication of any places that had been travel before (Khalil, et al., 2017). The challenging part of TSP is the complexity of solving this problem is getting increase exponentially when the place to visit or travel is increase (Khalil, et al., 2017). In real-world this problem may be hard to solve without the help of any system, human next location to visit is based on his previous knowledge and experience, and it may have the chances to select the route which is more costly. There may have several existing algorithms had been done by the researchers like Genetic Algorithm, Ant Colony

Algorithm, Pick Near Point Algorithm. In this project, genetic algorithm will be selected to implement in this system and the details of the algorithm will be further discussed.

Genetic Algorithm (GA)

Genetic algorithm is a search based algorithms which involve natural selection and evolution of the generation (Fan, 2006). The concept of this algorithm is continually undergoing the biological evolution process in order to increase the chance of getting generation which is better than their parents. In genetic algorithm, a possible solution to solve a particular problem is store in a collection which called as a population (Fan, 2006). Each individual in the population is known as chromosome and each of them has its own fittest value which may be used to generate new generation by going through several processes which are initial population, parent selection, crossover and mutation (Fan, 2006). According to Fan (2006), genetic algorithm is a repeating process until reaching the number of the generation that had been initialized at the beginning. The process of genetic algorithm can be stopped when there is no further improvement is observed or the algorithm had reached the number of iterations depending on speed and resource criteria.

i) Initial population phase (Ahn and Ramakrishna, 2002)**:**

In this phase, a population size which contains N numbers of chromosomes will be generated. Chromosome/route is formed by randomly selecting and combining all genes/cities that the user needed to visit without any duplication of them, and the order of the genes may play an important role in travelling salesman problem.

ii) Selection phase (Ahn and Ramakrishna, 2002)**:**

There are two kinds of selection in this phase, one of them is parent selection while the other is survivor selection. In parent selection, M numbers of individuals will be selected to pass their genes or characteristics to their next generation by undergoing crossover process. Parent selection must favor to fittest chromosome, therefore there is a higher chance to generate a better offspring. But there must also have chances for weak chromosome to be select, therefore the diversity of the solution can be maintained. If only the fitness individuals are selected, then there may have chance that the final solution only reaches the local maximum instead of the global one. In

survivor selection, X numbers of individuals will be select based on the result of fitness value. In survivor selection, only those individual who has a better ranking in the population will be select and retain until the next generation without performing crossover process. This is to make sure that there are always some individuals with better fitness value.

iii) Crossover phase (Ahn and Ramakrishna, 2002)**:**

Two parents will be selected and pass part of their genes to their new generation. Therefore, the new offspring inherit both parent's genes in their chromosome. In travelling salesman problem, crossover is quite different from the other because there cannot have duplicate gene/city in one chromosome and the order of the cities play an important role in the chromosome. Crossover is important because it enables the algorithm to generated new offspring, hence the chances of getting back the same route in for the solution will be reduced.

iv) Mutation phase (Ahn and Ramakrishna, 2002)**:**

In nature, mutation is the result of errors in the change of DNA. Most of the change is negative but some may have a positive impact on the individual. In Genetic Algorithm the diversity of the population must be maintained, therefore the final solution may not easily get trap in the local maxima. In other words, chances to get the global maxima point will be increased. In genetic algorithm, the probability for mutation to happen is very low, this is because if the probability of mutation is high then the final result may not be that stable in terms of reliability due to the solution provided by the system may be worst. In conclusion, mutation with low probability may benefit this algorithm, while mutation with high probability may have a high chance to convert the good solution into the bad one.

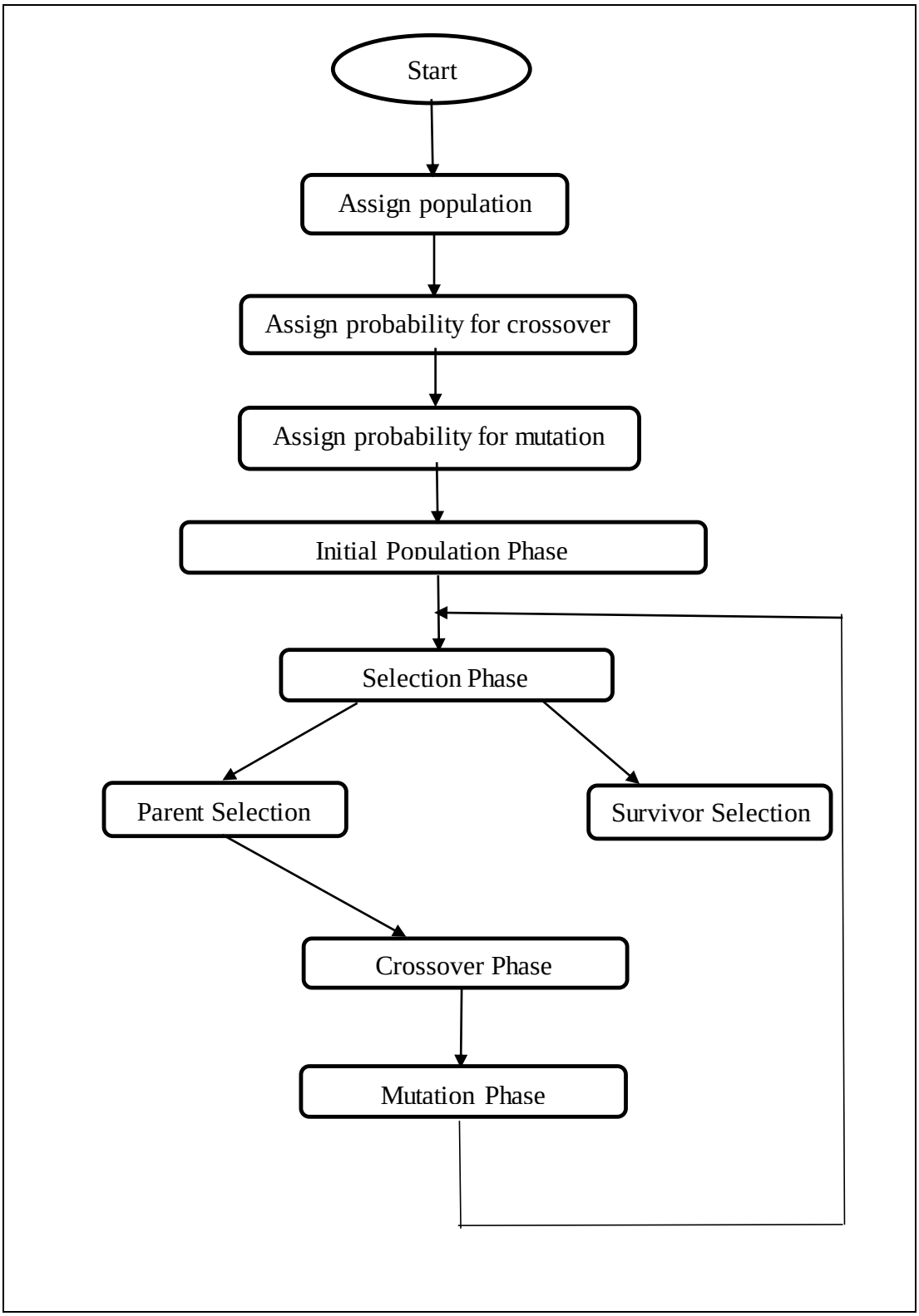


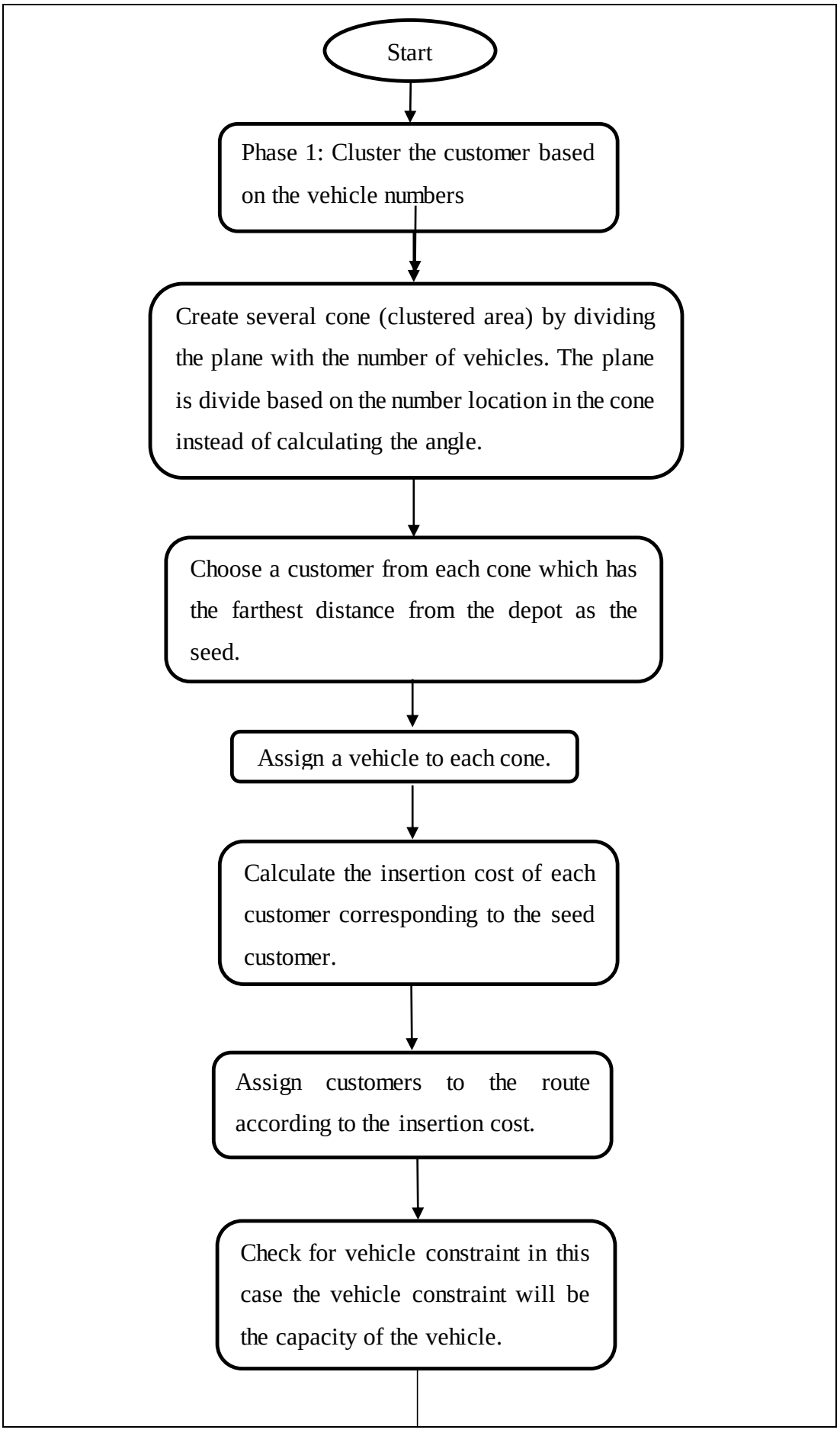
Figure 2.26 Overview of Genetic Algorithm. (Ahn and Ramakrishna, 2002)

2.3.2 Capacitated Vehicle Routing Problem (CVRP)

Vehicle routing problem also known as truck dispatching problem is a variant of travelling salesman problem. In vehicle routing problem we aim to design an optimal solution of vehicle routes to deliver cargoes or services to the customer with a given number of vehicles (Carwalo, Thankappan and Patil, 2017). Same as travelling salesman problem, vehicle routing problem is not a new problem in transportation, distribution and logistics. The aims of solving vehicle routing problem are to minimize the total travelling cost and at the same time maintaining some constraints like vehicle capacity, total travel times, time windows of customers and so on (Carwalo, Thankappan and Patil, 2017). Capacitated Vehicle Routing Problem is to find a set of optimal routes with the constraints of vehicle capacity. There are two popular approaches that have been found by researchers which are constructive approach and clustering approach. The algorithm that is going to discuss is a variant of Fisher and Jaikuamr Algorithm which is a type of clustering approach.

Variant Fisher and Jaikuamr (VFJ) Algorithm (Carwalo, Thankappan and Patil, 2017)

Carwalo, Thankappan and Patil (2017) stated that most of the clustering approach including this algorithm consists of two phases, the first phase will be generating a wise clustering method based on number and vehicles and the second phase will be a travelling salesman problem that can be solved by any TSP optimizing algorithm. One of the optimizing algorithm had already been discussed hence only the first part of the VFJ algorithm will be discussed in this section.


Start
Phase 1: Cluster the customer based on the vehicle numbers
Create several cone (clustered area) by dividing the plane with the number of vehicles. The plane is divide based on the number location in the cone instead of calculating the angle.
Choose a customer from each cone which has the farthest distance from the depot as the seed.
Assign a vehicle to each cone.
Calculate the insertion cost of each customer corresponding to the seed customer.
Assign customers to the route according to the insertion cost.
Check for vehicle constraint in this case the vehicle constraint will be the capacity of the vehicle.

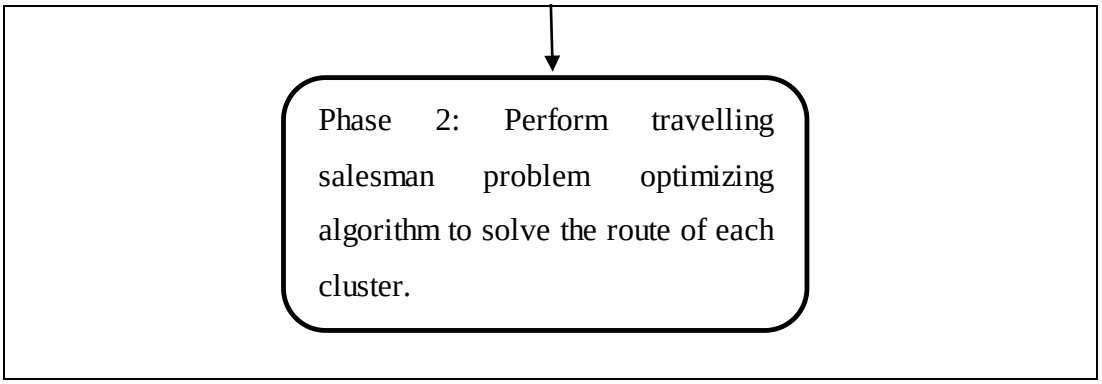


Figure 2.27 Overview of Variant Fisher and Jaikuamr (VFJ) Algorithm.
(Carwalo, Thankappan and Patil, 2017)

2.3.3 Container Loading Problem

The Container Loading Problem (CLP) is one of the variants of the Bin Packing Problem which focus on implementing a 3D optimization algorithm to pack a set of rectangular boxes into a container, in such a way that some optimization constraint has been satisfied (Martínez, Alvarez-Valdes and Parreño, 2015). The aim of solving this problem is to maximize the available space in the truck when cargoes is loaded into the truck. There will be at least two factors needed to be considered in this CLP. One of them is the boxes orientation. There may have exact three orientations that the boxes can be arranged. The second factor will be the volume of the container and cargoes. We may need to make sure that the total volume of the cargoes does not exceed the container volume in the algorithm. Martínez, Alvarez-Valdes and Parreño (2015) stated that CLP is not a new problem and may need to solve by any supply chain on their daily basis. For example, varying types of cargoes to be delivered based on the FILO sequence, number of trucks to be used and more. To solve this problem, research is perform to find for existing algorithm which is able to minimize the empty spaces in the truck. A GRASP algorithm which is a constructive approach will be discussed in this section and the overview of the algorithm had been stated below.

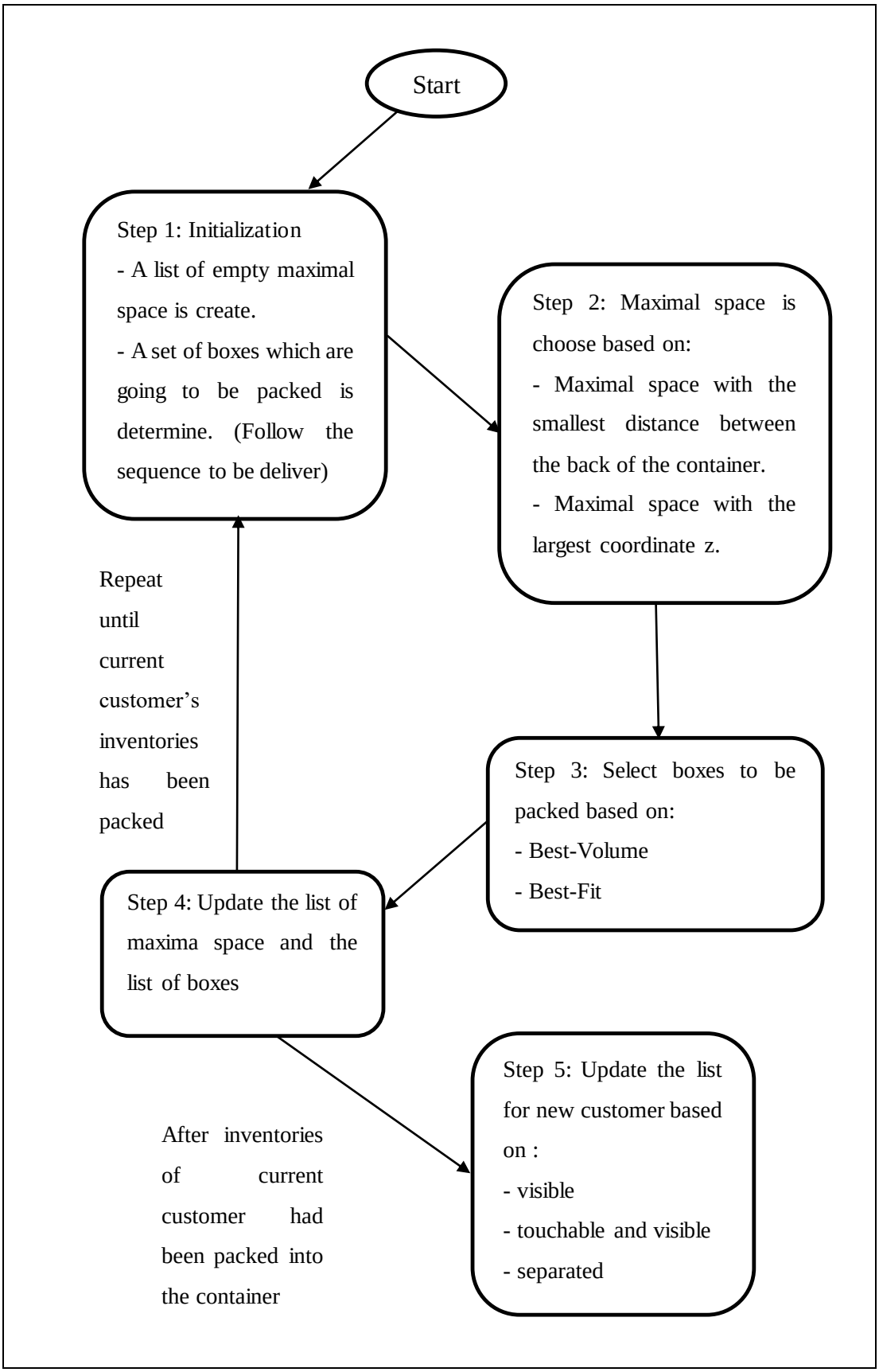


Figure 2.28 Overview of GRASP Algorithm. (Martínez, Alvarez-Valdes and Parreño, 2015)

2.3.4 Hierarchical Clustering

Clements (2019) stated that clustering is used to find similarity on the data and then cluster them by grouping those data which have a similar characteristic with each other. Clements (2019) also mentioned that hierarchical clustering is an alternative algorithm which is able to produce 1 to n set of clusters, n clusters mean data all data serve as their own cluster. There is two types of hierarchical clustering which are Divisive clustering and Agglomerative clustering.

2.3.4.1 Divisive Clustering (Clements, 2019):

Divisive clustering is a top-down approach which starts by treating the entire data as one cluster. The process then continues by reassigning a new group to the data which has a high value of dissimilarity to the cluster. This process will repeats until the number of clusters which is needed had been met.

2.3.4.2 Agglomerative Clustering (Clements. 2019):

Different from divisive clustering, agglomerative clustering is a bottom-up approach which starts by treating each particular data as a cluster. Which means that if the system contain of 10000 particular data then the cluster that the system has at the beginning will be 10000. The process then begins by calculating the similarity between each point of data. Two unique data which has the least value of dissimilarity will then be group together. This process repeats until the number of the clusters reduces to the number of clusters that had been set at the beginning.

In this project, Agglomerative clustering will be used to group the locations to deliver based on the number of trucks needed. This mean that each unique address will be threatened as one cluster, the clustering process will then start by grouping two locations (clusters) which have the smaller dissimilarity value. In this case, the metric that used to measure similarity and dissimilarity will be the distance between each location. The clustering process then repeats until the number of clusters is equal to the number of trucks/cargoes that selected by the user. The reason agglomerative clustering is needed in this project is due to the number of the truck that the company owned and wish to use at the particular day must be considered. In order to have a better planning on the route without overlapping areas that responsible by a particular truck, then clustering method may be needed to apply in the algorithm.

2.4 Software Development Methodology

Software development life cycle (SDLC) consists of several standard procedures which is planning, analysis, designing, implementation and testing. Geambasu, et al. (2011) stated that the structure of the software development life cycle plays an important role in determining the success rate of a project and at the same time he also mentioned that the success rate of a project can be increased by selecting an appropriate software development methodologies. According to CHAOS (2014), they report that there may have overall 16.2% of IT project was successfully be delivered since 1985, 52.7% of them was late to deliver, over budget, and incomplete of features and 31.1% of the project has been cancelled. Therefore appropriate methodology can assure that the project is able to complete within schedule, low cost is able to maintain and meet the requirement of target users.

There may have a wide range of methodologies can be used by developer. In this literature review, only several methodologies will be selected for further discussion and comparison. They are Waterfall, V-Model, Prototyping and Agile Methodology.

2.4.1 Waterfall Methodology

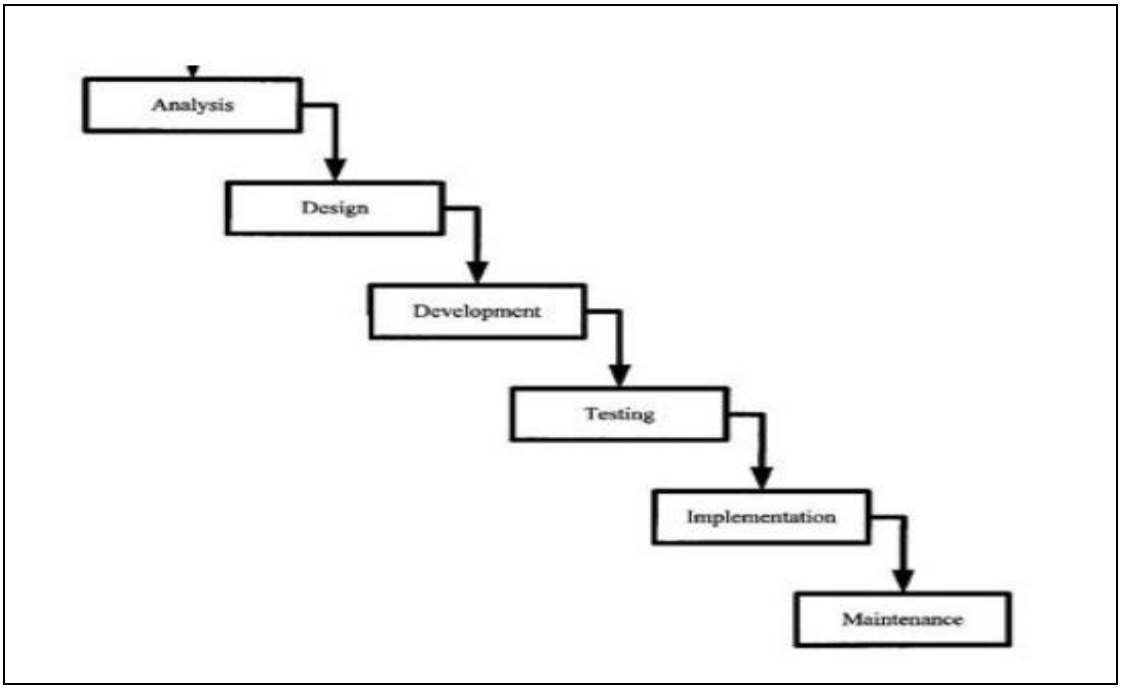


Figure 2.29 Waterfall Model Life Cycle (Balaji and Murugaiyan, 2012)

Waterfall is a traditional methodology that focuses on cascade procedures (Balaji and Murugaiyan, 2012). This means that each phase of the SDLC has to be finished before moving to the next phase. For example, the requirements of the project must be fully collected before continue to the design stage. The condition is the same for every phase in the SDLC. In other words, each phase of SDLC moving in the order without any interference. The advantages of waterfall methodology are developers are able to understand the requirements clearly before proceeding to the development (Balaji and Murugaiyan, 2012). Therefore, the system developed will meet the requirement of the user at the end of the project. But the disadvantages are, each phase may take a long period to process and this actually causes a frozen in the whole development process (Balaji and Murugaiyan, 2012). This is very time consuming and high cost if there are any issues unable to solve at the particular phase. According to Balaji and Murugaiyan (2012), waterfall is a unidirectional process and doesn't allow any reverse process to take part hence the project is not able to make any changes at this current development process if the client wishes to change certain requirements.

Based on the above review on the waterfall methodology, this methodology is not suitable for this project hence waterfall methodology will not be adapted in this project. As mentioned, waterfall model is an irreversible model, due to this characteristic it may not well perform in this project due to there may have many uncertainty factors when researching and implementing algorithm. Besides, an interactive and simple design of user interfaces may play an important role in this project. The system should be designed in the way that user can have all the visualization view on one screen instead of keep switching between different tabs or pages. In this case, methodologies which produce a partial workable prototype or system and methodology which allow several turns of iterations is more suitable because there has at least a product to show the user in order to get their feedback hence improvement on the algorithm and user interfaces can be done at the next iteration. In addition, unclarified requirement and miscommunication issues can also be solved easily.

In conclusion, waterfall methodology is not suitable due to requirements that are not well-known and clear and there may also have possibility in changing of certain requirements. Waterfall development may only bring benefits to the project which has

a clear requirement at the beginning and project whose requirements are not constantly changes.

2.4.2 V-Model

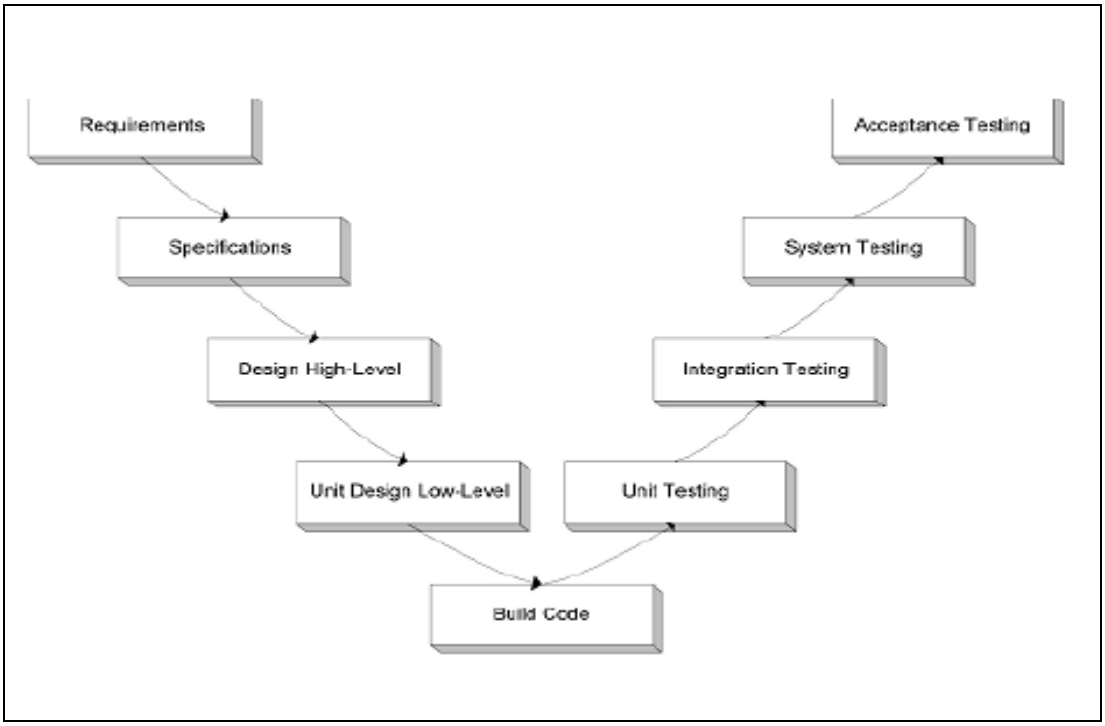


Figure 2.30 V-Model Life Cycle (Balaji and Murugaiyan, 2012)

V-model also known as validation & verification model is a variant of waterfall methodology and has a v shape model which is able to fold symmetrically. Balaji and Murugaiyan (2012) stated that the left leg of the v-model represents the verification phase of the development process and there is a decomposition of analysis and design phase starting from requirement analysis, system design, architecture design and then module design. The right leg of the v-model represents the validation phase which consists of unit testing, integration testing, system testing and user acceptance testing (Balaji and Murugaiyan, 2012). Each phase in the validation phase has a corresponding stage in the verification phase through the horizontal axis. Which means that the product of verification phase needs to be check by the corresponding stage in the validation phase before moving forward (Balaji and Murugaiyan, 2012).

In v-model, developers and testers are must work in parallel (Balaji and Murugaiyan, 2012). For example, a test case which corresponding to a particular

design in verification phase can be prepared before the testing process starts. Once coding is done, the testing process can be started without any delay due to the test case has already been prepared. The advantages of V-model is same as waterfall methodology which are able to understand user requirements before the development begins, remain a high quality of documentation and easy to implement (Balaji and Murugaiyan, 2012). While the disadvantages of v-model are this model is very rigid and least flexible (Balaji and Murugaiyan, 2012). Any changes in the user requirements may lead to failure of any part of the designs and test cases hence design and test case must always be updated once there is a change in the requirement.

Due to the rigid design of this model, this methodology will not be considered in this project although it is able to make modifications during the process of software development life cycle, but it may need a huge amount of time to change all other related deliverables including the test case. This project may have several uncertainties on the factors for the algorithm and may result in different outputs of the system, therefore if the test case is too early to be developed then it may need to change those test cases frequently. Developer may end up lack of time for this project due to they may need to change other parts of deliverables instead of just modifying the ones that got affected. In conclusion, V-Model is not suitable for those project which has a short schedule and high uncertainty on requirements and factors.

2.4.3 Prototyping Methodology

Prototyping is known as activities which involve the process of implementing a trial version of the system that is able to let users or developers to view the overall requirements and design of the system (Gordon and Bieman, 1995). There may be two methodologies in prototyping which are throw-away and evolutionary. Throw-away prototyping is used to check the satisfaction of user's requirement and reduce the risk by confirming certain issues before the system is built. In throw-away prototyping, the prototype may be discarded once those issues have been resolve (Gordon and Bieman, 1995). While this is difference in evolutionary prototyping, prototype which is defined at the beginning of the SDLC will continue adding new features, evolve and retain until the final product (Gordon and Bieman, 1995). The advantages of prototyping are it can actually overcome certain vulnerabilities of the traditional methodology. For example weaknesses like difficult to revisit the previous phase, a key deliverable for

each phase may require a long time period, unclarified user's requirement, miscommunication between users and developers and dissatisfaction of user interface may be easily solved by using prototyping methodology (Carter, et al., 2001).

In prototyping methodology, there may have several turns of iteration until users are satisfied with the prototype. In each iteration of the SDLC, developers are able to get feedback from the users regarding the design of the user interface and also the satisfaction of the user's requirement. Feedback from the users helps to clarify the requirements which are previously unclear. Feedback from the users also proves that the involvement of users in SDLC has been improved. In addition, prototyping also helps developer to develop a better quality and better user-interface system. As long as all the requirements have been clarified and user is satisfied with it, developer can proceed to the next step which is implementation of the real system.

In this project, evolutionary prototype model is going to adapt as my development methodology due to the characteristics of prototyping is more suitable for this project. As we may not get the full detail of requirements from the client, this may due to miscommunication or misunderstanding and sometime we may never notice that there are some requirements which are still unclarified until we have started to implement something hence prototyping model which provide several iterations will be helpful as we can trial an error on the prototype and get the feedback from client. In this project, an algorithm is going to implement in order to find the optimal route to deliver and at the same time maximize the available space in the truck. In order for the developed system to perform well without having an error on the algorithm, feedback from the user may be important in order to clarify those factors which had not be found yet. After reviewing on existing system we found that an interactive and a simple design user interface may need to develop to let user have a better experience and have the feeling of 'ease to use'. Prototyping model may be the best choice for improving user interface of this project by getting user feedback. Last, the reason this project using evolutionary prototyping instead of throw-away prototyping is due to reuse and integration of the existing prototype until the final product is developed, and this can minimal the overall time and cost compare to throw away those existing prototype and develop a separately new system.

2.4.4 Agile Methodology

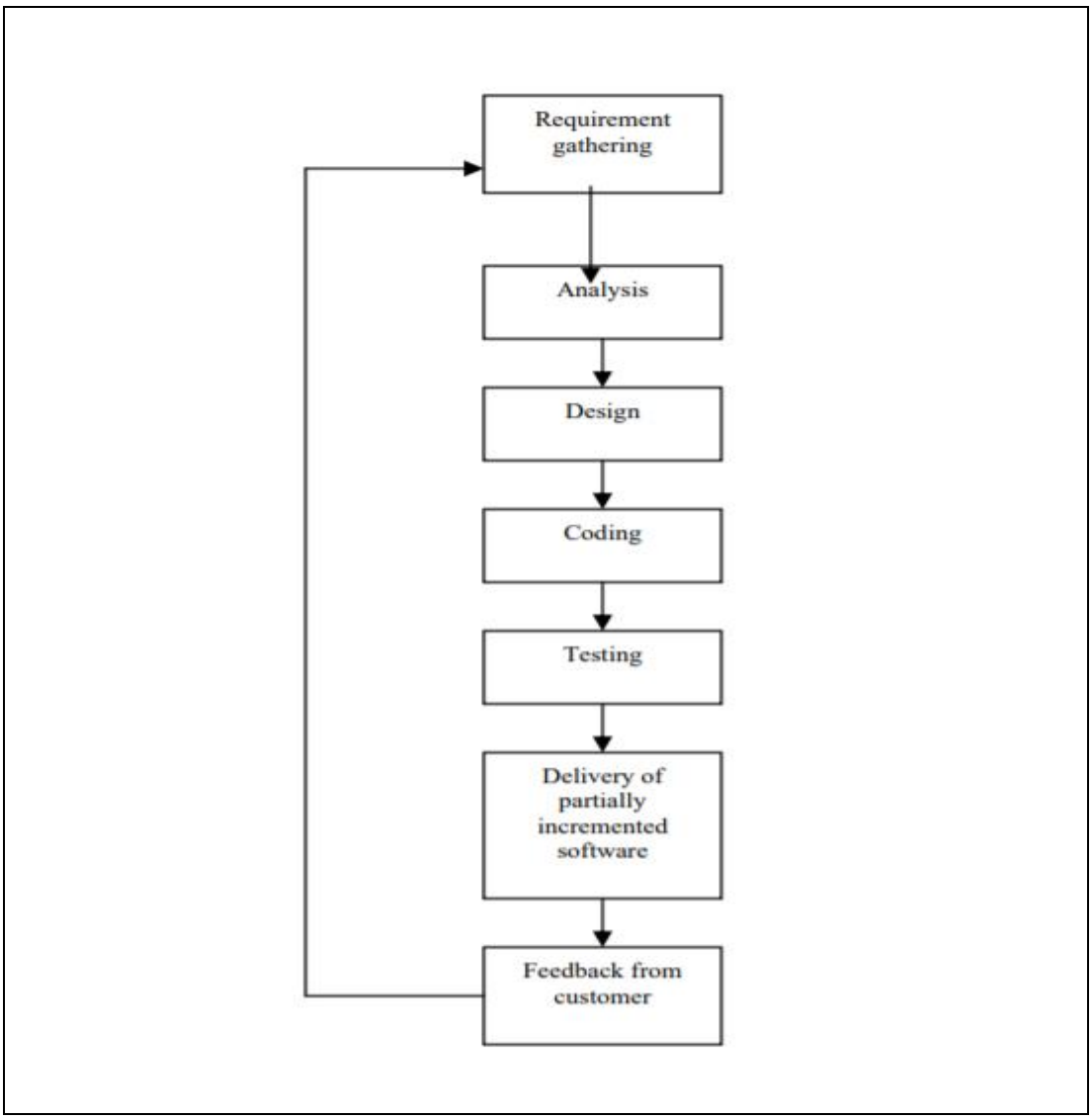


Figure 2.31 Agile Life Cycle (Sharma, Sarkar and Gupta, 2012)

The concept of agile methodology is breaking the project into a smaller sub-project therefore the process of the SDLC can be speed up. Similar to prototyping methodology, there may have several iterations of cycle until the final product is deliverables. In agile model, there may have small incremental release in each iteration and those incremental feature is building on the previous functionality. In agile methodology, each phase of software development life cycle can easily be revisited until full-filled user requirements. In order word, agile methodology is very flexible and is able to deal with the project which has requirements that constantly change. The advantages of this model are it has the ability to respond to the changing requirements

of the project, able to speed up the overall software development life cycle and able to meet customer's satisfaction by providing rapid and continuous delivery of small useful software (Balaji and Murugaiyan, 2012). The only disadvantage of agile model is it is not suitable for a large or complex project, because it is hard to the effort and duration required due to the project may break down into too many sub-projects (Balaji and Murugaiyan, 2012). There have several popular methodologies derived from agile model, they are DSDM, Scrum, XP, ASD, Crystal, and FDD.

After going through the review of the agile methodology, we found out that this methodology has some similar properties with the evolutionary prototyping methodology as both of them are going to undergo several iterations which require to gather the feedback of the system from their client in order to clarify the user requirements. The only difference is one of them is developing a prototype while the other is developing a real system but in some cases prototype can also be considered as the first version of the system. The reasons that agile methodology is not selected in this project are agile methodology may need developer to work closely and have a constant communication with the client therefore they are able to change their system immediately when there is a requirement change. In this project, we may not constantly meet with the client or targeted user therefore prototype which is able to present the overall workflow will be more suitable. In agile methodology there may have less documentation compare to other methodology, therefore it may not suitable for a non-experience developer to undergo this methodology without the guild of an expert. Due to lack of documentation, new and unexperienced, developer may felt hard to understand the whole project in terms of objectives, scopes, system design and business domain.

2.4.5 Comparison of Software Development Methodologies

Table 2.3 Comparison on the factors of different Software Development Methodology

Factors for comparison	**Waterfall** (Balaji and Murugaiyan, 2012)	**V-Model** (Balaji and Murugaiyan, 2012)	**Prototyping** (Gordon and Bieman, 1995), (Carter, et al., 2001)	**Agile** (Balaji and Murugaiyan, 2012)
Unclear user requirements	**Poor.** Phases in waterfall methodology is irreversible hence it is hard to clarify the requirements again after proceeding into design and implementation phase.	**Poor.** This may due to the rigid design of this methodology. Although the requirements can still be modified or update but changes in the requirements may need to change all other related deliverables and their test case.	**Good.** May have several iterations in this model. Trial an error strategy can be done on the prototype in order to get feedback from the client. After getting feedback developer can have a better understanding on the requirement and an improvement of the system can be done.	**Good.** Same with prototyping there may have many iteration for developer to redefine the requirement to user again until all the requirement is fully clarified.

System complexity	**Good.** Each phases of the waterfall methodology is long hence developer have more time to deal with a big and complex project.	**Poor.** Products from each verification phase is linked with the validation phase through horizontal axis. Changes in the requirements may need to update all relevant design/ documentation and test case. Many works is going to be perform therefore is more suitable to a small or medium project.	**Poor for evolutionary prototyping but good for throwaway prototyping.** By using evolutionary prototyping there may have chances that the problematic logic forget to remove from the prototype and may cause logic error in the future if the features or functions continue to add on while in throwaway methodology prototype is only used to clarify the requirement none of the prototype is included in the real system. Fortunately, this project is consider small or medium therefore evolutionary	**Poor.** Agile methodologies emphasis interaction with the client and therefore a complex and larger project may involve of many different group of people it is impossible for development team to deal with all of them.

			prototyping is still appropriate in this project.	
Short Time Schedule	**Poor.** Each phase is conducting of many activities in order to meet the objectives of each phase hence each phase may take a long period of time and it is not suitable for project which only have a short time schedule.	**Poor.** Each stages in the model may spent a lot of time in reviews.	**Good.** Does not spent more time on gathering all the requirements at the beginning and more willing to trial and error on the prototype in order to get feedbacks from the user. With prototype user may have a better understanding on what he actually want and developer can update the change immediately.	**Good.** Same as prototyping, spent less time on clarified all the requirements at the beginning. Going through several short iteration with client in order to clarify the requirement and at the same time update the system after the requirement has been clarified.

Involvement of user	**Low.** Only at the beginning of the phase. In the analysis phase developer may perform some activities in order to clarify the requirements before proceeding into design or construction phase.	**Average.** At the beginning of phase which is requirement analysis phase and user acceptance testing phase.	**High.** There may have several iterations in this methodology. In each iterations developer team may need to get feedback from the user on the prototype therefore developer will work closely with the client.	**High.** In order to speed up the process of the overall software development life cycle, developers are more willing to constantly communicate with the clients in order to clarify the requirements before he/she continue in adding any new features or functionalities.
Incorporation of requirements changes	**Poor.** Waterfall methodology is irreversible and each phase may take a long period to meet its objectives. The product will only be shown to the user at the end of the SDLC cycle and there is	**Average.** Update is available in this methodology at any time but all the related documents, designs and also test cases must also be update.	**Good.** There may have several iterations in this methodology if the requirements has been changes then developer are able to know when they show their prototype to the user.	**Good.** In agile methodology developer is work closely with the client therefore if there is any changes in the requirements client can easily inform those developer and

	possible that the requirements has been out of date and developer may need to redo it again to meet the new requirements.			developer can changes the requirements immediately.
With unfamiliar technology	**Good.** There may have sufficient time for developer to learn a new languages due to training will also be consider as an activities to accomplish the objectives of the project.	**Good.** There may not have any iterations in this methodology therefore no need to deliver any prototype or first version of the system to client. Thus time provided is sufficient to learn new technology.	**Poor.** Each iterations in this methodology is short and developer may lack of time to master a related technology.	**Poor.** Same as prototyping methodology, each iteration is short and may not have time to learn for a new technology.

2.5 Research Methodology

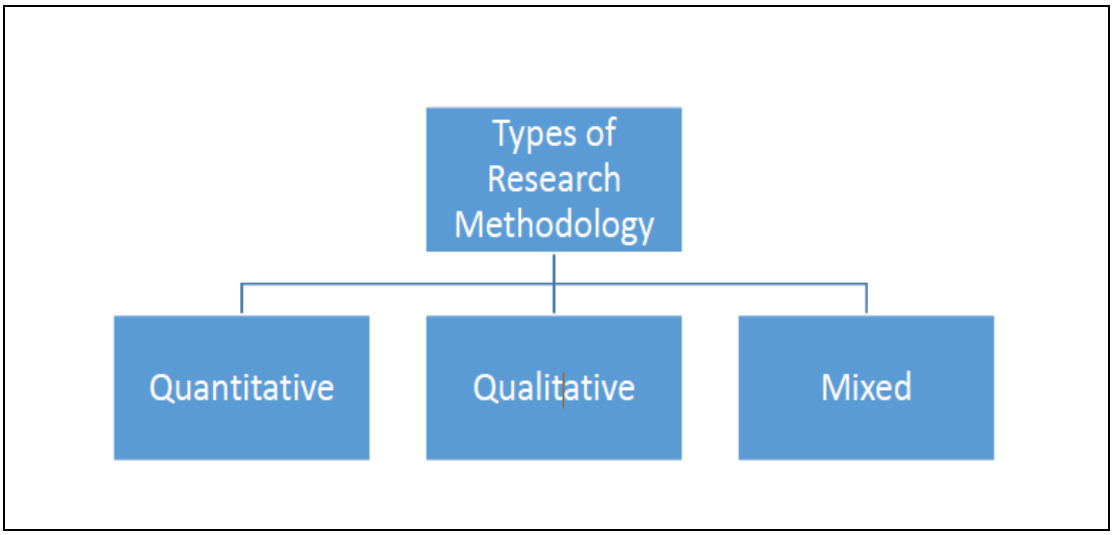


Figure 2.32 Types of Research Methodology

Research is a process of searching for information and knowledge which are able to prove the assumption that had been made for a particular project (Grover, 2015). When performing research researchers may need to go through several phases which are data collection, data analysis and data interpretation in order to gain a better understanding of the related phenomena (Grover, 2015). In order words, research is a method that collects information from a size of samples. There may have three types of research methodologies which are, quantitative methodology, qualitative methodology and mixed methodology. Different types of project may require different types of research approaches. This means that which research methodology selected is highly dependent on the project's objective and also the data that the system wish to collect.

The objective of performing qualitative approach is aims to have a deep understanding toward a given phenomenon. Taylor & Bogdan (1984 cited in Prikladnicki, et al., 2008, p.3) stated that "qualitative methodology is widely used by educational researcher and other social scientist to investigate the complexities of human behavior". Which mean that if the objective of the project has been to evaluate a more specific and complex processes or individual's strategies on making a certain decision like how an expert manage their company, what is the steps or procedures to complete certain tasks, perhaps the qualitative methodology will be more suitable

compare to quantitative. The advantages of qualitative are it offers a complete description and analysis of that particular research project and the data collected will not be loss as data is not quantified (Langkos, 2014). Based on Langkos (2014) declaration, there may have several limitations when conducting a qualitative approach which are the data collected cannot represent the opinion of wider population due to the sample size was relatively small hence data collected may also not reliable and may be biased toward the participants, the result of this approach is heavily based on researchers'' skills and abilities, and in some cases participant may be refused to speak against their experience. Examples of data collection methods in qualitative methodology are interview, literature review, observation, diaries, case study and ethnography.

The difference between quantitative and qualitative approach is quantitative approach try to convert data collected into a statistical model in an attempt to prove the assumptions that had been made or explain in a scientific way which consisting of figures, numbers and several charts. This method aims to answer the phenomena about 'how many' or 'how much' rather than 'how', why' or 'what' (McCusker and Gunaydin, 2015). In quantitative methodology, researcher may need to know what the data he wishes to collect and then plan and design for it in order to collect useful and quality data (McCusker and Gunaydin, 2015). In this method most of the data will be presented in the form of graphs, charts or any statistical models therefore researcher can have an overall picture of the phenomena and conclusions can be drawn easily. The advantages of quantitative are data collected may not biased toward certain opinions due to the sample size of the participant is big and the questions asked will be more generalized (McCusker and Gunaydin, 2015). The time taken for quantitative methodology is fast compare to qualitative methodology. However, this approach does not provide a deep explanation toward the data collected and is not suitable to cover those issues which require a better understanding. Questionnaire is the tool used by researcher to gather information in quantitative methodology.

Mixed methodology referring to the integration of quantitative and qualitative data in a particular project (Agency for Healthcare Research and Quality, 2013). The main focus of this methodology is the utilization of two different kinds of data in order to produce a more reliable and good quality data. The advantages of performing this

methodology are researchers are able to validate their finding by comparing both quantitative and qualitative data (Agency for Healthcare Research and Quality, 2013), this gives researcher more understanding on the relationship between both data sources. Besides, qualitative data can use to explore quantitative findings (Agency for Healthcare Research and Quality, 2013). Which means that quantitative data can have a better explanation when using qualitative data as supporting details. Agency for Healthcare Research and Quality (2013) mentioned that mixed methodology also provides flexibility toward researchers and study. Researcher can have an alternative choices when performing data collection and at the same time this method is able to adapt many different kinds of researches and studies. However Agency for Healthcare Research and Quality (2013) also stated that this methodology increases the complexity of evaluation, researcher may need to plan carefully to avoid contradiction of data during data collection and analysis phase. Mixed methodology may require greater resources and time to conduct a study.

In this project, the research approach which is going to adapt in this project is qualitative methodology. The reason qualitative approach is select is because of the system that is going to develop in this project consist of finding the shortest route to deliver and at the same time optimize the cargo space of the particular truck. Those features may need an algorithm to search for the optimal result, therefore a better understanding on how the manager of those cargo delivery related industry to plan for their route and arrange the placement of cargoes in the truck must be gathered before starting to implement the algorithm by reference on those existing algorithm. Besides, by using qualitative approach the factors that may be needed to consider when implementing the algorithm in order to find the optimal route and maximize the cargo space is able to identify. Quantitative approach may not suitable for this project due to the algorithm which is going to implement may need a detail of the industry overall process on route planning and cargo loading in order to solve the situation mentioned above. From the review, Langkos (2014) stated that the result of qualitative approach like interview may be biased toward a small range of population however this property is just suited to this project due to this route optimization system is mainly developed for those users who involve in the cargo delivery related industry. As interviewer can find someone who has experience in performing those tasks then their opinion and suggestion may be very useful in this project.

In conclusion, the information which is going to collect is basically how those industries operates their daily business and this is hard to convert into a statistical model. Another reason that quantitative methodology is not conducted in this project is because quantitative methodology may need a population to support the topic of study, and it is hard to find those experience people who have involved in planning the route to deliver and at the same time involving in loading the inventories into the truck. Therefore, the only research approach which is suitable for this project is qualitative methodology.

2.5.1 Interview

An interview is a conversation or discussion for gathering information. According to Easwaramoorthy and Zarinpoush (2011), participants in the interview section are interviewer (people who conduct the interview section and play the role of asking questions and writing down the appropriate answer for the particular questions) and interviewee (people who respond and provide answer to the interviewer). There may have a different form of interview been widely used by other researchers which are face-to-face interview, phone-call interview and some of them conduct the interview through internet. Easwaramoorthy and Zarinpoush (2011) stated that face-to-face interview will be the most suitable ones when comparing to other forms of interviews due to the interviewer can get a more detail and specific answer from the respondent.

Besides, the interviewer is able to clarify those answers that provided by the respondent immediately hence the chances of getting misunderstood answer will be reduced (Easwaramoorthy and Zarinpoush, 2011). Interview will only apply when there is a need for the project to collect in-depth information. For example, people's thoughts, feelings, behaviours and experiences. In this project, experience of the manager regarding how they plan the route to deliver and how they try to optimize the capacity of the truck will be the areas that this system needed to focus on. The advantages of interview are the outcome of interview consist of richest and details data, provide opportunity to explore the topics in-depth and able to collect a more quality and useful data due to interviewer will help to clarify the questions of interview to the respondent (Frechtling, et al., 2002). While the disadvantages of interview are it is time-consuming, need a well-trained interviewers, it is hard for interviewer to translate topics regarding the experience, behaviour or feeling of people into words (Frechtling,

et al., 2002). Interview questions can be designed in different ways depend on the data the project wishes to collect. There are three types of interviews design which are structured interview, semi-structured interview and unstructured interview (Easwaramoorthy and Zarinpoush, 2011).

Structured Interview (Easwaramoorthy and Zarinpoush, 2011)**:**
In structured interview, all of the questions being to ask is standard, predetermined and must follow the sequence of the order. Respondent may only need to select the answer which is able to fit with its current situation from a predefined list of options.

Semi-structured Interview (Easwaramoorthy and Zarinpoush, 2011)**:**
In this interview, interviewers may seek to have an in-depth information from the respondent, therefore the questions of this interview may design in such a way which encourage respondent to answer in their own words. The advantages of having semi-structure interview is interviewer is able to ask for clarification or explanation based on the answers given by the respondents or interviewer can ask for a supplementary questions to clarify the answer given.

Unstructured Interview (Easwaramoorthy and Zarinpoush, 2011)**:**
Unstructured interview does not follow any specific guidelines, predetermined questions and a list of options. In this interview, interviewer will ask for several questions which are general and broad and encourage respondents to provide an answer in an open, informal and spontaneous discussion. Similar to semi-structure interview, interviewer is able to ask for clarification by asking further questions which are more specific to the particular topic. This design of interview may be helpful when interviewer wants to know more about the respondents' experience in a particular fields and is also suitable when the information regarding the topic is little. The disadvantage of performing this type of interview is this interview is heavily relied on interviewer skills.

To make sure that the algorithm that going to implement is able to solve issues efficiently, a better understanding on the on the overall process of how the route was planned to deliver and how those cargoes were loaded into the truck are required.

After reviewing on different types of interview's design, the one which is most suitable to this project will be semi-structured interview due to rich information can be gathered from the interviewee. Therefore an in-depth interview may be preferred in order to gather requirements or to declare the problem they faced. Structured interview may lack of flexibility and the responded answer is based on the options given to the interviewee hence the answer that can be gathered is limited and fixed. To gather a rich and quality information, interviewer may want to encourage respondents to answer in their own words and make sure that the answer can clearly describe his experience and those processes. Semi-structure and unstructured interview is able to gather this kind of information well, but unstructured interview is not chosen to be adapted in this project due to unstructured interview is heavily depends on the interview skills. We are inexperienced in gathering information, therefore semi-structure which has a proper guideline, predetermined questions and supplementary questions is able to help interviewer to have a better control during interview section. In conclusion, a semi-structured interview will be conducted with manager of Hondail Trading Sdn Bhd to gather for related information.

CHAPTER 3

METHODOLOGY AND WORK PLAN

3.1 Introduction

Section 3.2 will discuss the details of each phase of the software development methodology that will be adapted in this project after comparing the advantages and disadvantages. Section 3.3 will introduce the analysis of the research method that had been used in this project corresponding to research methodology which had been chosen. Last, development tools, work breakdown structure and gantt chart will be states in section 3.3.

3.2 Software Development Methodology

A summary is able to make after reviewing on four different types of software development methodologies, which is different types of methodology is suitable for different kind of projects which depends on the project's characteristics. By adapting an appropriate methodology developer can assure that the project is able to complete within schedule and the requirements can also be met at the end of the project. Evolutionary prototyping is going to adapt to this project due to some requirements that may still remain unclarified until the implementation process had started. Evolutionary prototype is able to clarify the requirement by frequently deliver the prototype to the targeted user. Then the next version of prototype will be generated after getting feedback from the targeted user. The main features of this project are finding the shortest route and optimize the capacity of the truck therefore evolutionary prototype is required to get feedback from the clients, therefore unclarified factors or requirements can be added into the algorithm to make sure that the algorithm is performed in the right way. Besides a simple and interactive user interface may be needed to let user have a better experience and easy to use. Therefore prototyping may be the most suitable development model for this project. The overall phases in evolutionary prototype model are stated in Figure 3.2.1.

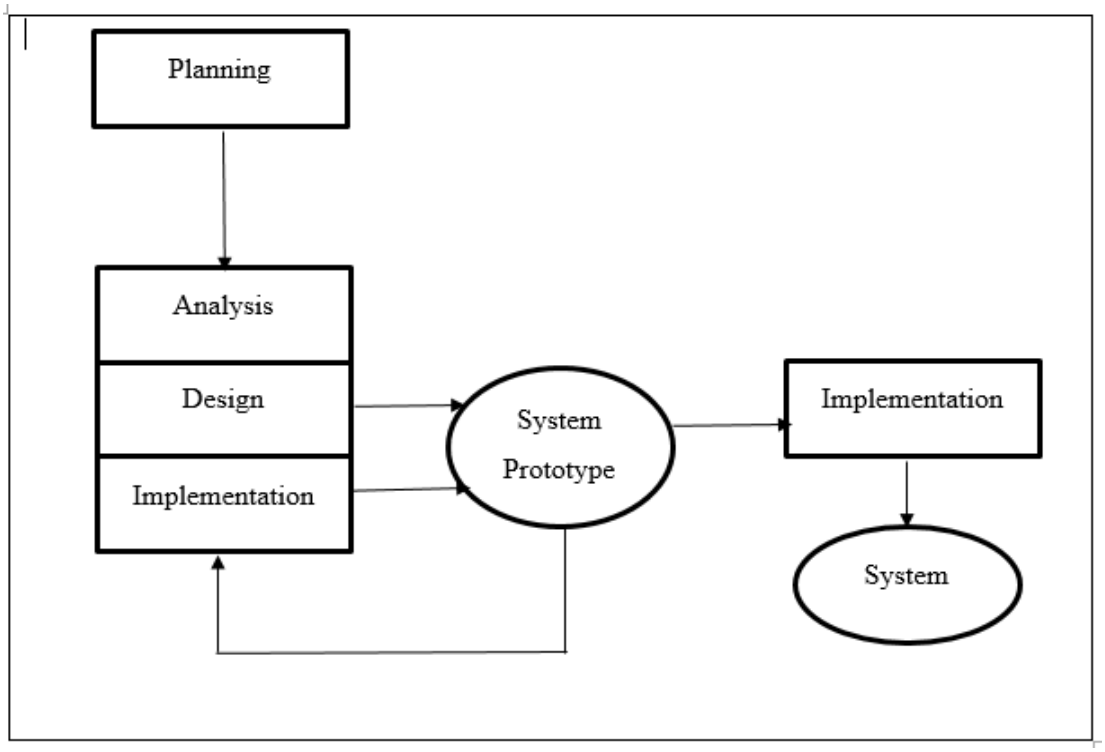


Figure 3.1 Evolutionary Prototyping Methodology (Mak, Hafit, Kasim, Md Fudzee et al., 2017)

3.2.1 Planning Phase

During this phase, the problem statement of this web-based route optimization system was determined then follow by objectives, scopes, proposed solutions and proposed approach. The problem statements of this project are company fail to utilize the resources in term of fuel cost and travelling cost due to the route planned by managers is based on their experience and may have the chance of selecting a high cost route. Another problem statement is inventories are unable to fit into the truck due to the truck is not fully maximize.

By reviewing on those problem statements, the objectives of this project were declared to ensure that the system that is going to develop is able to solve the issues that were faced by those logistic company as stated above. The main objectives of this project are to find an optimal route to deliver cargoes to their customer and at the same time maximize the available space in the truck therefore logistic company can fully utilize their resources. The targeted user for this route optimization system will be those logistic company, managers, drivers and workers who have involving in company daily operations likes planning the route to deliver and loading the cargoes

to the truck. A project with clear objectives and appropriate target users may help us to understand about the problem statements and also able to increase the success rate of the project.

Project scopes, proposed solution and proposed approach for this project were developed after the problem statements and objectives have been clarified. Project scopes were determined to clarify the project's requirements in order to achieve the project's objectives. The proposed solution is also determined to have an overview of how the system actually solves the problem faced by the user. Then proposed approach like research approach and development methodology that can be adapt in this project was declared hence this project can stick with those methodologies to gather relevant data which is able to enhance this system and have a better guild during system implementation.

After project objectives had been determined, a work plan which includes the work breakdown structure, gantt chart and milestones of the project is developed. Those plans are created at the beginning of this project to keep track of the progress of the system and also get a better understanding on the overview of the processes/activities that needed to perform in order to achieve the objectives of this project. By having a work plan, management and control on the time, quality, changes and risk of this project can be handled in a more effective way. Project with a well-developed plan can be used as a guideline for development team to complete the project within the due date and at the same time maintaining the cost. By having a work plan, developer is able to know which activities have a high weightage comparing to others. Activities with high priority may be needed to be completed first then follow by other sub-activities.

3.2.2 Requirements Gathering Phase

In this phase, requirements of this project were gathered through reviewed on existing related application, performed research on related existing research and its algorithm, conducting an interview with manager who role is to plan a route for delivery and also involve in supervision drivers or workers in arranging and loading the inventories into the truck.

Reviewed on existing related application is performed by searching, observing, accessing on several online systems which have a certain similarity with this project. Besides, testing or walkthrough of those existing systems had also been performed in this phase. This is to have a better understanding of how those features or functionalities actually work therefore relevant features can consider being implemented in this project. Meanwhile, by observing those similar systems we can get a better idea on how should the user interface of the system be implement and design hence user can have a better experience and control on the system. Furthermore, a comparison on those existing systems had also been done in order to get a better insight into which features or designs are more suitable to develop in this project. My Route Online, an online route planner system and two online cargo loading systems which are EasyCargo and LoadCalculator were studied, analyzed and evaluated on their features and functionalities.

Two features that will be covered in this route optimization are finding the optimal route for trucks to deliver cargoes to their customers and maximize the available space in the truck during loading and arranging cargoes in a cargo space. At the end of this project, only feature "finding the optimal route to deliver" was developed. Another feature which is "maximize the available space in the truck" will only be developed in the future which had been listed at the out of scope section. Both of them are NP-hard problems that have been solved by other researchers. In order for this system to solve those problems, an algorithm was implemented based on the information gathered when performing literature review. Literature review was done on several related research problems and related algorithms like travelling salesman problem, vehicle routing problem and bin packing problem.

Besides, a face-to-face interview was conducted on 7 July 2019 at Timez Business Hotel which located at No. 21, Jalan C180/1, Dataran C180, Kuala Lumpur with the manager of Hondail Trading Sdn Bhd. An interview's questions were generated and then verified by my supervisor Dr.Winnie before the interview section. During the interview, predetermined questions was asked to have a better understanding on how they actually run their daily operations which involve planning on the route to deliver and loading the cargoes. The responses, opinions and suggestions were then recorded on a paper for later analysis and report generation. Last,

the functional and non-functional specification were determined based on the gathered requirements.

3.2.3 Prototype Implementation Phase

During prototype development phase, several activities were carried out continuously based on the number of iterations which are design, implementation of prototype, customer review and evaluation. For this project, there will be three iterations of prototypes going to be developed. After each iteration, the prototype will be closer to the final product.

3.2.3.1 First Iteration

In the first iteration, a prototype that only consists of the user interface was developed while the back-end implementation was planned to develop on the second iteration. An iterative web-based prototype which involves the overall screen flow of the system was implemented in the first iteration. The purpose of the first iteration is to make sure that the user interface designed is able to meet the satisfaction level of the targeted user. During the design phase, a use case diagram and use case description were developed to capture the dynamic behaviour between targeted user and system. By developing a use case diagram developer can have a better understanding on the requirement gathered, an overview of the functionalities that will be provided by the system and also the overview interaction between user and the system. Then based on the use case diagram and gathered information, a preliminary prototype was developed. The preliminary prototype was developed using webflow. The preliminary prototype was then presented to the targeted user in order to get feedback from them. Those comments and suggestions were recorded for improvement in the second iteration.

3.2.3.2 Second Iteration

In this iteration, activity diagram, data flow diagram and ERD diagram were drawn to have a better understanding of the overall flow of the data and events during backend implementation. Improvement like adding further use cases and also adding missing role which is company owner/ company admin to the use case diagram and also use case description had been done in this iteration. The front-end screens were continued to develop and those unsatisfactory parts were modified after getting feedback from supervisor and also moderator.

In this iteration, the standard flow of the system which includes CRUD operations were developed. The database that used to interact and store data of this system is cloud firestore which is one of the NoSQL databases. Besides, algorithm which used to suggest the optimal route to deliver those cargoes also developed in this iteration. Another algorithm with the purpose of maximizing the available space in the truck will only be developed in the future. By developing the backend logic, all the static code which used in the first iteration were removed and replaced with dynamic and workable code.

After improvement on the front-end design, the back-end implementation has also been done, the second evaluation phase was carried out to get user feedback on the performance of the implemented algorithm and also the specification requirements of this system.

3.2.3.3 Third Prototype

In this iteration, requirements that had been missed or misunderstood during development process in the second iteration had been listed out. Those listed out requirements had then been added and fixed in these iterations.

The prototype is again sent to user for evaluating whether the system has met their requirements. The prototype will keep on refining until the latest requirements have been met or the user is satisfied with the current design of the system. Then the prototype will be developed into the final deliverables of this project.

3.2.4 Testing Phase

In this phase, a series of testing like unit testing, integration testing and user acceptance testing were performed to make sure that the quality of the system is achieved and also the outcome produce is able to address those issues which had determined in the problem statement. After all testing was passed, the system proceeds for further improvement and adding of new features.

3.3 Development Tools

3.3.1 JQuery

JQuery is a lightweight JavaScript framework which provides a set of reusable tasks which are easy to access by developing several lines of codes. In JQuery many complicated tasks like AJAX calls, HTML manipulation, DOM manipulation and animations was simplified. The purpose of JQuery is to make developer coding life easier due to most of the common tasks had already predefined. The reason JQuery is selected instead of others JavaScript frameworks is due to JQuery is small but has a rich features of library, it is more extendable compare to other framework and it is more reliable due to big companies like Google, Microsoft, IBM and Netflix are also using JQuery for their web development.

3.3.2 Bootstrap

Bootstrap is known as a front-end framework which is able to help developer to speed up their web development progress by providing HTML and CSS based design templates for creating a responsive web layout. Bootstrap has been chosen in this project due to it is compatible with all modern web browsers likes Safari, Chrome, Firefox, Opera and Internet Explorer. Besides, bootstrap provides several basic templates for design purpose and all of them are fully customizable. Bootstrap also provides useful helper classes which are able to let developer have a better control on the styling and make the development of responsive website easy. Bootstrap also provides great online documentation and tutorials which help developer to have a better understanding on their features and functionalities.

3.3.3 React

React is a web framework that using JavaScript library for building a component-based user interface. Which mean that the whole application is built by those reusable class component. The reason react is used in this project is because react has the latest documentation which is useful. Besides, react also provide different kind of UI component which has been developed by those third parties and most of them are open sources and can be used without charge. Another reason that react is used for this project is because of the concept of reusable component. It provides reusability, flexibility and scalability for developer to design project's user interface. The used of

JavaScript in react make developer more easy to call to the backend server like firebase and other online services.

3.3.4 WebGL

WebGL is a JavaScript API that allows developer to create an interactive 3D objects directly on the browser. The reason WebGL had been used in this project is due to the visualization of the orientation of cargoes in the truck is one of the features that will be implemented in this project.

3.3.5 Google Maps API

Google API is a set of programming interface which implemented by Google which enables developers to interact with the services that had been located at the Google cloud platform. Google Maps API is one of the services that provided by Google in order to access to the data regarding maps, locations, distance, directions, address and so on. Google Maps API will be used in this project to allow the system to access to the visualization of map. Besides, with the help of Google Maps API the system is able to know the exact distance between one address to another address hence system is able to find the optimal (shortest) route to deliver cargoes to their customers.

3.3.6 Firebase Authentication

Firebase Authentication is a cloud service that provided by the firebase for developer to easily handle all the authentication process. Firebase authentication support several authentication methods like passwords, Google, email, phone number, twitter and so on. This helps developer to smooth the authentication process and developer may not need to implement the code.

3.3.7 Cloud Firestore

Cloud Firestore is a NoSQL cloud database that provided by the firebase to store and sync data for client and also the server side development. The reason Cloud Firestore is used in this project is due to all of the relevant data can be store in a single document. Therefore, developer doesn't have to join the table one by one. With the concept of NoSql developer can easily store or retrieve the data by only a single call.

3.3.8 WebFlow

WebFlow is an application that allows developer to build a responsive website without any knowledge on web development. WebFlow is used as a prototyping tool in this project due to WebFlow has the ability to auto generate HTML, CSS and Javascript code once the designer has completed their jobs. Therefore, during the development phase developer can just refer to the code that generated by the WebFlow. Besides, WebFlow allows developer illustrates a 3D object easily within a limited time range. In the system, the orientation of the cargoes in the truck may be needed to visualize in the 3D format. Construct a 3D model may need a lot of time. Due to time constraints and also inexperience in constructing a 3D models, WebFlow becoming the best prototyping tool to demonstrate the overall workflow of my system.

3.4 Preliminary User Interface Design

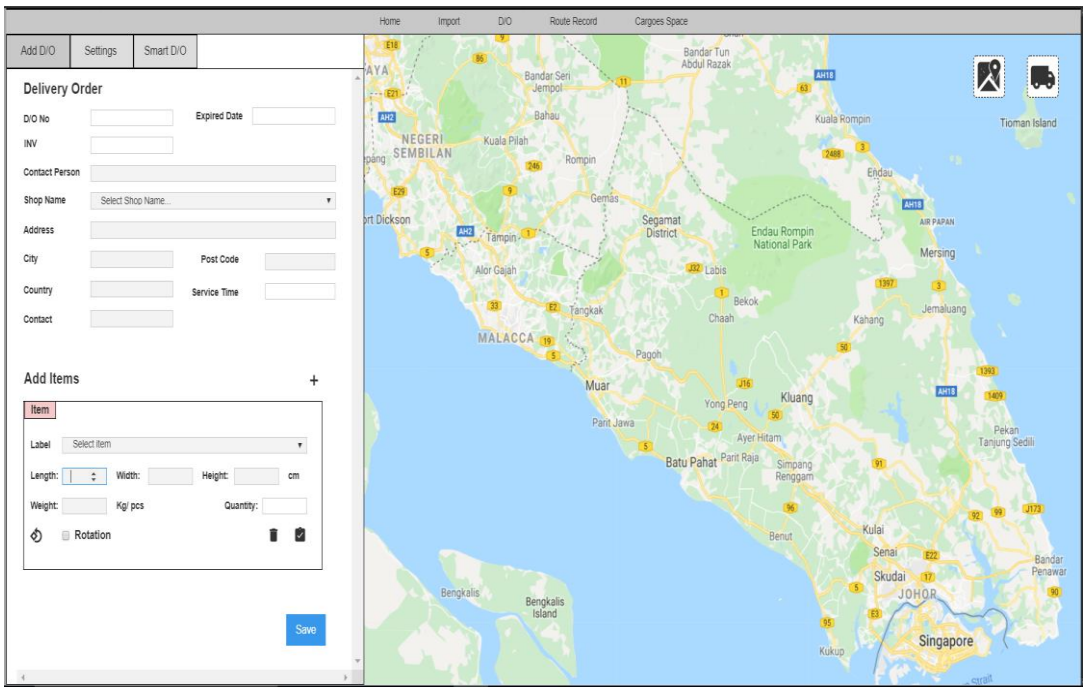


Figure 3.2 UI – Overview Layout of the system

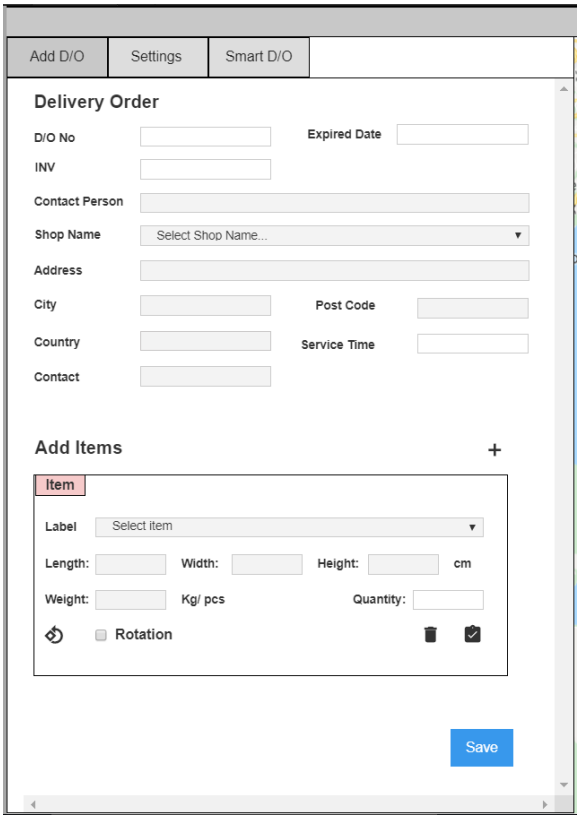


Figure 3.3 UI – Add D/O manually

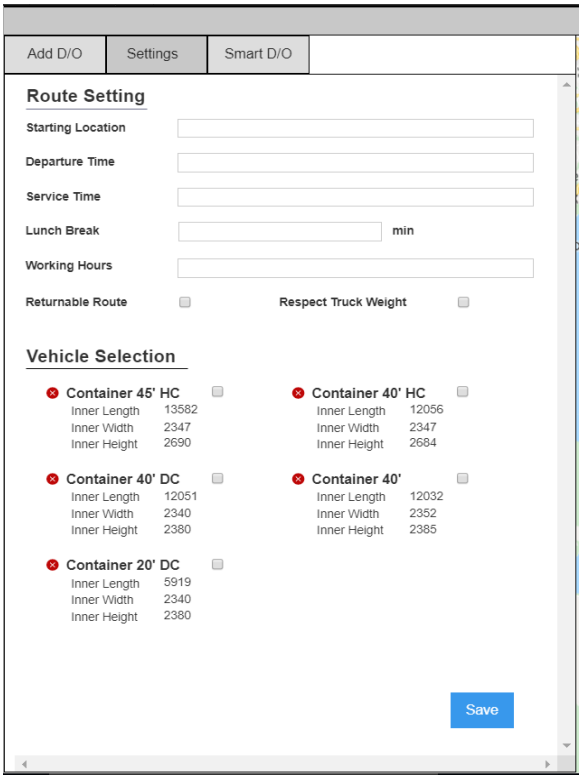


Figure 3.4 UI – Route Settings

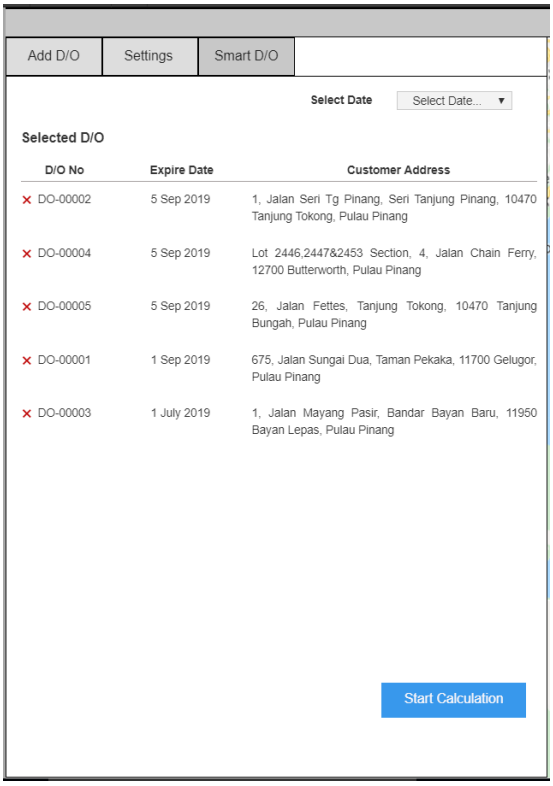


Figure 3.5 UI – Show selected D/O based on selected date

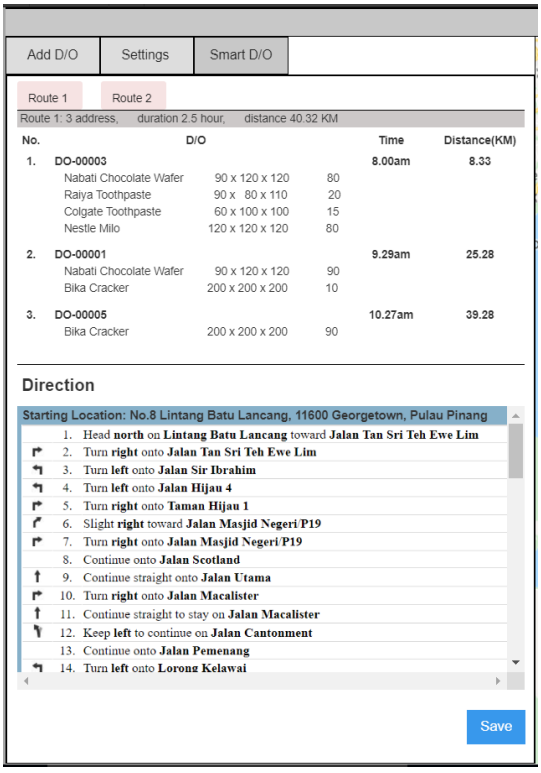


Figure 3.6 UI - Show result of the route on 'SMART D/O' tab

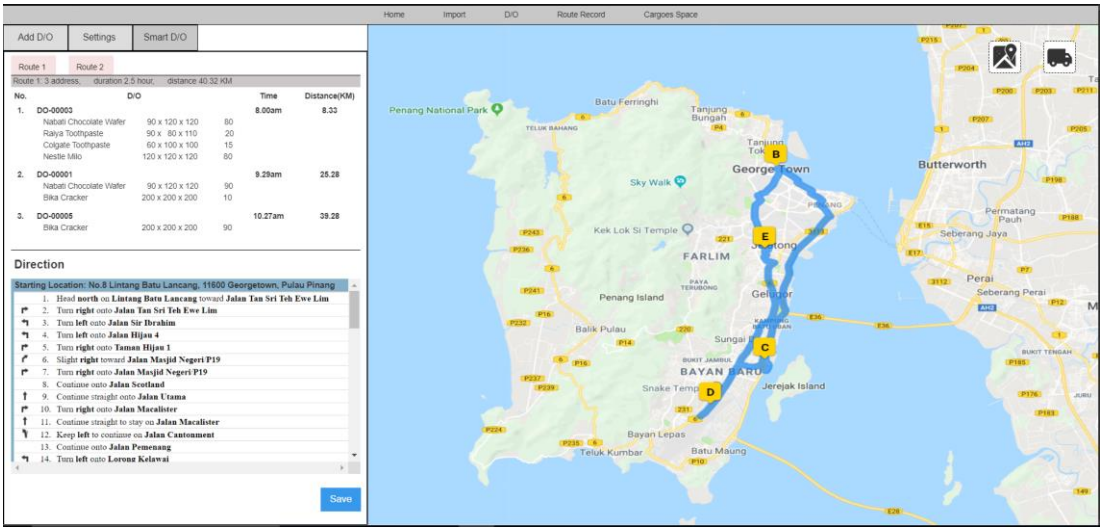


Figure 3.7 UI - Show the particular route on the map

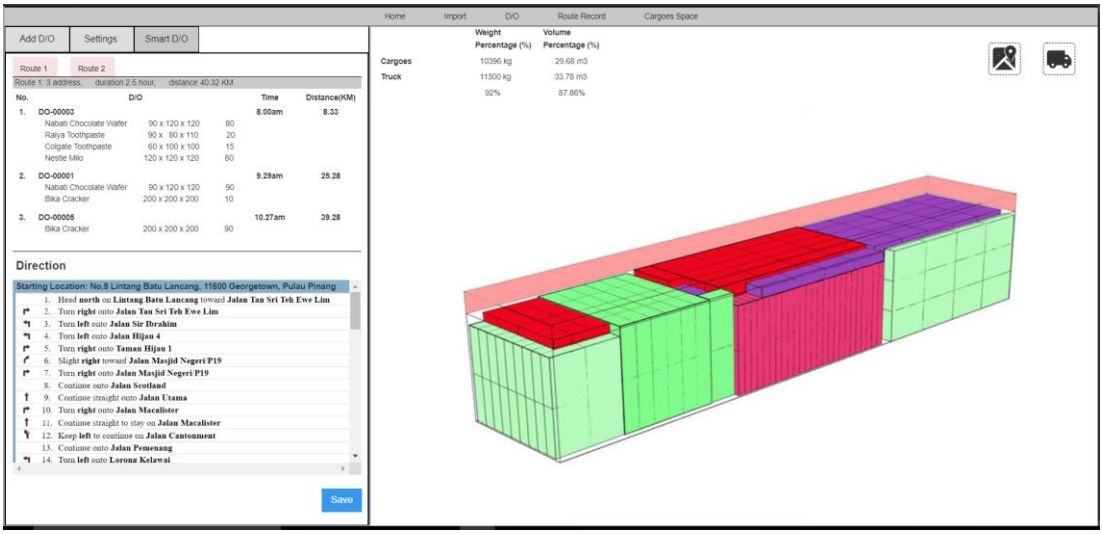


Figure 3.8 UI - Cargoes visualization of a particular truck which responsible for a particular route

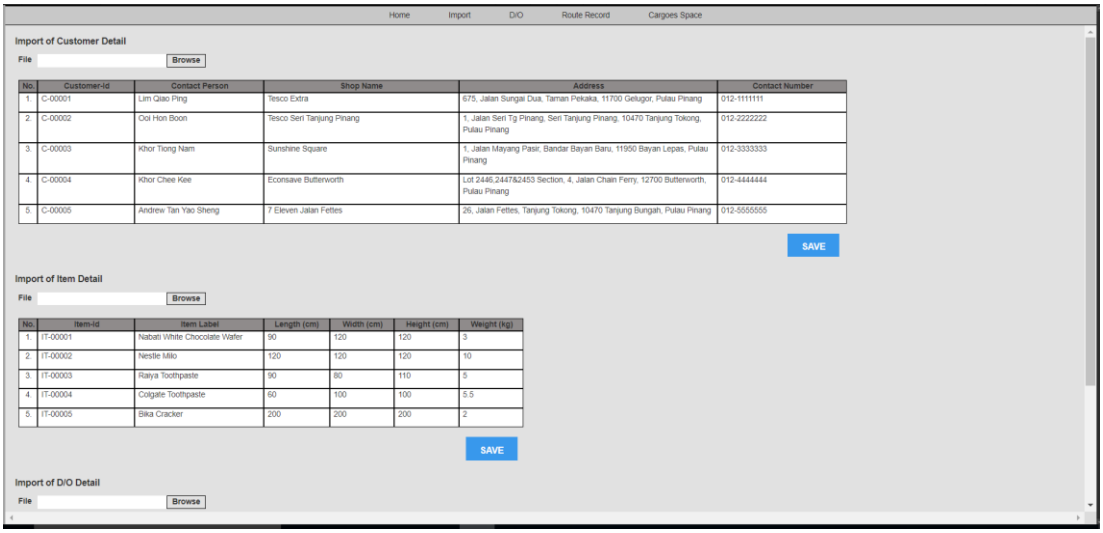


Figure 3.9 UI - Page for user to input data through Excel format

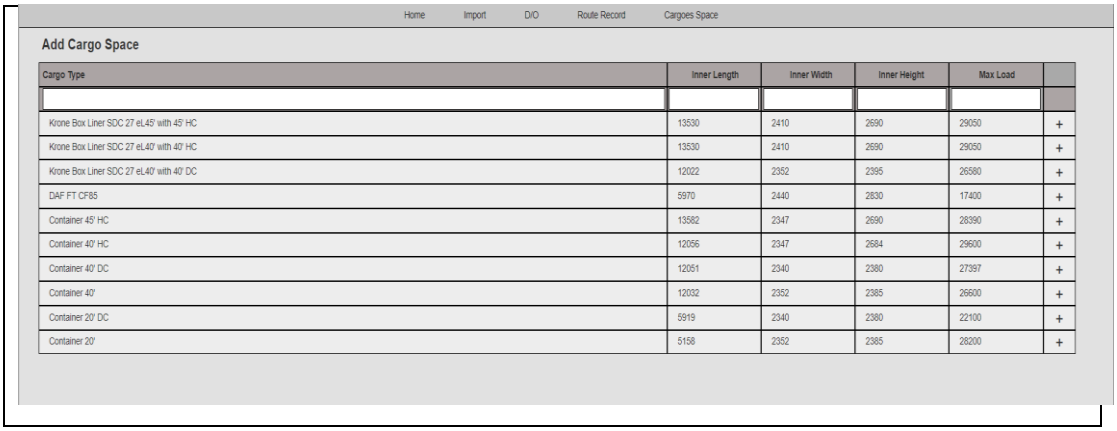


Figure 3.10 UI - Page to add cargoes spaces when the type of cargo's space user own did not find at "Setting" tab

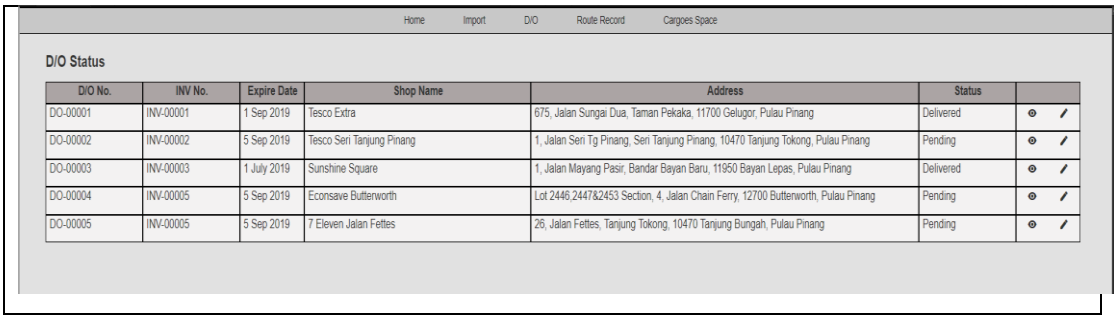


Figure 3.11 UI – Page to view the status of added D/O

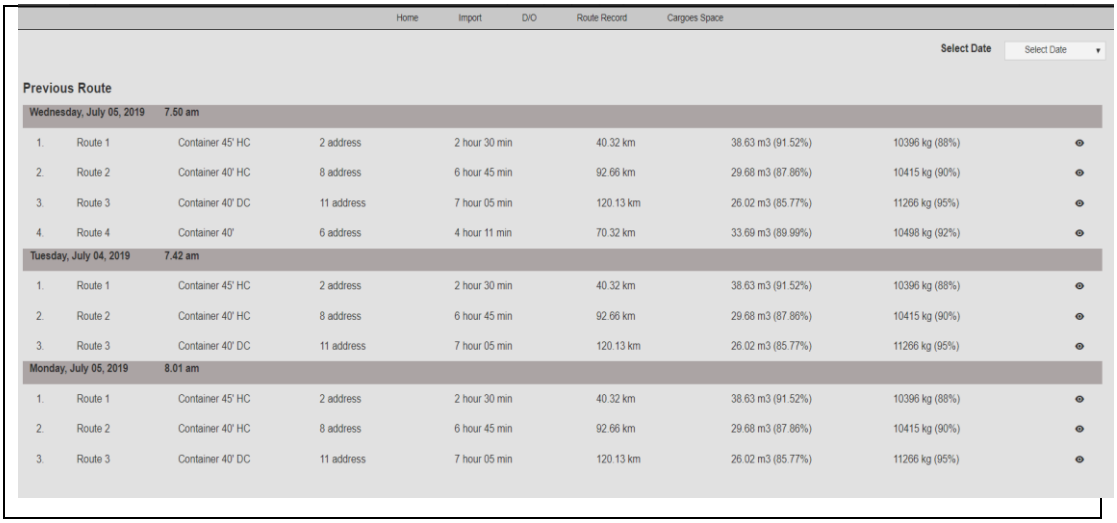


Figure 3.12 UI – Page to view the previous route

3.5 Project Planning

3.5.1 Work Breakdown Structure (WBS)

Work Breakdown Structure (WBS) is a vehicle for breaking down the whole projects into many subprojects, tasks and subtasks. The aim of defining WBS is to make sure that the particular project is under control and able to perform well as stated in the plan. Based on the Software Development Life Cycle the WBS will break down into 5 phases which are planning phase, requirement gathering phase, design, implementation and testing.

	Task Name	Duration	Start	Finish
1	**Web-Based Route Optimization System For Logistic**	**314 days**	**Mon 5/27/19**	**Sat 4/4/20**
2	**1.0 Planning Phase**	**32 days**	**Mon 5/27/19**	**Thu 6/27/19**
3	**1.1 Develop Work Plan**	**6 days**	**Mon 5/27/19**	**Sat 6/1/19**
4	1.1.1 Determine Project Milestones	1 day	Mon 5/27/19	Mon 5/27/19
5	1.1.2 Develop Project's Work Breakdown Structure	2 days	Tue 5/28/19	Wed 5/29/19
6	1.1.3 Develop Project's Gantt Chart	3 days	Thu 5/30/19	Sat 6/1/19
7	1.2 Determine Problem Statement	3 days	Tue 6/4/19	Thu 6/6/19
8	1.3 Determine Project's Objectives	1 day	Fri 6/7/19	Fri 6/7/19
9	**1.4 Determine Project's Scopes**	**3 days**	**Sat 6/8/19**	**Mon 6/10/19**
10	1.4.1 Identify Scope Covered	1 day	Sat 6/8/19	Sat 6/8/19
11	1.4.2 Identify Topic Out of Scope	1 day	Sun 6/9/19	Sun 6/9/19
12	1.4.3 Identify Assumption of Scope	1 day	Mon 6/10/19	Mon 6/10/19
13	1.5 Determine Project's Proposed Solution	2 days	Tue 6/11/19	Wed 6/12/19
14	**1.6 Determine Project's Proposed Approach**	**7 days**	**Thu 6/13/19**	**Wed 6/19/19**
15	**1.6.1 Determine Software Development Methodology**	**4 days**	**Thu 6/13/19**	**Sun 6/16/19**
16	1.6.1.1 Research on Waterfall Methodology	1 day	Thu 6/13/19	Thu 6/13/19
17	1.6.1.2 Research on V-Model Methodology	1 day	Fri 6/14/19	Fri 6/14/19
18	1.6.1.3 Research on Prototyping Methodology	1 day	Sat 6/15/19	Sat 6/15/19
19	1.6.1.4 Research on Agile Methodology	1 day	Sun 6/16/19	Sun 6/16/19
20	**1.6.2 Determine Research Methodology**	**3 days**	**Mon 6/17/19**	**Wed 6/19/19**
21	1.6.2.1 Research on Qualitative Approach	1 day	Mon 6/17/19	Mon 6/17/19
22	1.6.2.2 Research on Quantitative Approach	1 day	Tue 6/18/19	Tue 6/18/19
23	1.6.2.3 Research on Mixed Approach	1 day	Wed 6/19/19	Wed 6/19/19
24	1.7 Preliminary Report Writing	8 days	Thu 6/20/19	Thu 6/27/19
25	**2.0 Requirement Gathering Phase**	**50 days**	**Sun 6/16/19**	**Sun 8/4/19**
26	**2.1 Perform Literature Review**	**43 days**	**Sun 6/23/19**	**Sun 8/4/19**
27	2.1.1 Research on Travelling Salesman Problem and its algorithm	10 days	Sun 6/23/19	Tue 7/2/19
28	2.1.2 Research on Vehicle Routing Problem and its algorithm	6 days	Wed 7/3/19	Mon 7/8/19
29	2.1.3 Research on Bin/Truck Packing Problem and its algorithm	27 days	Tue 7/9/19	Sun 8/4/19
30	**2.2 Conduct an Interview**	**39 days**	**Sat 6/22/19**	**Tue 7/30/19**
31	2.2.1 Generation of Interview Questions	2 days	Sat 6/22/19	Sun 6/23/19
32	2.2.2 Conduct Interview with manager of Hondail Trading Sdn Bhd	1 day	Mon 6/24/19	Mon 6/24/19
33	2.2.3 Analysis of findings	9 days	Mon 7/22/19	Tue 7/30/19
34	**2.3 Review on Existing Application**	**13 days**	**Sun 6/16/19**	**Fri 6/28/19**
35	2.3.1 Review on My Route Online	2 days	Sun 6/16/19	Mon 6/17/19
36	2.3.2 Review on Easy Cargo	2 days	Tue 6/25/19	Wed 6/26/19
37	2.3.3 Review on LoadCalculator	2 days	Thu 6/27/19	Fri 6/28/19
38	2.4 Writing Literature Review	50 days	Thu 6/13/19	Thu 8/1/19

Figure 3.13 First part of WBS

39	**3.0 Prototype Development Phase**	**237 days**	**Mon 7/22/19**	**Sat 3/14/20**
40	**3.1 First Iteration**	**25 days**	**Mon 7/22/19**	**Thu 8/15/19**
41	3.1.1 Determine Use Case Diagram	6 days	Mon 7/22/19	Sat 7/27/19
42	3.1.2 Determine Use Case Descriptions	7 days	Sun 7/28/19	Sat 8/3/19
43	**3.1.3 Build Prototype**	**22 days**	**Mon 7/22/19**	**Mon 8/12/19**
44	3.1.3.1 Design User Interface	22 days	Mon 7/22/19	Mon 8/12/19
45	3.1.4 Collect User Feedback	3 days	Tue 8/13/19	Thu 8/15/19
46	**3.2 Second Iteration**	**51 days**	**Mon 1/13/20**	**Tue 3/3/20**
47	3.2.1 Improve on use case and use case description	1 day	Mon 1/13/20	Mon 1/13/20
48	3.2.2 Draw Activity Diagram	3 days	Tue 1/14/20	Thu 1/16/20
49	3.2.3 Draw Data Flow Diagram	4 days	Fri 1/17/20	Mon 1/20/20
50	**3.2.4 Draw ERD Diagram**	**1 day**	**Sun 1/19/20**	**Sun 1/19/20**
51	3.2.4.1 User Collection	1 day	Sun 1/19/20	Sun 1/19/20
52	3.2.4.2 Customer Collection	1 day	Sun 1/19/20	Sun 1/19/20
53	3.2.4.3 Item Collection	1 day	Sun 1/19/20	Sun 1/19/20
54	3.2.4.4 D/O Collection	1 day	Sun 1/19/20	Sun 1/19/20
55	3.2.4.5 Cargoes Collection	1 day	Sun 1/19/20	Sun 1/19/20
56	3.2.4.6 Route Setting Collection	1 day	Sun 1/19/20	Sun 1/19/20
57	3.2.4.7 Route Record Collection	1 day	Sun 1/19/20	Sun 1/19/20
58	**3.2.5 Build Prototype**	**39 days**	**Wed 1/22/20**	**Sat 2/29/20**
59	**3.2.5.1 Login and SignUp Feature**	**5 days**	**Wed 1/22/20**	**Sun 1/26/20**
60	3.2.5.1.1 Create Login screen	1 day	Wed 1/22/20	Wed 1/22/20
61	3.2.5.1.2 Create Sign Up screen	1 day	Thu 1/23/20	Thu 1/23/20
62	3.2.5.1.3 Connect with Firebase Authentication	2 days	Fri 1/24/20	Sat 1/25/20
63	3.2.5.1.4 Connect and store user data into Cloud Firestore	1 day	Sun 1/26/20	Sun 1/26/20
64	**3.2.5.2 Import Data Feature**	**6 days**	**Mon 1/27/20**	**Sat 2/1/20**
65	**3.2.5.2.1 Create Home Screen**	**3 days**	**Mon 1/27/20**	**Wed 1/29/20**
66	**3.2.5.2.1.1 "Add D/O" tab**	**1 day**	**Mon 1/27/20**	**Mon 1/27/20**
67	3.2.5.2.1.1.1 Create UI for Add D/O view	1 day	Mon 1/27/20	Mon 1/27/20
68	3.2.5.2.1.1.2 Enable user to create new D/O and store data to firestore	1 day	Mon 1/27/20	Mon 1/27/20
69	3.2.5.2.1.1.3 Enable system to retrieve customer and item data for selection purpose when creating new D/O	1 day	Mon 1/27/20	Mon 1/27/20
70	**3.2.5.2.1.2 "Settings" tab**	**1 day?**	**Wed 1/29/20**	**Wed 1/29/20**
71	3.2.5.2.1.2.1 Create UI for Settings view	1 day	Wed 1/29/20	Wed 1/29/20
72	3.2.5.2.1.2.1 Store route setting information into the firestore	1 day	Wed 1/29/20	Wed 1/29/20
73	3.2.5.2.1.2.3 Retrieve cargo information and display its on the Vehicle Selection area	1 day	Wed 1/29/20	Wed 1/29/20
74	**3.2.5.2.2 Import data from excel file (.csv) format**	**2 days**	**Fri 1/31/20**	**Sat 2/1/20**
75	3.2.5.2.2.1 Create UI for Import Data Screen	1 day	Fri 1/31/20	Fri 1/31/20

Figure 3.14 Second part of WBS

76	3.2.5.2.2.2 Research and implement react-excel-renderer on this screen	2 days	Fri 1/31/20	Sat 2/1/20
77	3.2.5.2.2.3 Enable the system to store data (customer, d/o and item) into the firestore	1 day	Sat 2/1/20	Sat 2/1/20
78	**3.2.5.3 Display and Update record for the system**	**28 days**	**Sun 2/2/20**	**Sat 2/29/20**
79	**3.2.5.3.1 Smart D/O screen**	**5 days**	**Sun 2/2/20**	**Thu 2/6/20**
80	3.2.5.3.1.1 Create UI for Smart D/O tab view	1 day	Sun 2/2/20	Sun 2/2/20
81	3.2.5.3.1.2 Enable to display the information of D/Os based on the selected date range	3 days	Mon 2/3/20	Wed 2/5/20
82	3.2.5.3.1.3 Remove a particular D/O from the displayed list	1 day	Thu 2/6/20	Thu 2/6/20
83	**3.2.5.3.2 Cargoes Space screen**	**1 day**	**Fri 2/7/20**	**Fri 2/7/20**
84	3.2.5.3.2.1 Create UI for Cargoes Space screen	1 day	Fri 2/7/20	Fri 2/7/20
85	3.2.5.3.2.2 Enable system to retrieve cargoes data from firestore and display them on screen	1 day	Fri 2/7/20	Fri 2/7/20
86	3.2.5.3.2.3 Enable user to update the satus of a particular cargo	1 day	Fri 2/7/20	Fri 2/7/20
87	**3.2.5.3.3 Customer Information screen**	**2 days**	**Sat 2/8/20**	**Sun 2/9/20**
88	3.2.5.3.3.1 Create UI for Customer Informationi screen	1 day	Sat 2/8/20	Sat 2/8/20
89	3.2.5.3.3.2 Enable system to retrieve	1 day	Sat 2/8/20	Sat 2/8/20
90	3.2.5.3.3.3 Enable user to edit a particular Customer then update the data to firestore	1 day	Sun 2/9/20	Sun 2/9/20
91	**3.2.5.3.4 Item Information screen**	**2 days**	**Mon 2/10/20**	**Tue 2/11/20**
92	3.2.5.3.4.1 Create UI for Item Informationi screen	1 day	Mon 2/10/20	Mon 2/10/20
93	3.2.5.3.4.2 Enable system to retrieve Item data from firestore and display them on the screen	1 day	Mon 2/10/20	Mon 2/10/20
94	3.2.5.3.4.3 Enable user to edit a particular Item then update the data to firestore	2 days	Mon 2/10/20	Tue 2/11/20
95	**3.2.5.3.5 D/O Information screen**	**2 days**	**Wed 2/12/20**	**Thu 2/13/20**
96	3.2.5.3.5.1 Create UI for D/O Informationi screen	1 day	Wed 2/12/20	Wed 2/12/20
97	3.2.5.3.5.2 Enable system to retrieve D/O data from firestore and display them on the screen	1 day	Wed 2/12/20	Wed 2/12/20
98	3.2.5.3.5.3 Display the detail of the D/O in PDF format	1 day	Thu 2/13/20	Thu 2/13/20
99	3.2.5.3.5.4 Enable user to edit a particular D/O then update the data to firestore	1 day	Thu 2/13/20	Thu 2/13/20
100	**3.2.5.3.6 Route Record Screen**	**2 days**	**Fri 2/28/20**	**Sat 2/29/20**
101	3.2.5.3.6.1 Create UI for Route Record screen	1 day	Fri 2/28/20	Fri 2/28/20
102	3.2.5.3.6.2 Enable system to retrieve	2 days	Fri 2/28/20	Sat 2/29/20

Figure 3.15 Third part of WBS

103	3.2.5.3.6.3 Display the detail of the route record in PDF format	1 day	Sat 2/29/20	Sat 2/29/20
104	**3.2.5.4 Implementation on algorithm**	**14 days**	**Thu 2/13/20**	**Wed 2/26/20**
105	3.2.5.4.1 Implement algorithm to cluster those customers' location based on the number of trucks	5 days	Thu 2/13/20	Mon 2/17/20
106	3.2.5.4.2 Implement Algorithm to find the shortes path (Genetic Algorithm)	5 days	Tue 2/18/20	Sat 2/22/20
107	3.2.5.4.3 Explore and apply Google Map Api in the algorithm	2 days	Sun 2/23/20	Mon 2/24/20
108	3.2.5.4.4 Implement algorithm to count on the total occupied space in the truck	1 day	Tue 2/25/20	Tue 2/25/20
109	3.2.5.4.5 Implement algorithm to count on the total travelled distance and total travel duration	1 day	Wed 2/26/20	Wed 2/26/20
110	3.2.5.5 Visualize the route on the Google Map	1 day	Thu 2/27/20	Thu 2/27/20
111	3.2.6 Collect User Feedback	3 days	Sun 3/1/20	Tue 3/3/20
112	**3.3 Third Iteration**	**11 days**	**Wed 3/4/20**	**Sat 3/14/20**
113	**3.3.1 Build Prototype**	**6 days**	**Wed 3/4/20**	**Mon 3/9/20**
114	**3.3.1.1 Add on missing requirement**	**6 days**	**Wed 3/4/20**	**Mon 3/9/20**
115	3.3.1.1.1 Enable user to create new D/O when click on the "+" icon at "D/O Information" screen	1 day	Wed 3/4/20	Wed 3/4/20
116	3.3.1.1.2 Enable user to delete a particular D/O by clicking on the "Trash" icon.	1 day	Wed 3/4/20	Wed 3/4/20
117	3.3.1.1.3 Enable user to create new Customer when click on the "+" icon at "Customer Information" screen	1 day	Wed 3/4/20	Wed 3/4/20
118	3.3.1.1.4 Enable user to delete a particular Customer by clicking on the "Trash" icon.	1 day	Thu 3/5/20	Thu 3/5/20
119	3.3.1.1.5 Enable user to create new Item when click on the "+" icon at "Item Information" screen	1 day	Thu 3/5/20	Thu 3/5/20
120	3.3.1.1.6 Enable user to delete a particular Item by clicking on the "Trash" icon.	1 day	Thu 3/5/20	Thu 3/5/20
121	3.3.1.1.7 Enable user to create new cargo when click on the "+" icon at "Cargoes Space" screen	1 day	Fri 3/6/20	Fri 3/6/20
122	3.3.1.1.8 Enable user to delete a particular Cargo by clicking on the "Trash" icon.	1 day	Fri 3/6/20	Fri 3/6/20
123	3.3.1.1.9 Enable user to delete a particular planned route record by clicking on the "Trash" icon.	1 day	Fri 3/6/20	Fri 3/6/20
124	3.3.1.1.10 Add on validation and spinner when retrieving and storing data.	2 days	Sat 3/7/20	Sun 3/8/20
125	3.3.1.1.11 Add on company admin role	1 day	Sun 3/8/20	Sun 3/8/20
126	3.3.1.1.12 Enable company admin to create staff account	1 day	Sun 3/8/20	Sun 3/8/20
127	3.3.1.1.13 Enable company admin to view staff information	1 day	Mon 3/9/20	Mon 3/9/20
128	3.3.1.1.14 Enable company admin to change the status of a particular staff account	1 day	Mon 3/9/20	Mon 3/9/20

Figure 3.16 Forth part of WBS

129	Collect User Feedback	2 days	Tue 3/10/20	Wed 3/11/20
130	Integrate prototype to final product	3 days	Thu 3/12/20	Sat 3/14/20
131	**4.0 Testing Phase**	**21 days**	**Sun 3/15/20**	**Sat 4/4/20**
132	**4.1 Unit Testing**	**7 days**	**Sun 3/15/20**	**Sat 3/21/20**
133	**4.1.1 Sign In feature**	**1 day**	**Sun 3/15/20**	**Sun 3/15/20**
134	4.1.1.1 Test Case ID: Login-1A	1 day	Sun 3/15/20	Sun 3/15/20
135	4.1.1.2 Test Case ID: Login-1B	1 day	Sun 3/15/20	Sun 3/15/20
136	**4.1.2 Sign Up feature**	**1 day**	**Sun 3/15/20**	**Sun 3/15/20**
137	4.1.2.1 Test Case ID: SignUp-1A	1 day	Sun 3/15/20	Sun 3/15/20
138	4.1.2.2 Test Case ID: SignUp-1B	1 day	Sun 3/15/20	Sun 3/15/20
139	**4.1.3 Manage D/O Information**	**1 day**	**Mon 3/16/20**	**Mon 3/16/20**
140	4.1.3.1 Test Case ID: CreateD/O-1A	1 day	Mon 3/16/20	Mon 3/16/20
141	4.1.3.2 Test Case ID: CreateD/O-1B	1 day	Mon 3/16/20	Mon 3/16/20
142	4.1.3.3 Test Case ID: ViewD/O-1A	1 day	Mon 3/16/20	Mon 3/16/20
143	4.1.3.4 Test Case ID: ViewD/O-1B	1 day	Mon 3/16/20	Mon 3/16/20
144	4.1.3.5 Test Case ID: EditD/O-1A	1 day	Mon 3/16/20	Mon 3/16/20
145	4.1.3.6 Test Case ID: EditD/O-1B	1 day	Mon 3/16/20	Mon 3/16/20
146	4.1.3.7 Test Case ID: DeleteD/O-1A	1 day	Mon 3/16/20	Mon 3/16/20
147	**4.1.4 Manage Item Information**	**1 day**	**Mon 3/16/20**	**Mon 3/16/20**
148	4.1.4.1 Test Case ID: CreateItem-1A	1 day	Mon 3/16/20	Mon 3/16/20
149	4.1.4.2 Test Case ID: CreateItem-1B	1 day	Mon 3/16/20	Mon 3/16/20
150	4.1.4.3 Test Case ID: ViewItem-1A	1 day	Mon 3/16/20	Mon 3/16/20
151	4.1.4.4 Test Case ID: ViewItem-1B	1 day	Mon 3/16/20	Mon 3/16/20
152	4.1.4.5 Test Case ID: EditItem-1A	1 day	Mon 3/16/20	Mon 3/16/20
153	4.1.4.6 Test Case ID: EditItem-1B	1 day	Mon 3/16/20	Mon 3/16/20
154	4.1.4.7 Test Case ID: DeleteItem-1A	1 day	Mon 3/16/20	Mon 3/16/20
155	**4.1.5 Manage Customer Information**	**1 day**	**Tue 3/17/20**	**Tue 3/17/20**
156	4.1.5.1 Test Case ID: CreateCustomer-1A	1 day	Tue 3/17/20	Tue 3/17/20
157	4.1.5.2 Test Case ID: CreateCustomer-1B	1 day	Tue 3/17/20	Tue 3/17/20
158	4.1.5.3 Test Case ID: ViewCustomer-1A	1 day	Tue 3/17/20	Tue 3/17/20
159	4.1.5.4 Test Case ID: ViewCustomer-1B	1 day	Tue 3/17/20	Tue 3/17/20
160	4.1.5.5 Test Case ID: EditCustomer-1A	1 day	Tue 3/17/20	Tue 3/17/20
161	4.1.5.6 Test Case ID: EditCustomer-1B	1 day	Tue 3/17/20	Tue 3/17/20
162	4.1.5.7 Test Case ID: DeleteCustomer-1A	1 day	Tue 3/17/20	Tue 3/17/20
163	**4.1.6 Manage Cargo Information**	**1 day**	**Tue 3/17/20**	**Tue 3/17/20**
164	4.1.6.1 Test Case ID: CreateCargo-1A	1 day	Tue 3/17/20	Tue 3/17/20
165	4.1.6.2 Test Case ID: CreateCargo-1B	1 day	Tue 3/17/20	Tue 3/17/20
166	4.1.6.3 Test Case ID: ViewCargo-1A	1 day	Tue 3/17/20	Tue 3/17/20
167	4.1.6.4 Test Case ID: ViewCargo-1B	1 day	Tue 3/17/20	Tue 3/17/20
168	4.1.6.5 Test Case ID: EditCargo-1A	1 day	Tue 3/17/20	Tue 3/17/20
169	4.1.6.6 Test Case ID: EditCargo-1B	1 day	Tue 3/17/20	Tue 3/17/20
170	4.1.6.7 Test Case ID: DeleteCargo-1A	1 day	Tue 3/17/20	Tue 3/17/20
171	**4.1.7 Import Data with Excel format (.csv)**	**1 day**	**Wed 3/18/20**	**Wed 3/18/20**
172	4.1.7.1 Test Case ID: ImportD/O-1A	1 day	Wed 3/18/20	Wed 3/18/20
173	4.1.7.2 Test Case ID: ImportD/O-1B	1 day	Wed 3/18/20	Wed 3/18/20
174	4.1.7.3 Test Case ID: ImportItem-1A	1 day	Wed 3/18/20	Wed 3/18/20
175	4.1.7.4 Test Case ID: ImportItem-1B	1 day	Wed 3/18/20	Wed 3/18/20
176	4.1.7.5 Test Case ID: ImportCustomer-1A	1 day	Wed 3/18/20	Wed 3/18/20

Figure 3.17 Fifth part of WBS

177	4.1.7.6 Test Case ID: ImportCustomer-1B	1 day	Wed 3/18/20	Wed 3/18/20
178	**4.1.8 Configure the route Setting**	**1 day**	**Wed 3/18/20**	**Wed 3/18/20**
179	4.1.8.1 Test Case ID: ConfigureRoute-1A	1 day	Wed 3/18/20	Wed 3/18/20
180	4.1.8.2 Test Case ID: ConfigureRoute-1B	1 day	Wed 3/18/20	Wed 3/18/20
181	**4.1.9 Display the D/O information which will be involved in later route calculation.**	**1 day**	**Thu 3/19/20**	**Thu 3/19/20**
182	4.1.9.1 Test Case ID: ViewSelectedD/O-1A	1 day	Thu 3/19/20	Thu 3/19/20
183	4.1.9.2 Test Case ID: ViewSelectedD/O-1B	1 day	Thu 3/19/20	Thu 3/19/20
184	**4.1.10 Calculate and display suggested route**	**1 day**	**Thu 3/19/20**	**Thu 3/19/20**
185	4.1.10.1 Test Case ID: CalculateRoute-1A	1 day	Thu 3/19/20	Thu 3/19/20
186	4.1.10.2 Test Case ID: CalculateRoute-1B	1 day	Thu 3/19/20	Thu 3/19/20
187	4.1.10.3 Test Case ID: CalculateRoute-1A	1 day	Thu 3/19/20	Thu 3/19/20
188	**4.1.11 Visualization of the calculated route on the map.**	**1 day**	**Fri 3/20/20**	**Fri 3/20/20**
189	4.1.11.1 Test Case ID: VisualizeRoute-1A	1 day	Fri 3/20/20	Fri 3/20/20
190	**4.1.12 Manage Previous Planned Route Record**	**1 day**	**Fri 3/20/20**	**Fri 3/20/20**
191	4.1.12.1 Test Case ID: CreateRoute-1A	1 day	Fri 3/20/20	Fri 3/20/20
192	4.1.12.2 Test Case ID: ViewRoute-1A	1 day	Fri 3/20/20	Fri 3/20/20
193	4.1.12.3 Test Case ID: ViewRoute-1B	1 day	Fri 3/20/20	Fri 3/20/20
194	4.1.12.4 Test Case ID: DeleteRoute-1A	1 day	Fri 3/20/20	Fri 3/20/20
195	**4.1.13 Manage Staff Information (Company Admin)**	**1 day**	**Fri 3/20/20**	**Fri 3/20/20**
196	4.1.13.1 Test Case ID: CreateStaff-1A	1 day	Fri 3/20/20	Fri 3/20/20
197	4.1.13.2 Test Case ID: CreateStaff-1B	1 day	Fri 3/20/20	Fri 3/20/20
198	4.1.13.3 Test Case ID: ViewStaff-1A	1 day	Fri 3/20/20	Fri 3/20/20
199	4.1.13.4 Test Case ID: ViewStaff-1B	1 day	Fri 3/20/20	Fri 3/20/20
200	4.1.13.5 Test Case ID: ChangeStaffStatus-1A	1 day	Fri 3/20/20	Fri 3/20/20
201	**4.2 Integration Testing**	**7 days**	**Sun 3/22/20**	**Sat 3/28/20**
202	4.2.1 Manage D/O Information	1 day	Sun 3/22/20	Sun 3/22/20
203	4.2.2 Manage Item Information	1 day	Mon 3/23/20	Mon 3/23/20
204	4.2.3 Manage Customer Information	1 day	Tue 3/24/20	Tue 3/24/20
205	4.2.4 Manage Cargo Information	1 day	Wed 3/25/20	Wed 3/25/20
206	4.2.5 Import Data	1 day	Thu 3/26/20	Thu 3/26/20
207	4.2.6 Route Calculation	1 day	Fri 3/27/20	Fri 3/27/20
208	4.2.7 Manage Route Record	1 day	Sat 3/28/20	Sat 3/28/20
209	4.2.8 Manage Staff Information (Company Admin)	1 day	Sat 3/28/20	Sat 3/28/20
210	**4.3 User Acceptance Testing**	**7 days**	**Sun 3/29/20**	**Sat 4/4/20**
211	4.3.1 Manage D/O Information	1 day	Sun 3/29/20	Sun 3/29/20
212	4.3.2 Manage Item Information	1 day	Mon 3/30/20	Mon 3/30/20
213	4.3.3 Manage Customer Information	1 day	Mon 3/30/20	Mon 3/30/20
214	4.3.4 Manage Cargo Information	1 day	Mon 3/30/20	Mon 3/30/20
215	4.3.5 Import Data with the Excel Format (.csv)	1 day	Tue 3/31/20	Tue 3/31/20
216	4.3.6 Route Setting Process	1 day	Wed 4/1/20	Wed 4/1/20
217	4.3.7 Route Calculation Process	1 day	Thu 4/2/20	Thu 4/2/20
218	4.3.8 Manage Previous Planned Route Record	1 day	Fri 4/3/20	Fri 4/3/20
219	4.3.9 Manage Staff Information (Company Admin)	1 day	Sat 4/4/20	Sat 4/4/20

Figure 3.18 Sixth part of WBS

3.5.2 Gantt Chart

Please refer to APPENDIX A: Gantt Chart

CHAPTER 4

PROJECT INITIAL SPECIFICATION AND DESIGN

4.1 Introduction

Section 4.2 and 4.3 will discuss on the functional requirements and non-functional requirements that have been gathered from the interview section with manager of Hondail Trading Sdn Bhd. Section 4.4 will introduce about the behavior of the user on the system by using use case diagram while section 4.5 is about the description of a particular use case in the use case diagram. A system architecture diagram was developed and displayed on the section 4.6. Besides, an ERD diagram used to explain the entities' relationship had designed and listed in section 4.7. Last, a data flow diagram and also the activity diagram which used to further explain the behaviour of the system had also been developed and listed in section 4.8 and 4.9 correspondingly.

4.2 Functional Requirements

1. The user shall be able to login to their account using his or her email and password.
2. The system shall be able to verify the user identity after the user enters his/her email and password.
3. The user shall be able to add a new D/O into the system manually.
4. The user shall be able to add a new item into the system manually.
5. The user shall be able to add a new customer into the system manually.
6. The user shall be able to add a new cargo space into the system manually.
7. The user shall be able to view the detail and make changes to a particular D/Os.
8. The user shall be able to make changes to a particular item.
9. The user shall be able to make changes to a particular customer.
10. The user shall be able to make changes to a particular cargo space.
11. The user shall be able to delete a particular D/Os.
12. The user shall be able to delete a particular item.
13. The user shall be able to delete a particular customer.
14. The user shall be able to delete a particular cargo space.
15. The user shall be able to import data (items, customers, D/O) into the system which is in Excel format (.csv).

16. The user shall be able to make changes on the route settings.
17. The user shall be able to select the types of trucks they own.
18. The system shall be able to calculate and search for the optimal route based on the customer address and D/O's expired date.
19. The user shall be able to remove a particular D/O before calculating for a route.
20. The user shall be able to view the results of the generated route on the SmartD/O tab and map.
21. The user shall be able to view the detail of the previous route.
22. The user shall be able to delete a particular route record.
23. Company admin shall be able to add staffs to access their company's information which include D/O detail, customer detail, item, detail and also cargo detail.
24. Company admin shall be able to change the status (activate/ deactivate) of their staff's account.
25. Company admin shall be able to view their company's staffs information,

4.3 Non-Functional Requirements

Security

1. The system shall restrict unauthorized user to access the system.
2. The system shall be able to backup the data time to time to prevent any loss of data.

Availability

1. The system shall be available to access at anytime and anywhere as long as user is able to connect to the internet.

Performance

1. The system shall be able to handle concurrent requests from multiple users without crashing.
2. The system shall be able to show the result to the user at a specific time range without delay.

Adaptability

1. The system shall be designed in the way which is able to modify and maintain easily by following software design principles.

Usability

1. The system shall be user-friendly by providing simple and consistent user interfaces which are able to let user to have a better experience and have the feeling of ‘ease to use’.
2. The user interfaces shall be designed to let user to have a better control over the system.
3. The user interfaces shall be able to guild the user to achieve their objectives.
4. The system shall be able to display error message when the input data is irrelevant.

4.4 Use Case Diagram

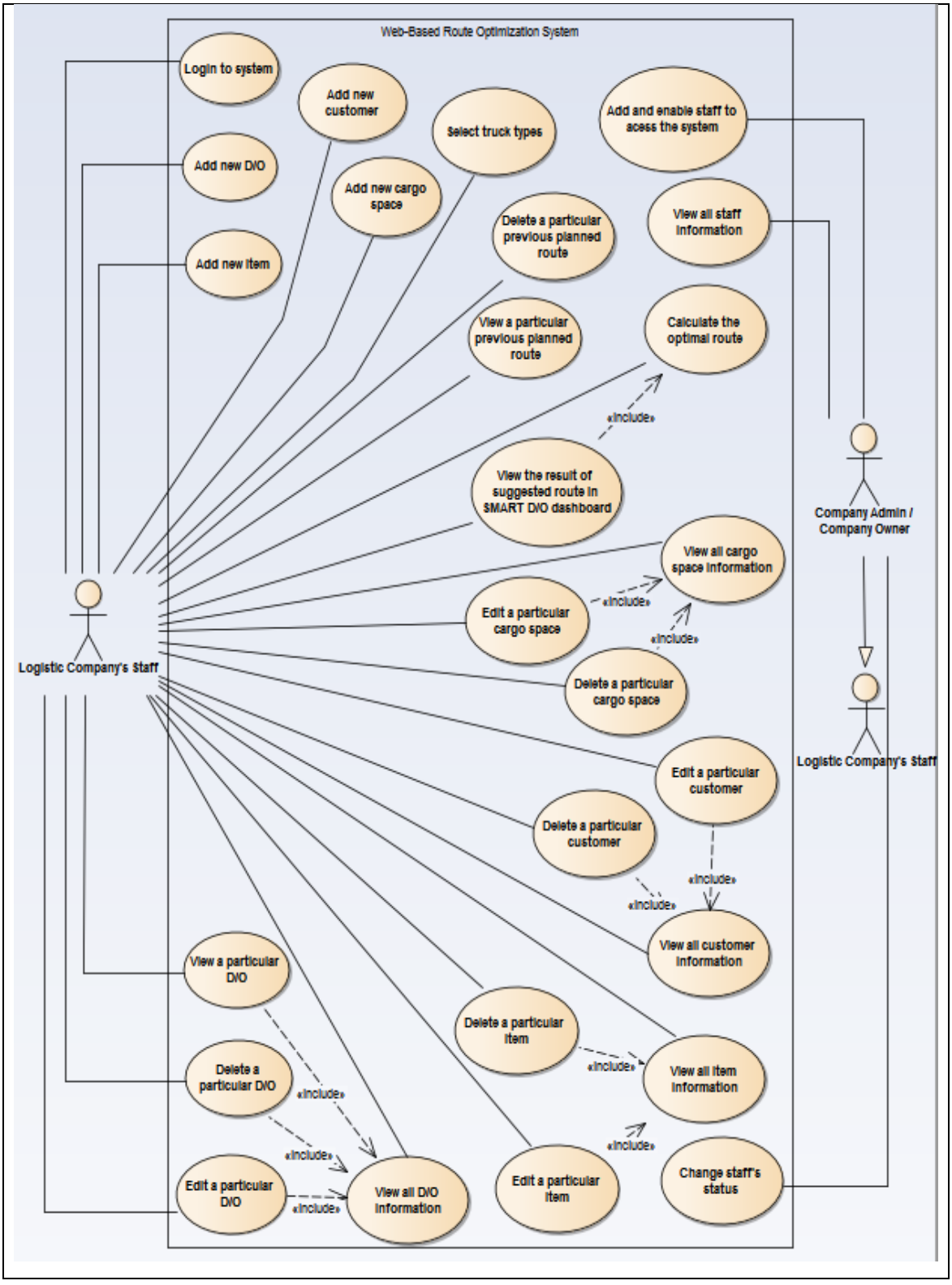


Figure 4.1 Use Case Diagram

4.5 Use Case Description

Use Case: Login to system	**ID:** 001	**Severity Level:** High
Actor: Logistic Company's Staff	**Type:** Detailed, Essential	
Stakeholders: Logistic Company's Staff – Staff wants to use this route optimization system.		
Description: This use case describe how company's manager/ staff login into the system.		
Trigger: Staff click on the "Login" button in navigation bar.		
Relationships: **Association:** Logistic Company's Staff **Include:** - **Extend:** - **Generalization:** -		
Normal Flow of Events: 1. User click on the "Login" button which located at the top right of the web page. 2. System prompt for user's email and password. 3. User enter a valid user's email and password then click on "Login" button. 4. System verify user's identity and redirect user to the home page of the system.		
SubFlows: -		
Alternate/Exceptional Flows: 3.1 User enters invalid email or password and clicks on "Login" button. 3.1.2. System displays "invalid user login" error message. 3.2 User leaves the input field empty and clicks on "Login" button. 3.2.2. System will prompt user to input the required information.		

Use Case: Add new D/O	**ID:** 002	**Severity Level:** High
Actor: Logistic Company's Staff	**Type:** Detailed, Essential	
Stakeholders: Logistic Company's Staff – Staff wants to add new D/O into the system.		
Description: This use case describe how staff add the generated D/O detail into the system.		
Trigger: A new D/O is generated and manager wish to add the detail of this D/O into the system in order to use the service of system.		
Relationships: **Association:** Logistic Company's Staff **Include:** - **Extend:** - **Generalization:** -		
Normal Flow of Events: 1. User enter the details of the D/O into the system manually and click on "Save" button. 2. System checks and validate for all input data. 3. System store those information into the database. 4. System display a success message to notify user that the process is success. 5. System reset the input field and user is able to enter the next D/O.		
SubFlows: -		
Alternate/Exceptional Flows: 1.1. Alternative option to input the D/O details. 1.1.1. User import the details of D/O which is in Excel format (.csv). 2.1. User enter invalid input type. 2.1.1. System display "invalid data type" error message to the user.		

Use Case: View all D/O information	**ID:** 003	**Severity Level:** High
Actor: Logistic Company's Staff	**Type:** Detailed, Essential	
Stakeholders: Logistic Company's Staff – Staff wish to view list of D/O records.		
Description: This use case describe how the user view all D/O information from the system.		
Trigger: Staff click on the "D/O Information" button which located on the navigation bar.		
Relationships: **Association:** Logistic Company's Staff **Include:** - **Extend:** - **Generalization:** -		
Normal Flow of Events: 1. User clicks on the "D/O Information" button on the navigation bar. 2. System redirect user to the pages which consist a list of D/O. 3. User can filter the D/O list by entering specific keyword on specific column field. 4. System show the narrowed D/O list to the user.		
SubFlows: -		
Alternate/Exceptional Flows: -		

Use Case: View a particular D/O	**ID:** 004	**Severity Level:** High
Actor: Logistic Company's Staff	**Type:** Detailed, Essential	
Stakeholders: Logistic Company's Staff – Staff wish to view a particular D/O.		
Description: This use case describe how the user view a particular D/O from the system.		
Trigger: Staff click on the "D/O Information" button which located on the navigation bar.		
Relationships: **Association:** Logistic Company's Staff **Include:** View all D/O information **Extend:** - **Generalization:** -		
Normal Flow of Events: 1. Perform use case "View all D/O information" 2. User click on the "Eye" icon on a particular D/O. 3. System show the detailed information of that particular D/O in PDF format.		
SubFlows: -		
Alternate/Exceptional Flows: -		

Use Case: Edit a particular D/O	**ID:** 005	**Severity Level:** High
Actor: Logistic Company's Staff	**Type:** Detailed, Essential	
Stakeholders: Logistic Company's Staff – Staff want to edit a particular D/O.		
Description: This use case describe how the user edit a particular D/O from the system.		
Trigger: Staff click on the "D/O Information" button which located on the navigation bar.		
Relationships: **Association:** Logistic Company's Staff **Include:** View all D/O information **Extend:** - **Generalization:** -		
Normal Flow of Events: 1. Perform use case "View all D/O information" 2. User click on the "Pencil" icon to edit the particular D/O. 3. User then click on the "Save" button. 4. System update the information to the database and display a success message.		
SubFlows: -		
Alternate/Exceptional Flows: -		

Use Case: Delete a particular D/O	**ID:** 006	**Severity Level:** High
Actor: Logistic Company's Staff	**Type:** Detailed, Essential	
Stakeholders: Logistic Company's Staff – Staff want to delete a particular D/O.		
Description: This use case describe how the user delete a particular D/O from the system.		
Trigger: Staff click on the "D/O Information" button which located on the navigation bar.		
Relationships: **Association:** Logistic Company's Staff **Include:** View all D/O information **Extend:** - **Generalization:** -		
Normal Flow of Events: 1. Perform use case "View all D/O information" 2. User click on the "Trash" icon to delete the particular D/O. 3. System will prompt user to confirm this action. 4. User then click on the "Confirm" button. 5. System delete the particular D/O record from the database and display a success message.		
SubFlows: -		
Alternate/Exceptional Flows: -		

Use Case: Add new item	**ID:** 007	**Severity Level:** High
Actor: Logistic Company's Staff	**Type:** Detailed, Essential	

Stakeholders:

Logistic Company's Staff – Staff wants to add new item into the system.

Description:

This use case describe how staff add the item information into the system.

Trigger: Manager wish to add new item's information into the system.

Relationships:

Association: Logistic Company's Staff

Include: -

Extend: -

Generalization: -

Normal Flow of Events:

1. User click on the "Item Information" button in the navigation bar.
2. User then click on the "Plus" icon beside the header "Item Detail".
3. User enter the details of the item into the system manually and click on "Save" button.
4. System checks and validate for all input data.
5. System store those information into the database.
6. System display a success message to notify user that the process is success.
7. System reset the input field and user is able to enter the next item.

SubFlows:

-

Alternate/Exceptional Flows:

1.1. Alternative option to input the Item details.

1.1.1. User import the details of Item which is in Excel format (.csv).

2.1. User enter invalid input type.

2.1.1. System display "invalid data type" error message to the user.

Use Case: View all item information	**ID:** 008	**Severity Level:** High
Actor: Logistic Company's Staff	**Type:** Detailed, Essential	
Stakeholders: Logistic Company's Staff – Staff wish to view list of Item records.		
Description: This use case describe how the user view all the items' information from the system.		
Trigger: Staff click on the "Item Information" button which located on the navigation bar.		
Relationships: **Association:** Logistic Company's Staff **Include:** - **Extend:** - **Generalization:** -		
Normal Flow of Events: 1. User clicks on the "Item Information" button on the navigation bar. 2. System redirect user to the pages which consist a list of Items. 3. User can filter the Item list by entering specific keyword on specific column field. 4. System show the narrowed items' list to the user.		
SubFlows: -		
Alternate/Exceptional Flows: -		

Use Case: Edit a particular item	**ID:** 009	**Severity Level:** High
Actor: Logistic Company's Staff	**Type:** Detailed, Essential	

Stakeholders:

Logistic Company's Staff – Staff want to edit a particular Item.

Description:

This use case describe how the user edit a particular Item in the system.

Trigger: Staff click on the "Item Information" button which located on the navigation bar.

Relationships:

Association: Logistic Company's Staff

Include: View all item information

Extend: -

Generalization: -

Normal Flow of Events:

1. Perform use case "View all item information"
2. User click on the "Pencil" icon to edit the particular item.
3. User then click on the "Save" button.
4. System update the information to the database and display a success message.

SubFlows:

-

Alternate/Exceptional Flows:

-

Use Case: Delete a particular item	**ID: 010**	**Severity Level:** High
Actor: Logistic Company's Staff	**Type:** Detailed, Essential	
Stakeholders: Logistic Company's Staff – Staff want to delete a particular Item.		
Description: This use case describe how the user delete a particular Item from the system.		
Trigger: Staff click on the "Item Information" button which located on the navigation bar.		
Relationships: **Association:** Logistic Company's Staff **Include:** View all item information **Extend:** - **Generalization:** -		
Normal Flow of Events: 1. Perform use case "View all item information" 2. User click on the "Trash" icon to delete the particular Item. 3. System will prompt user to confirm this action. 4. User then click on the "Confirm" button. 5. System delete the particular item record from the database and display a success message.		
SubFlows: -		
Alternate/Exceptional Flows: -		

Use Case: Add new customer	**ID:** 011	**Severity Level:** High
Actor: Logistic Company's Staff	**Type:** Detailed, Essential	

Stakeholders:

Logistic Company's Staff – Staff wants to add new customer into the system.

Description:

This use case describe how staff add the new customer information into the system.

Trigger: A new customer had approached to the company and manager wish to add his/her detail into the system.

Relationships:

Association: Logistic Company's Staff

Include: -

Extend: -

Generalization: -

Normal Flow of Events:

1. User click on the "Customer Information" button in the navigation bar.
2. User then click on the "Plus" icon beside the header "Customer Detail".
3. User enter the details of the customer into the system manually and click on "Save" button.
4. System checks and validate for all input data.
5. System store those information into the database.
6. System display a success message to notify user that the process is success.
7. System reset the input field and user is able to enter the next customer.

SubFlows:

-

Alternate/Exceptional Flows:

1.1. Alternative option to input the Customer details.

1.1.1. User import the details of Customer which is in Excel format (.csv).

2.1. User enter invalid input type.

2.1.1. System display "invalid data type" error message to the user.

Use Case: View all customer information	**ID:** 012	**Severity Level:** High
Actor: Logistic Company's Staff	**Type:** Detailed, Essential	
Stakeholders: Logistic Company's Staff – Staff wish to view list of Customer records.		
Description: This use case describe how the user view all the customers' information from the system.		
Trigger: Staff click on the "Customer Information" button which located on the navigation bar.		
Relationships: **Association:** Logistic Company's Staff **Include:** - **Extend:** - **Generalization:** -		
Normal Flow of Events: 1. User clicks on the "Customer Information" button on the navigation bar. 2. System redirect user to the pages which consist a list of Customers. 3. User can filter the Customer list by entering specific keyword on specific column field. 4. System show the narrowed customers' list to the user.		
SubFlows: -		
Alternate/Exceptional Flows: -		

Use Case: Edit a particular customer	**ID:** 013	**Severity Level:** High
Actor: Logistic Company's Staff	**Type:** Detailed, Essential	
Stakeholders: Logistic Company's Staff – Staff want to edit a particular Customer.		
Description: This use case describe how the user edit a particular Customer in the system.		
Trigger: Staff click on the "Customer Information" button which located on the navigation bar.		
Relationships: **Association:** Logistic Company's Staff **Include:** View all customer information **Extend:** - **Generalization:** -		
Normal Flow of Events: 1. Perform use case "View all customer information" 2. User click on the "Pencil" icon to edit the particular customer. 3. User then click on the "Save" button. 4. System update the information to the database and display a success message.		
SubFlows: -		
Alternate/Exceptional Flows: -		

Use Case: Delete a particular customer	**ID: 014**	**Severity Level:** High
Actor: Logistic Company's Staff	**Type:** Detailed, Essential	
Stakeholders: Logistic Company's Staff – Staff want to delete a particular Customer.		
Description: This use case describe how the user delete a particular Customer from the system.		
Trigger: Staff click on the "Customer Information" button which located on the navigation bar.		
Relationships: **Association:** Logistic Company's Staff **Include:** View all customer information **Extend:** - **Generalization:** -		
Normal Flow of Events: 1. Perform use case "View all customer information" 2. User click on the "Trash" icon to delete the particular Customer. 3. System will prompt user to confirm this action. 4. User then click on the "Confirm" button. 5. System delete the particular customer record from the database and display a success message.		
SubFlows: -		
Alternate/Exceptional Flows: -		

Use Case: Add new cargo space	**ID:** 015	**Severity Level:** High
Actor: Logistic Company's Staff	**Type:** Detailed, Essential	

Stakeholders:

Logistic Company's Staff – Staff wants to add new cargo space into the system.

Description:

This use case describe how staff add new cargo space into the system.

Trigger:

- Staff wish to add the truck space that his/her company own into the system.

Relationships:

Association: Logistic Company's Staff

Include: -

Extend: -

Generalization: -

Normal Flow of Events:

1. User click on the "Cargoes Space" button in the navigation bar.
2. User then click on the "Plus" icon beside the header "Add Cargo Space".
3. User enter the details of the cargo space into the system manually and click on "Save" button.
4. System checks and validate for all input data.
5. System store those information into the database.
6. System display a success message to notify user that the process is success.
7. System reset the input field and user is able to enter the next cargo space.

SubFlows:

-

Alternate/Exceptional Flows:

1.1. Alternative option to input the Cargo Space's details. (Admin only)

1.1.1. User import the details of Cargoes space which is in Excel format (.csv).

2.1. User or Admin enter invalid input type.

2.1.1. System display "invalid data type" error message to the user.

Use Case: View all cargo space information	**ID:** 016	**Severity Level:** High
Actor: Logistic Company's Staff	**Type:** Detailed, Essential	

Stakeholders:

Logistic Company's Staff – Staff wish to view list of cargo space records.

Description:

This use case describe how the user view all the cargoes space information from the system.

Trigger: Staff click on the "Cargoes Space" button which located on the navigation bar.

Relationships:

Association: Logistic Company's Staff

Include: -

Extend: -

Generalization: -

Normal Flow of Events:

1. User clicks on the "Cargoes Space" button on the navigation bar.
2. System redirect user to the pages which consist a list of Cargoes Space.
3. User can filter the Cargoes Space list by entering specific keyword on specific column field.
4. System show the narrowed Cargoes Space list to the user.

SubFlows:

-

Alternate/Exceptional Flows:

-

Name: Edit a particular cargo space	**ID:** 017	**Severity Level:** High
Actor: Logistic Company's Staff	**Type:** Detailed, Essential	

Stakeholders:

Logistic Company's Staff – Staff want to edit a particular cargo space.

Description:

This use case describe how the user edit a particular cargo space in the system.

Trigger: Staff click on the "Cargoes Space" button which located on the navigation bar.

Relationships:

Association: Logistic company's Staff

Include: View all cargoes space information

Extend: -

Generalization: -

Normal Flow of Events:

1. Perform use case "View all cargoes space information"
2. User click on the "Pencil" icon to edit the particular cargo space.
3. User then click on the "Save" button.
4. System update the information to the database and display a success message.

SubFlows:

-

Alternate/Exceptional Flows:

-

Use Case: Delete a particular cargo space	**ID:** 018	**Severity Level:** High
Actor: Logistic Company's Staff	**Type:** Detailed, Essential	

Stakeholders:

Logistic Company's Staff – Staff want to delete a particular cargo space.

Description:

This use case describe how the user delete a particular cargo space from the system.

Trigger: Staff click on the "Cargoes Space" button which located on the navigation bar.

Relationships:

Association: Logistic Company's Staff

Include: View all cargo space information

Extend: -

Generalization: -

Normal Flow of Events:

6. Perform use case "View all cargo space information"
7. User click on the "Trash" icon to delete the particular cargo space.
8. System will prompt user to confirm this action.
9. User then click on the "Confirm" button.
10. System delete the particular cargo space's record from the database and display a success message.

SubFlows:

-

Alternate/Exceptional Flows:

-

Use Case: Select truck type	**ID:** 019	**Severity Level:** High
Actor: Logistic Company's Staff	**Type:** Detailed, Essential	

Stakeholders:

Logistic Company's Staff – Staff wish to select the truck type that his company own.

Description:

This use case describe how the user can add different type of truck into the system.

Trigger: Staff click on the "Add" icon which located in "Setting" tab bar.

Relationships:

Association: Logistic Company's Staff

Include: -

Extend: -

Generalization: -

Normal Flow of Events:

1. User click on the "Add" icon which located in "Setting" tab bar, just besides the heading of "Vehicle Selection".
2. System redirect the user to another page which show a list of predefined trucks, containers and other cargoes space.
3. User click on the trucks' type which own by his company.
4. The selected trucks will then display on the "Vehicle Selection" section which is in "Setting" tab bar.
5. User click on the check box to confirm that those vehicles selected will involve in the calculation.

SubFlows:

-

Alternate/Exceptional Flows:

-

Name: Calculate the optimal route	**ID:** 020	**Severity Level:** High
Actor: Logistic Company's Staff	**Type:** Detailed, Essential	

Stakeholders:

Logistic Company's Staff – Staff wants to find the optimal route to deliver cargoes and at the same time maximize the capacity in the truck.

Description:

This use case describe how company's manager find the optimal route to deliver.

Trigger: Staff click on the "Start Calculate" button which located at the Smart D/O tab bar.

Relationships:

Association: Logistic Company's Staff

Include: Maximize the capacity of truck

Extend: -

Generalization: -

Normal Flow of Events:

1. User select a specific range of time.
2. Expire date of D/O which match the specific time range will be display on the dashboard.
3. User is able to remove a particular D/O from the dashboard.
4. After user satisfied with the D/Os displayed, user then click on the "Start Calculate" button.
5. System find and search for the optimal route.
6. Performed use case "Maximize the capacity of truck". (Future Development)
7. Performed use case "View the result of suggested route in SMART D/O dashboard".

SubFlows:

-

Alternate/Exceptional Flows:

-

Name: View the result of suggested route in SMART D/O dashboard.	**ID:** 021	**Severity Level:** High
Actor: Logistic Company's Staff	**Type:** Detailed, Essential	
Stakeholders: Logistic Company's Staff – Staff wants to view the optimal route that had been calculated by the system.		
Description: This use case describe how the system display the generated route to user.		
Trigger: Staff click on the "Start Calculate" button on the "Result" tab bar.		
Relationships: **Association:** Logistic Company's Staff **Include:** Calculate the optimal route. **Extend:** - **Generalization:** -		
Normal Flow of Events: 1. Perform use case "Calculate the optimal route". 2. After the optimal route is selected, system display the result on the SMART D/O which is located at the "SmartD/O" tab.		
SubFlows: -		
Alternate/Exceptional Flows: -		

Use Case: View a particular previous planned route	**ID:** 022	**Severity Level:** High
Actor: Logistic Company's Staff	**Type:** Detailed, Essential	
Stakeholders: Logistic Company's Staff – Staff wish to view a particular previous planned route.		
Description: This use case describe how the user view the list of the previous record and the information of each record.		
Trigger: Staff click on the "Route Record" button which located on the navigation bar.		
Relationships: **Association:** Logistic Company's Staff **Include:** - **Extend:** - **Generalization:** -		
Normal Flow of Events: 1. User clicks on the "Route Record" button on the navigation bar. 2. System redirect user to the pages which consist of those previous route. 3. User search for a particular route by selecting a specific date. 4. System show the particular date's route to the user. 5. User click on the "Eye" icon which located at the back of the particular route. 6. System show the detailed information of that particular route.		
SubFlows: -		
Alternate/Exceptional Flows: -		

Use Case: Delete a particular previous planned route	**ID:** 023	**Severity Level:** High
Actor: Logistic Company's Staff	**Type:** Detailed, Essential	

Stakeholders:

Logistic Company's Staff – Staff wish to delete a particular previous planned route.

Description:

This use case describe how the user delete a particular previous planned route.

Trigger: Staff click on the "Route Record" button which located on the navigation bar.

Relationships:

Association: Logistic Company's Staff

Include: -

Extend: -

Generalization: -

Normal Flow of Events:

1. User clicks on the "Route Record" button on the navigation bar.
2. System redirect user to the pages which consist of those previous route.
3. User search for a particular route by selecting a specific date.
4. System show the particular date's route to the user.
5. User click on the "Trash" icon which located at the back of the particular route.
6. System will prompt user to confirm this action
7. User then click on the "Confirm" button.
8. System delete the particular route record from the database and display a success message.

SubFlows:

-

Alternate/Exceptional Flows:

-

Use Case: Add and enable staff to access the system.	**ID:** 024	**Severity Level:** High
Actor: Company Admin/ Company Owner	**Type:** Detailed, Essential	

Stakeholders:

Company Admin/ Company Owner – Company Admin wish to add staffs to access the route optimization system with company's information (D/O, item, customer, cargo).

Description:

This use case describe how company admin or company owner add the staff into the system and allow them to access company information.

Trigger: Company Admin clicked on the "Staff" button which located at navigation bar.

Relationships:

Association: Company Admin/ Company Owner

Include: -

Extend: -

Generalization: -

Normal Flow of Events:

1. User login with the company admin account.
2. Company Admin clicked on the "Staff" button which located at the navigation bar.
3. Company Admin clicked on the "Add People" icon.
4. Company Admin enter staff detail with staff's validated email.
5. Company Admin then clicked on the "Create" button.
6. System displayed a message to notify the user the action has been success.

SubFlows:

-

Alternate/Exceptional Flows:

2.1. User enter invalid input type.

2.1.1. System display "invalid data type" error message to the user.

Use Case: View staff's information	**ID:** 025	**Severity Level:** High
Actor: Company Admin/ Company Owner	**Type:** Detailed, Essential	
Stakeholders: Company Admin/ Company Owner – Company Admin wish to view staffs' information.		
Description: This use case describe how company admin or company owner add view the staffs' information.		
Trigger: Company Admin clicked on the "Staff" button which located at navigation bar.		
Relationships: **Association:** Company Admin/ Company Owner **Include:** - **Extend:** - **Generalization:** -		
Normal Flow of Events: 1. User login as a Company Admin. 2. Company Admin clicks on the "Staff" button on the navigation bar. 3. System redirect user to the pages which consist a list of staffs. 4. Company Admin can filter the Staff list by entering specific keyword on specific column field. 5. System show the narrowed staff's list to the admin.		
SubFlows: -		
Alternate/Exceptional Flows: -		

Use Case: Change staff's status	**ID:** 026	**Severity Level:** High
Actor: Company Admin/ Company Owner	**Type:** Detailed, Essential	

Stakeholders:

Company Admin/ Company Owner – Company Admin wish to active/deactivate a particular staff's account.

Description:

This use case describe how company admin or company owner activate or deactivate the staff's account.

Trigger: Company Admin clicked on the "Staff" button which located at navigation bar.

Relationships:

Association: Company Admin/ Company Owner

Include: -

Extend: -

Generalization: -

Normal Flow of Events:

1. User login as a Company Admin.
2. Company Admin clicks on the "Staff" button on the navigation bar.
3. System redirect user to the pages which consist a list of staffs.
4. Company Admin clicks on the "tick" icon or "cross" icon which located at the status column.
   - 4.1. "Tick" icon indicated the particular staff is able to access the system by using he/she email address.
   - 4.2. "Cross" icon indicated the particular staff is not able to access the system using the particular email address.

SubFlows:

-

Alternate/Exceptional Flows:

-

4.6 System Architecture Diagram

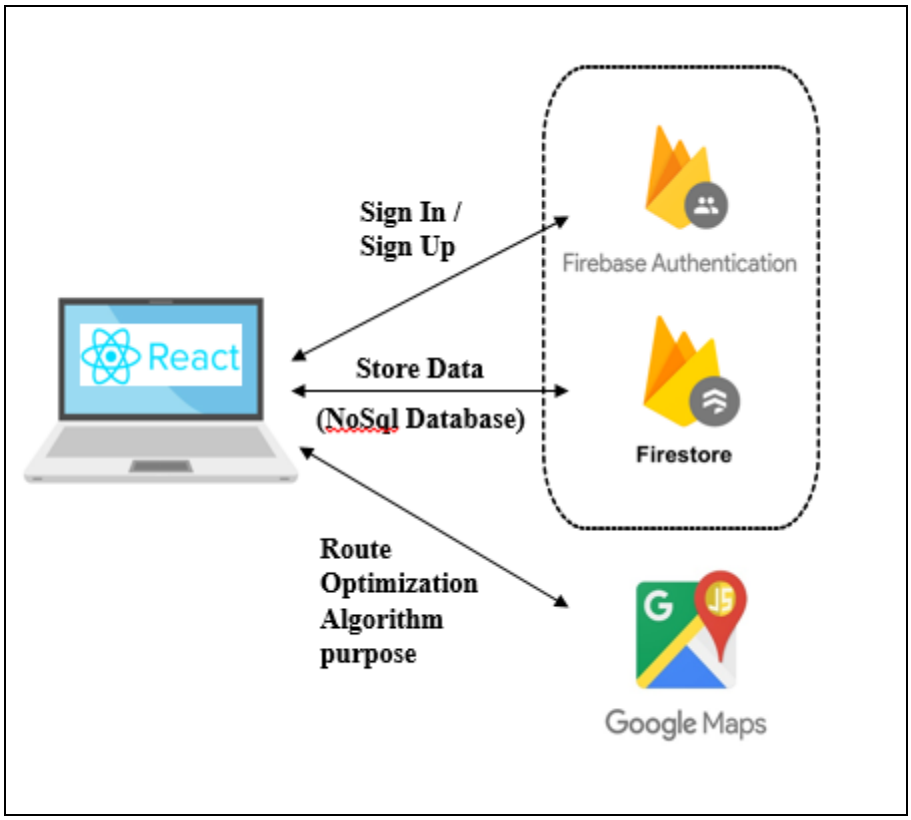


Figure 4.2 System Architecture Design (Project Compilation)

React which is known as a Javascript web-framework had been used for the front-end development for this route optimization system. React has been selected for this project due to it is simple and easy to learn. React also provides a huge range of libraries which are reusable during the implementation phase and this actually help developer too speed up their process instead of creating those components from the sketch. The duration for this project is short, thus react is suitable for this project due to it simple and reusable characteristic. Besides, React framework can also use as a medium to develop the backend logic which uses Javascript to call those online services like Firebase and Google Map Api.

In this project, two services from the firebase had been used for the development of this web-based system. One of them is Firebase Authentication which used to validate whether the particular account is valid for this system, else user can also sign up their account by using this

Firebase Authentication. Another service that had been used in this project is Cloud Firestore which serves as the system's database to store user's data in NoSQL structure.

Some services from Google Map Api also had been used in this project with the purpose of developing an algorithm to find the shortest route to deliver. Those services from the Google Map API are Map Javascript API which used to render the Map on the screen. Direction API which used to calculate the total route distance and travelled time after the sequence to deliver had been done by the algorithm. Direction API also helps to visualize the route and the sequence to deliver on the map. In this system, Geocoding API is used to retrieve the coordinate of a particular location. Thus, the system is able to get the estimated distance from one location to another location in the route optimization algorithm.

After using react on this project, I found out that there is no proper standard guideline on how should developer structure their file for a project. It is quite flexible and easy to manage at the beginning, but when the system getting complex those files may become messy and hard to maintain. Thus, developer should find out on how others structure theirs react project before starting to implement. Although there are many open sources libraries which developer can reuse during their implementation, developer must aware that some of the libraries had already been depreciated and out of support from their owner. Besides, those reusable libraries are also hard to customize and may not fit to your project. In conclusion, I learned React in a hard way but still prefer it for development due to its reusability characteristic and it can speed up the development process after you familiar with it.

4.7 ERD Diagram

4.7.1 Logical ERD Diagram

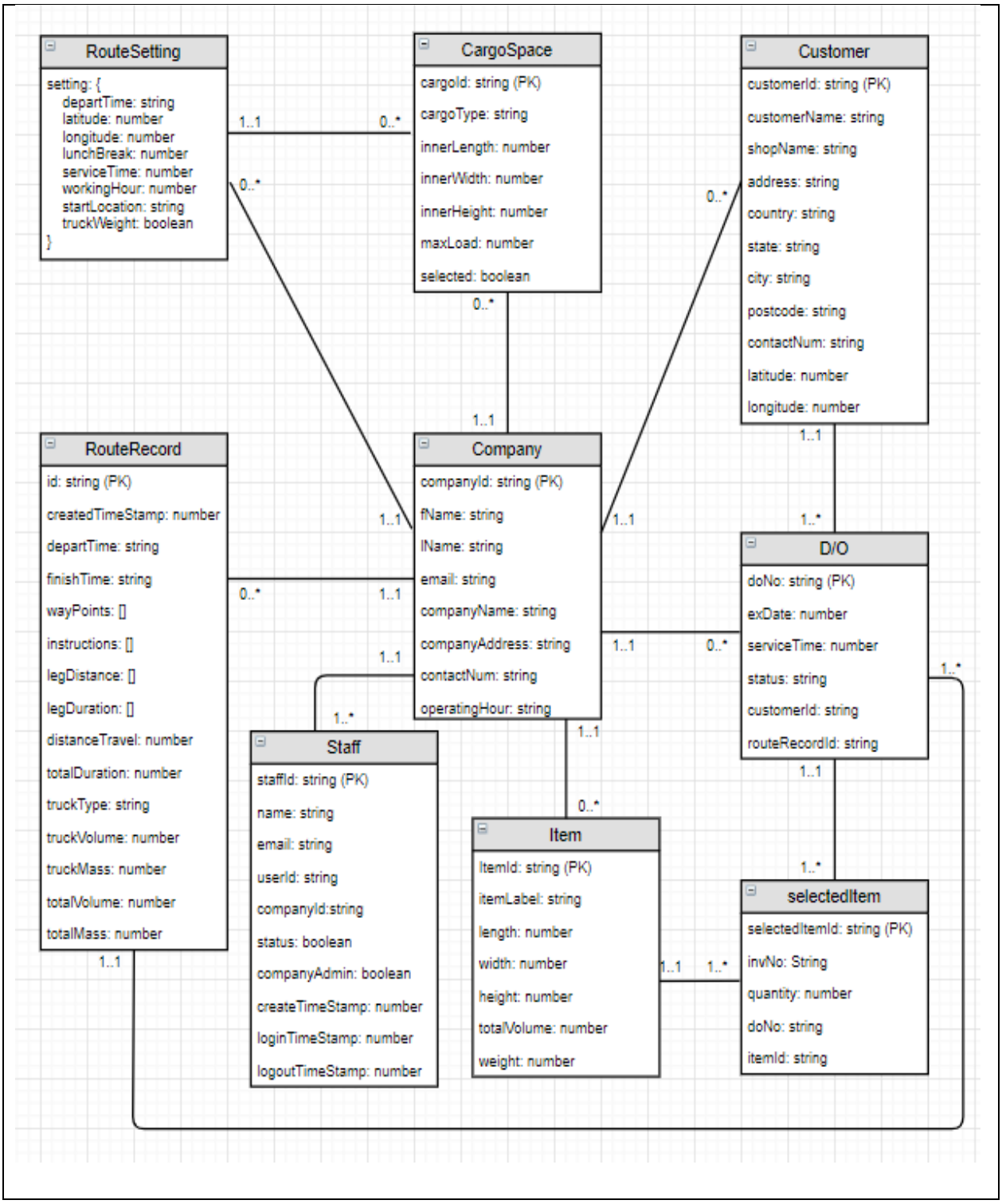


Figure 4.3 Logical ERD Diagram

4.7.2 Physical ERD Diagram

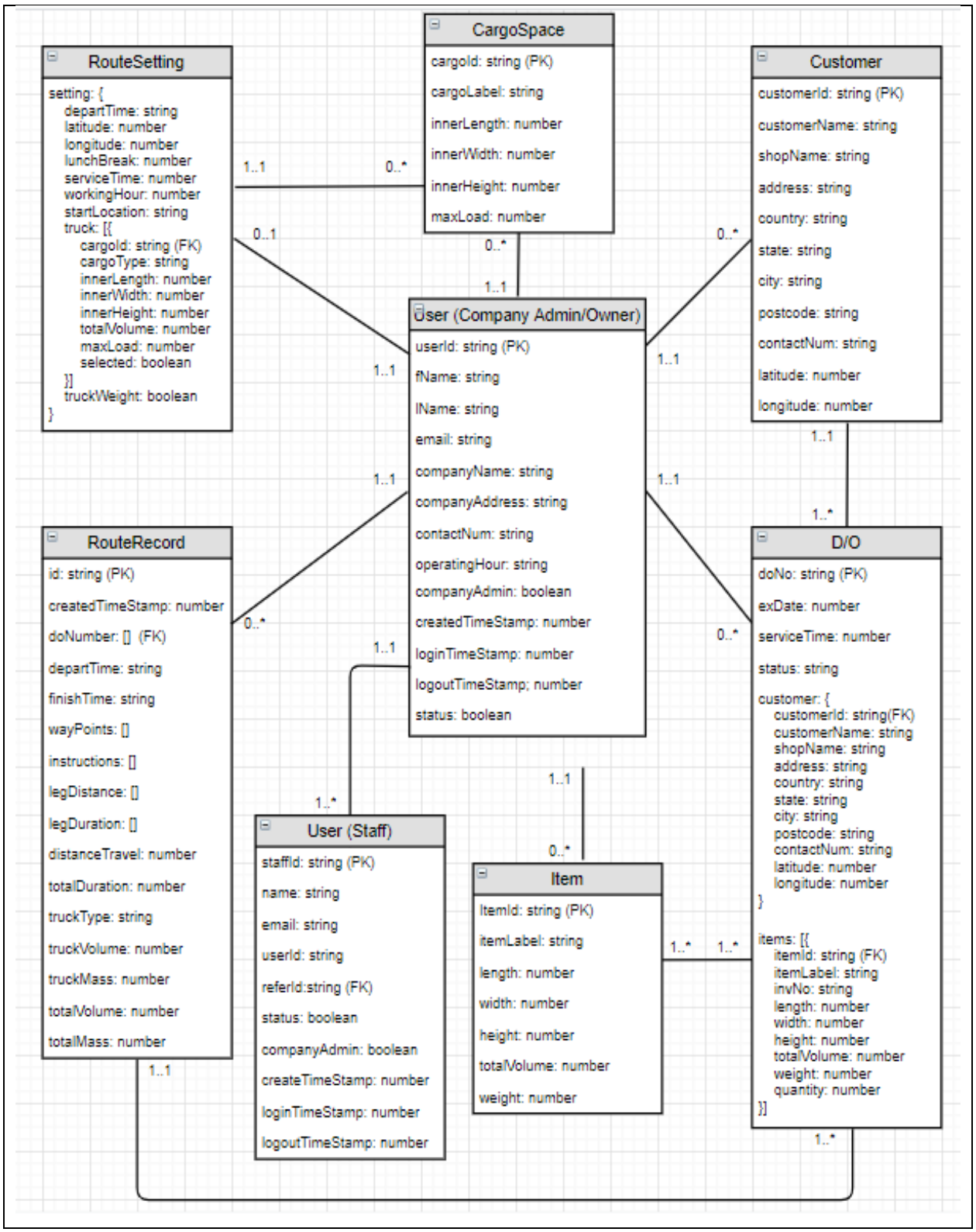


Figure 4.4 Physical ERD Diagram

For this project, the attributes of customer and also the attributes of the item are embedded into the D/O's collection which can be seen by comparing the logical ERD diagram from Figure 4.3 with the physical ERD diagram from Figure 4.4. This is due to the customer information and also the item information is always retrieved and displayed together with the D/O information. Thus, having them combine together may have a better performance on the speed of retrieving the data and also the process of filtering and combining of different tables can be neglected.

Besides, the cargo space's attributes are embedded into the RouteSetting's collection. This is due to system may need those cargo space information together with the RouteSetting's information to calculate on the route to deliver when performing route optimizer algorithm. It is best to retrieve them together from the database with only one query. Therefore, processing time can largely be reduced.

There are two roles exists in this system, which are the company's staff and also the company admin or owner. Both of them are located under the same user collection, the difference between both of them is that some of the fields in the collection are different from each other. For example, the company collection which exists at the logical ERD diagram had been embedded into the company admin / owner's document. The reason company detail is embedded under company admin / owner document is because the logistics company is also treated as the system's user just that it may need a person to in charge of that particular account. The person who has this account is treated as a company admin or company owner who has the permission to create an account for their staff. And those staff's account that had been created under this account is allowed to access the company information like D/O, item, customer and cargoes to perform optimization on the route to deliver.

4.7.3 Data Dictionary

Table 4.1 User (Company Admin/ Owner) Data Dictionary

Attributes	Description	Data Type	PK/FK	Nulls
userId	Unique identifier for user.	string	PK	N
fName	First name of the company admin / company owner.	string		N
lName	Last name of the company admin / company owner.	string		N
Email	Company email.	string		N
companyName	The company name.	string		N
companyAddress	The full address of the company.	string		N
contactNum	The contact number of the company.	string		N
operatingHour	The operating hour of the company.	string		N
status	Status of the user to access the system (activate/deactivate)	boolean		N
companyAdmin	Indicator to check user's role.	boolean		N
createdTimeStamp	Timestamp when the user account is created to access the system.	number		N
loginTimeStamp	Timestamp when the user login into the system.	number		N
logoutTimeStamp	Timestamp when the user logout the system.	number		N

Table 4.2 User (Staff) Data Dictionary

Attributes	Description	Data Type	PK/FK	Nulls
staffId	Unique identifier for a particular staff.	string	PK	N
name	Name of the staff.	string		N
email	Email of the staff.	string		N
userId	A unique id given be firebase when account is created.	string		N
referId	Id which reference to the company account.	string	FK	N
status	Status of the user to access the system (activate/deactivate)	boolean		N
companyAdmin	Indicator to check user's role.	boolean		N
createdTimeStamp	Timestamp when the user (staff) account is created to access the system.	number		N
loginTimeStamp	Timestamp when the user (staff) login into the system.	number		N
logoutTimeStamp	Timestamp when the user (staff) logout the system.	number		N

Table 4.3 Customer's Data Dictionary

Attributes	Description	Data Type	PK/FK	Nulls
customerId	Unique identifier for company's customer.	string	PK	N
customerName	Customer's name.	string		N
shopName	Customer's shop.	string		N
address	The full address of customer's shop.	string		N
country	The country where the shop allocated.	string		N
state	The state where the shop allocated.	string		N
city	The city where the shop allocated.	string		N
postcode	The postcode of the shop allocated.	string		N
contactNum	The contact number of the customer.	string		N
latitude	The latitude of the customer's shop.	number		N
longitude	The longitude of the customer's shop.	number		N

Table 4.4 Item's Data Dictionary

Attributes	Description	Data Type	PK/FK	Nulls
itemId	The unique identifier for a particular item.	string	PK	N
itemLabel	The label/ name of the item.	string		N
length	The length of the package (box) of the item.	number		N
width	The width of the package (box) of the item.	number		N
height	The height of the package (box) of the item.	number		N
totalVolume	Total volume of the particular tem.	number		N
weight	The weight of the particular item.	number		N

Table 4.5 CargoSpace's Data Dictionary

Attributes	Description	Data Type	PK/FK	Nulls
cargoId	The unique identifier for a particular cargo space.	string	PK	N
cargoLabel	The label /type of the cargo space (prefer to be the plate number of the truck).	string		N
innerLength	The inner length of a particular truck.	number		N
innerWidth	The inner width of a particular truck.	number		N

innerHeight	The inner height of a particular truck.	number		N
maxLoad	The maximum weightage that the truck can afford.	number		N

Table 4.6 D/O's Data Dictionary

Attributes	**Description**	**Data Type**	**PK/FK**	**Nulls**
doNo	The unique identifier for a particular D/O.	string	PK	N
exDate	The due date of the D/O that must be sent.	number		N
serviceTime	The estimate service time that set by the manager for the driver to unload the inventories.	number		N
status	The status of the particular D/O. Can be ("Delivered"/"Pending"/Cancelled).	string		N
customer	The information of a particular customer that the inventories will be sent to.	object	FK (customerId)	N
items	The information of the items that will be delivered in this D/O.	array of objects	FK (itemId)	N

Table 4.7 RouteSetting's Data Dictionary

Attributes	Description	Data Type	PK/FK	Nulls
departTime	The time where the truck leave the warehouse/company.	string	PK	N
latitude	The latitude of the starting location.	number		N
longitude	The longitude of the starting location.	number		N
lunchBreak	The lunch time that the manager set for the driver.	number		N
serviceTime	The default service time to unload the inventories from the truck.	number		N
workingHour	The operation hour of the company.	number		N
startLocation	The starting location of the route.	string		N
truck	The information of the trucks/cargoes space that enable user to select. The selected trucks will be involved in the route optimization algorithm.	array of objects	FK (cargoId)	N
truckWeight	Field that used to check weather user want to respect the truck weight when there are still available spaces but the total mass of inventories involved had exceeded the maximum weightage of the truck.	boolean		N

Table 4.8 RouteRecord's Data Dictionary

Attributes	Description	Data Type	PK/FK	Nulls
id	Unique identifier for a particular planned route record.	string	PK	N
createdTimeStamp	The unique time stamp created when the route is calculated. Used to let user know when a particular route had been generated.	number		N
doNumber	An array of doNo to indicate which D/O had been involved in a particular route record.	array	FK (doNo)	N
departTime	The starting time of the truck to deliver those inventories.	string		N
finishTime	The time where the truck return back to the starting location.	string		N
wayPoints	The location (customer's location) where the truck stop to unload the inventories.	array		N
instructions	The directions provide for driver to reach all customers' location then return back to the origin start point.	array		N
legDistance	The distance from on location to another location.	array		N
legDuration	The time needed to travel from one location to another location.	array		N
distanceTravel	The total distance travelled for a particular planned route.	number		N
totalDuration	The total duration used to deliver all inventories to the customer then back to the starting location. (may include the service time and also the breaking hour)	number		N
truckType	The truck that used to deliver those inventories for a particular planned route.	string		N

truckVolume	The total volume of the truck involved.	number		N
truckMass	The total mass that the involved truck can afford.	number		N
totalVolume	The total volume that will be occupied by the inventories.	number		N
totalMass	The total mass that will be occupied by the inventories.	number		N

4.8 Data Flow Diagram

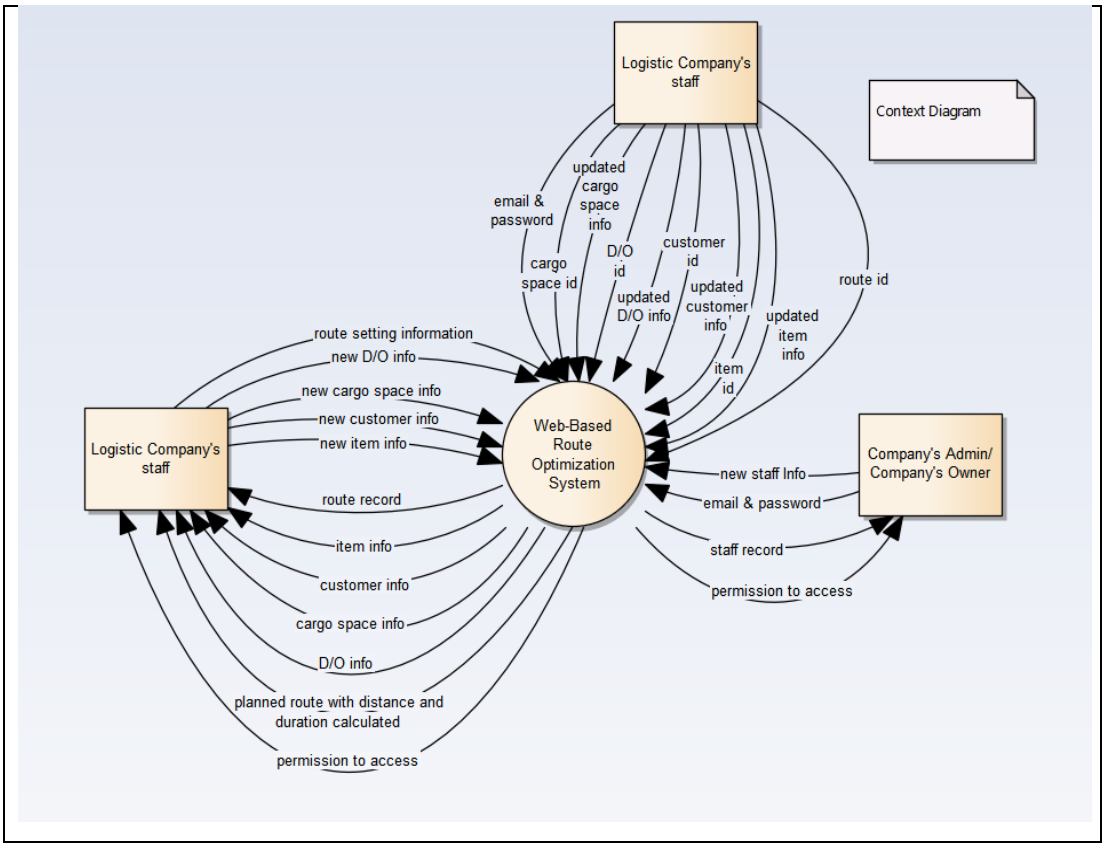


Figure 4.5 Context Diagram

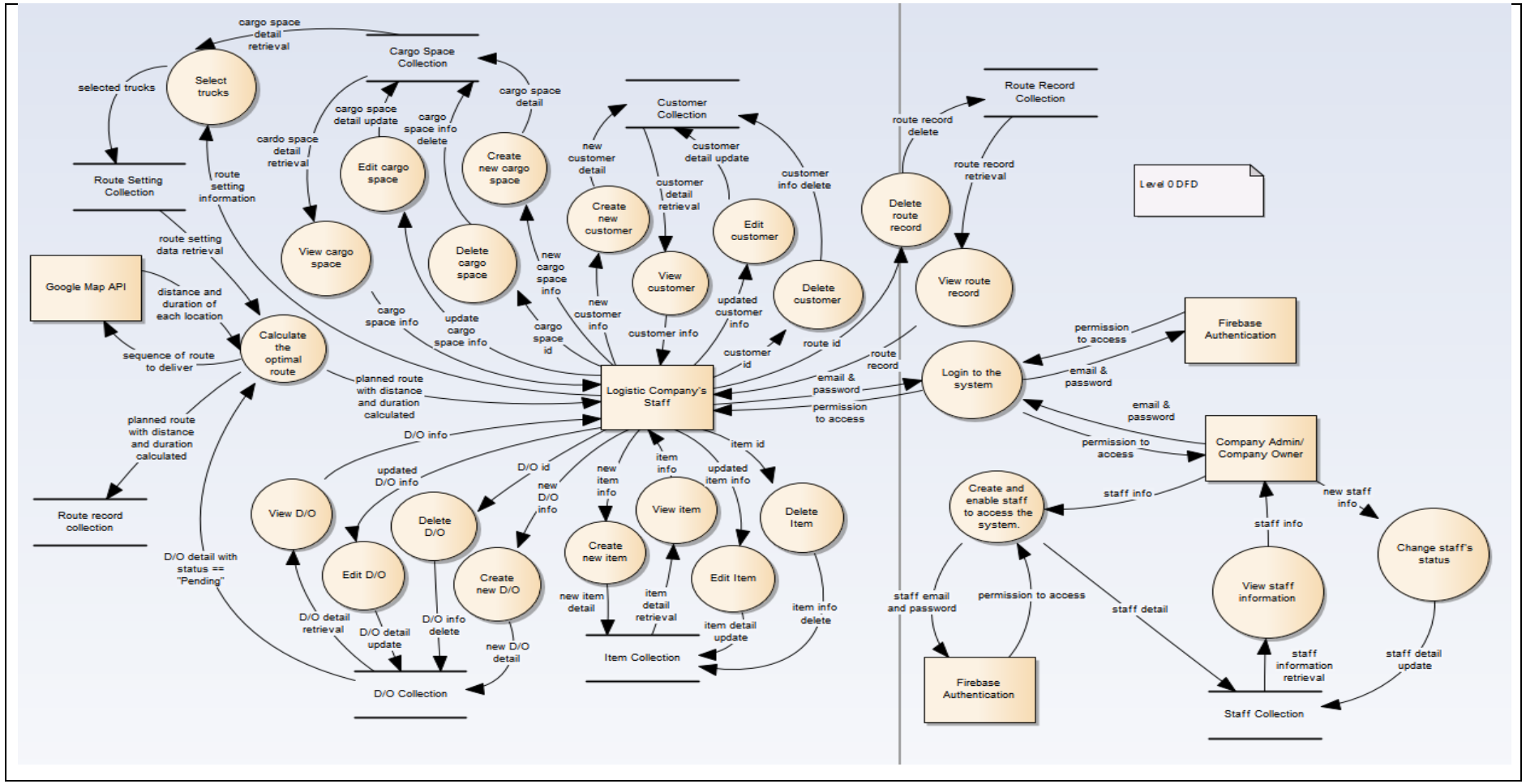


Figure 4.6 Level 0 DFD

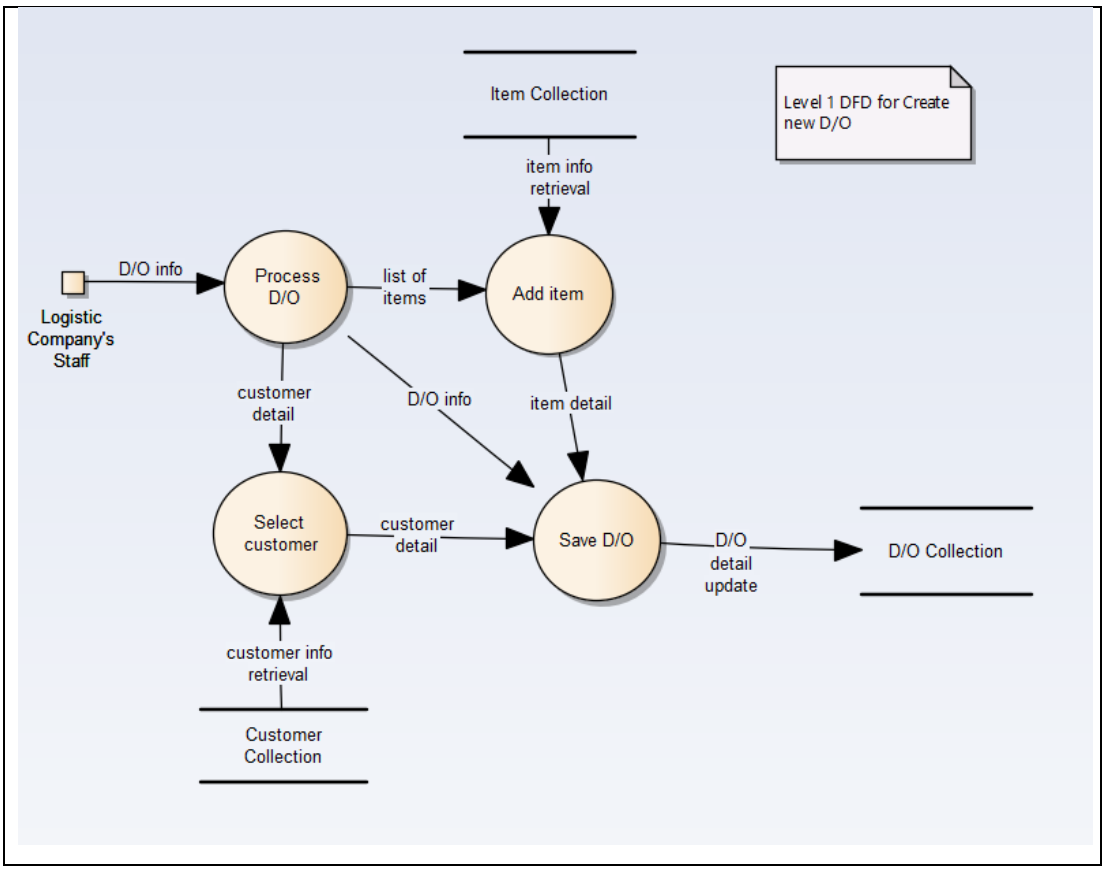


Figure 4.7 Level 1 DFD for Create new D/O

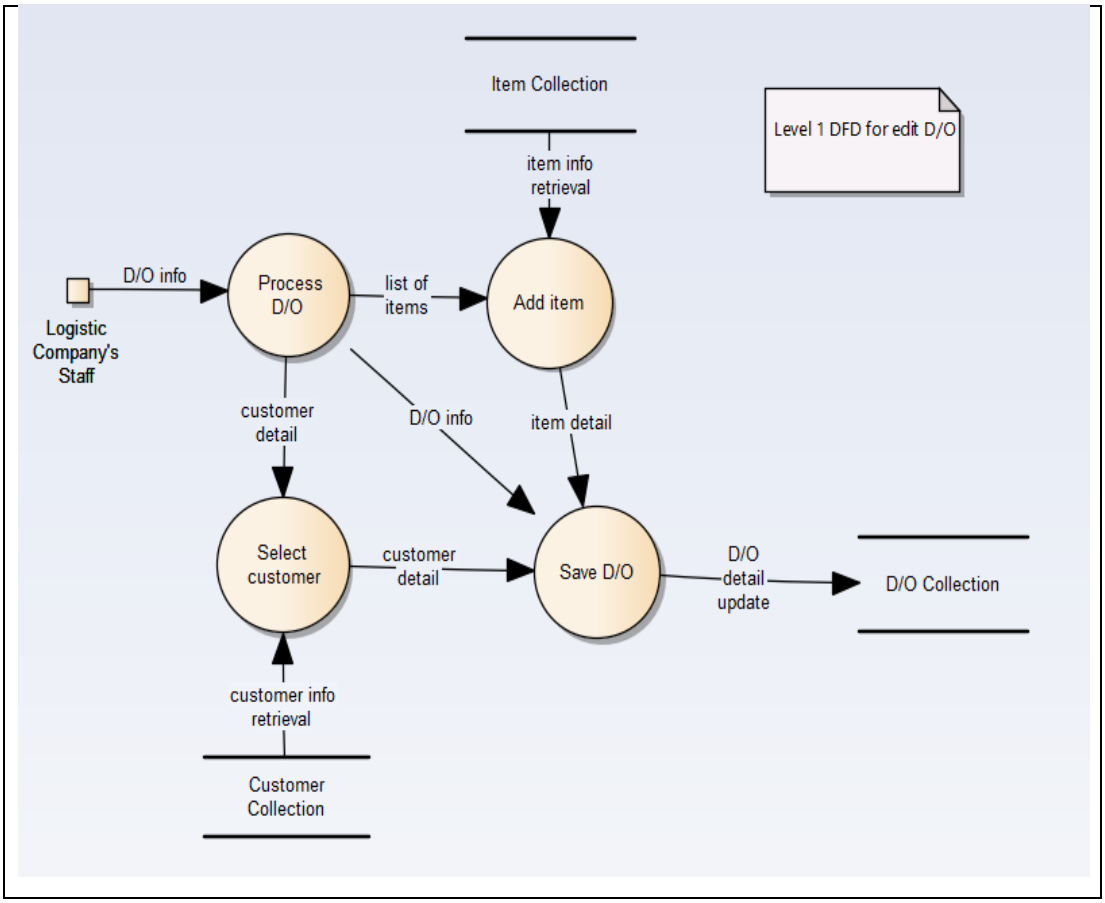


Figure 4.8 Level 1 DFD for Edit D/O

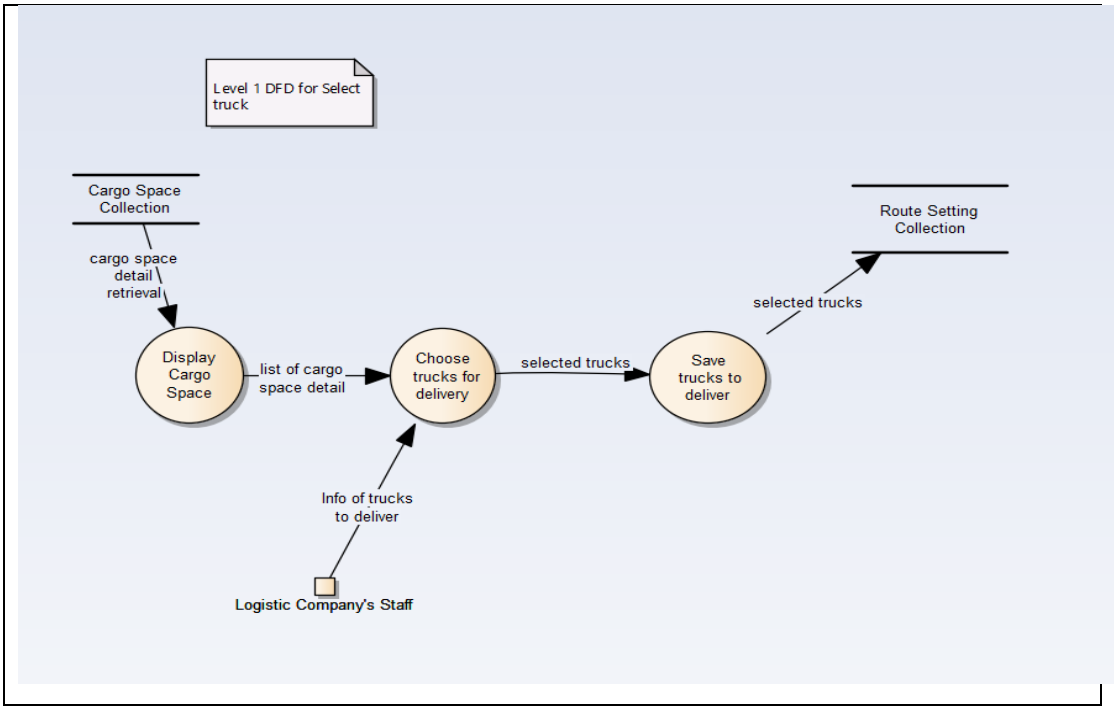


Figure 4.9 Level 1 DFD for Select truck

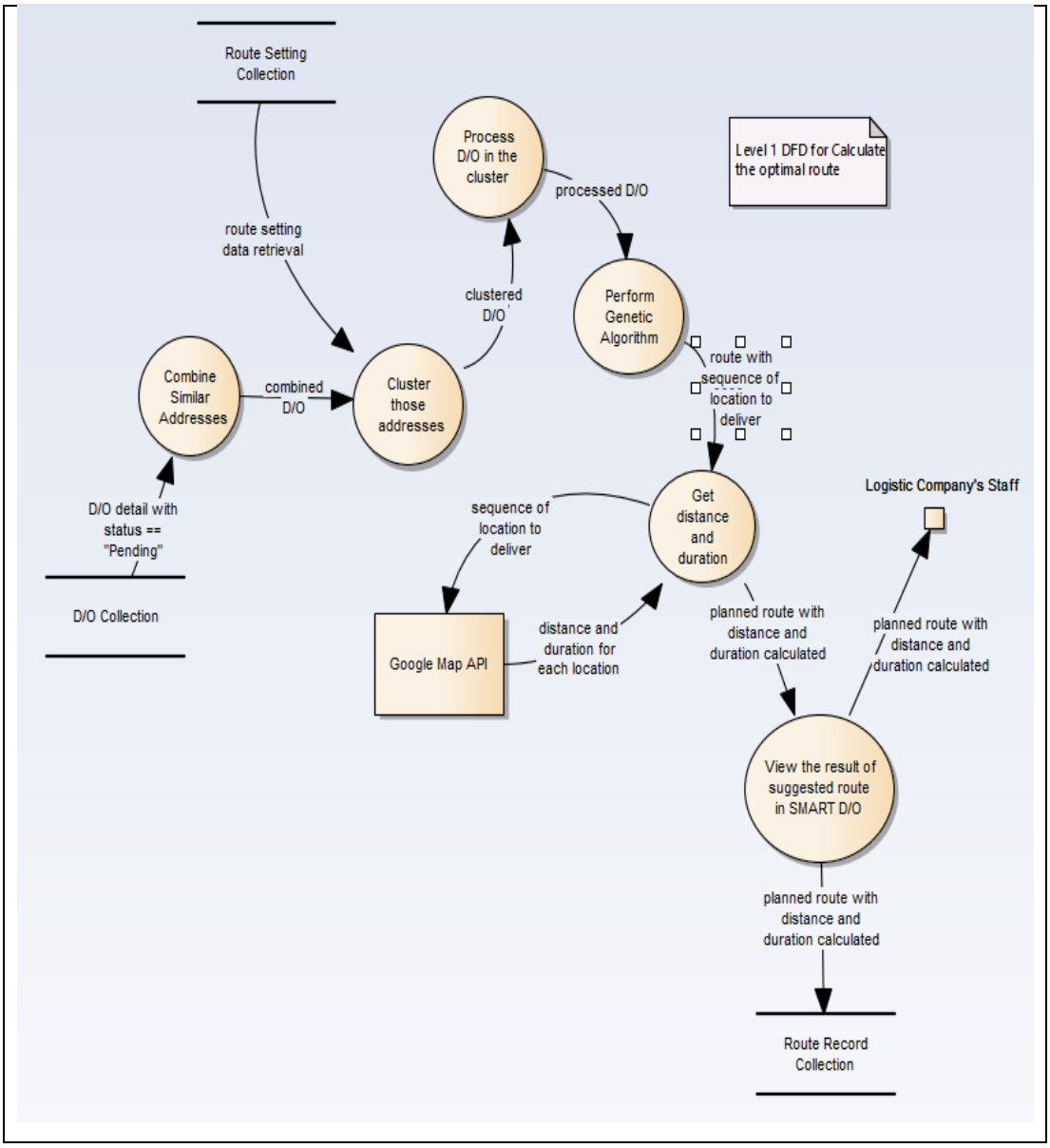


Figure 4.10 Level 1 DFD for Calculate the optimal route

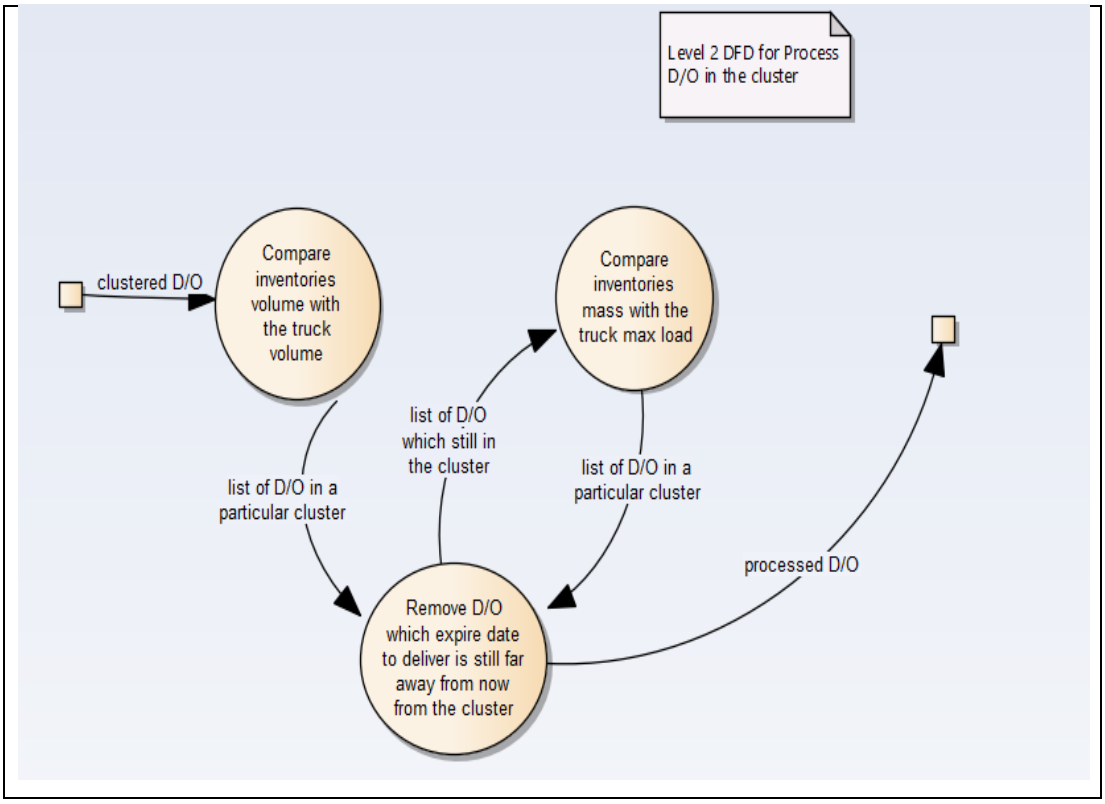


Figure 4.11 Level 2 DFD for Process D/O in the cluster

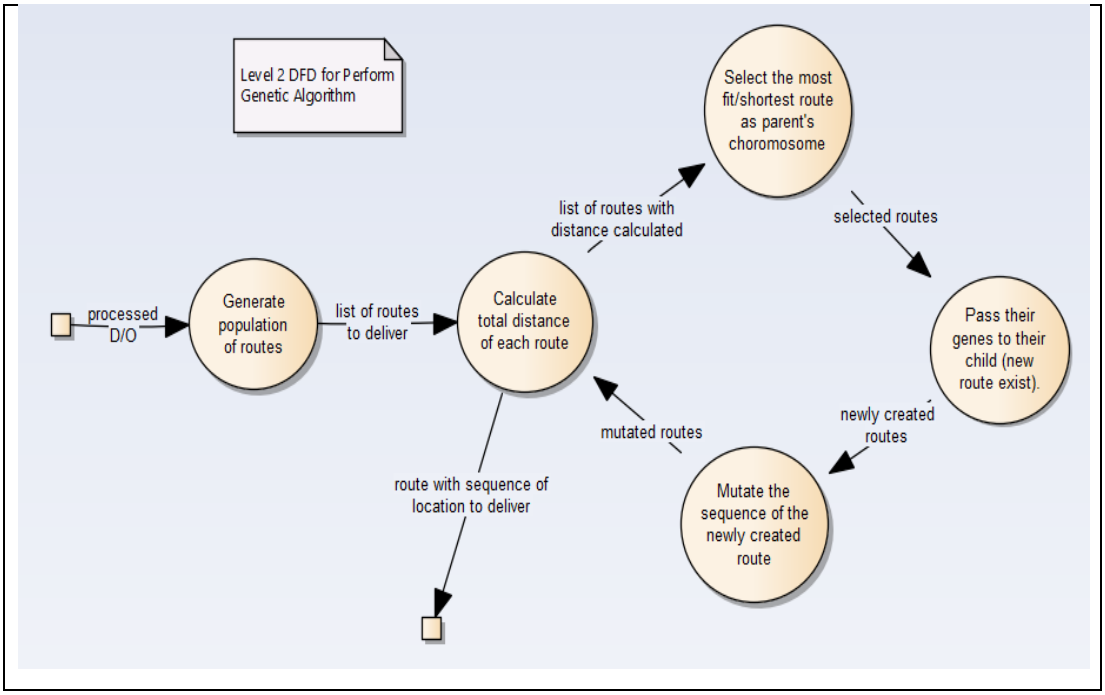


Figure 4.12 Level 2 DFD for Perform Genetic Algorithm

4.9 Activity Diagram

4.9.1 "Login to system" Activity Diagram

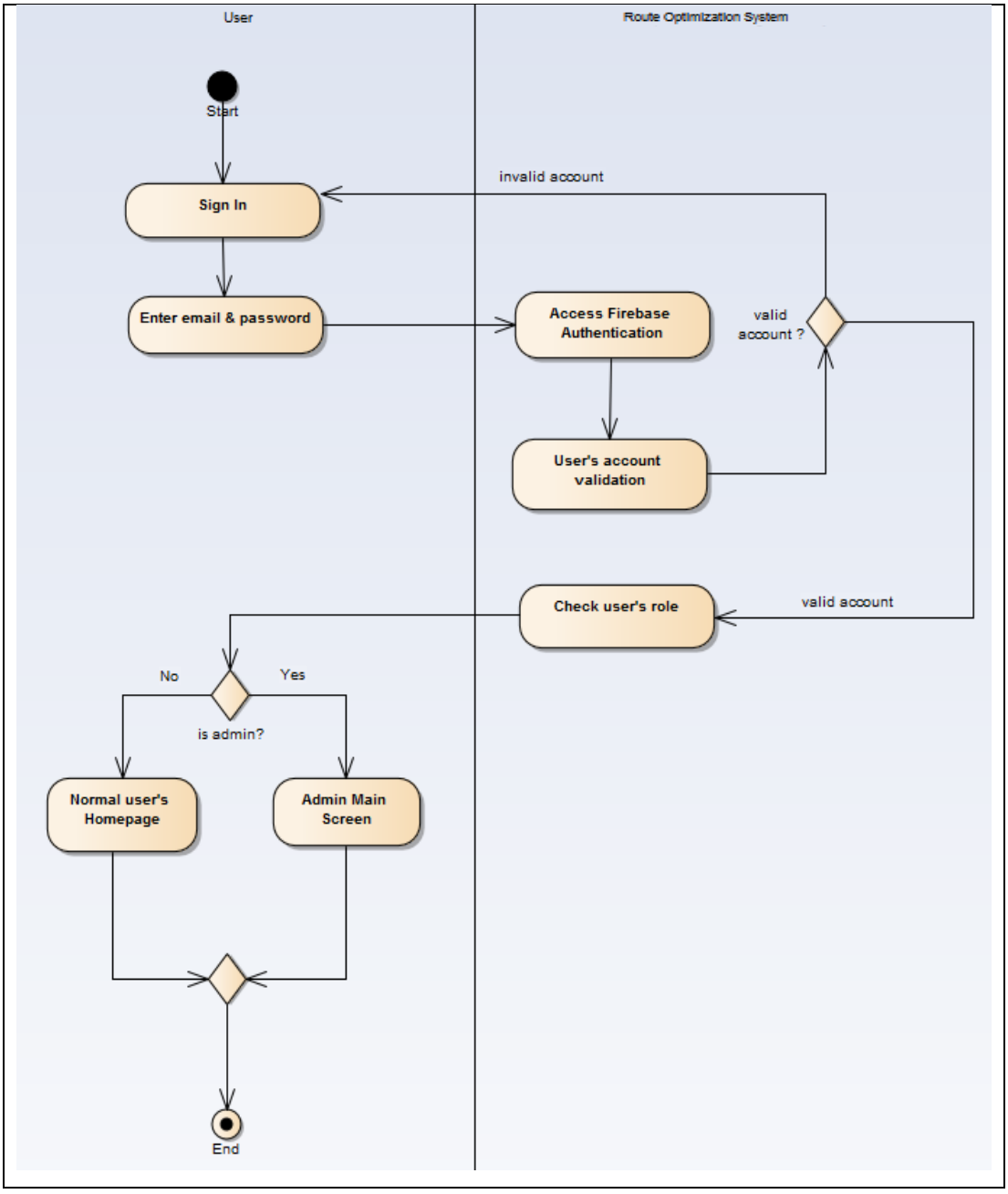


Figure 4.13 "Login to system" Activity Diagram

4.9.2 "Add new D/O" Activity Diagram

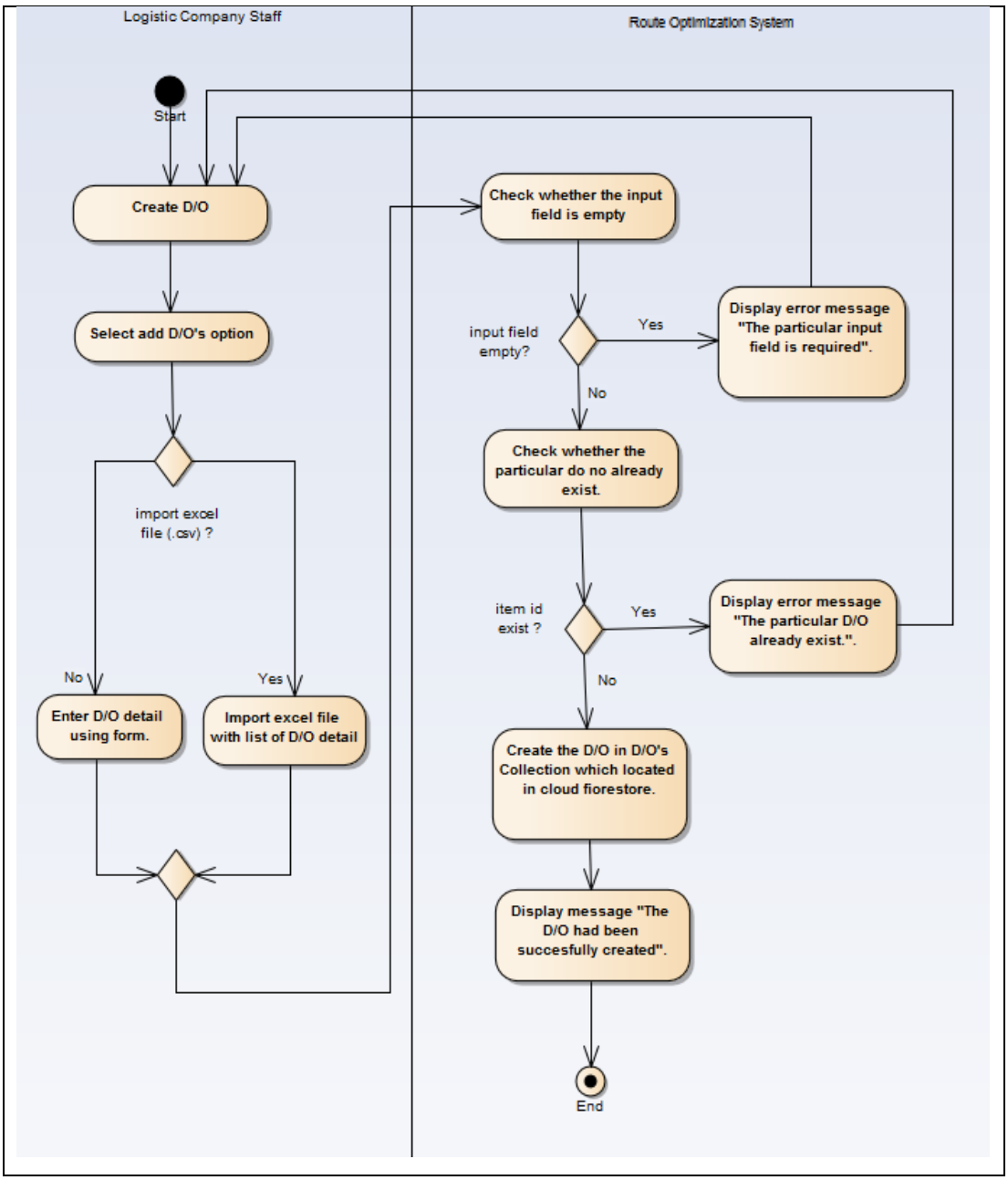


Figure 4.14 "Add new D/O" Activity Diagram

4.9.3 “View D/O” Activity Diagram

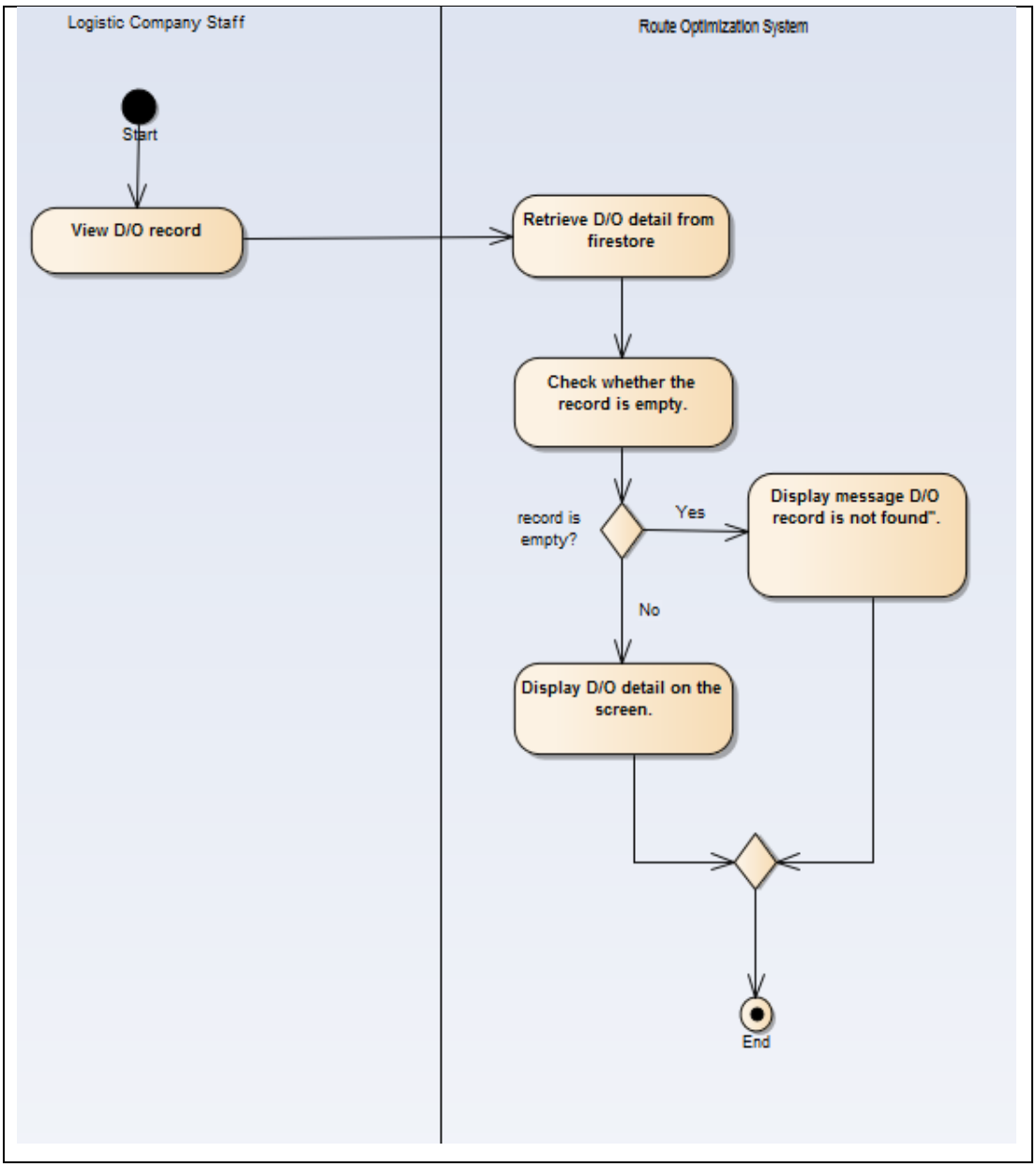


Figure 4.15 “View D/O” Activity Diagram

4.9.4 "Edit a particular D/O" Activity Diagram

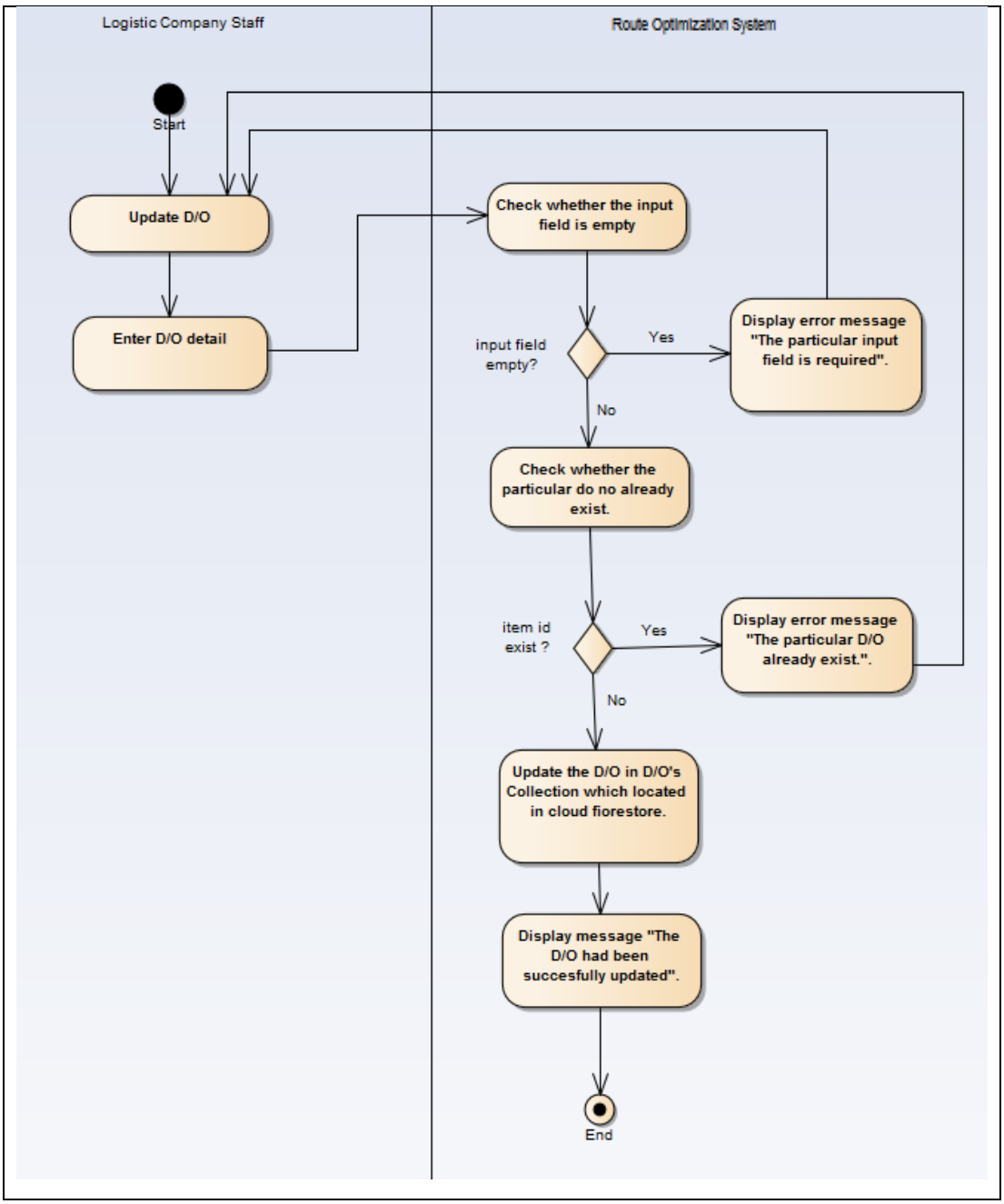


Figure 4.16 "Edit a particular D/O" Activity Diagram

4.9.5 “Delete a particular D/O” Activity Diagram

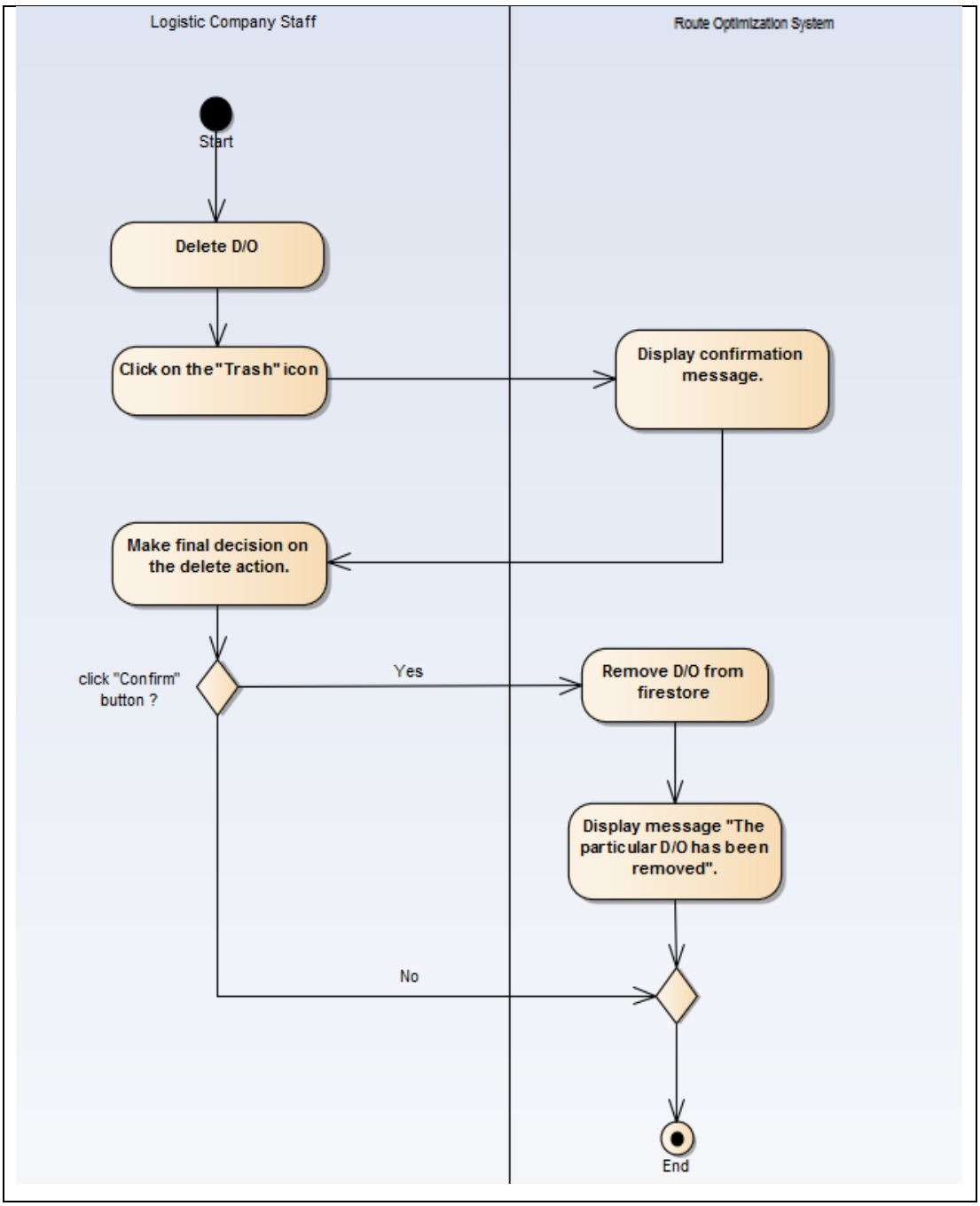


Figure 4.17 “Delete a particular D/O” Activity Diagram

4.9.6 "Add new Item" Activity Diagram

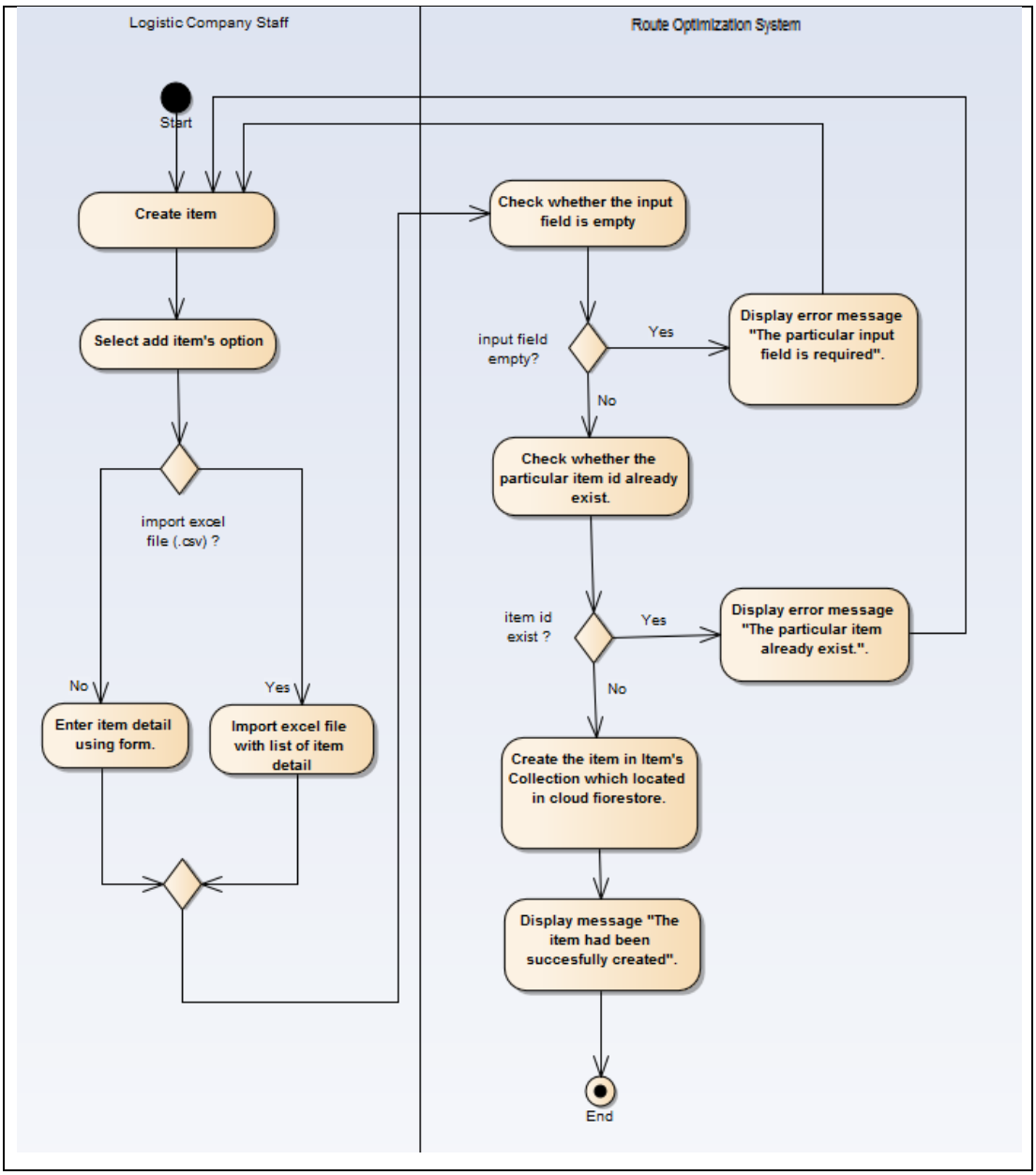


Figure 4.18 "Add new Item" Activity Diagram

4.9.7 “View Item” Activity Diagram

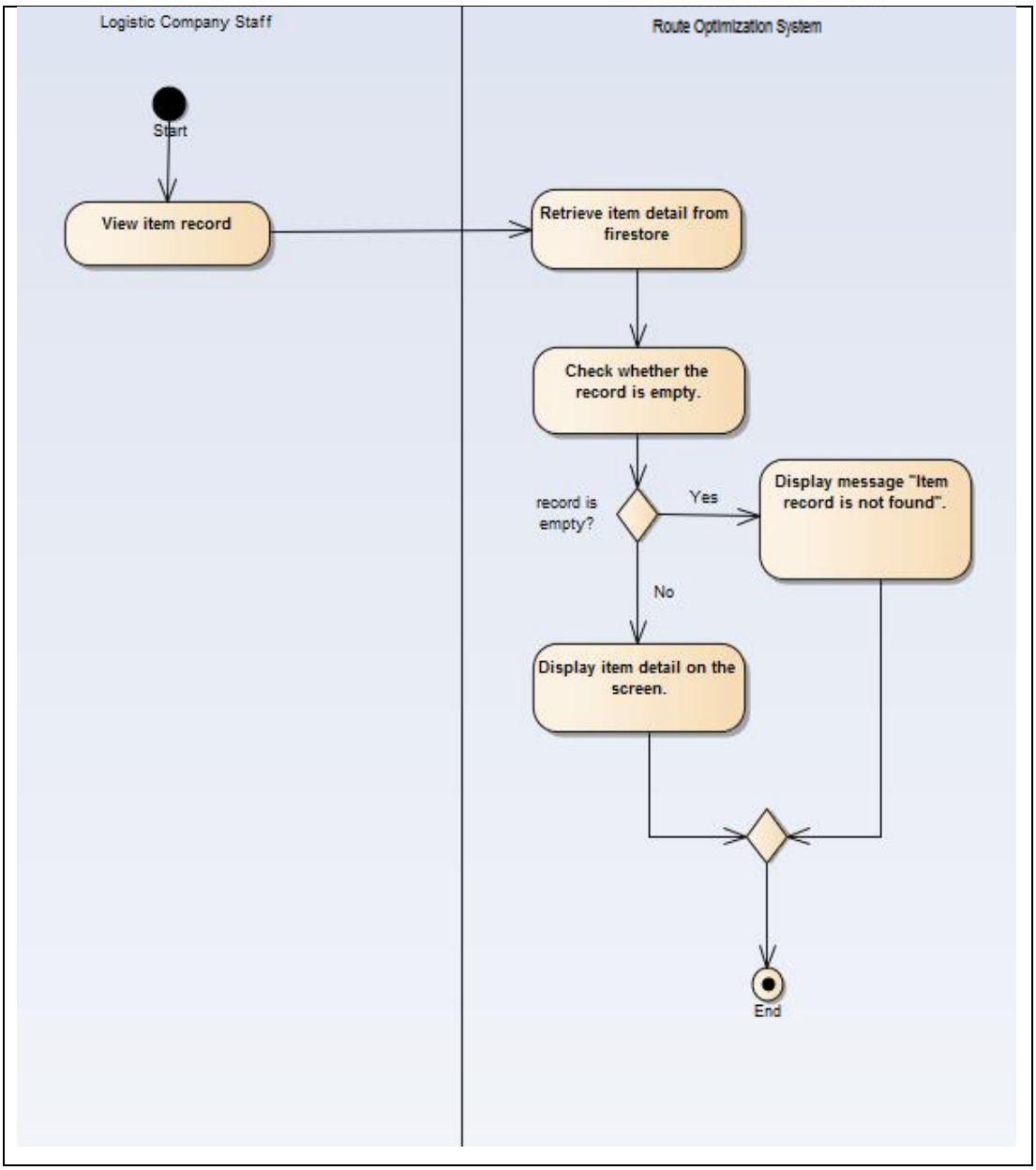


Figure 4.19 “View Item” Activity Diagram

4.9.8 “Edit a particular Item” Activity Diagram

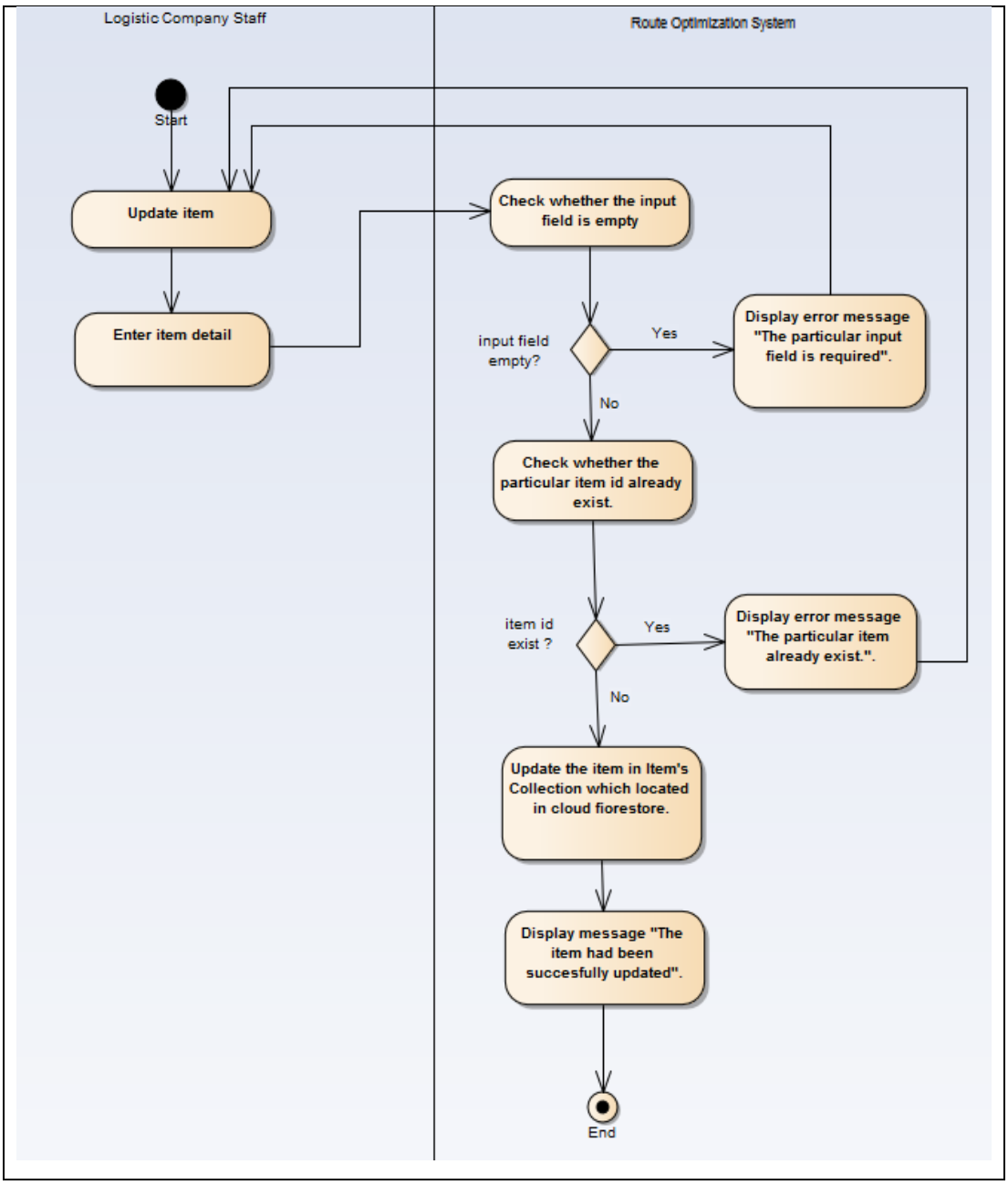


Figure 4.20 “Edit a particular Item” Activity Diagram

4.9.9 “Delete a particular Item” Activity Diagram

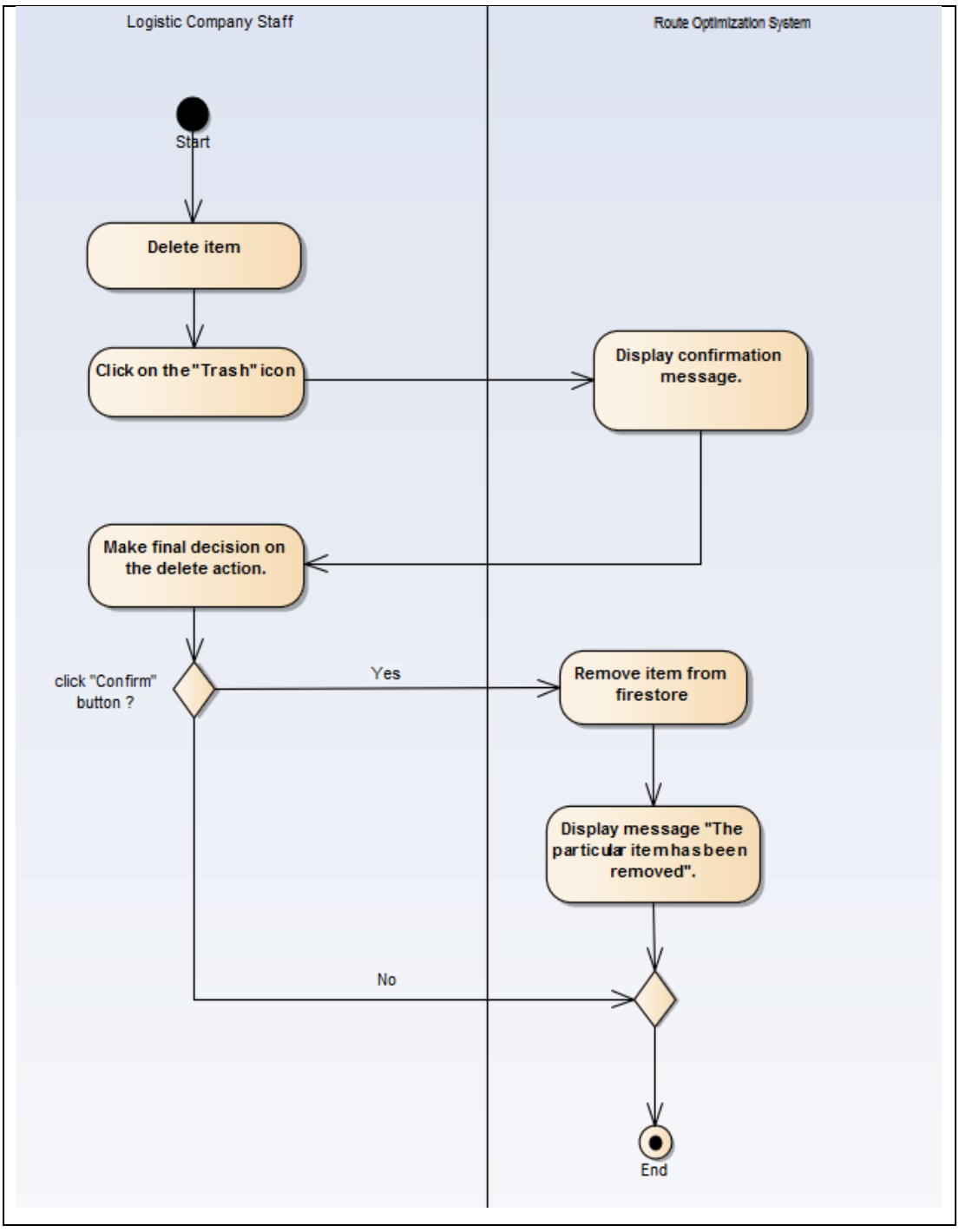


Figure 4.21 “Delete a particular Item” Activity Diagram

4.9.10 "Add new Customer" Activity Diagram

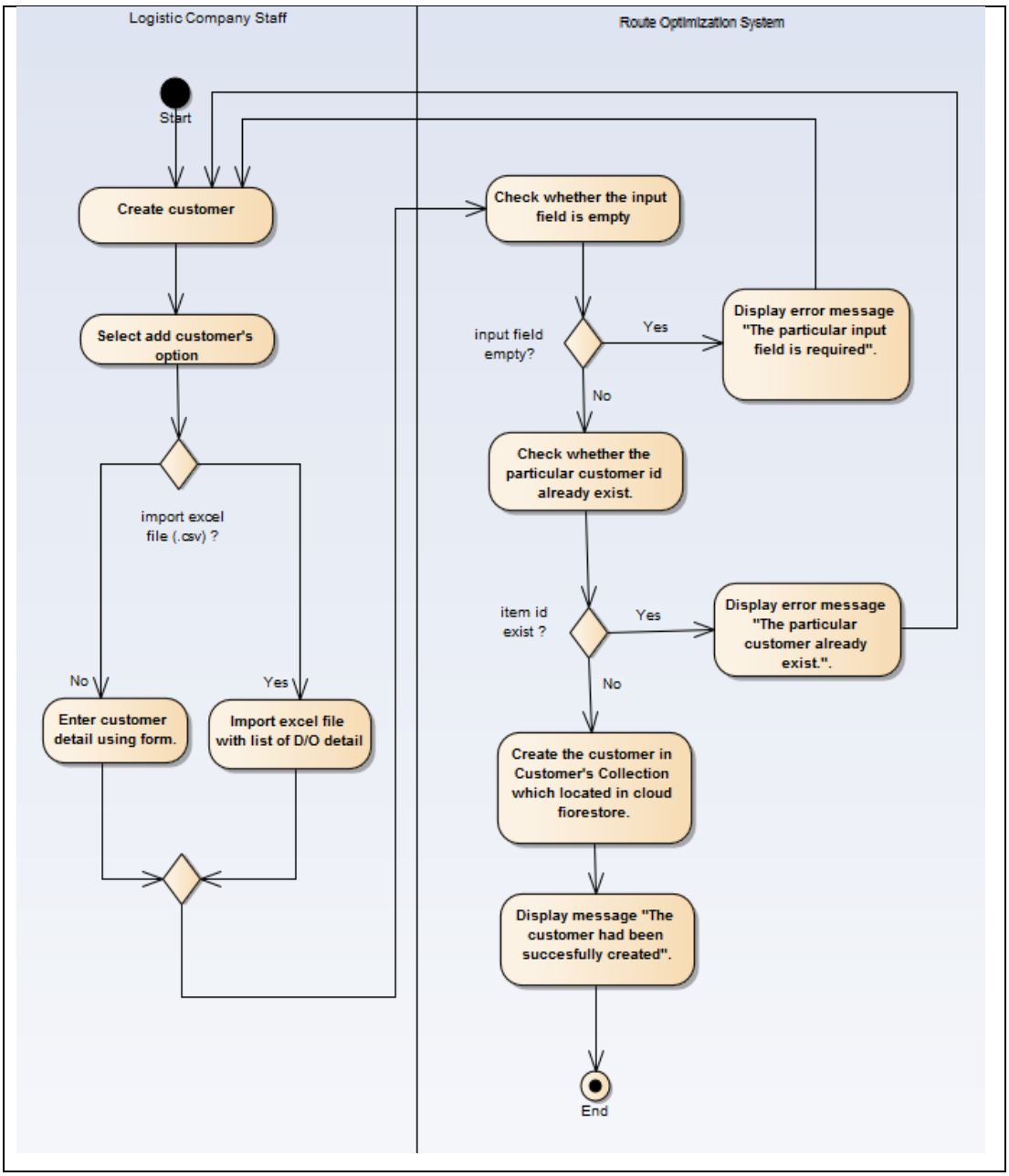


Figure 4.22 "Add new Customer" Activity Diagram

4.9.11 “View Customer” Activity Diagram

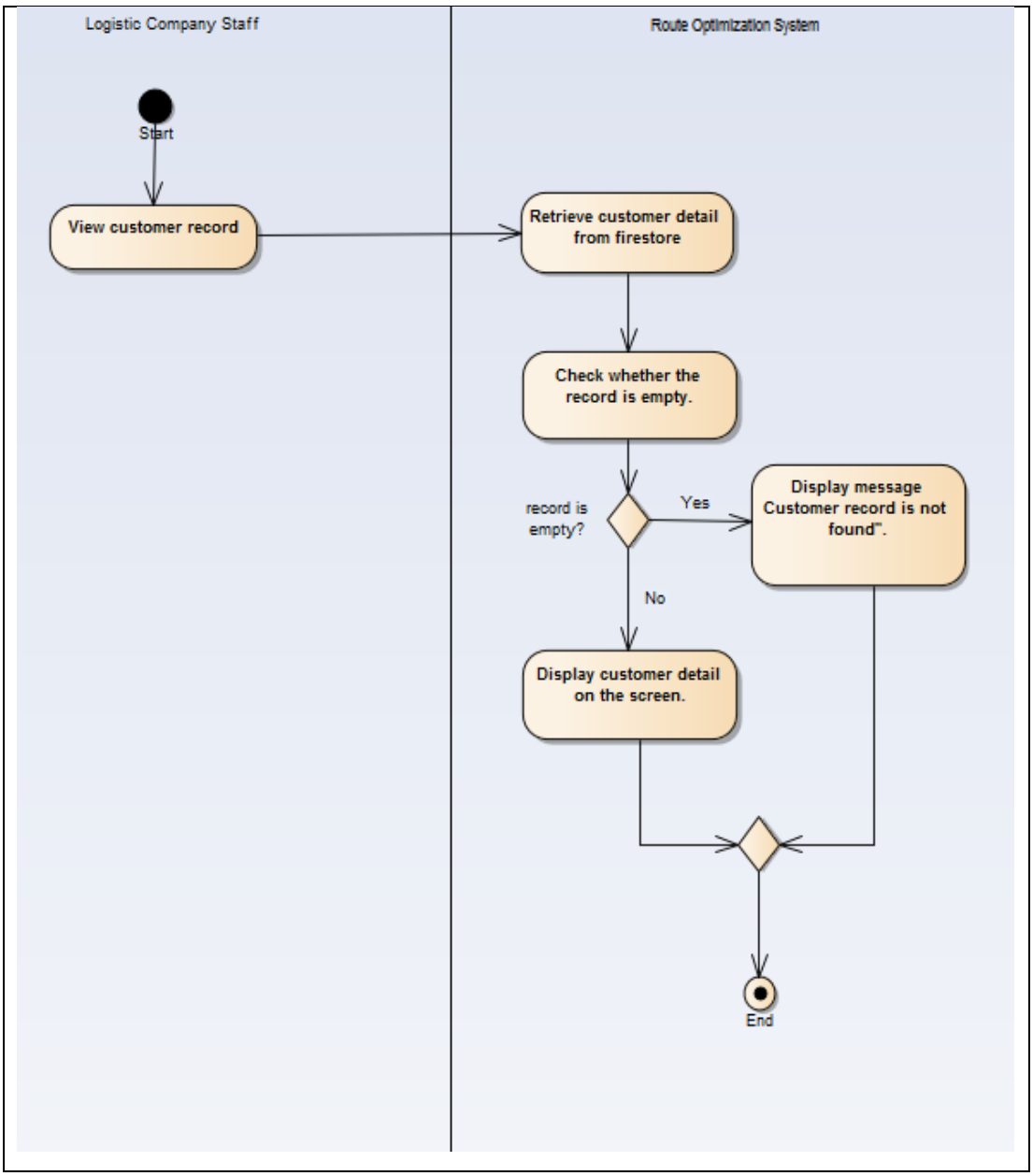


Figure 4.23 “View Customer” Activity Diagram

4.9.12 "Edit a particular Customer" Activity Diagram

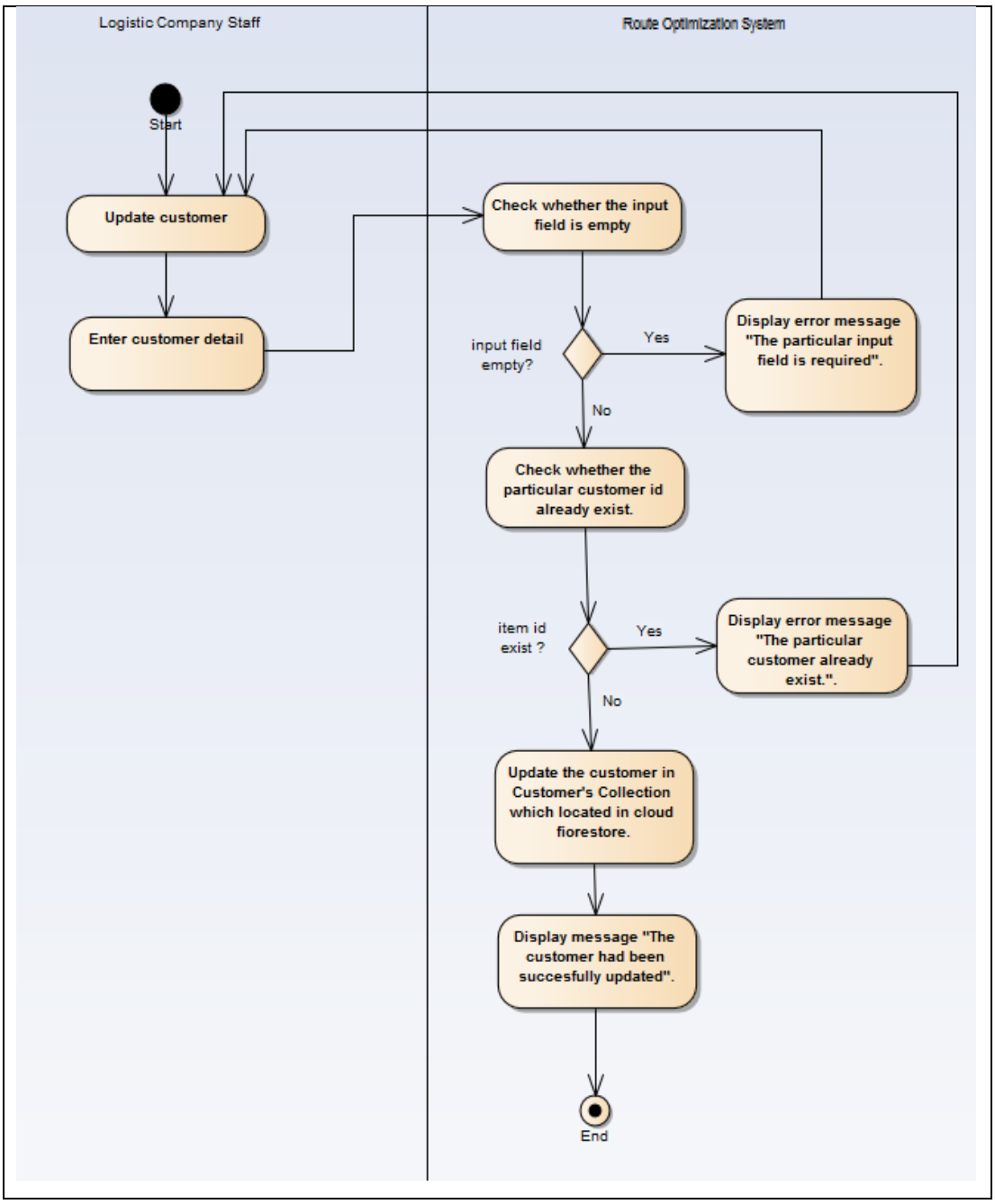


Figure 4.24 "Edit a particular Customer" Activity Diagram

4.9.13 “Delete a particular Customer” Activity Diagram

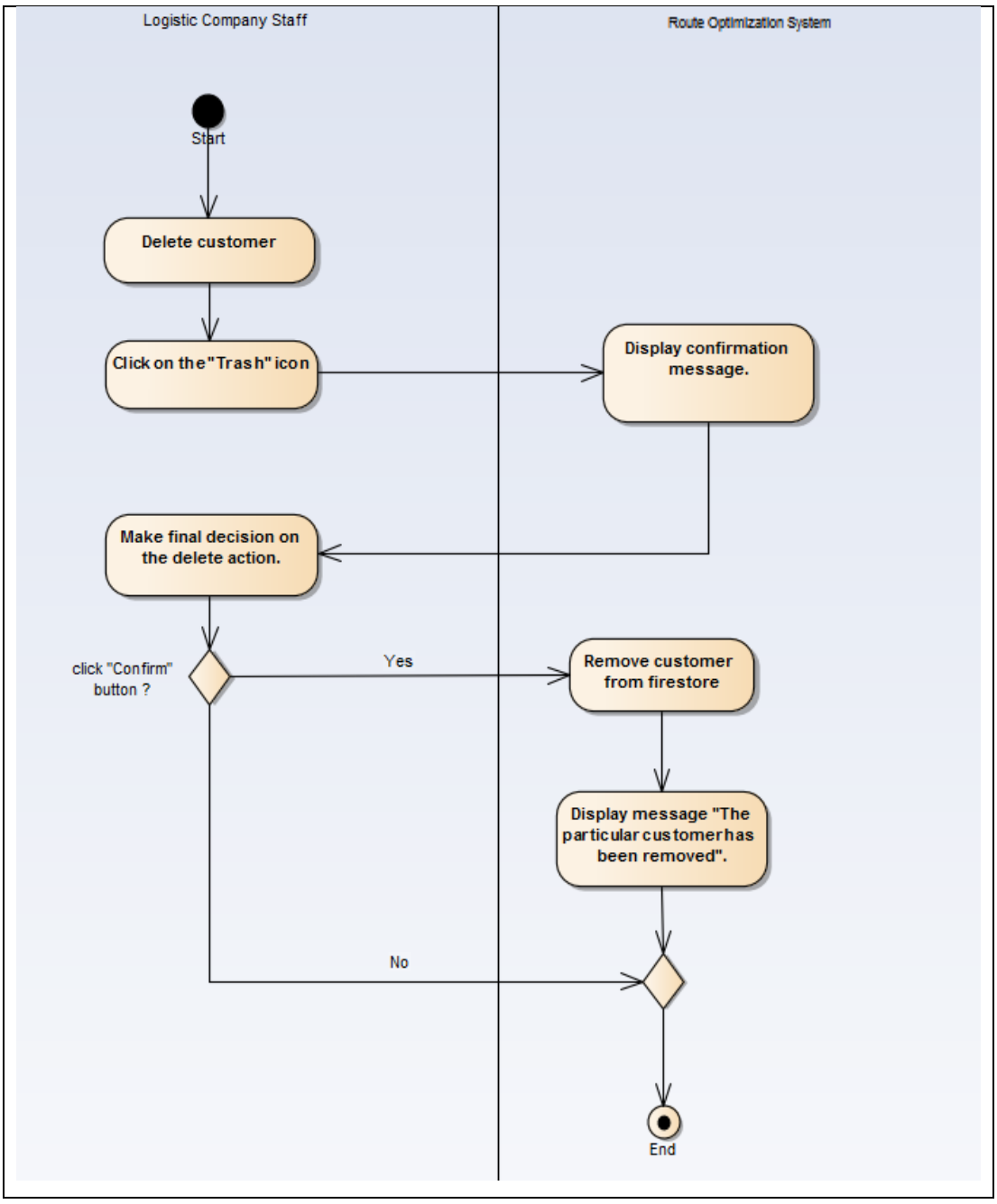


Figure 4.25 “Delete a particular Customer” Activity Diagram

4.9.14 “Add new Cargo Space” Activity Diagram

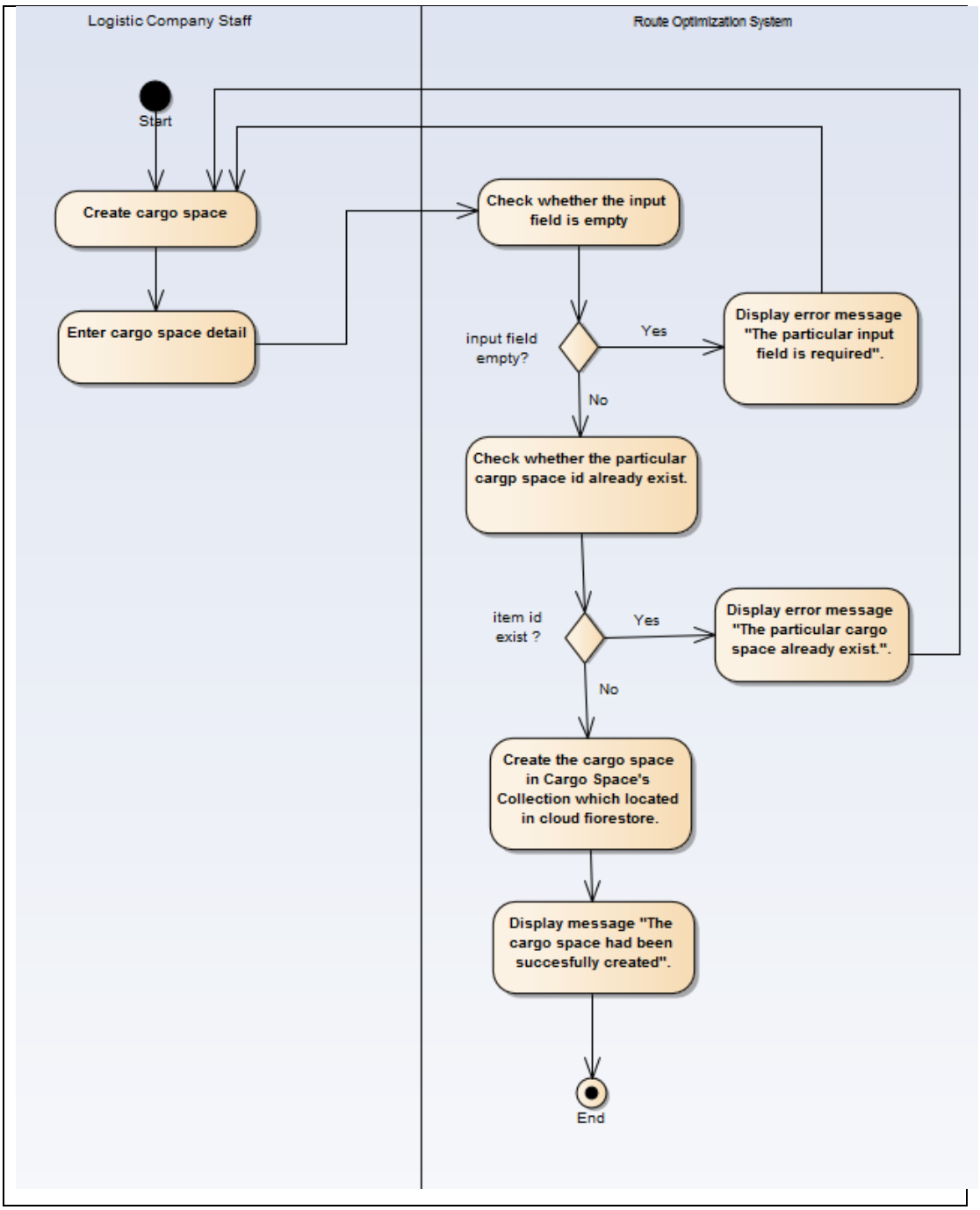


Figure 4.26 “Add new Cargo Space” Activity Diagram

4.9.15 "View Cargo Space" Activity Diagram

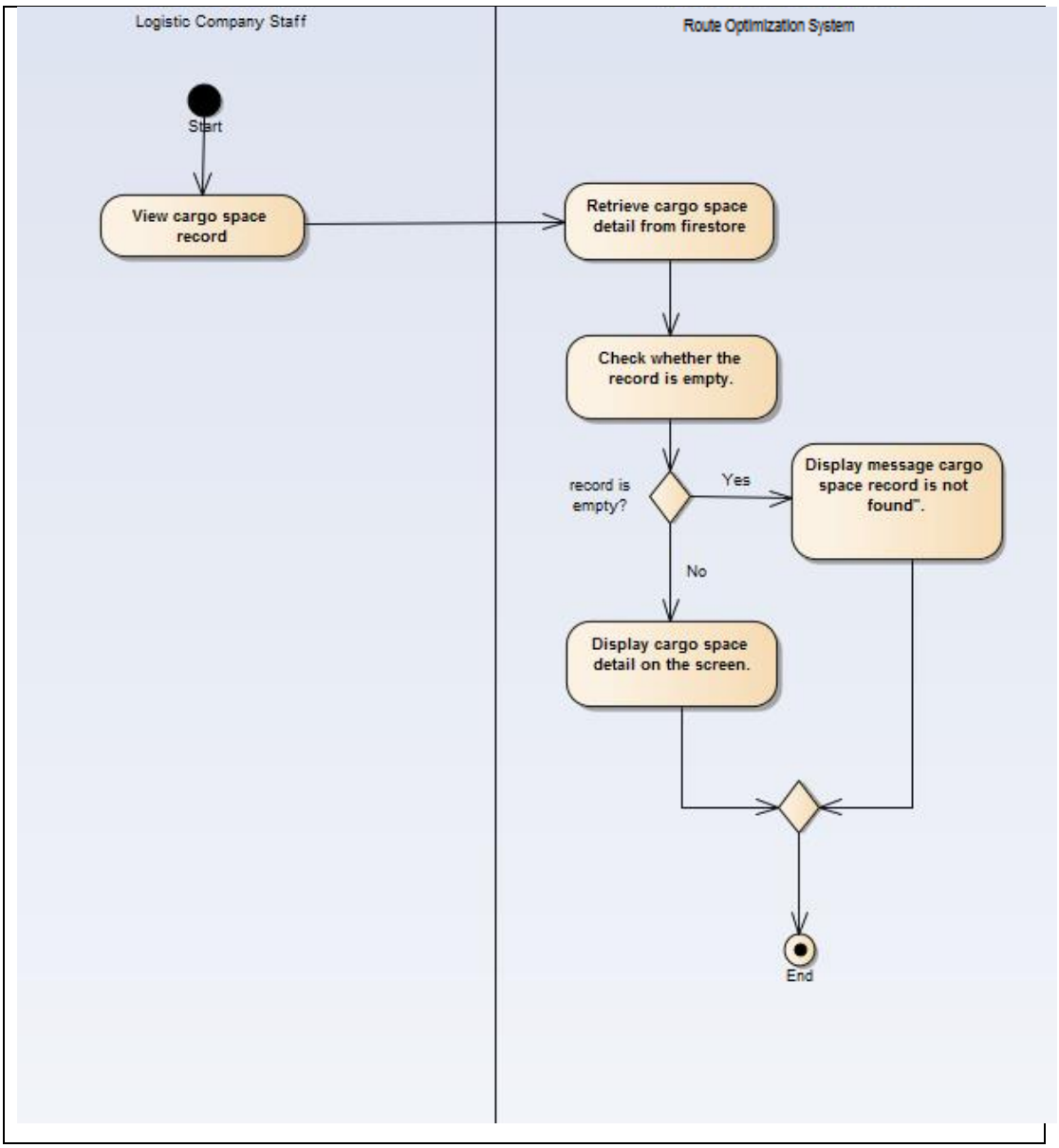


Figure 4.27 "View Cargo Space" Activity Diagram

4.9.16 "Edit a particular Cargo Space" Activity Diagram

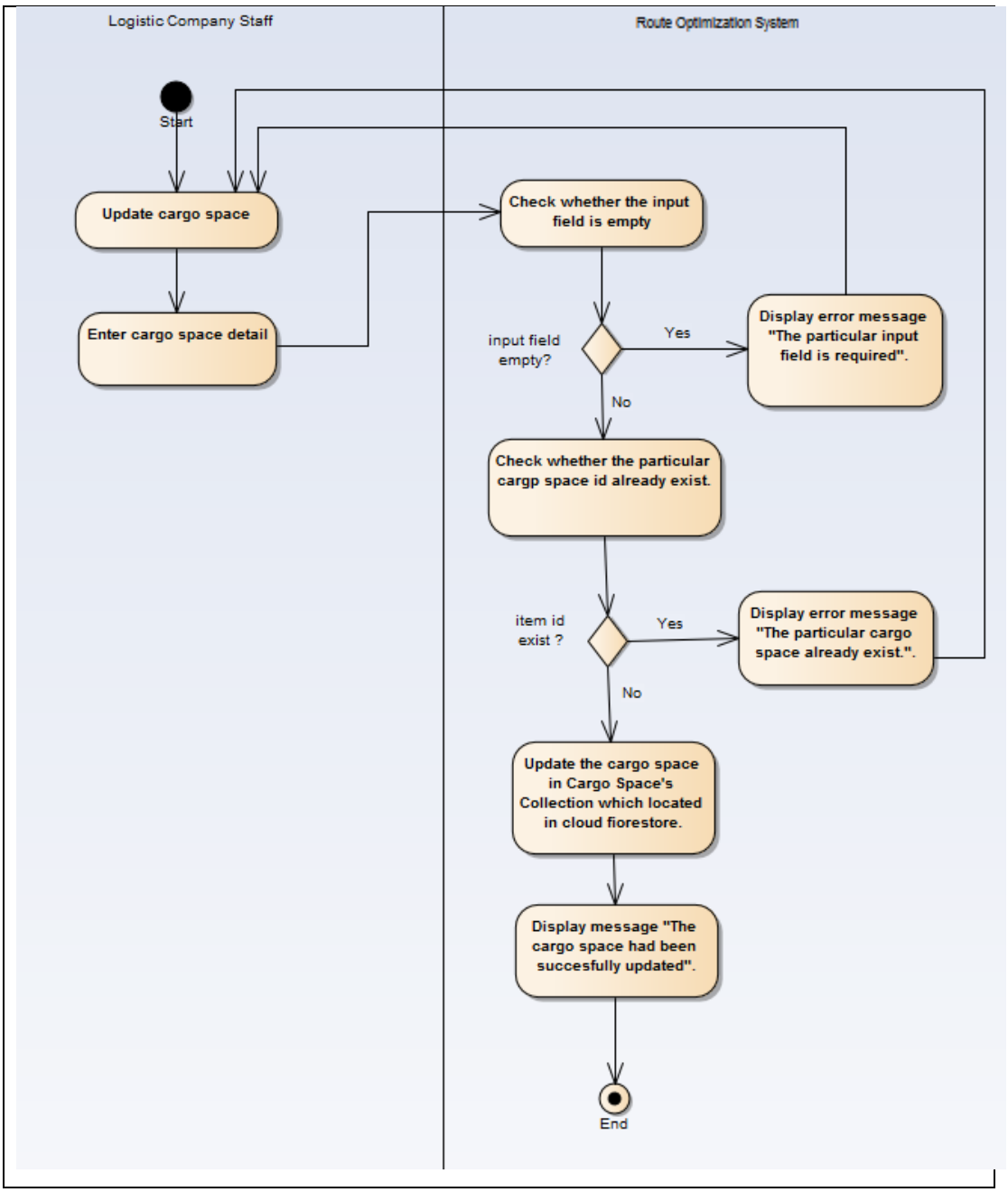


Figure 4.28 "Edit a particular Cargo Space" Activity Diagram

4.9.17 “Delete a particular Cargo Space” Activity Diagram

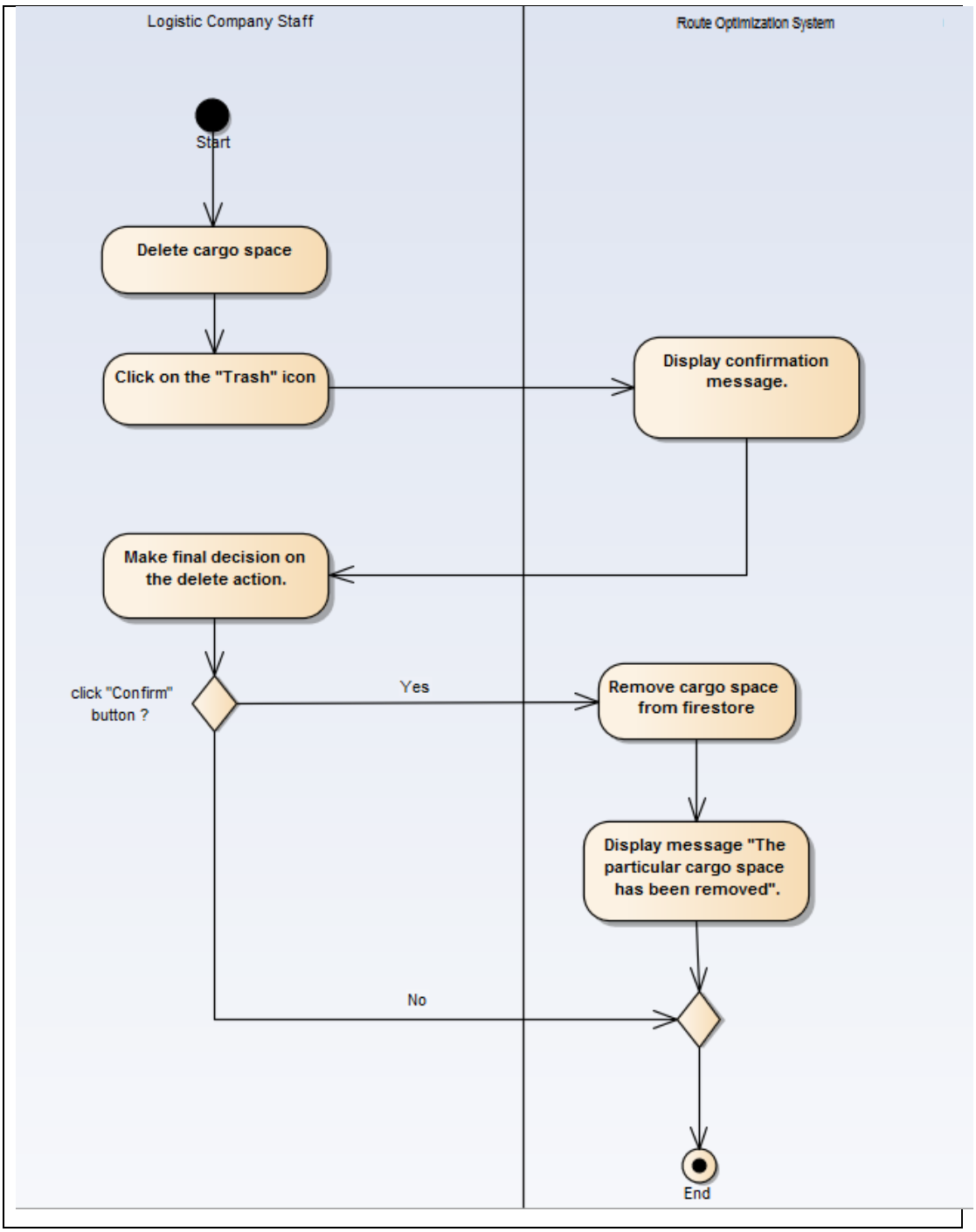


Figure 4.29 “Delete a particular Cargo Space” Activity Diagram

4.9.18 “Select truck” Activity Diagram

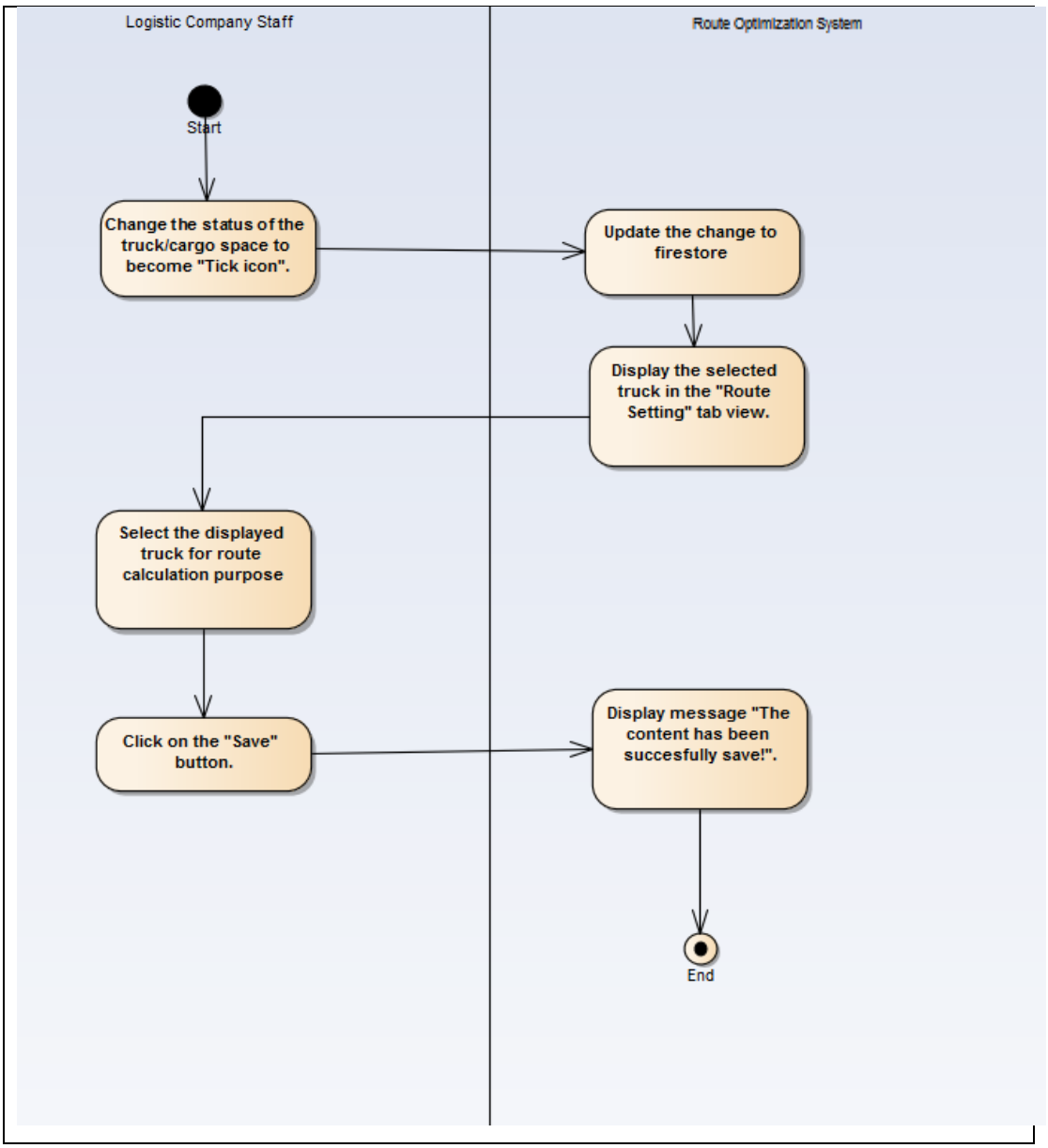
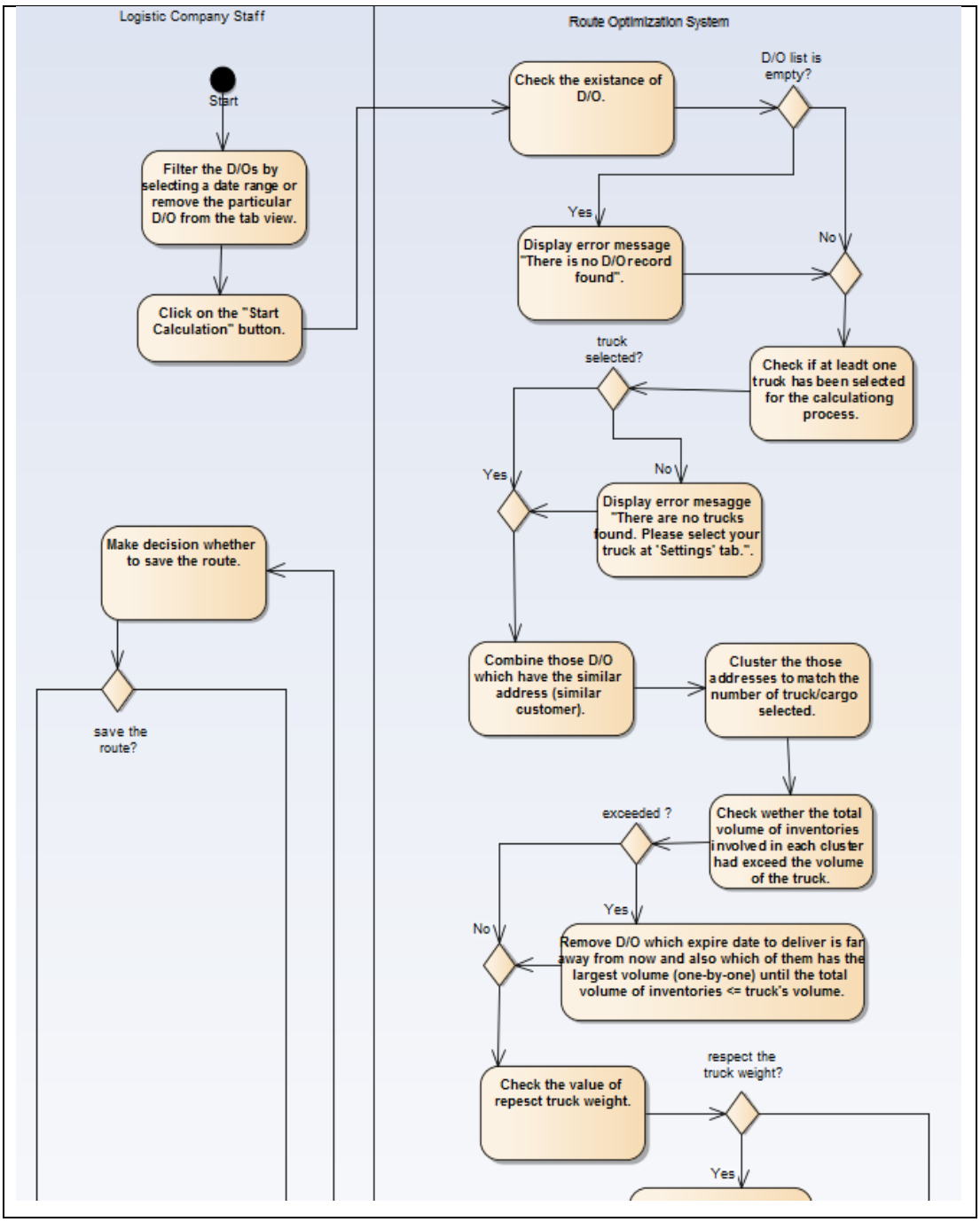


Figure 4.30 “Select Truck” Activity Diagram

4.9.19 "Calculate the optimal route" Activity Diagram



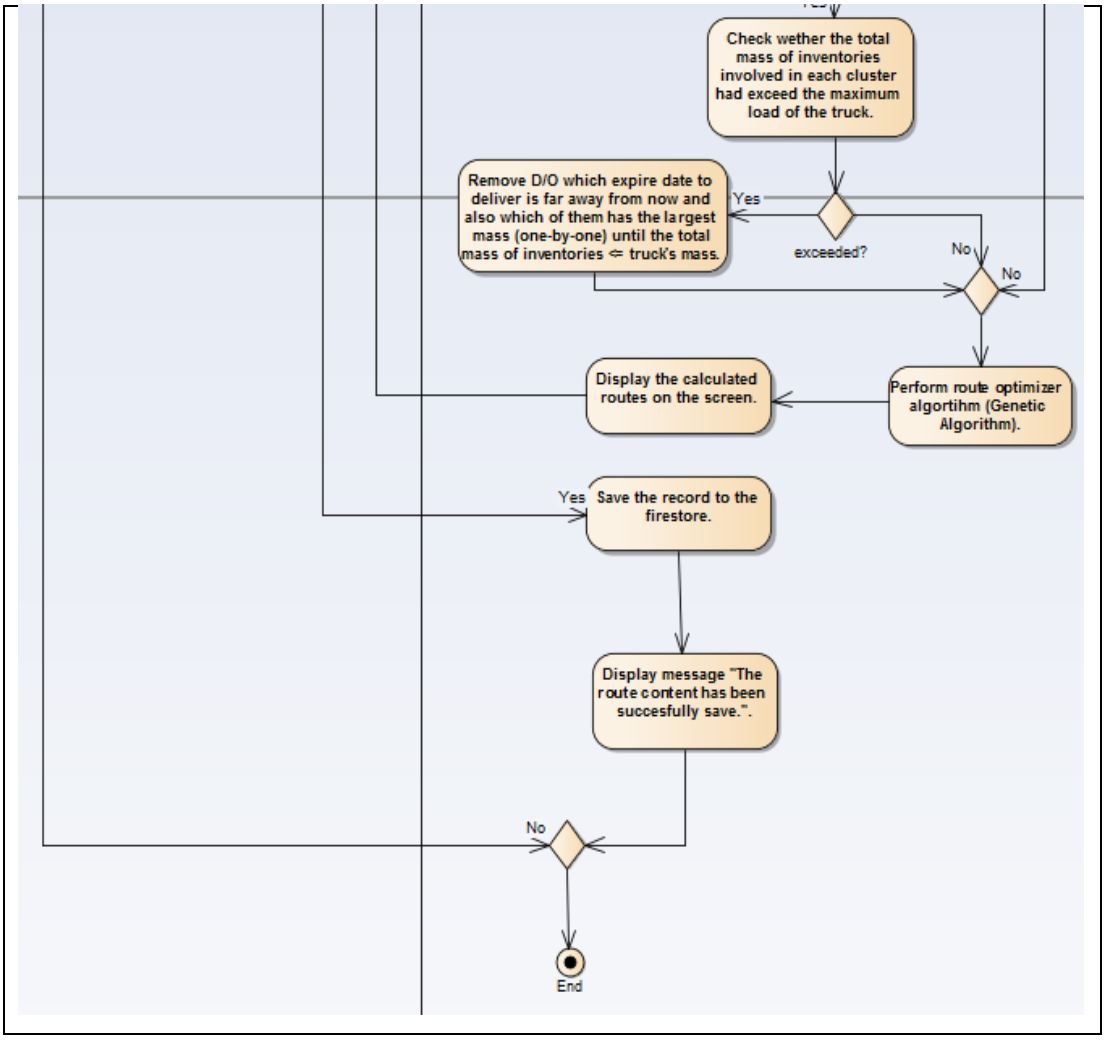


Figure 4.31 "Calculate the optimal route" Activity Diagram

4.9.20 "View planned route" Activity Diagram

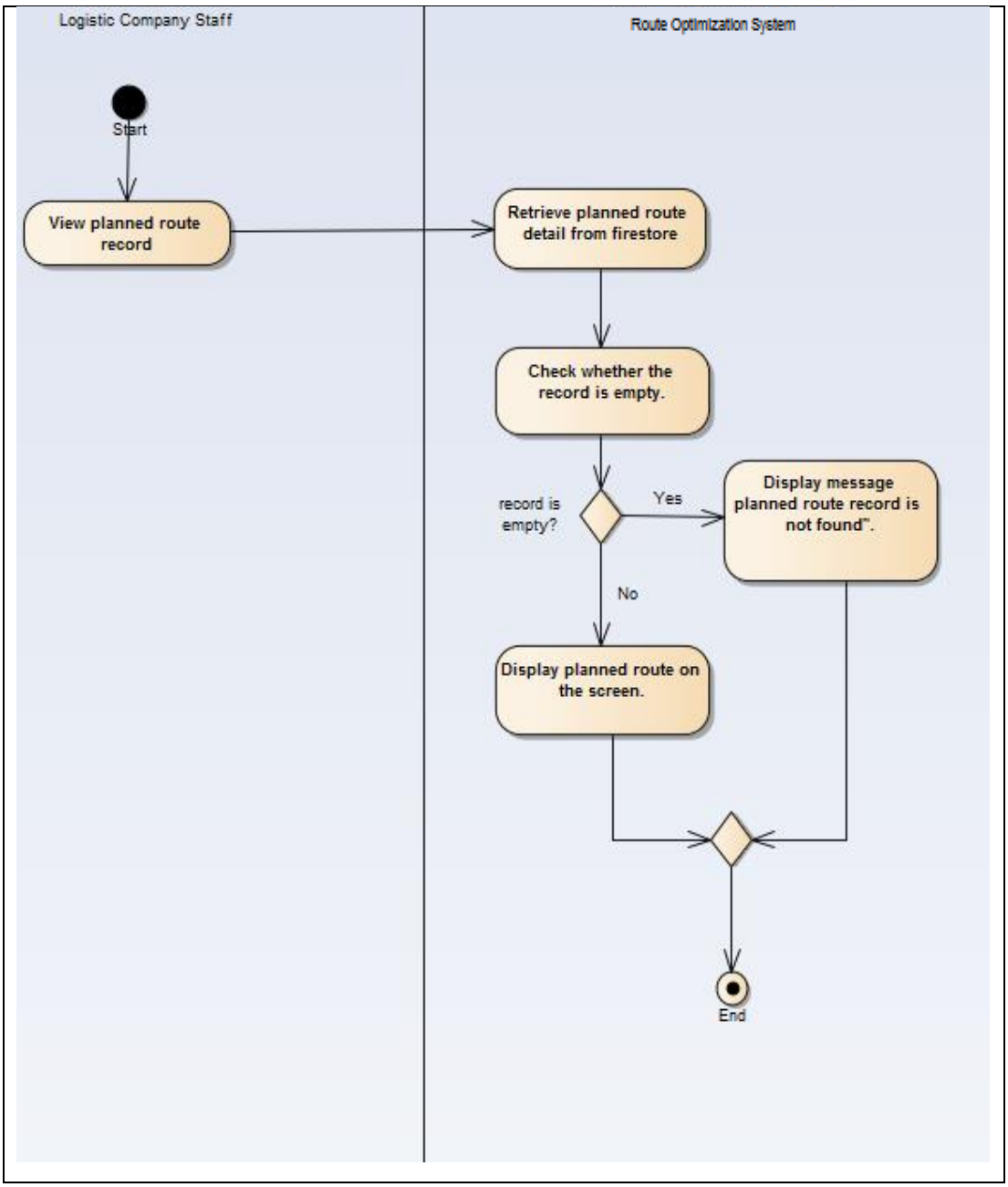


Figure 4.32 "View planned route" Activity Diagram

4.9.21 “Delete a particular planned route” Activity Diagram

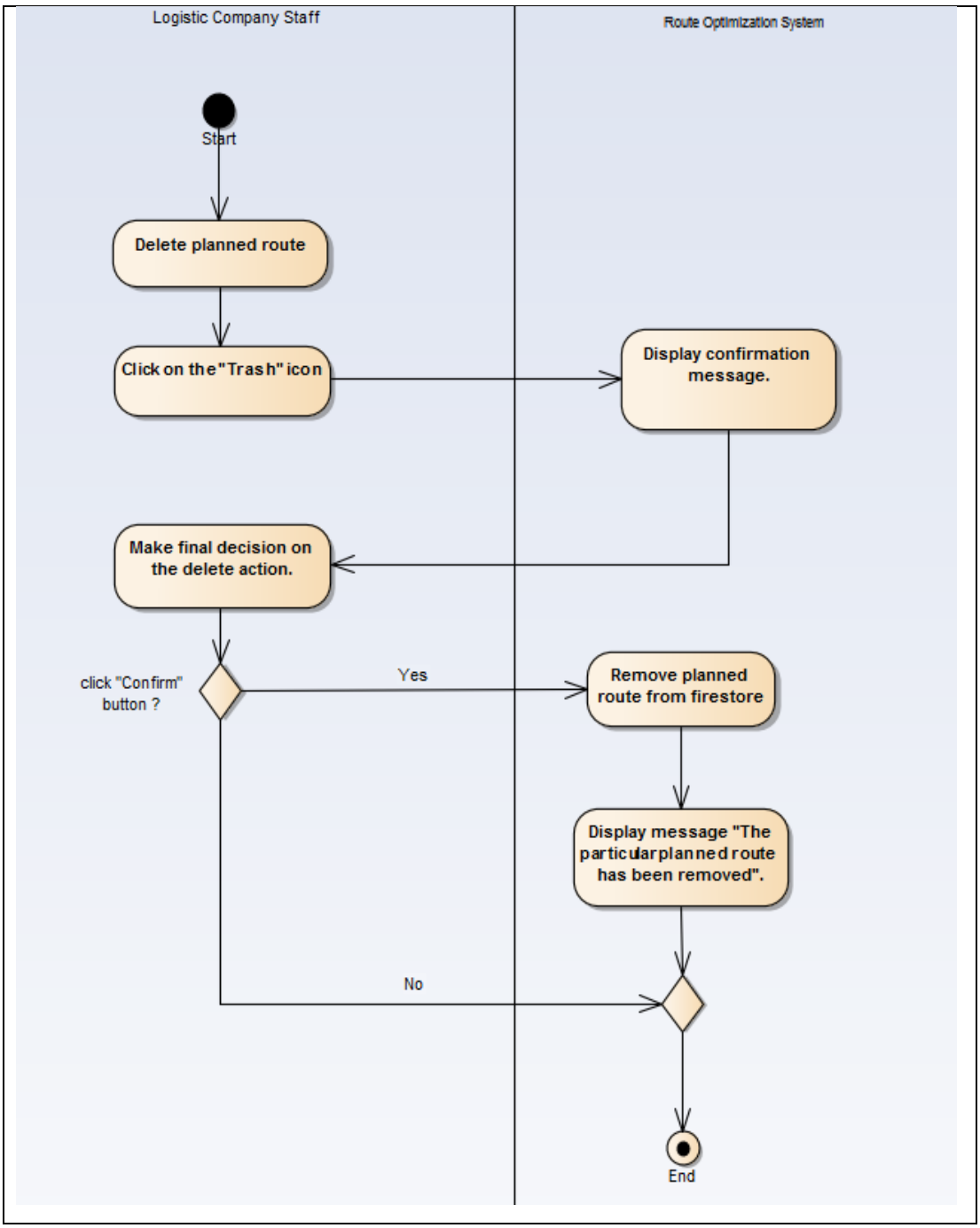


Figure 4.33 “Delete a particular planned route” Activity Diagram

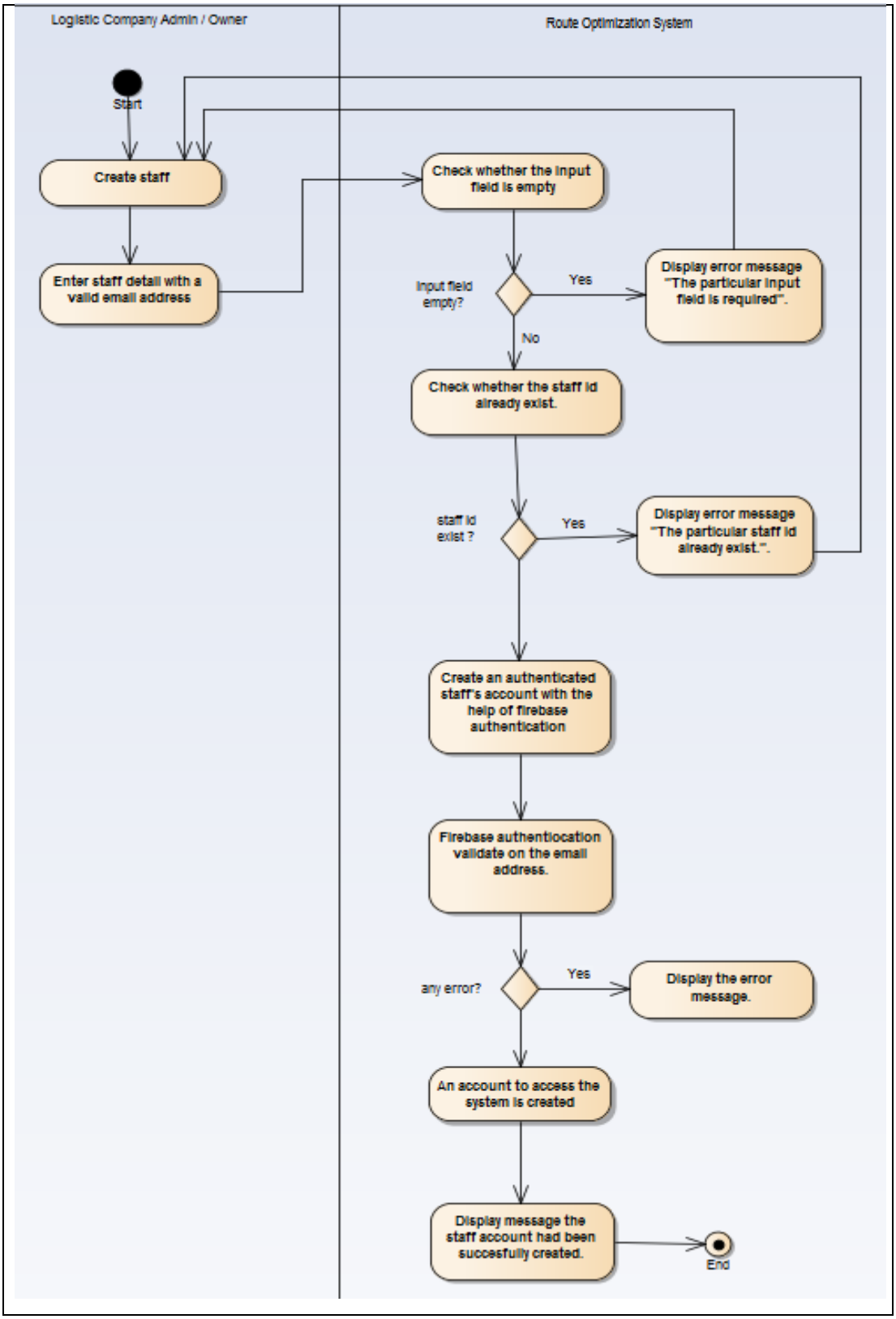


Figure 4.34 "Add and enable staff to access the system" Activity Diagram

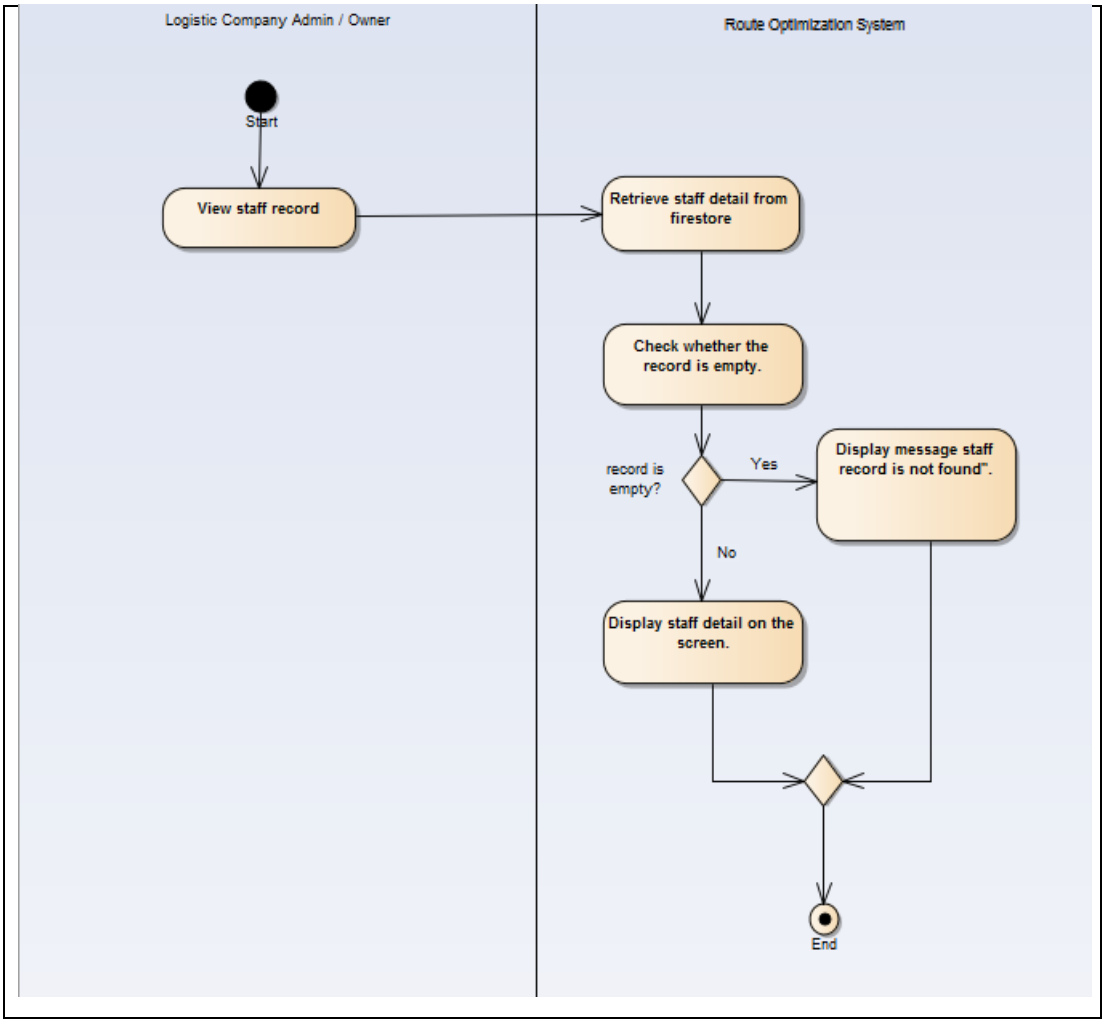


Figure 4.35 "View all staff information" Activity Diagram

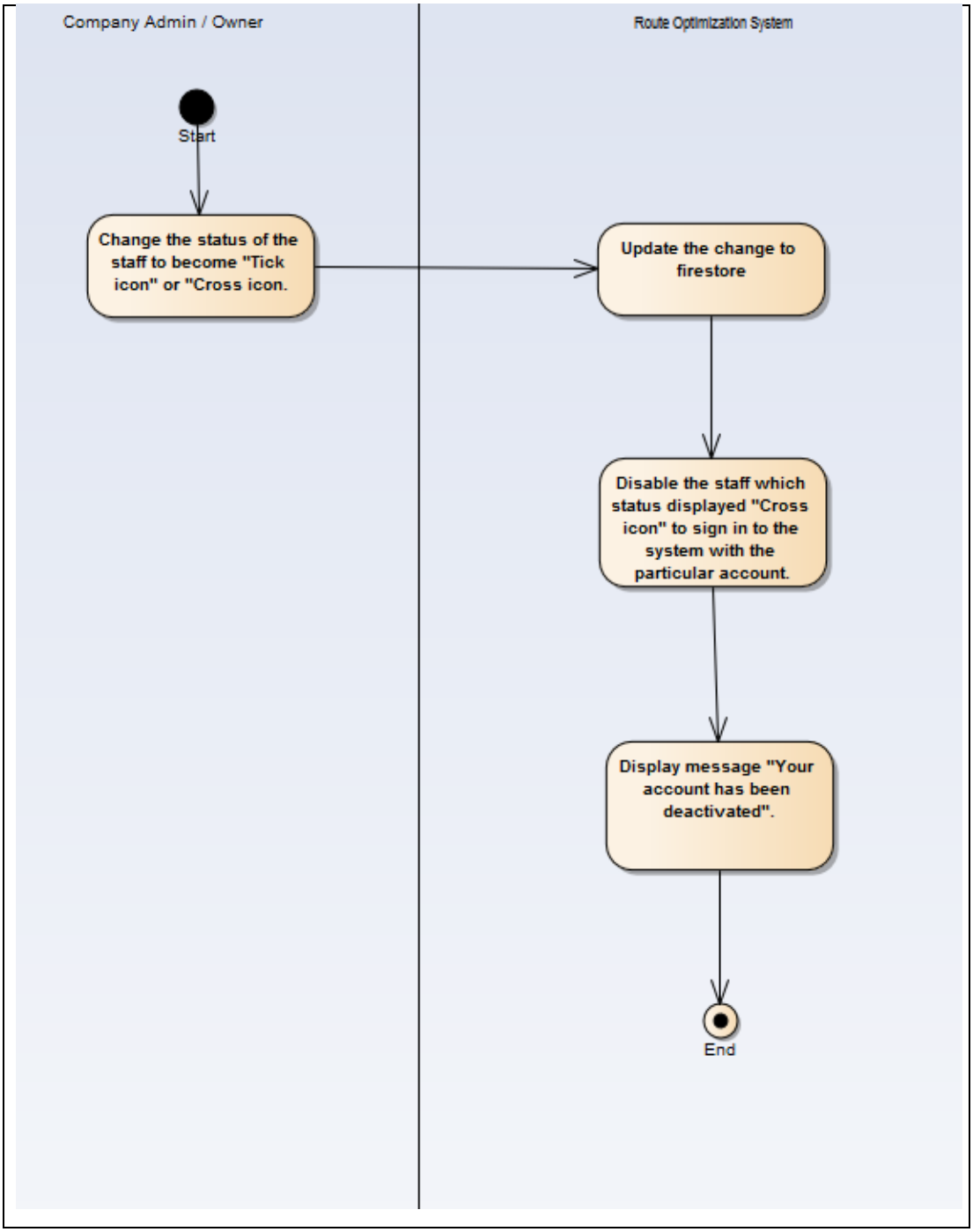


Figure 4.36 "Change staff's status" Activity Diagram

CHAPTER 5
IMPLEMENTATION

5.1 Implementation of System

After getting all requirements from the user, the system's features were broken into smaller tasks which can be seen in the work breakdown structure. Each task is then arranged and planned by adding the sequence to develop, the starting time and also the estimated end time. By having the starting time and estimated end time, the status of each task and also the status of the project can keep track. Besides, having a detailed work breakdown structure can help to confirm whether a particular task had already been performed or yet to be performed.

By looking at the sequence of tasks to be performed, the implementation process started with designing the database by figured out all the possible relationships between each entity and also with their attributes. A Logical ERD diagram was drawn to show the logic of the relationship between each entity. Then a physical ERD diagram that indicates how the real database is design was developed. The database for this project is then created by referring to this physical ERD diagram. A data flow diagram and activity diagram were drawn before started to implement the backend logic. The procedures together with the flow of the data in the system are illustrated by both data flow and activity diagram.

After all the behaviors and flow of the data had been illustrated, the project then proceeds for coding. In this phase, the user interfaces of this system were developed then followed by the backend logic which includes interacting with the firebase authentication for a user to create and access with an authenticated account, firebase cloud firestore to store data which needed for this system and also Google Map API to get the final distances after the route optimizer algorithm had arranged the location to deliver.

An account is then created to manually test this system to make sure that a happy flow can be achieved. If any bugs arise, then the particular bug will be solved immediately. The happy flow process is then continued until it achieved the final expected outcome. After a happy flow is achieved the system is then sent to testing.

5.2 Functionalities of the System

In this route optimization system, there are 2 different types of users involved, one of them are logistics company's staff who got assigned by their manager to plan for the route to deliver while another user in this system is the Company Admin/ Owner who responsible for managing company's staff information. Each user may have different roles in accessing the functionality provided by this system. The list of the functionalities that can be assessed by each user are stated below:

Table 5.1 Roles and Functionalities

User	Functionalities
Logistic Company's Staff and Logistic Company Admin / Owner	• Create, Update and View and Delete D/O information. • Create, Update and View and Delete Item information. • Create, Update and View and Delete Customer information. • Create, Update, View and Delete Cargo Space information. • Import data (D/O, Item, Customer) with Excel format (.csv). • Configure the route setting (route calculation purpose). • Display the D/O information which will involve in later route calculation. • Calculate and display the suggested route on SmartD/O screen. • Visualization of the calculated route on the map. • Create, view and delete system's planned route record.
Logistic Company Admin / Owner	• Create and enable staff to access the system. • View staff's information. • Change staff's status.

Besides functionality above, the system also provides sign-in and also the sign-up functionality which is the standard for all of the system.

5.2.1 Registration

5.2.1.1 Sign In and Sign Up

In this route optimization system, users (logistic company's staff) which considered as normal staff does not have the permission to perform the functionalities that only company's admin / owner can perform. Another user who is the company admin or company owner is able to perform those functionalities that normal staff can perform. Besides, he or she is able to help the staff to create an account for them to access the system with the company's information. Logistic company admin or owner must register their account by clicking on the "sign up" button which is located at the navigation bar. A totally new account without any information had been created. Staff account can only be created by the company admin or owner, therefore those staffs can view the existing information that had been created by another staff or the company admin / owner. After company admin has signed up an account he or she will redirect to the system's homepage instead of needed to sign in one more time.

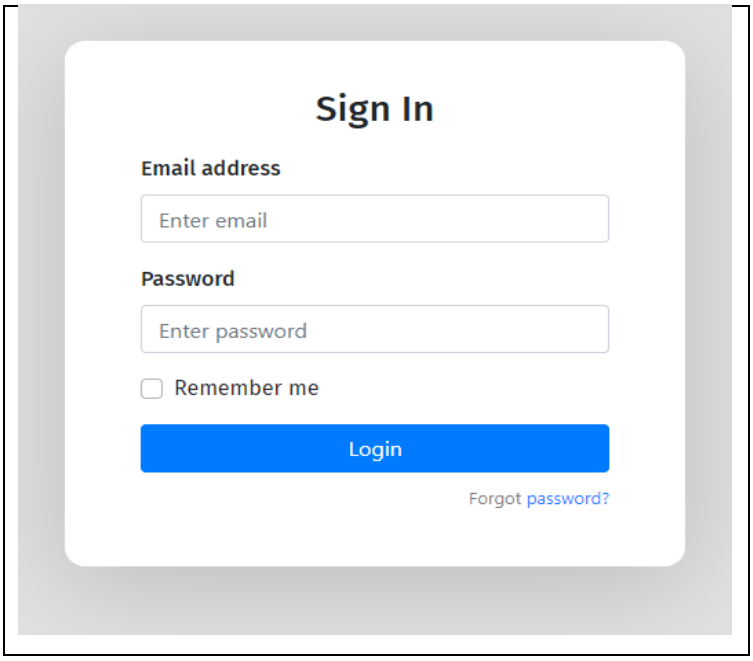


Figure 5.1 Sign in Form at Sign In Page

```
login = (e) => {

    this.props.openSpinnerStatus();

    e.preventDefault();
    fire.auth().signInWithEmailAndPassword(this.state.userEmail, this.state.userPass)
        .then(async (u)=>{
            // this.props.onLoginChange(true);
            if(u.user){
                let userId = u.user.uid;

                await this.callUser(userId);

                if(this.admin !== null){
                    if(this.admin){
                        this.props.history.push('/adminBoard')
                    }
                    else{
                        this.props.history.push('/home')
                    }
                }

            }
        })
        .catch((error) => {
        console.log(error);
    });
}
```

Figure 5.2 Section of code to validate user's identity with the help of firebase authentication during login process.

If the user was successfully validated then the particular user will go further to validate on him/her role. If the user is company admin then an extra "Staff" button will exist on the navigation bar to let the company admin control the staff account.

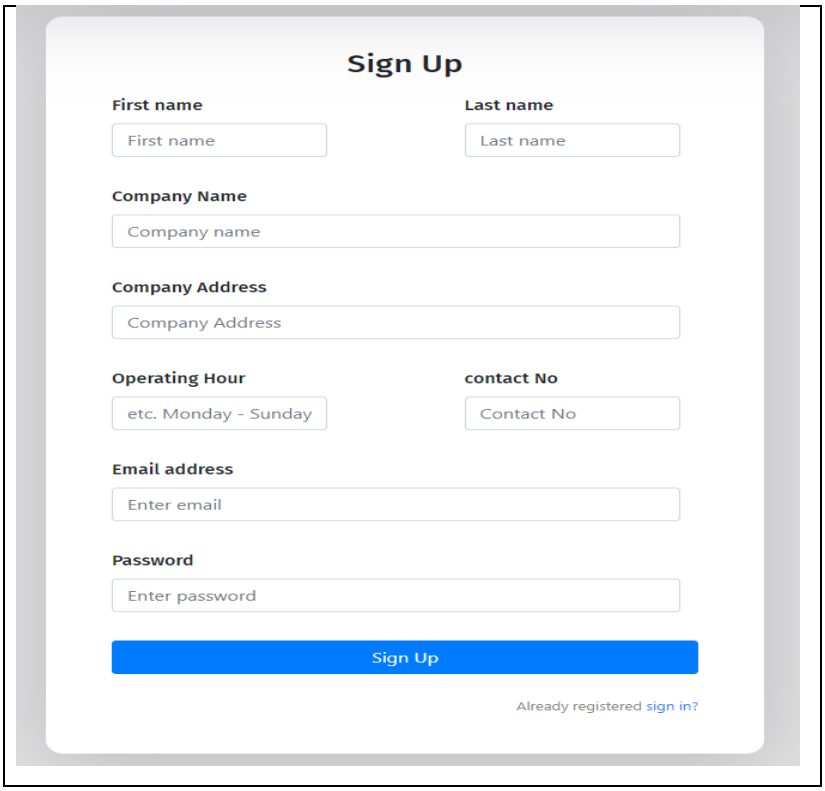


Figure 5.3 Sign Up Form at Sign Up Page

User who signs up their account on this page will be treated as company admin / owner. Therefore this particular user has the permission to add normal staff to access the system and also the company information (D/O, item, customer, cargo).

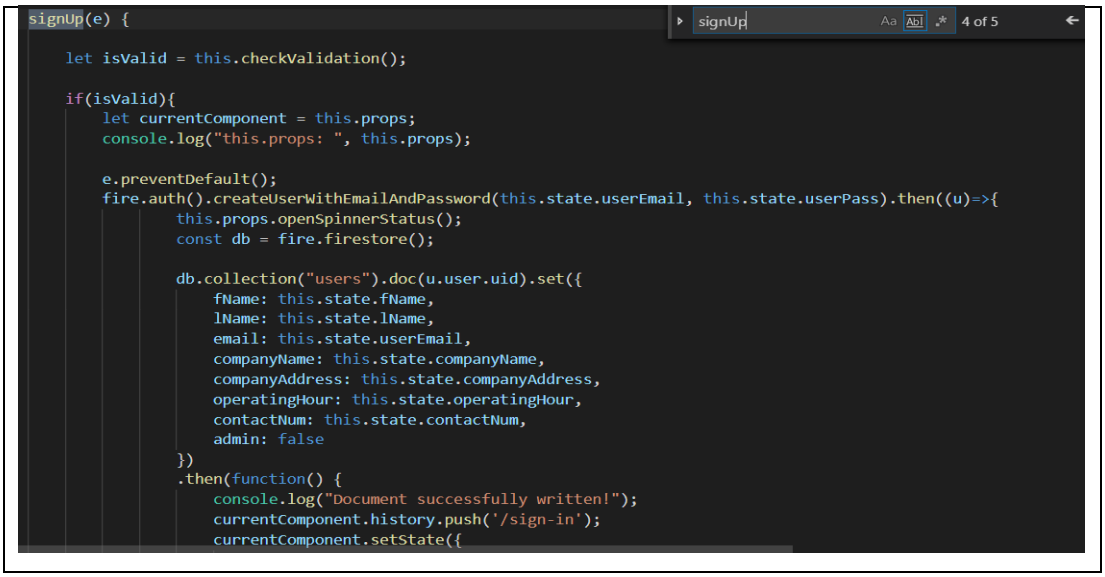


Figure 5.4 Section of code to create user's account with the help of firebase authentication.

With the help of firebase authentication, the route optimization system is able to validate whether a particular email has been registered. If the email already existed in the system's database then an error message will be displayed to let the user know that this particular email has been registered. Therefore user can select a new email for registration. If user successfully registered without any error, then the system will automatically direct this user to the home page.

5.2.2 Functionalities (Logistic company's staff and Company Admin)

5.2.2.1 Manage D/O Informaton

5.2.2.1.1 Create D/O information

In order for the user to calculate on the route, user must first have some D/O information in the system. User is allowed to create the D/O information at the "Add D/O" tab which is located at the "Home" page. The design of the UI can be seen in Figure 5.5.

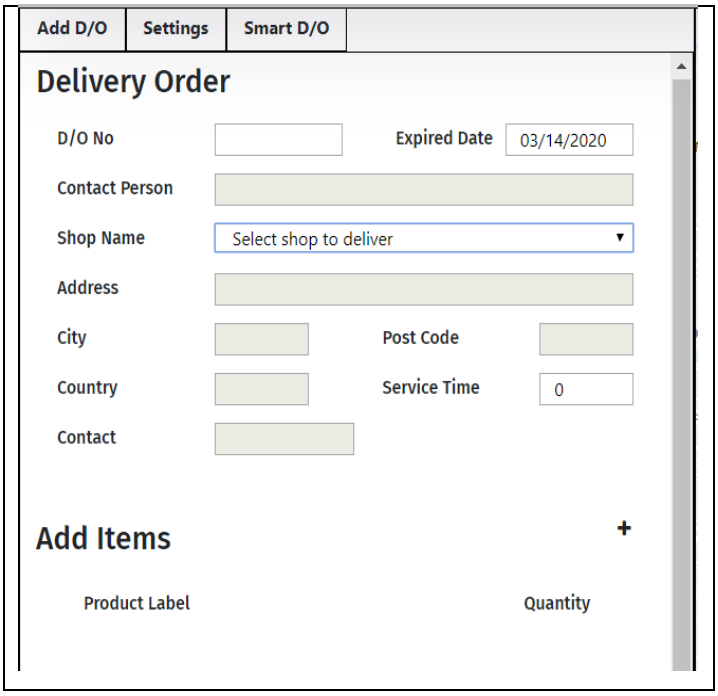


Figure 5.5 Add D/O form

In order for the convenience of the user, the form is designed to automatically display the information of the customer by selecting the shop name of that customer. After selecting the customer, all the relevant data that needed in this form will be populated automatically. User only needed to enter several fields which are D/O No which is unique, service time and also the expired date for that particular D/O. In order for the system to display all the customer's shop name in the selection picker, the customer data must first be imported into the system's database.

```
db.collection('users').doc(this.userId).collection('customers').get()
.then(function(querySnapshot){
    console.log(querySnapshot)
    return Promise.all(
        querySnapshot.docs.map(function(doc) {

            customers.push(doc.data());

        })
    ).then(_ => {
        return customers
    })
})
.then(function(datas){
    console.log('Getting Customers data: ', datas);
    localStorage.setItem('customers', JSON.stringify(datas));
})
```

Figure 5.6 Section code of retrieving customers' data from the firestore

```
onDropdownSelected = e => {
    let selectedData = this.state.cusDatas.filter(function(cus){
        return cus.customerId == e.target.value;
    })

    this.setState({
        selectedCus:{
            address: selectedData[0].address,
            contactNum: selectedData[0].contactNum,
            customerId: selectedData[0].customerId,
            customerName: selectedData[0].customerName,
            shopName: selectedData[0].shopName,
            country: selectedData[0].country,
            city: selectedData[0].city,
            postCode: selectedData[0].postCode,
            latitude: selectedData[0].latitude,
            longitude: selectedData[0].longitude
        }
    })
}
```

Figure 5.7 Section code of populating the customer information when user select on the particular shop name from the selection picker.

Before saving the created D/O into the system, user may also need to add the items that corresponding to the particular D/O No. Figure 5.8 shows the design of the Add item form.

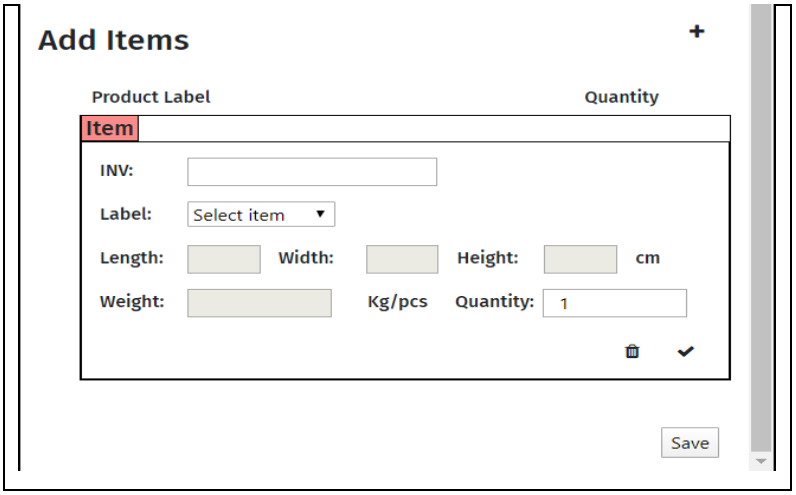


Figure 5.8 Add item form

Similar to the form in Figure 5.5, user only needed to fill in the INV no which corresponding to the particular item and the quantity of the item needed to be delivered in this particular D/O. Other information like length, width and height of the item will automatically be populated on the screen after the user selects a particular label from

the selection picker. Similar to the customer information, the data of those items must also be imported into the system before adding an item into the particular D/O.

```
if(this.userId !== null){
    db.collection('users').doc(this.userId).collection('items').get()
    .then(function(querySnapshot){
        console.log(querySnapshot)
        return Promise.all(
            querySnapshot.docs.map(function(doc) {

                items.push(doc.data());

            })
        ).then(_ => {
            return items
        })
    })
    .then(function(datas){
        console.log('Getting items data: ', datas);
        localStorage.setItem('items', JSON.stringify(datas));
        currentComponent.setState({
            itemDatas: datas,
            spinnerStatus: false
        })

    })
    .catch(function(error){
        console.log("Error getting items collection: ", error);
```

Figure 5.9 Section code of retrieving items' data from the firestore

```
onDropdownSelected = e => {

    let selectedData = this.state.itemDatas.filter(function(item){
        return item.itemId == e.target.value;
    })

    this.setState({
        selectedItem:{
            itemId: selectedData[0].itemId,
            height: selectedData[0].height,
            itemLabel: selectedData[0].itemLabel,
            length: selectedData[0].length,
            weight: selectedData[0].weight,
            width: selectedData[0].width,
            invNo: '',
            quantity: '',
            total: selectedData[0].total
        }
    })
}
```

Figure 5.10 Section code of populating the item information when user select on a particular item's label from the selection picker.

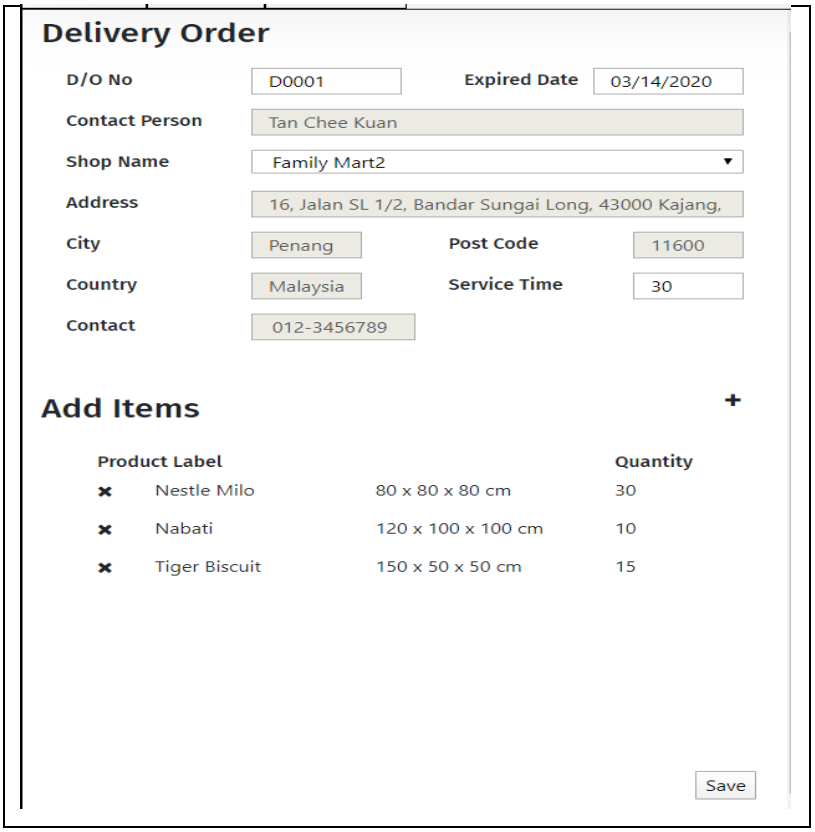


Figure 5.11 Final view of the form when all require information is filled.

After the user had key in all the required information, user can click on the "Save" button. All relevant data will then be stored at the firestore. If some of the input fields is not filled and user proceeds to click on the "Save" button, an error message will prompt out to ask user to fill in the particular input field. If the D/O No that enter by the user is already exists in the firestore, then thw system will ask user to change the D/O No in order for this particular D/O to successfully save into the firestore.

```
addItems = (item) => {
    this.itemList.push(item);

    this.setState({
        itemsToDisplay: this.itemList
    }, () => {
        this.changeAddItemStatus("2");
    })
}
```

Figure 5.12 Section code of add item into a list for displaying purpose.

```
createSelectItems() {
    let items = [];
    for (let i = 0; i < this.state.itemDatas.length; i++) {
        items.push(<option key={this.state.itemValue[i]} value={this.state.itemValue[i]}>{this.state.itemLabel[i]}</option>);
    }
    return items;
}
```

Figure 5.13 Section code of displaying multiple selected items in the "Add Items" section.

```
db.collection("users").doc(userId).collection("DOs").doc(this.state.doNo).set({
    doNo: this.state.doNo,
    exDate: Number(this.state.expireTimeStamp),
    items: this.state.itemsToDisplay,
    customer: this.state.selectedCus,
    serviceTime: Number(this.state.serviceTime),
    status: "Pending"
})
.then(function() {
console.log("Document successfully written!");
})
.catch(function(error) {
console.error("Error writing document: ", error);
});
```

Figure 5.14 Section code of create D/O's data on the firestore.

5.2.2.1.2 View D/O information

In this system, user is able to view the list of D/O records by clicking on the "D/O Information which located at the navigation bar. Figure 5.15 shows the design of how the D/Os' record are displayed.

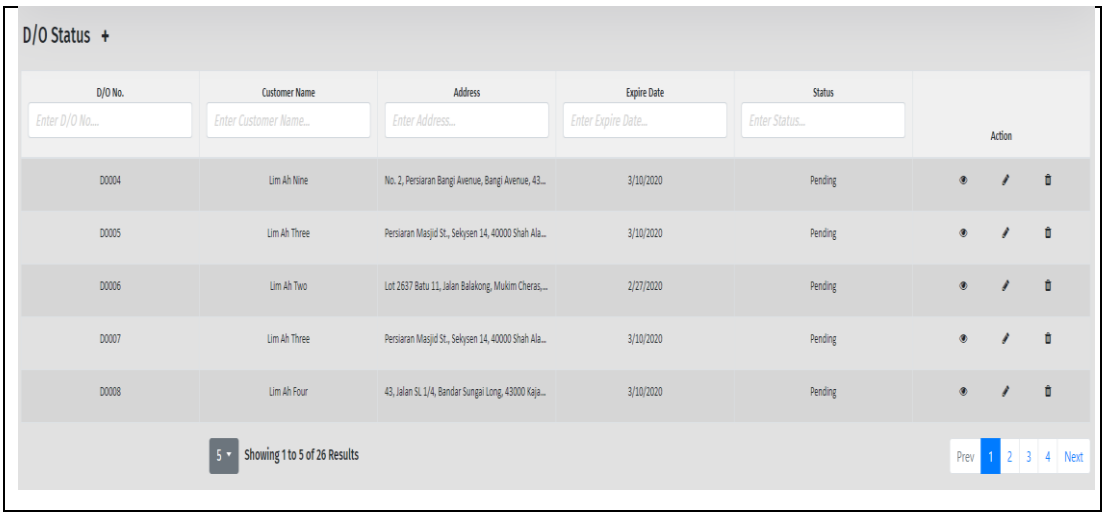


Figure 5.15 D/O Information screen

If there are too many records display on the screen, then user may be needed to spend some time on finding a particular record. For example, user may want to find a D/O record with the customer name of "Albert tan" and with the status of "Delivered", it is difficult for user to find the particular record from the start to the end. This is not efficient due to finding a record from hundreds of them may waste a lot of time and also sometime user may overlook the particular record. To overcome this problem, this system provides filters on each column, therefore user can narrow down the list of records by entering the specific keyword which relevant to the particular record. This system also allows user to perform multiple filtering processes on the list of D/O records.

5.2.2.1.3 View the detail of a particular D/O

Table is not enough to display all the detail of a particular D/O, therefore this system enables user to view all the detail of a particular D/O by clicking on the "Eye: icon which located at the action column of the table in Figure 5.15. The detail of the particular D/O will be displayed to the user with PDF format which can be seen in Figure 5.16. If the user wishes to have a hard copy of the particular D/O user can download or print it from the system.

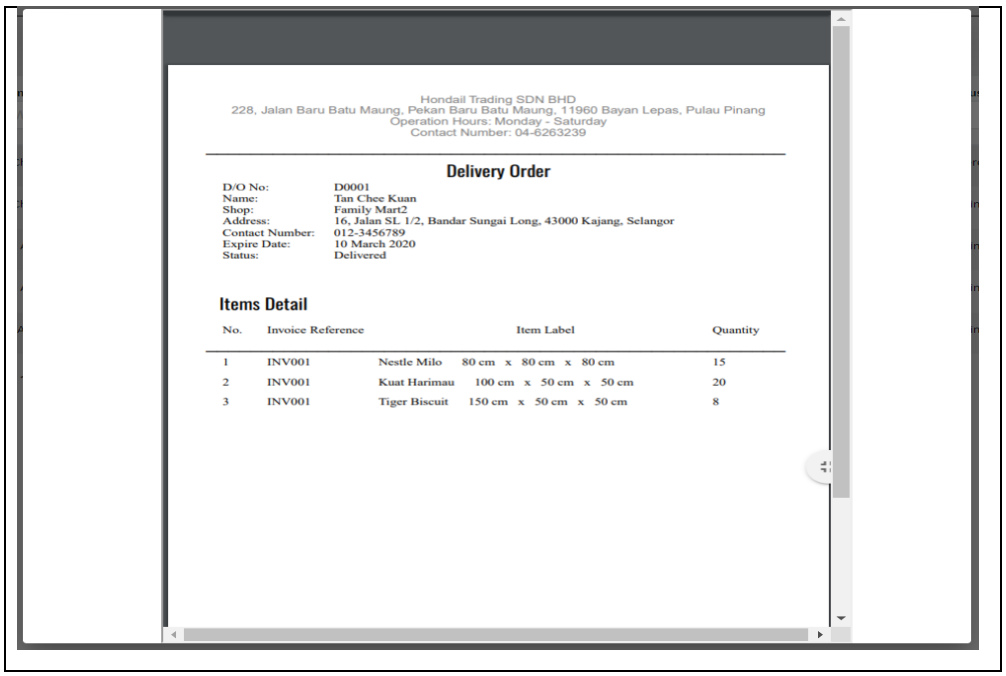


Figure 5.16 Detail of a particular D/O in the PDF format.

```
openDOView = (cell, row, index) => {

    let selectedData = this.state.doDatas.filter(function(data){
        return data.doNo == row.doNo;
    })

    this.setState({
        selectedDO: {...selectedData[0]},
    }, () => {
        this.setState({
            openDO: true
        })

    })
}
```

Figure 5.17 Section code which set the state of the indicator that used to display the detail of a particular D/O in PDF format when user click on the "Eye" icon.

```
{this.state.openDO ?
    <ViewDO
        handleClose={() => this.handleClose1("1")}
        open={this.state.openDO}
        deliveryOrder={this.state.selectedDO}
    /> : null}
```

Figure 5.18 Section code to render the PDF View.

5.2.2.1.4 Edit on a particular D/O

In this system, user is allowed to edit the detail of the particular D/O by clicking on the "Edit" icon which located at the action column of the table. The detail of the particular icon will be populated on the dialog box, user may edit the particular D/O by changing the customer detail or adding and removing items from the particular D/O. The design of the UI can be view in Figure 5.17.

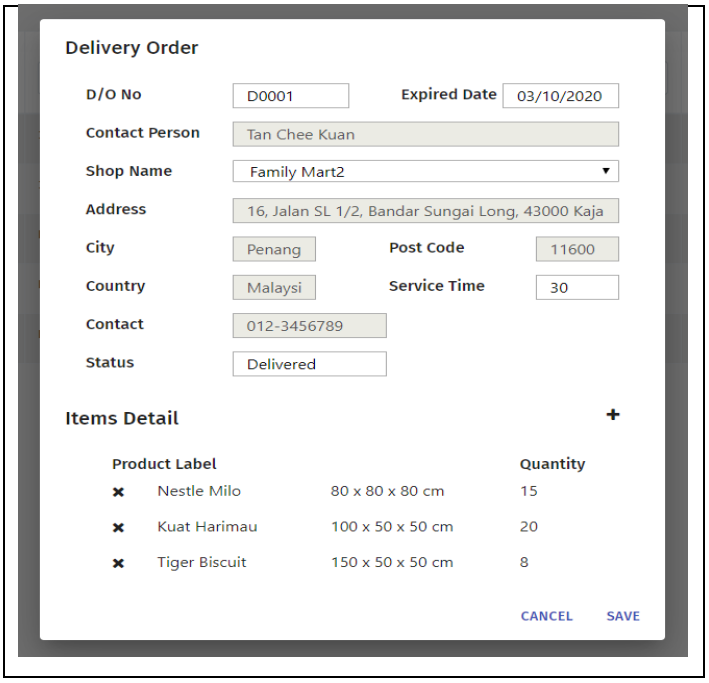


Figure 5.19 The Dialog Box which contain the D/O detail to be modified.

After edit on the particular D/O, user can proceed to update the information in the firestore by clicking on the "Save" button which located below the dialog box. Validation process will be takeover to make sure that all the required fields had been filled with value before update to the firestore. An error message will be displayed if there is an empty input field and the system will not allow the user to proceed. If the information is successfully updated, the system will display a popup dialog to let user know that his/her record had been updated successfully.

```
openDODialog = (cell, row, index) =>{

    let selectedData = this.state.doDatas.filter(function(data){
        return data.doNo == row.doNo;
    })

    this.setState({
        selectedDO: {...selectedData[0]},
        selectedCus: {...selectedData[0].customer}
    }, () => {
        this.setState({
            open: true
        })

    })
}
```

Figure 5.20 Section code which set the state of the indicator that used to display the edit form when user click on the "Edit" icon.

5.2.2.1.5 Delete a particular D/O

This system enables user to delete a particular D/O record, a dialog message will be pop up to prompt user whether he/she wants to continue the following action. If the user clicks on the "Confirm" button, then the particular D/O record will be deleted permanently from the system.

```
removeDO = (cell, row, index) => {
    this.openSpinner();
    let currentComponent = this;
    let tempDoDatas = [...this.state.doDatas];
    let tempFinalDatas = [...this.state.finalDatas];
    let userId = localStorage.getItem('user');
    const db = fire.firestore();

    tempDoDatas = tempDoDatas.filter((deliveryOrder, index) => {
        return deliveryOrder.doNo !== row.doNo
    })

    tempFinalDatas = tempFinalDatas.filter((finalOrder, index) => {
        return finalOrder.doNo !== row.doNo
    })


    db.collection("users").doc(userId).collection("DOs").doc(row.doNo).delete()
    .then(function() {

        console.log("Document successfully deleted!");

        currentComponent.closeSpinner("2");

        currentComponent.setState({
            deleteIndicator: false,
            successDelete: true,
            doDatas: tempDoDatas,
            finalDatas: tempFinalDatas
        }, () => {
            localStorage.setItem("dos", JSON.stringify(currentComponent.state.doDatas));
```

Figure 5.21 Section code of removing a particular D/O record from the system.

5.2.2.2 Manage Item Information

5.2.2.2.1 Create Item Information

As mentioned in 5.1.2.1.1, in order to add items into a new D/O, there must have items' data in the firestore. Therefore, user is able to add a particular item into the new D/O by selecting an item from the selection picker. This system allows user to create new item information and store it into the firestore. User may need to go for the particular page by clicking the "Item Information" button which located at the navigation bar. After enter to the page user may need to click on the "Plus" icon which located beside the header "Item Detail". A popup dialog will appear to prompt user for the detail of item's information. The design of the popup dialog can be view in figure 5.22.

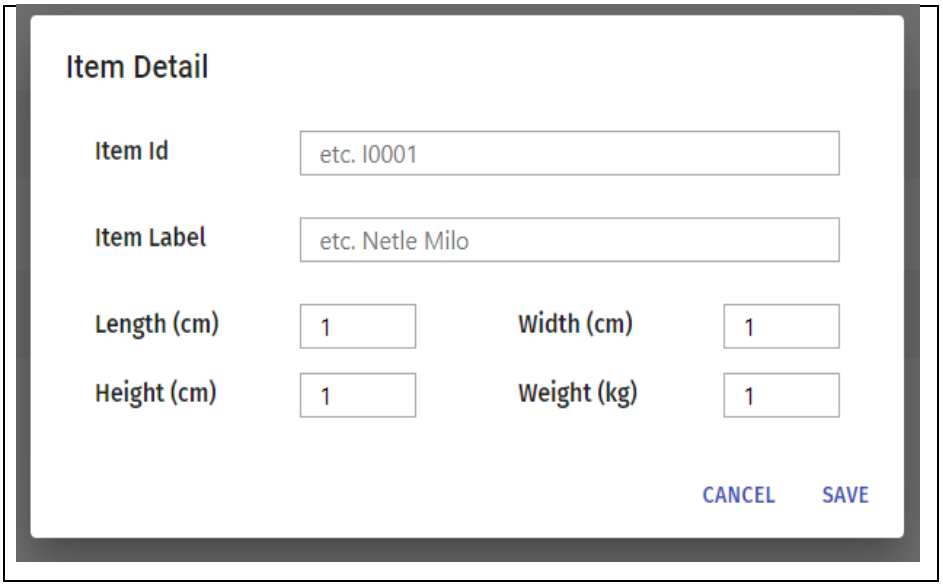


Figure 5.22 Dialog Form for item's detail

Similar to other forms mentioned above, the validation process will take place when the user clicks on the save button. User is disabled to proceed and an error message will be displayed when there is an empty field. Another popup dialog will be shown when the record has been successfully created.

5.2.2.2.2 View Item Information

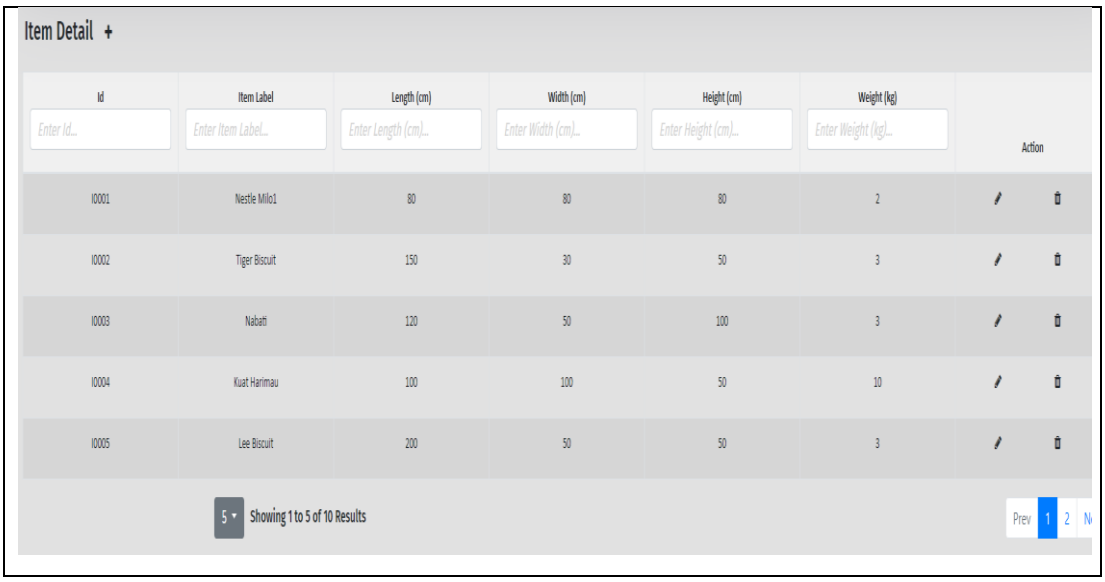


Figure 5.23 Item Information Screen

The process is similar to the “D/O Information Screen” which located in section 5.1.2.1.2.

5.2.2.2.3 Edit on Item Information

The process for edit item’s information is similar to “Edit on D/O Information” which located in section 5.1.2.1.4 and the design of the UI is similar to Figure 5.22.

5.2.2.2.4 Delete a particular Item

The process to delete a particular item’s record is similar to “Delete a particular D/O record” which located in section 5.1.2.1.5.

5.2.2.3 Manage Customer Information

5.2.2.3.1 Create Customer Information

Similar to section 5.1.2.2.1. Customer information needed to be created when user wishes to create a particular D/O using this system. User may be needed to click on the “Customer Information” button which located at the navigation bar to direct to the particular page. Similar to section 5.1.2.2.1 user may be needed to click on the “Plus” icon which located beside the “Customer Detail” header. The design of the UI can be view in Figure 5.24.

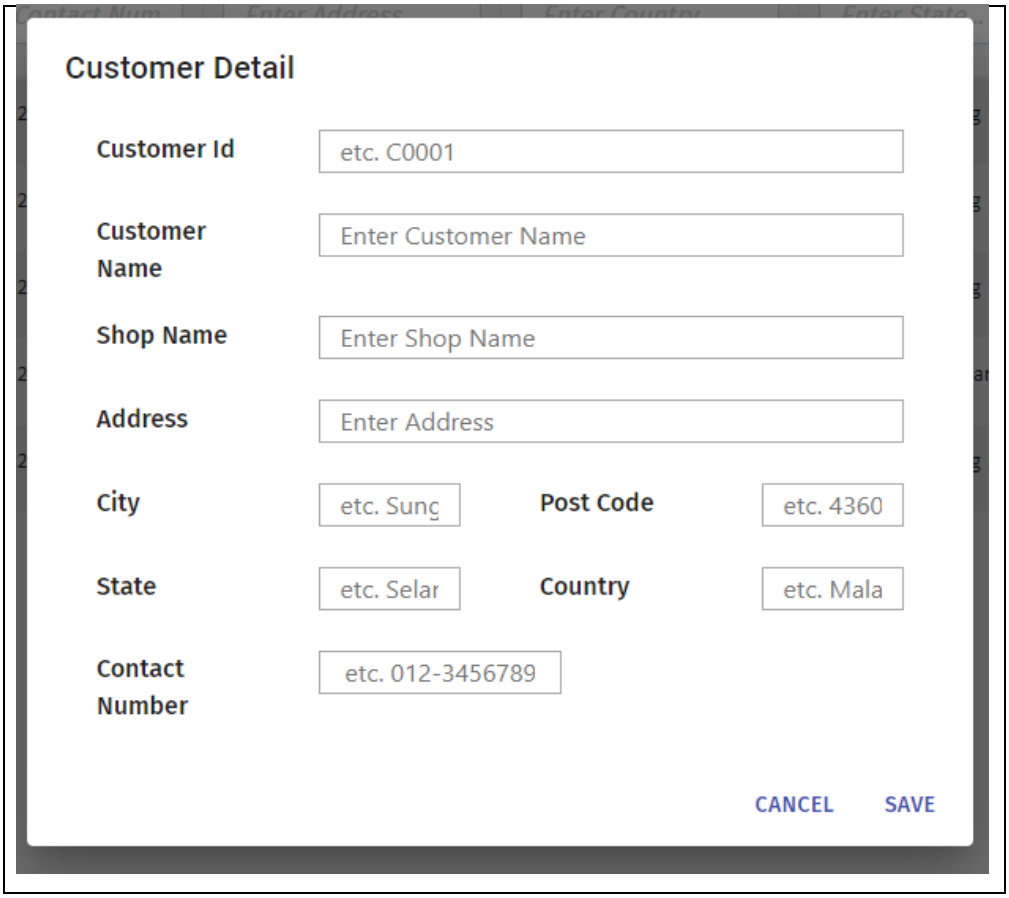


Figure 5.24 Dialog Form for customer's detail

```
let lat = 0;
let lng = 0;
const geocoder = new google.maps.Geocoder();

geocoder.geocode({
    'address': selectedCus.address,
}, function(response, status){
    lat = response[0].geometry.location.lat();
    lng = response[0].geometry.location.lng();
    console.log("Latitude: ", lat);
    console.log("Longitude: ", lng);

    db.collection("users").doc(userId).collection("customers").doc(selectedCus.customerId).set({
        address: selectedCus.address,
        city: selectedCus.city,
        contactNum: selectedCus.contactNum,
        country: selectedCus.country,
```

Figure 5.25 Section code of getting the coordinate of the customer's address with the help of Google Map API and store them together with the customer's data.

5.2.2.3.2 View Customer Information

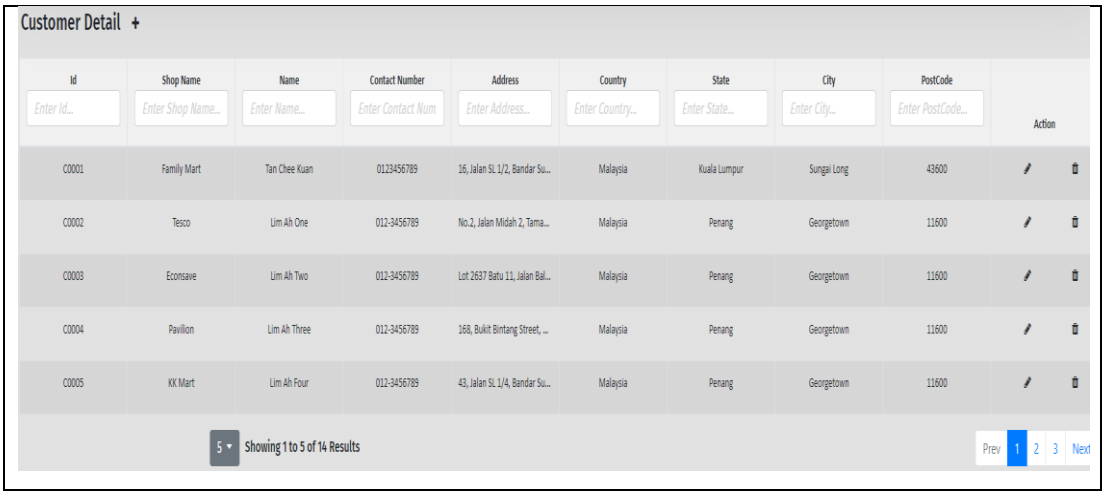


Figure 5.26 Customer Information Screen

The process is similar to the "D/O Information Screen" which located in section 5.1.2.1.2.

5.2.2.3.3 Edit on Customer Information

The process for edit item's information is similar to "Edit on D/O Information" which located in section 5.1.2.1.4 and the design of the UI is similar to Figure 5.24.

5.2.2.3.4 Delete on a particular Customer

The process to delete a particular customer's record is similar to "Delete a particular D/O record" which located in section 5.1.2.1.5.

5.2.2.4 Manage Cargo Space Information

5.2.2.4.1 Create Cargo Space Information

In order to let the system perform the route calculation process, user must need to have at least one cargo space in the system. User may need to click on the "Cargoes Space" button which located at the navigation bar to direct to the particular page. Similar to section 5.1.2.2.1 user may be needed to click on the "Plus" icon which located beside the "Customer Detail" header. The design of the UI can be view in Figure 5.27.

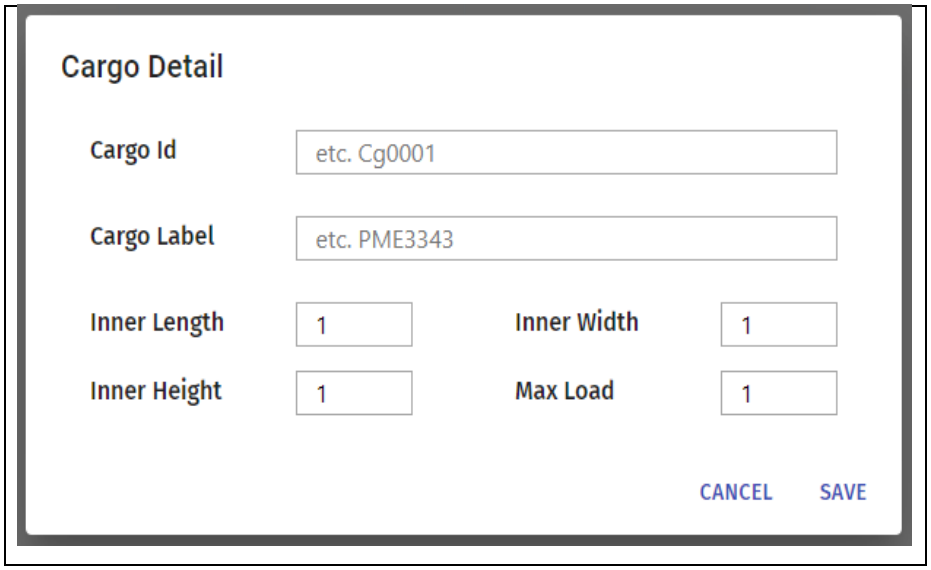


Figure 5.27 Dialog Form for cargo space's detail

5.2.2.4.2 View Cargo Space Information

Figure 5.28 Cargo Space Information Screen

The process is similar to the "D/O Information Screen" which located in section 5.1.2.1.2. Besides, on this page users are able to change the status of each cargo space. If the status column displaying with a "Tick" icon. "Tick" icon indicates that the particular cargo space will be displayed on the vehicle selection in the "Vehicle Selection" section which located at the "Home" page's "Setting" tab. "Cross" icon indicates that the particular cargo space will not be displayed on the "Vehicle Selection"

section. Once the status of the cargo space had been a change, the particular cargo id will be saved into the firestore.

```
if(newValue !== oldValue){
    let currentComponent = this;
    let userId = this.userId;
    cargoes.forEach(function(cargo){
        if(cargo.cargoType === row.cargoType){
            const db = fire.firestore();
            cargo.status = newValue;
            cargoId = cargo.cargoId;

            if(currentComponent.state.cargoSelected.indexOf(cargoId) == -1 && newValue == "true"){
                tempCargoSelected.push(cargoId);
            }
            else if(currentComponent.state.cargoSelected.indexOf(cargoId) !== -1 && newValue == "false"){
                let index = currentComponent.state.cargoSelected.indexOf(cargoId);
                tempCargoSelected.splice(index,1);
            }

            currentComponent.setState({
                cargoSelected: tempCargoSelected
            }, () => {

                console.log("****** currentComponent.state.cargoSelected: ", currentComponent.state.cargoSelected);
                db.collection("users").doc(userId).collection("CargoToDisplay").doc("cargoes").set({
                    cargoesId: currentComponent.state.cargoSelected
                })
                .then(function() {
                console.log("Document successfully written!");
```

Figure 5.29 Section code to store the cargoes' id to firestore for later reference when user change the status of a particular cargo space in the Cargoes Space Screen.

5.2.2.4.3 Edit Cargo Space Information

The process for edit item's information is similar to "Edit on D/O Information" which located in section 5.1.2.1.4 and the design of the UI is similar to Figure 5.27.

5.2.2.4.4 Delete Cargo Space Information

The process to delete a particular cargo space's record is similar to "Delete a particular D/O record" which located in section 5.1.2.1.5.

5.2.2.5 Import data (D/O, Item, and Customer) with Excel format (.csv).

It is not convenient for the user to key in those data one by one into the system if there are hundreds of data waiting to be entered into the system especially when user first time using this system. Therefore, this route optimization system provides an

alternative way to let the user store their data which is import convert the data into excel format (.csv) then import those excel files into the system. The view of the UI can be seen in Figure 5.30.

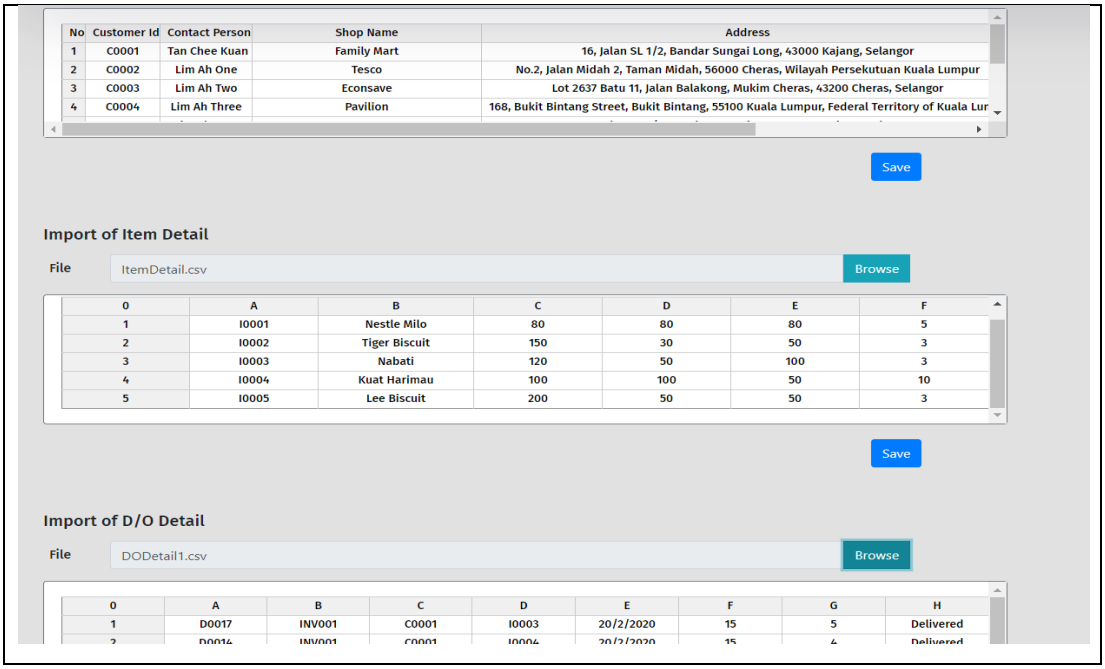


Figure 5.30 Import Data Page

Similar to other pages, when storing data to the system, the validation process may take place to check on each field of the data. The system will display an error if there is a missing field in the Excel file. System will also check if a particular data already exists on the firestore. If the data already exists another error message will be shown to ask the user to make changes to the particular D/O. Users can directly make changes on this particular page. The excel format provided by this page is able to edit directly. User may not need to re-import the excel file again.

5.2.2.6 Configure the route setting

After all relevant data had been imported into the system's firestore. User may be needed to configure the route setting which prompts the user on what is the starting location, how many numbers of truck users planned to use to deliver the D/Os and also the depart time and so on. The detail can be seen in Figure 5.31. The user would be needed to click on the "Home" page button on the navigation bar and then further click on the "Settings" button which located at the "Home" page.

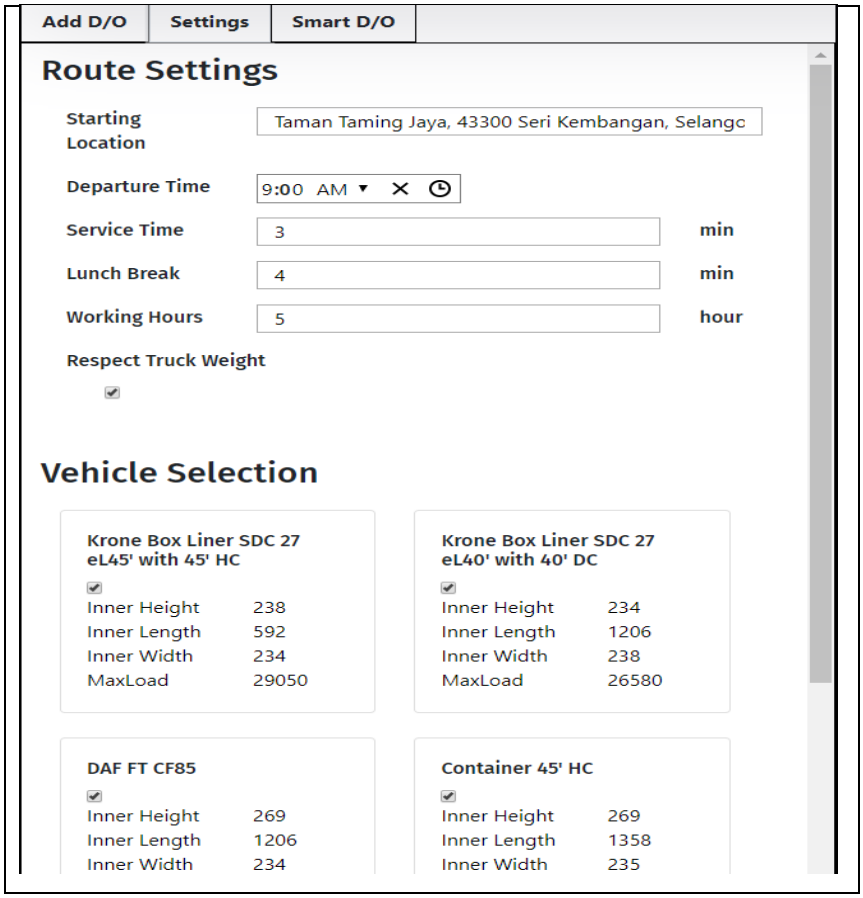


Figure 5.31 Route Settings View

This following page only needed a one-time configure after the user had signed in to the system. User can also re-configure the parameters again every time the user wishes to calculate the route. The Staring location indicates the starting point of the trip. The departure time indicates the time where the truck leaves the warehouse. The service time indicates the default service time to unload the inventories to the customer if the value of the service time for a particular D/O is 0. The lunch break indicates that the driver can have how many minutes of break for a particular delivery trip. The respect truck weight indicates whether the user wishes to follow the maximum weight of the truck when loading inventories into the truck. There may have the condition like the inventories loaded into the system already exceeded the maximum weight load of the truck but there is still much space in the truck that can be occupied. The respect truck weight checks whether the user wishes to continue to load the inventories into the truck when the maximum weight load of the truck had been exceeded.

```
saveData = () =>{
    const {google} = this.props;

    const geocoder = new google.maps.Geocoder();

    let settings = {...this.state.settings};

    const db = fire.firestore();
    let userId = localStorage.getItem('user');

    geocoder.geocode({
        'address': settings.startLocation,
    }, function(response, status){
        console.log("Status: ",status);
        console.log("Response: ", response[0]);
        settings.latitude = response[0].geometry.location.lat();
        settings.longitude = response[0].geometry.location.lng();

        db.collection("users").doc(userId).collection("RouteSetting").doc("Setting").set({
            setting: settings
        })
        .then(function() {
            console.log("Document successfully written!");
        })
        .catch(function(error) {
            console.error("Error writing document: ", error);
        });
```

Figure 5.32 Section of code which retrieve the coordinate of the starting location and store the data to the firestore.

After user had configured all the relevant data, user may be needed to click on the "Save" button which located at the bottom right corner. The validation process will be taken place. If there is no error, then the system will call the Google Map API to retrieve the coordinates of the starting location and store them together with the data which gathered in the "Route Settings" tab. Coordinates of the starting location will then be used when the system calculating on the route to deliver.

5.2.2.7 Display the D/Os' record which involved in the route calculation.

After complete the configuration process in the "Settings" tab, user can proceed to perform route calculation in order to get the suggested route to deliver. User may be needed to click on the "Smart D/O" button which was located beside the "Settings" tab. After the user had clicked at the "Smart D/O" tab user will see a list of D/Os. In the screen, only those D/O with the status of "Pending" will be displayed. D/Os with the status of "Delivered" will not be retrieved from the firestore. The view of the screen can be seen in Figure 5.33.

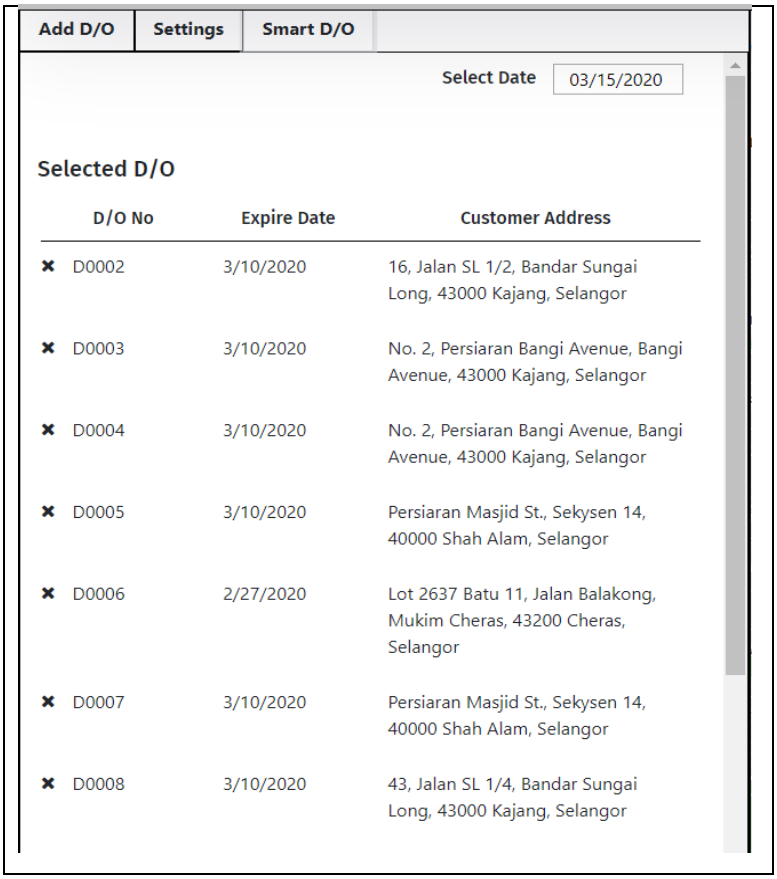


Figure 5.33 List of pending D/Os which displayed on the "Smart D/O" tab

The D/Os which listed here will be involved in the route calculation after the user press on the "Start Calculation" button. If the user just wants D/Os within an expired date range to be involved in the calculation, the user can select a particular date by clicking on the date input field located at the top right corner of the tab view. After selecting on a specific date, only D/Os with the expired date before the date that select by the user will be displayed and involved in the calculation. Besides, the user can also remove a particular D/O by clicking on the "Cross" icon beside the D/O No. After finish all the filtering process, user can click on the "Start Calculation" button which located at the bottom right of the "Smart D/O" tab view.

```
handleChange = date => {
    let timeStamp = date.getTime();
    let tempDoDatas = [...this.state.tempDODatas];

    tempDoDatas = tempDoDatas.filter( (doData, index) => {
        return (Number(doData.exDate) * 1000) < timeStamp;
    })

    this.setState({
        selectedDate: date,
        doDatas: [...tempDoDatas]
    });
};
```

Figure 5.34 Section code of filtering the list of D/Os by the user selected date.

```
removeDo = (i, doNo) => {
    this.setState((state, props) => {
        let doDatas = [...this.state.doDatas];
        let tempDoDatas = [...this.state.tempDODatas];

        tempDoDatas = tempDoDatas.filter(function(value, index) {
            return  value.doNo !== doNo;
        })

        doDatas = doDatas.filter(function(value, index) {
            return  value.doNo !== doNo;
        })

        return {
            tempDODatas: [...tempDoDatas],
            doDatas: [...doDatas]
        }
    })
}
```

Figure 5.35 Section code of removing the particular D/O's record when user click on the "Cross" Icon.

5.2.2.8 Calculate and display the suggested route on "Smart D/O" tab view.

After user click on the "Start Calculation" button, a series of activities will be performed to get the final result of the route suggested. First, the system will combine those D/O which has the same address. Next, the system will cluster all the location based on the coordinates that retrieve from the Google Map API. The number of the cluster formed will be decided by the number of the trucks that user planned to use during configure on the "Route Settings". After clusters were formed the system will be assigned a truck to each cluster based on what user had been selected. The cluster which consists of the greater total volume will be assigned with a truck which has a bigger volume. Then the system will start to compare the total volumes of the inventories in the particular cluster with the total volume of the truck. If the volume of

the truck is exceeded then the D/O with the latest expired date and larger volume will be removed. This process is continued until the volume of the truck has not been exceeded. After comparing the volume, the system will then check on whether user wishes to respect the truck weight. If the user wishes to respect the truck weight then the system will proceed to compare the maximum load weight of the truck with the total mass of inventories in the particular cluster. Last the system will pass the remaining location to the genetic algorithm to calculate the optimal route to travel. After finish on the calculation, the system will display the final result on the "Smart D/O" tab view. The result displayed can be view in Figure 5.36.

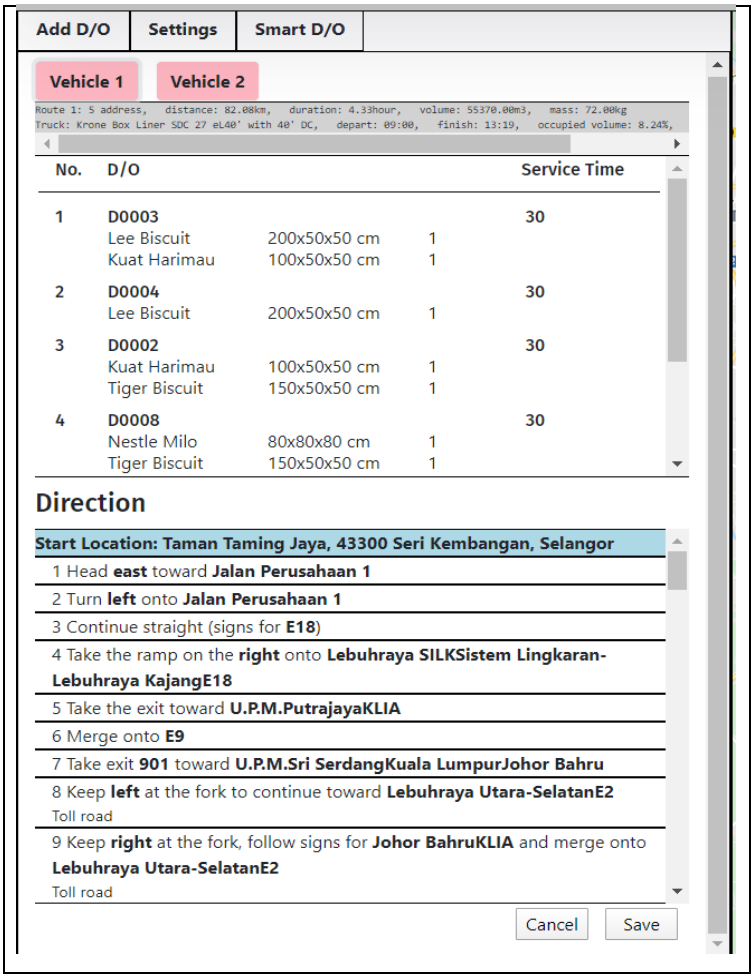


Figure 5.36 The route result after performing calculation.

User can choose to save or cancel the result. Once user clicks on the save button the status of the D/O which involved in those routes will be updated to "Delivered". This

is to avoid the particular D/O which had been delivered will not show again when performing the next calculation.

```
if(tempLocations.length > 1){
    for(let i = 0; i < number - j; i++){
        if(indicator){
            i = index;
        }

        if(tempLocations[i].address == tempLocations[i + 1].address){
            tempLocations[i].do.push(tempLocations[i + 1].do[0]);
            tempLocations.splice(i+1, 1);
            j++;
            index = i;
            indicator = true;

            if(i+1 == number-j){
                finalLocations.push(tempLocations[i]);
            }
        }

        if(i+1 !== number-j){
            if(tempLocations[i].address !== tempLocations[i + 1].address){
                indicator = false;
                finalLocations.push(tempLocations[i]);

                if(i+1 == number-j){
                    finalLocations.push(tempLocations[i + 1]);
                }
            }
        }
    }
}
else{
    finalLocations.push(tempLocations[0]);
```

Figure 5.37 Section code for combining those D/O which have the same address.

```
while (clusters.length > number) {
    // Find two closest clusters
    let minClusterDistance = Number.POSITIVE_INFINITY;
    let minClusterI = -1;
    let minClusterJ = -1;
    for (let i = 0; i < clusters.length; ++i) {
      for (let j = i + 1; j < clusters.length; ++j) {
        const dist = distance(clusters[i].center, clusters[j].center);
        if (dist < minClusterDistance) {
          minClusterDistance = dist;
          minClusterI = i;
          minClusterJ = j;
        }
      }
    }

    // Combine the two closest clusters by adding minClusterJ into minClusterI...
    clusters[minClusterI].locs = clusters[minClusterI].locs.concat(clusters[minClusterJ].locs);
    // `centroid()` takes a GeoJSON FeatureCollection and spits out a point,
    // so need to convert our list of Features into a FeatureCollection

    clusters[minClusterI].center = centroid({
      type: 'FeatureCollection',
      features: clusters[minClusterI].locs
    });

    // and removing minClusterJ
    clusters.splice(minClusterJ, 1);
}
```

Figure 5.38 Section code for grouping the location of all D/Os into several clusters.

```
        let volStatus = Number(settings.truck[i].totalVolume) < Number(newClusters[i].totalVolume);
        let weigthStatus = Number(settings.truck[i].maxLoad) < Number(newClusters[i].totalMass);

        status.volStatus = volStatus;
        status.weigthStatus = (weigthStatus);
        statusHolder.push(status);
    }

    for(let i = 0; i < number; i++){
        if(statusHolder[i].volStatus == true){
            this.removeDOBasedOnVolume(i, newClusters, settings, removedLocations);
        }
    }

    if(respectWeight == true){
        console.log("Going to removed Mass function");
        for(let i = 0; i < number; i++){
            if(statusHolder[i].weigthStatus == true){
                this.removeDOBasedOnMass(i, newClusters, settings, removedLocations);
            }
            else{
                console.log("No weight has been exist");
            }
        }
    }
    else{
        console.log("Mass Function has been skip.");
    }
```

Figure 5.39 Section code to remove D/O out from the cluster when the truck volume or truck's maximum load weight had been exceeded.

```
        var geneticAlgorithmConstructor = require('geneticalgorithm')
        var geneticAlgorithm = geneticAlgorithmConstructor({
            mutationFunction: this.mutationFunction,
            crossoverFunction: this.crossoverFunction,
            fitnessFunction: this.fitnessFunction,
            population: [ cluster.locations ],
            populationSize:100
        });

        var best = []
        var previousBestScore = 0
        var currentResult = []
        for( var a = 0 ; a < 100 ; a++ ) {
            for( var i = 0 ; i < 25 ; i++ )
                geneticAlgorithm.evolve()

            var score = geneticAlgorithm.bestScore()
            if ( score == previousBestScore ) {
                break;
            }
            previousBestScore = score;

            currentResult.push(score);
        }
        best = geneticAlgorithm.best()
        newRoute.push(best);
        currentResult.push(this.fitnessFunction(best));
```

Figure 5.40 Section code to trigger the genetic algorithm.

```
    console.log("Document successfully written!");
    directionObj.doNumber.forEach((doNo, index) => {
        console.log("@@@@@@ doNo in update: ", doNo);
        db.collection("users").doc(userId).collection("DOs").doc(doNo).update({
            status: "Delivered"
        })
        .then(function() {
            console.log("DOs document successfully updated!");
        })
        .catch(function(error) {
            console.error("Error writing DOs document: ", error);
        });

    })
)
```

Figure 5.41 Section code to update the status of D/Os in the firestore.

5.2.2.9 Visualization of the calculated route on the map.

It is good to have the visualization of the route, therefore user can check whether the solution provided by the system is trustable or not. With the help of Google Map API, this system is able to provide the view of the calculated route on the map. The visualization of the calculated result can be seen in Figure 5.42.

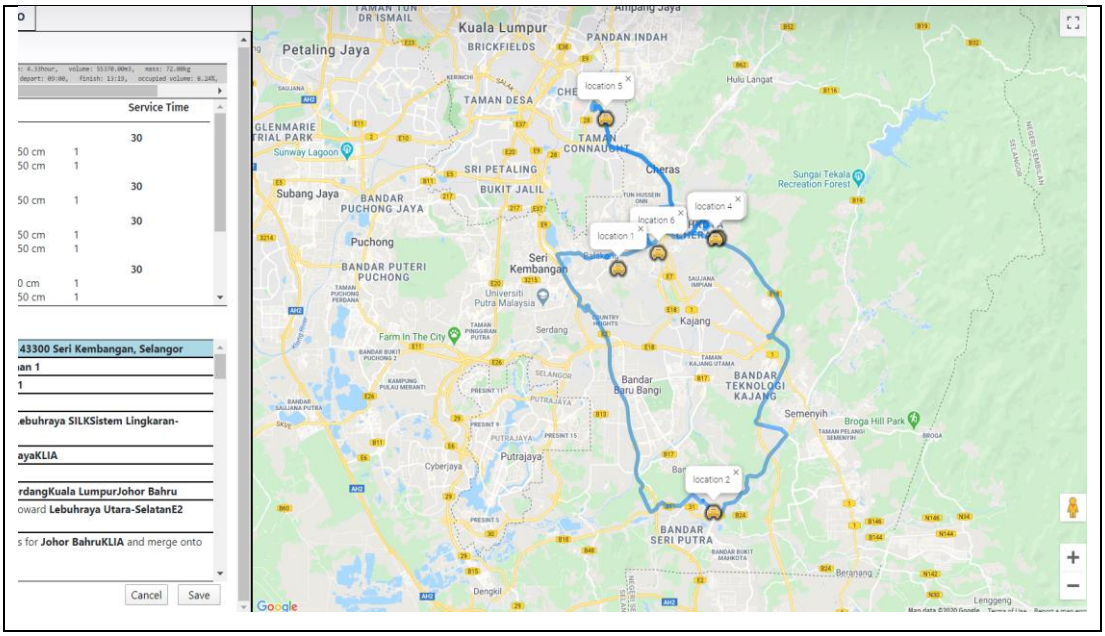


Figure 5.42 Visualization of calculated route on the map.

Besides, this system also creates the marker for each location which is going to deliver. The markers will then displayed on the Map together with the route. Therefore user is

able to have a better view on the sequence of the route to deliver from the starting point to multiple waypoint then back to the origin again.

```
    var directionsRenderer = new google.maps.DirectionsRenderer({
        suppressMarkers: true
    });

    let map = new google.maps.Map(document.getElementById('mobilemap'), {
    mapTypeControl: false,
    center: {
        lat: latitude,
        lng: longitude
    },
    zoom: 100
    });

    directionsRenderer.setMap(map);

    directionsRenderer.setDirections(response);

    let num = response.routes[0].legs.length;

    response.routes[0].legs.forEach( (leg,index) => {
        let string = "location ";
        index ++;
            string += index;
            this.createMarker(leg.start_location, index , string, directionsRenderer.getMap(), infowindow);
    })
}
```

Figure 5.43 Section code to display the route on the map.

```
createMarker = (location, label, content, map, id) => {
    const {google} = this.props;
    var marker = new google.maps.Marker({
      position: location,
      title: label,
      id: id,
      icon: {
        url: 'https://maps.google.com/mapfiles/kml/pal4/icon31.png',

        anchor: new google.maps.Point(0, 0)
      },
      map: map
    });
    infowindow = new google.maps.InfoWindow({
      content: content,
      maxWidth: 350
    });
    infowindow.setContent(content);
    infowindow.open(map, marker);
}
```

Figure 5.44 Section code to create marker on the map.

5.2.2.10 Save system's planned route

After the system displayed the result on the "Smart D/O" tab view, user has the choice to save the result. Once user clicked on the "Save" button, the route results will be store into the firestore.

```
db.collection("users").doc(userId).collection("routesRecord").doc(id).set({
    id: id,
    createdTimeStamp: parseInt(timeStamp),
    wayPoints: [...directionObj.wayPoints],
    distanceTravel: Number(directionObj.distance),
    totalDuration: Number(directionObj.duration),
    instructions: [...tempInstructions],
    totalVolume: Number(directionObj.totalVolume),
    totalMass: Number(directionObj.totalMass),
    truckVolume: Number(directionObj.truckVolume),
    truckMass: Number(directionObj.truckMass),
    truckType: directionObj.truckType,
    departTime: directionObj.departTime,
    finishTime: directionObj.finishTime,
    legDuration: [...directionObj.legDuration],
    legDistance: [...directionObj.legDistance],
    doNumber: [...directionObj.doNumber]
})
.then(function() {
    console.log("Document successfully written!");
    directionObj.doNumber.forEach((doNo, index) => {
```

Figure 5.45 Section code of storing the route record to the firestore.

5.2.2.11 View system's planned route

This system enables user to view back the previously planned route by clicking on the "Route Record" button which located at the navigation bar. The UI of the screen can be seen in Figure 5.46.

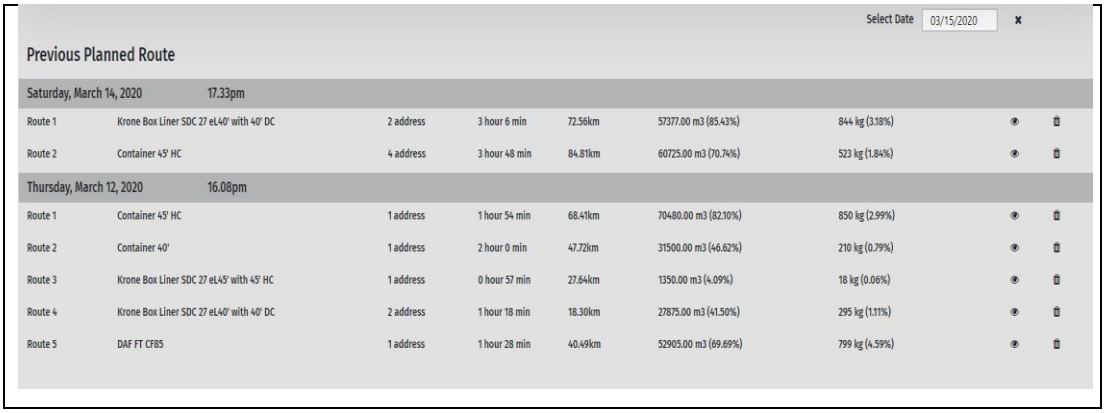


Figure 5.46 Route Record Screen

The system enables user to select a particular day's record by selecting a date on the date picker. After selecting on a particular date, the record which matches to the date selected by the user will be displayed else the system will display a message which is "No Records Can Be Found In This Particular Date". From Figure 5.46 user is able to know that the system provides functionality like view the particular route record by clicking on the "Eye" icon and delete a particular route record by clicking on the "Trash" icon.

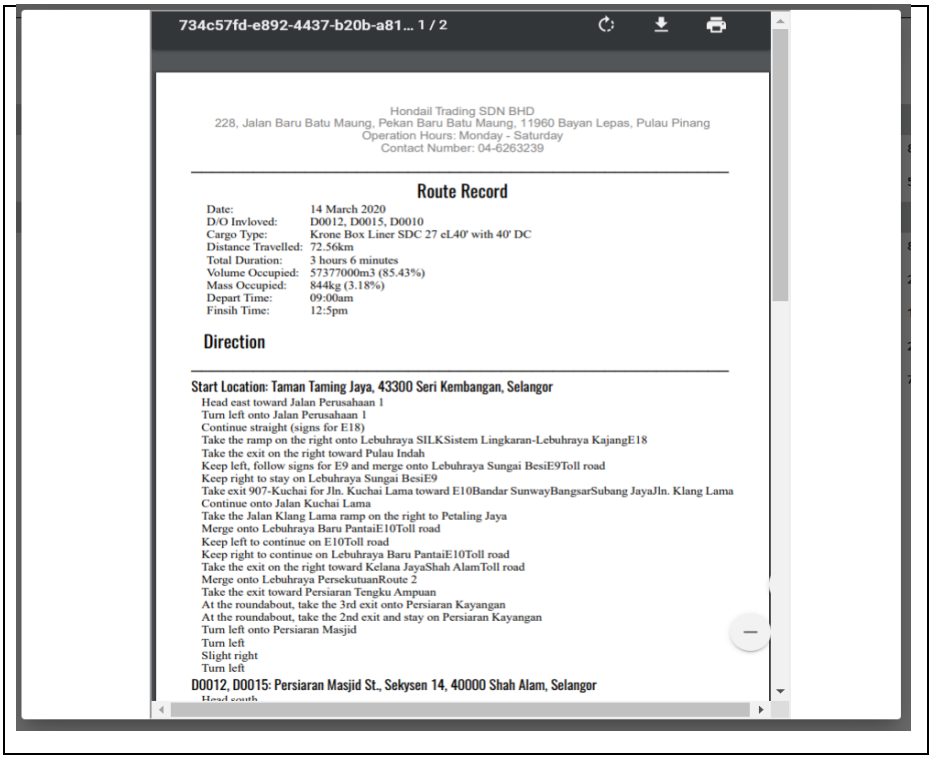


Figure 5.47 Detail of a planned route in the PDF format.

The flow is much more similar to section 5.1.2.1.3.

5.2.2.12 Delete system's planned route

This system enables user to delete the route record, a dialog message will be pop up to prompt user whether he/she wants to continue the following action. If user clicks on the "Confirm" button, then the particular route record will be deleted permanently from the system.

```
db.collection("users").doc(userId).collection("routesRecord").doc(route.id).delete()
.then(function() {

    tempRoutes = tempRoutes.filter((tempRoute, index) => {
        return tempRoute.id !== route.id;
    })

    tempRecordHolder = tempRecordHolder.filter((tempRoute, index) => {
        return tempRoute.id !== route.id;
    })

    currentComponent.setState({
        routesRecord: tempRoutes,
        routedRecordHolder: tempRecordHolder
    })

    console.log("Document successfully deleted!");
})
.catch(function(error) {
    console.error("Error removing document: ", error);
});
```

Figure 5.48 Section of code that used to delete the route record from the firestore.

5.2.3 Company Admin's functionality

5.2.3.1 Add and enable staff to access the system and also company information.

In order for staff within the same company to plan the route by using company information such as managing a company's D/O, company's customer, company's items and company's cargoes. User must first sign in with a company admin account which has a "Staff"' button located at the navigation bar. This system allows the company admin to add staff to access the system with company's information shown to them. Company's Admin may need to go for the particular page and the clicks on the "Add People" icon which located beside the header "Staff Detail". A popup dialog will appear to prompt the staff's id, staff's name, validated email and then the password. The staff's email and password will then pass to firebase authentication for validation. If the email provided is a valid email then an account to access the system will be created. Then the staff detail will be saved into the firestore. The design of the popup dialog can be view in figure 5.49.

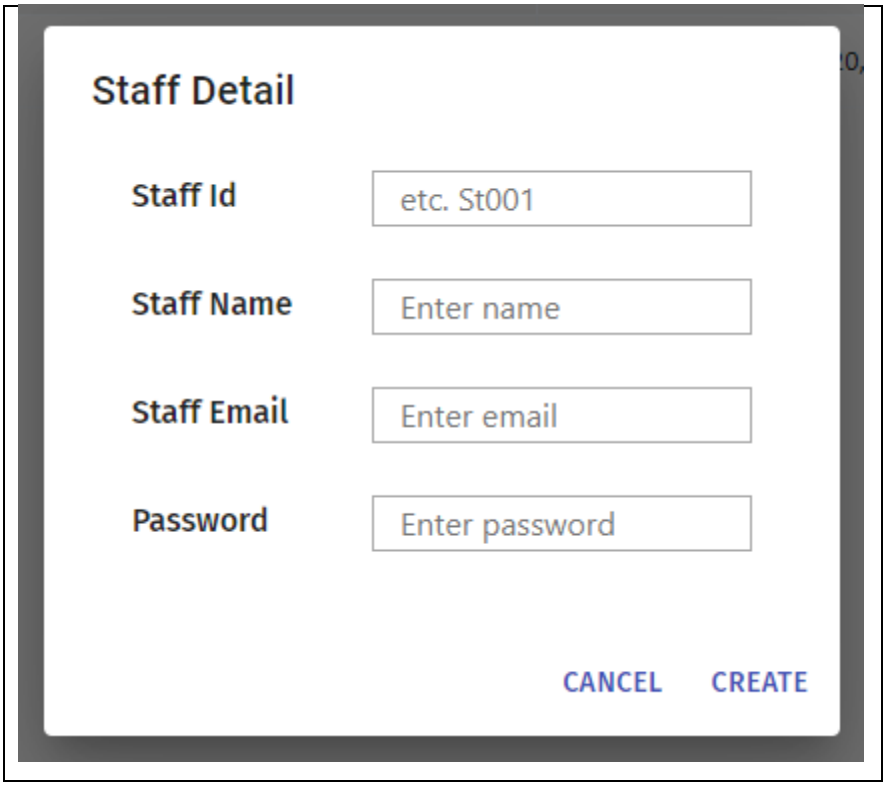


Figure 5.49 Dialog Form to create the staff

```
e.preventDefault();
fire.auth().createUserWithEmailAndPassword(selectedStaff.staffEmail, selectedStaff.staffPass)
.then((u)=>{

    db.collection("users").doc(u.user.uid).set({
        userId: u.user.uid,
        staffId: selectedStaff.staffId,
        name: selectedStaff.staffName,
        email: selectedStaff.staffEmail,
        admin: false,
        companyAdmin: false,
        referId: this.userId,
        createTimeStamp: Number(createdDateTimeStamp),
        loginTimeStamp: Number(0),
        logoutTimeStamp: Number(0),
        status: "true"
    })
    .then(function() {
        console.log("Document successfully written!");
        let staffs = [];
        db.collection('users').where("referId", "==", currentComponent.userId).get()
        .then(function(querySnapshot){
            console.log(querySnapshot)
            return Promise.all(
                querySnapshot.docs.map(function(doc) {
```

Figure 5.50 Section code of creating an authenticated staff account by using firebase authentication and store the data into firestore.

5.2.3.2 View staff information and change staff's status

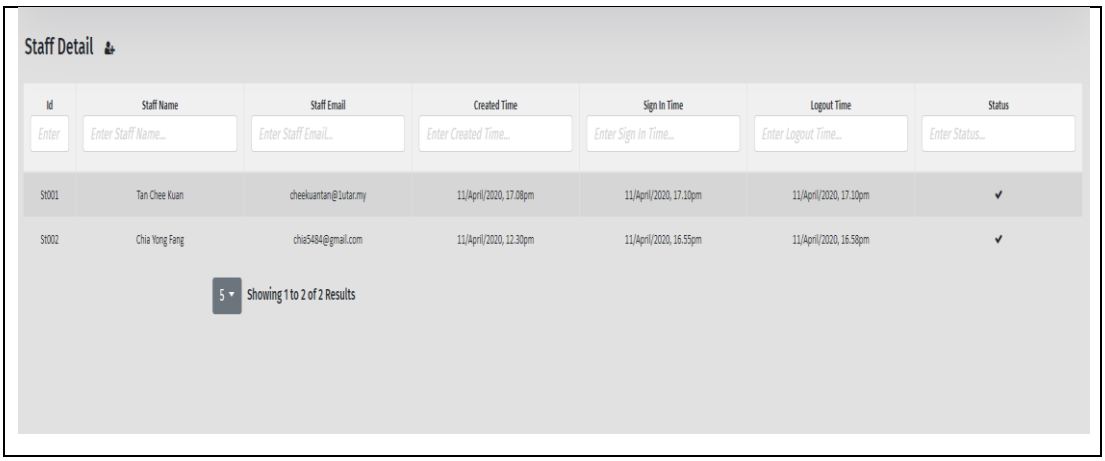


Figure 5.51 Staff Information Screen

The process is similar to the "D/O Information Screen" which located in section 5.1.2.1.2. Besides, in this page company admin is able to change the status of the staff. Once the status of the staff has been a change, the status of the staff will update on the firestore also. "Tick" icon indicated that the particular staff account had been activated while the "Cross" icon indicated that the particular staff account had been deactivated. Staff with an activated account can login to the system and use this system to calculate the route to deliver with company's information. Staff with a deactivated account is not able to access the system.

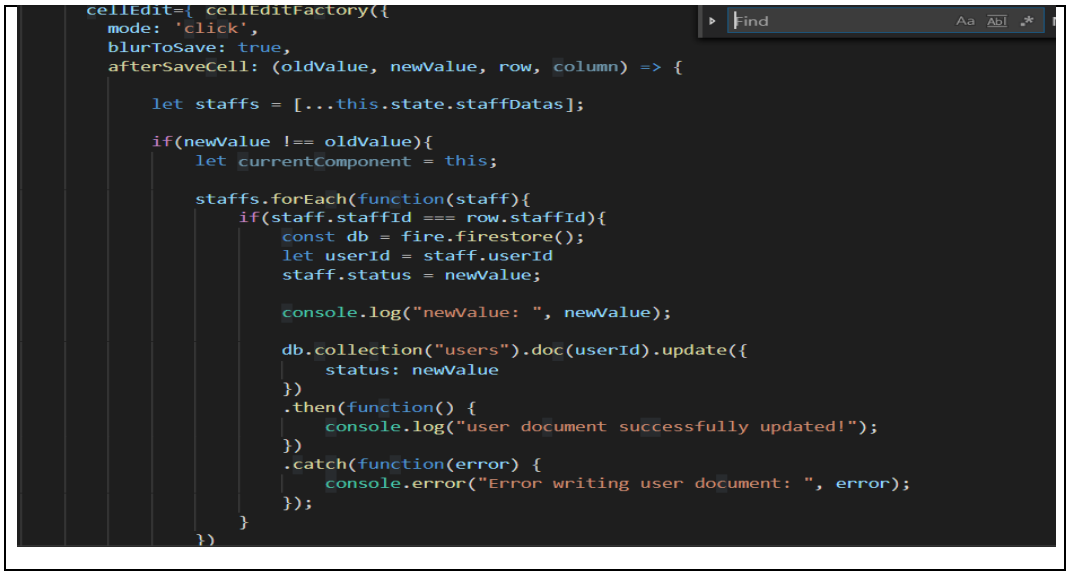


Figure 5.52 Section code to update the status of the staff.

CHAPTER 6
SYSTEM TESTING

6.1 Unit Testing

Unit testing is a low level of software testing which used to make sure that each component/unit of the application can perform well as what the developer had been designed (Software Testing Fundamentals, 2020). Different kinds of programming practices/styles will have a slightly different definition on the term “unit”. In procedural programming, “unit” is defined as the individual program, procedure or function. While in object-oriented programming, “unit” is referring to the class and the method. When performing unit testing, developer may need to make sure that each unit testing is independent toward each other. Which means, the input or output of a particular unit test case must not be able to affect other test cases.

Before performing unit testing, developer may need to have a list of unit test cases which including all the possible conditions. By having a unit test case, developer can easily know what is the input and output required to pass all the test. Each test-case must have a specific test case id to differentiate the uniqueness of each test case.

6.1.1 Test Coverage

Table 6.1 Test coverage for each project scope

Roles	Project Scope	Test Case ID	Status
Logistic Company Staff and Company Admin / Owner	Provide user interface for user to import their data.	ImportD/O -1A	Pass
		ImportD/O -1B	Pass
		ImportItem -1A	Pass
		ImportItem -1B	Pass
		ImportCustomer -1A	Pass
		ImportCustomer -1B	Pass
	Manage Customer information.	CreateCustomer -1A	Pass
		CreateCustomer -1B	Pass
		ViewCustomer -1A	Pass
		ViewCustomer -1B	Pass
		EditCustomer -1A	Pass
		EditCustomer -1B	Pass
		DeleteCustomer -1A	Pass
	Manage Item information.	CreateItem -1A	Pass
		CreateItem -1B	Pass
		ViewItem-1A	Pass

		ViewItem -1B	Pass
		EditItem-1A	Pass
		EditItem -1B	Pass
		DeleteItem -1A	Pass
	Manage Cargo Space Information	CreateCargoSpace -1A	Pass
		CreateCargoSpace -1B	Pass
		ViewCargoesSpace -1A	Pass
		ViewCargoesSpace -1B	Pass
		EditCargoSpace -1A	Pass
		EditCargoSpace -1B	Pass
		DeleteCargoSpace -1A	Pass
	Manage D/O information.	CreateD/O -1A	Pass
		CreateD/O -1B	Pass
		ViewD/O-1A	Pass
		ViewD/O-1B	Pass
		EditD/O-1A	Pass
		EditD/O-1B	Pass
		DeleteD/O -1A	Pass
	Calculate the shortest route to deliver.	ConfigureRoute -1A	Pass

		ConfigureRoute -1B	Pass
		ViewSelectedD/O-1A	Pass
		ViewSelectedD/O-1B	Pass
		CalculateRoute -1A	Pass
		CalculateRoute -1B	Pass
		CalculateRoute -1C	Pass
		VisualizeRoute -1A	Pass
		CreateRoute -1A	Pass
	Route Record page which used to display the information of the previous calculated route.	CreateRoute -1A	Pass
		ViewRoute -1A	Pass
		ViewRoute -1B	Pass
		DeleteRoute -1A	Pass
Company Admin / Owner	6.1.3.1 Manage Staff Information	CreateStaff -1A	Pass
		CreateStaff -1B	Pass
		ViewStaff -1A	Pass
		ViewStaff -1B	Pass
		ChangeStaffStatus -1A	Pass

6.1.2 Unit Test Cases (Basic Feature)

6.1.2.1 Sign In feature

Generally, user without a valid account is not able to access on the other screen of this system. The only screens they are able to view are Sign-in screen and Sign-up screen. Besides, this system consists of two roles which are normal user (logistic company's staff) and the company admin. Therefore, some validation processes must be done to differentiate on these two roles because each role is only able to access on specific functionalities assigned to them. For example, company admin is able to add/create staff authenticated account for company's staff to access the system which is not able to perform by normal company's staff.

Test Scenario ID	Login-1	**Test Case ID**	Login-1A
Test Case Description	Login – Positive test case	**Test Priority**	High
Pre-Requisite	A valid user's account	**Post-Requisite**	N/A

Test Execution Steps:

S.No	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Test Comments**
1	1. Enter correct Email & Password for normal user and	Email: cheekuantan@1utar.my	Login success and then system direct user to the home page.	User login into the system.	Pass	-

	click on the "Sign In" button.	Password: ******				

Test Scenario ID	Login-1	**Test Case ID**	Login-1B
Test Case Description	Login – Negative test case	**Test Priority**	High
Pre-Requisite	N/A	**Post-Requisite**	N/A

Test Execution Steps:

S.No	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Test Comments**
1	1. Enter invalid email & any password. 2. Click on the "Sign In" button.	Email: cheekuantan@1utar.my Password: ******	Display error message "*User Not Found. Please Sign Up for an account.*". User is retain at the Sign-In page.	"*User Not Found. Please Sign Up for an account.*". User is retain at the Sign-In page.	Pass	-

2	1. Enter valid email & incorrect password. 2. Click on the "Sign In" button.	Email: cheekuantan@1utar.my Password: ******	Display error message "*Password is invalid. Please try again*.". User is retain at the Sign-In page.	"*Password is invalid. Please try again*.". User is retain at the Sign-In page	Pass	-

6.1.2.2 Sign Up feature

Generally, user without a valid account will need to sign up for an account before accessing to the system. The system will check on whether the email entered by the user already registered or not. User will register for a company's admin account when performing sign up through this system. Account for normal staff can only be created when company admin login into this system.

Test Scenario ID	SignUp-1	**Test Case ID**	SignUp -1A
Test Case Description	SignUp – Positive test case	**Test Priority**	High
Pre-Requisite	N/A	**Post-Requisite**	N/A
Test Execution Steps:			

S.No	Action	Inputs	Expected Output	Actual Output	Test Result	Test Comments
1	1. Enter all the require input at the sign-up page. 2. Click on the "Sign Up" button.	First name: Tan Last name: Chee Kuan Company name: Hondail Trading Company address: 228, Jln Baru Batu Maung, Pekan Baru Batu Maung, 11960 Bayan Lepas, Pulau Pinang Operating hour: Monday – Saturday	SignUp success and direct user to the system "Home" page.	User was directed to the home page.	Pass	-

		Contact no: 012-3456789 Email address: cheekuantan1998@gmail.com Password: ******				

Test Scenario ID	SignUp-1		**Test Case ID**	SignUp -1B	
Test Case Description	SignUp – Negative test case		**Test Priority**	High	
Pre-Requisite	N/A		**Post-Requisite**	N/A	
Test Execution Steps:					
S.No	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Test Comments**

1	1. Enter an invalid email address with all the require input at the sign-up page. 2. Click on the "Sign Up" button.	First name: Tan Last name: Chee Kuan Company name: Hondail Trading Company address: 228, Jln Baru Batu Maung, Pekan Baru Batu Maung, 11960 Bayan Lepas, Pulau Pinang Operating hour: Monday – Saturday	Display error message "*Invalid email address*.". User retain at the Sign Up page.	"*Invalid email address*.". User retain at the Sign Up page.	Pass	-

		Contact no: 012-3456789 Email address: cheekuantan Password: ******				
2	1. Enter a valid email address which already exist in the system with all the require input at the sign-up page.	First name: Tan Last name: Chee Kuan Company name: Hondail Trading Company address:	Display error message "*The email address is already in use by another account.*". User retain at the Sign Up page.	"*The email address is already in use by another account.*". User retain at the Sign Up page.	Pass	-

	2. Click on the "Sign Up" button.	228, Jln Baru Batu Maung, Pekan Baru Batu Maung, 11960 Bayan Lepas, Pulau Pinang Operating hour: Monday – Saturday Contact no: 012-3456789 Email address: Cheekuantan1998@gmail.com Password: ******				

3	1. Enter a valid email address with some of the input field remain empty at the sign-up page. For example: Company address is empty. 2. Click on the "Sign Up" button.	First name: Tan Last name: Chee Kuan Company name: Hondail Trading Company address: "" Operating hour: Monday – Saturday Contact no: 012-3456789 Email address:	Display error message "*Company address is require.*". User retain at the Sign Up page.	"*Company address is require.*". User retain at the Sign Up page.	Pass	-

		Cheekuantan1998@gmail.com Password: ******				

6.1.3 Unit Test Cases (Logistic Company's Staff and Company Admin)

6.1.3.1 Manage D/O Information

6.1.3.1.1 Create D/O Information

User is able to create D/O one by one by clicking on the "Add D/O" tab which located at the "Home" screen.

Test Scenario ID	CreateD/O-1	**Test Case ID**	CreateD/O -1A
Test Case Description	CreateD/O – Positive test case	**Test Priority**	High
Pre-Requisite	Logged in with a valid user account	**Post-Requisite**	N/A
Test Execution Steps:			

S.No	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Test Comments**
1	1. Enter all the require input at the "Add D/O" tab which located at "Home" screen. 2. Click on the "Save" button.	D/O No: D0001 Expire Date: 3/20/2020 Contact Person: Tan Chee Kuan Shop Name: Family Mart Address: 16, Jalan SL 1/2, Bandar Sungai Long, 43000 Kajang, Selangor	Display message "D/O content has been successfully created!"	"D/O content has been successfully created!"	Pass	-

		State: Kuala Lumpur Post Code: 43000 Country: Malaysia Contact: 0123456789 Service Time: 30 Items: At least one item is added				

Test Scenario ID	CreateD/O-1	**Test Case ID**	CreateD/O -1B
Test Case Description	CreateD/O – Negative test case	**Test Priority**	High
Pre-Requisite	Logged in with a valid user account	**Post-Requisite**	N/A

Test Execution Steps:

S.No	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Test Comments**
1	1. Some of the input field remain empty at "Add D/O" tab which located at "Home" screen. For example: D/O No is empty 2. Click on the "Save" button.	D/O No: "" Expire Date: 3/20/2020 Contact Person: Tan Chee Kuan Shop Name: Family Mart	Display error message "*D/O No is require*"	"*D/O No is require*"	Pass	-

		Address: 16, Jalan SL 1/2, Bandar Sungai Long, 43000 Kajang, Selangor State: Kuala Lumpur Post Code: 43000 Country: Malaysia Contact: 0123456789 Service Time:				

		30 Items: At least one item is added				
2	1. All input field had been filled while there is no item added to that D/O. 2. Click on the "Save" button.	D/O No: D0001 Expire Date: 3/20/2020 Contact Person: Tan Chee Kuan Shop Name: Family Mart	Display error message "*Minimum one item should be added*"	"*Minimum one item should be added*"	Pass	-

		Address: 16, Jalan SL 1/2, Bandar Sungai Long, 43000 Kajang, Selangor State: Kuala Lumpur Post Code: 43000 Country: Malaysia Contact: 0123456789 Service Time:				

		30 Items: []				
3	1. Enter the D/O No which already exist in the system. 2. Click on the "Save" button.	D/O No: D0001 Expire Date: 3/20/2020 Contact Person: Tan Chee Kuan Shop Name: Family Mart Address:	Display error message "D/O No already exist! Please change the D/O No."	"D/O No already exist! Please change the D/O No."	Pass	-

		16, Jalan SL 1/2, Bandar Sungai Long, 43000 Kajang, Selangor State: Kuala Lumpur Post Code: 43000 Country: Malaysia Contact: 0123456789 Service Time: 30				

		Items: At least one item is added				

6.1.3.1.2View D/O Information

User is enables to read the information of the D/O by clicking on the "D/O Information" which located at the navigation bar.

Test Scenario ID	ViewD/O-1		**Test Case ID**	ViewD/O-1A	
Test Case Description	ViewD/O – Positive test case		**Test Priority**	High	
Pre-Requisite	- Logged in with a valid user account - System already have D/Os information in the database		**Post-Requisite**	N/A	
Test Execution Steps:					
S.No	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Test Comments**

1	Click on the "D/O Information" button at the navigation tab.	D/Os' information from the firestore.	Display a list of D/O information to user.	List of D/O were displayed.	Pass	-
2	Click on the "eye" icon which located at the 'Action" column of each row of D/O.	D/Os' information from the firestore.	Display the particular D/O information in PDF format.	Detail of a particular D/O was displayed	Pass	-

Test Scenario ID	ViewD/O-1	**Test Case ID**	ViewD/O-1B
Test Case Description	ViewD/O – Negative test case	**Test Priority**	High
Pre-Requisite	- Logged in with a valid user account - System does not have any D/Os information in the database.	**Post-Requisite**	N/A

Test Execution Steps:

S.No	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Test Comments**
1	Click on the "D/O Information" button at the navigation tab.	empty	Display message "No Records Can Be Found. Please create your D/O by clicking on the + icon."	"No Records Can Be Found. Please create your D/O by clicking on the + icon."	Pass	-

6.1.3.1.3Edit D/O Information

User is enables to edit the information of the D/O by clicking on the "Pencil" icon which located at the "D/O Information" screen.

Test Scenario ID	EditD/O-1	**Test Case ID**	EditD/O-1A
Test Case Description	EditD /O – Positive test case	**Test Priority**	High
Pre-Requisite	- Logged in with a valid user account - System already have D/Os information in the database	**Post-Requisite**	N/A

Test Execution Steps:

S.No	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Test Comments**
1	1. Click on the "eye" icon which located at the "Action" column of each row of D/O.	- D/Os' information from the firestore. - Service Time: 50	Display message "D/O content has been successfully save!"	"D/O content has been successfully save!"	Pass	-

	2. Modify the data. For example, user wish to change the Service Time of the particular D/O to 50 min. 3. Click on the "Save" button.					

Test Scenario ID	EditD/O-1	**Test Case ID**	EditD/O-1B
Test Case Description	EditD /O – Negative test case	**Test Priority**	High
Pre-Requisite	- Logged in with a valid user account - System already have D/Os information in the database	**Post-Requisite**	N/A

Test Execution Steps:

S.No	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Test Comments**
1	1. Click on the "eye" icon which located at the "Action" column of each row of D/O. 2. Modify the data. For example, user wish to change	- D/Os' information from the firestore. - Status: ""	Display error message *"Status is require"*	*"Status is require"*	Pass	-

	the status of the particular D/O. 3. Leave the status field empty. 4. Click on the “Save” button.					

6.1.3.1.4Delete D/O Information

User is allows to delete the information of a particular D/O by clicking on the "Trash" icon which located at the "D/O Information" screen.

Test Scenario ID	DeleteD/O-1			**Test Case ID**	DeleteD/O -1A	
Test Case Description	DeleteD/O – Positive test case			**Test Priority**	High	
Pre-Requisite	- Logged in with a valid user account - System already have D/Os' information in the database			**Post-Requisite**	N/A	
Test Execution Steps:						
S.No	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Test Comments**
1	1. Click on the "trash" icon which located at the "Action" column of each row of D/O.	- D/Os' information from the firestore.	Display message "D/O content has been successfully removed!"	"D/O content has been successfully removed!"	Pass	-

	2. System prompt user to confirm the action. 3. Click on the "Confirm" button.					

6.1.3.2 Manage Item Information

6.1.3.2.1 Create Item Information

After user had login, they are able to create Item one by one by clicking on the "+" icon which located at the "Item Information" screen.

Test Scenario ID	CreateItem-1	**Test Case ID**	CreateItem -1A
Test Case Description	CreateItem – Positive test case	**Test Priority**	High
Pre-Requisite	Logged in with a valid user account	**Post-Requisite**	N/A

Test Execution Steps:

S.No	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Test Comments**
1	1. Click on the "+" icon which located beside the "Item Detail" header.	ItemId: I0001 ItemLabel: Lee Biscuit Length:	Display message "Item content has been successfully created!"	"Item content has been successfully created!"	Pass	-

	2. Enter all the require input into the “Item Detail” form. 3. Click on the “Save” button.	100 Width: 80 Height: 80 Weight: 2				

Test Scenario ID	CreateItem -1	**Test Case ID**	CreateItem -1B
Test Case Description	CreateItem – Negative test case	**Test Priority**	High
Pre-Requisite	Logged in with a valid user account	**Post-Requisite**	N/A

Test Execution Steps:

S.No	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Test Comments**
1	1. Some of the input field in the Item Detail form remain empty. For example: Item Id is empty 2. Click on the "Save" button.	ItemId: "" ItemLabel: Lee Biscuit Length: 100 Width: 80	Display error message "*Item Id is require*"	"*Item Id is require*"	Pass	-

		Height: 80 Weight: 2				
2	1. Enter the Item Id which already exist in the system. 2. Click on the "Save" button.	ItemId: I0001 ItemLabel: Lee Biscuit Length: 100 Width: 80	Display error message "I0001 already exist! Please change the Item Id."	"I0001 already exist! Please change the Item Id."	Pass	-

		Height: 80 Weight: 2				

6.1.3.2.2View Item Information

User is enables to read the information of the Item by clicking on the “Item Information” which located at the navigation bar.

Test Scenario ID	ViewItem-1		**Test Case ID**	ViewItem-1A	
Test Case Description	ViewItem – Positive test case		**Test Priority**	High	
Pre-Requisite	- Logged in with a valid user account - System already have items’ information in the database		**Post-Requisite**	N/A	
Test Execution Steps:					
S.No	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Test Comments**

1	Click on the "Item Information" button at the navigation tab.	Items' information from the firestore.	Display a list of Item's information to user.	List of Item's information were displayed.	Pass	-

Test Scenario ID	ViewItem -1		**Test Case ID**	ViewItem -1B	
Test Case Description	ViewItem – Negative test case		**Test Priority**	High	
Pre-Requisite	- Logged in with a valid user account - System does not have any items' information in the database.		**Post-Requisite**	N/A	
Test Execution Steps:					
S.No	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Test Comments**
1	Click on the "Item Information"	empty	Display message "No Records Can Be Found. Please create	"No Records Can Be Found. Please create your item	Pass	-

	button at the navigation tab.		your item by clicking on the + icon."	by clicking on the + icon."		

6.1.3.2.3Edit Item Information

User is enables to edit the information of a particular item by clicking on the "Pencil" icon which located at the "Item Information" screen.

Test Scenario ID	EditItem-1		**Test Case ID**	EditItem-1A	
Test Case Description	EditItem – Positive test case		**Test Priority**	High	
Pre-Requisite	- Logged in with a valid user account - System already have items' information in the database		**Post-Requisite**	N/A	
Test Execution Steps:					
S.No	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Test Comments**
1	1. Click on the "eye" icon which located at the	- Items' information from the firestore.	Display message "Item content has been successfully save!"	"Item content has been successfully save!"	Pass	-

	"Action" column of each row of item. 2. Modify the data. For example, user wish to change the length of the particular item to 120 cm. 3. Click on the "Save" button.	- Length: 120				

Test Scenario ID	EditItem-1		**Test Case ID**	EditItem -1B	
Test Case Description	EditItem – Negative test case		**Test Priority**	High	
Pre-Requisite	- Logged in with a valid user account - System already have items' information in the database		**Post-Requisite**	N/A	
Test Execution Steps:					
S.No	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Test Comments**
1	1. Click on the "eye" icon which located at the "Action" column of each row of item. 2. Modify the data. For example, user wish to change	- Items' information from the firestore. - Item Label: ""	Display error message "Item Label is require"	"Item Label is require"	Pass	-

	the item label of a particular item. 3. Leave the item label field empty. 4. Click on the "Save" button.					

6.1.3.2.4Delete Item Information

User is allows to delete the information of a particular item by clicking on the "Trash" icon which located at the "Item Information" screen.

Test Scenario ID	DeleteItem-1	**Test Case ID**	DeleteItem -1A
Test Case Description	DeleteItem – Positive test case	**Test Priority**	High
Pre-Requisite	- Logged in with a valid user account - System already have items' information in the database	**Post-Requisite**	N/A

Test Execution Steps:						
S.No	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Test Comments**
1	1. Click on the "trash" icon which located at the "Action" column of each row of cargo space. 2. System prompt user to confirm the action. 3. Click on the "Confirm" button.	- Items' information from the firestore.	Display message "Item content has been successfully removed!"	"Item content has been successfully removed!"	Pass	-

6.1.3.3 Manage Customer Information

6.1.3.3.1 Create Customer Information

User is allows to add Customer information into the system one by one by clicking on the "+" icon which located at the "Customer Information" screen.

Test Scenario ID	CreateCustomer-1	**Test Case ID**	CreateCustomer -1A
Test Case Description	CreateCustomer – Positive test case	**Test Priority**	High
Pre-Requisite	Logged in with a valid user account	**Post-Requisite**	N/A

Test Execution Steps:

S.No	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Test Comments**
1	1. Click on the "+" icon which located beside the "Customer Detail" header.	CustomerId: C0001 CustomerName: Andrew Kim	Display message "Customer content has been successfully created!"	"Customer content has been successfully created!"	Pass	-

	2. Enter all the require input into the "Customer Detail" form. 3. Click on the "Save" button.	Shop Name: Family Mart Address: 16, Jalan SL 1/2, Bandar Sungai Long, 43000 Kajang, Selangor City: Sungai Long Post Code: 43600 State: Kuala Lumpur Country: Malaysia				

		Contact Number: 012-3456789				

Test Scenario ID	CreateCustomer -1	**Test Case ID**	CreateCustomer -1B
Test Case Description	CreateCustomer – Negative test case	**Test Priority**	High
Pre-Requisite	Logged in with a valid user account	**Post-Requisite**	N/A

Test Execution Steps:

S.No	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Test Comments**
1	1. Some of the input field in the Customer Detail form remain empty.	CustomerId: “” CustomerName: Andrew Kim	Display error message “*Customer Id is require*”	“*Customer Id is require*”	Pass	-

	For example: CustomerId is empty 2. Click on the "Save" button.	Shop Name: Family Mart Address: 16, Jalan SL 1/2, Bandar Sungai Long, 43000 Kajang, Selangor City: Sungai Long Post Code: 43600 State: Kuala Lumpur Country:				

		Malaysia Contact Number: 012-3456789				
2	1. Enter the Customer Id which already exist in the system. 2. Click on the "Save" button.	CustomerId: C0001 CustomerName: Andrew Kim Shop Name: Family Mart Address: 16, Jalan SL 1/2, Bandar Sungai Long, 43000 Kajang, Selangor	Display error message "C0001 already exist! Please change the Customer Id."	"C0001 already exist! Please change the Customer Id."	Pass	-

		City: Sungai Long Post Code: 43600 State: Kuala Lumpur Country: Malaysia Contact Number: 012-3456789				

6.1.3.3.2View Customer Information

User is enables to read the information of their customers by clicking on the "Customer Information" which located at the navigation bar.

Test Scenario ID	ViewCustomer-1	**Test Case ID**	ViewCustomer -1A
Test Case Description	ViewCustomer – Positive test case	**Test Priority**	High
Pre-Requisite	- Logged in with a valid user account - System already have customers' information in the database	**Post-Requisite**	N/A

Test Execution Steps:

S.No	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Test Comments**
1	Click on the "Customer Information" button at the navigation tab.	Customers' information from the firestore.	Display a list of Customers' information to user.	List of Customers' information were displayed.	Pass	-

Test Scenario ID	ViewCustomer -1	**Test Case ID**	ViewCustomer -1B
Test Case Description	ViewCustomer – Negative test case	**Test Priority**	High
Pre-Requisite	- Logged in with a valid user account - System does not have any customers' information in the database.	**Post-Requisite**	N/A

Test Execution Steps:

S.No	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Test Comments**
1	Click on the "Customer Information" button at the navigation tab.	empty	Display message "No Records Can Be Found. Please create your customer by clicking on the + icon."	"No Records Can Be Found. Please create your customer by clicking on the + icon."	Pass	-

6.1.3.3.3Edit Customer Information

User is enables to edit the information of a particular customer by clicking on the “Pencil” icon which located at the “Customer Information” screen.

Test Scenario ID	EditCustomer-1			**Test Case ID**	EditCustomer -1A	
Test Case Description	EditCustomer – Positive test case			**Test Priority**	High	
Pre-Requisite	- Logged in with a valid user account - System already have customers’ information in the database			**Post-Requisite**	N/A	
Test Execution Steps:						
S.No	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Test Comments**
1	1. Click on the “eye” icon which located at the “Action” column of each row of item. 2. Modify the data. For	- Customers’ information from the firestore. - Shop Name: Giant	Display message “Customer content has been successfully save!”	“Customer content has been successfully save!”	Pass	-

	example, user wish to change the Shop Name of the particular customer's information to Giant. 3. Click on the "Save" button.					

Test Scenario ID	EditCustomer -1	**Test Case ID**	EditCustomer -1B
Test Case Description	EditCustomer – Negative test case	**Test Priority**	High
Pre-Requisite	- Logged in with a valid user account - System already have customers' information in the database	**Post-Requisite**	N/A
Test Execution Steps:			

S.No	Action	Inputs	Expected Output	Actual Output	Test Result	Test Comments
1	1. Click on the "eye" icon which located at the "Action" column of each row of customer's record. 2. Modify the data. For example, user wish to change the address of a particular customer's record.	- Customers' information from the firestore. - Address: ""	Display error message *"Address is require"*	*"Address is require"*	Pass	-

	3. Leave the address field empty. 4. Click on the "Save" button.					

6.1.3.3.4Delete Customer Information

User is allows to delete the information of a particular customer by clicking on the "Trash" icon which located at the "Customer Information" screen.

Test Scenario ID	DeleteCustomer-1	**Test Case ID**	DeleteCustomer -1A
Test Case Description	DeleteCustomer – Positive test case	**Test Priority**	High
Pre-Requisite	- Logged in with a valid user account - System already have customers' information in the database	**Post-Requisite**	N/A
Test Execution Steps:			

S.No	Action	Inputs	Expected Output	Actual Output	Test Result	Test Comments
1	1. Click on the "trash" icon which located at the "Action" column of each row of cargo space. 2. System prompt user to confirm the action. 3. Click on the "Confirm" button.	- Customers' information from the firestore.	Display message "Customer content has been successfully removed!"	"Customer content has been successfully removed!"	Pass	-

6.1.3.4 Manage Cargo Space Information

6.1.3.4.1 Create Cargo Space Information

User is allows to add their own cargo space information into the system by clicking on the "+" icon which located at the "Cargoes Space" screen.

Test Scenario ID	CreateCargoSpace-1	**Test Case ID**	CreateCargoSpace -1A
Test Case Description	CreateCargoSpace – Positive test case	**Test Priority**	High
Pre-Requisite	Logged in with a valid user account	**Post-Requisite**	N/A

Test Execution Steps:

S.No	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Test Comments**
1	1. Click on the "+" icon which located beside the "Add Cargo Space" header.	Cargo Id: Cg0001 Cargo Type: PME 9235 Inner Length:	Display message "Cargo Space content has been successfully created!"	"Cargo Space content has been successfully created!"	Pass	-

	2. Enter all the require input into the “Cargo Space Detail” form. 3. Click on the “Save” button.	1290 Inner Width: 300 Inner Height: 350 Max Load: 1920				

Test Scenario ID	CreateCargoSpace -1		**Test Case ID**	CreateCargoSpace -1B	
Test Case Description	CreateCargoSpace – Negative test case		**Test Priority**	High	
Pre-Requisite	Logged in with a valid user account		**Post-Requisite**	N/A	
Test Execution Steps:					
S.No	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Test Comments**

1	1. Some of the input field in the Cargo Space Detail form remain empty. For example: Cargo Type is empty 2. Click on the "Save" button.	Cargo Id: Cg0001 Cargo Type: "" Inner Length: 1290 Inner Width: 300 Inner Height: 350 Max Load: 1920	Display error message "*Cargo Type is require*"	"*Cargo Type is require*"	Pass	-

2	1. Enter the Cargo Id which already exist in the system. 2. Click on the "Save" button.	Cargo Id: Cg0001 Cargo Type: PME 9235 Inner Length: 1290 Inner Width: 300 Inner Height: 350 Max Load: 1920	Display error message "Cg0001 already exist! Please change the Cargo Id."	"Cg0001 already exist! Please change the Cargo Id."	Pass	-

6.1.3.4.2View Cargo Space Information

User is enables to read the information of the cargo space by clicking on the "Cargoes Space" which located at the navigation bar.

Test Scenario ID	ViewCargoesSpace-1	**Test Case ID**	ViewCargoesSpace -1A
Test Case Description	ViewCargoesSpace – Positive test case	**Test Priority**	High
Pre-Requisite	- Logged in with a valid user account - System already have cargoes space' information in the database	**Post-Requisite**	N/A

Test Execution Steps:

S.No	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Test Comments**
1	Click on the "Cargoes Space" button at the navigation tab.	Cargoes Space' information from the firestore.	Display a list of Cargoes Space' information to user.	List of Cargoes Space' information were displayed.	Pass	-

Test Scenario ID	ViewCargoesSpace -1	**Test Case ID**	ViewCargoesSpace -1B
Test Case Description	ViewCargoesSpace – Negative test case	**Test Priority**	High
Pre-Requisite	- Logged in with a valid user account - System does not have any cargoes' information in the database.	**Post-Requisite**	N/A

Test Execution Steps:

S.No	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Test Comments**
1	Click on the "Cargoes Space" button at the navigation tab.	empty	Display message "No Records Can Be Found. Please create your own cargo space by clicking on the + icon."	"No Records Can Be Found. Please create your own cargo space by clicking on the + icon."	Pass	-

6.1.3.4.3Edit Cargo Space Information

User is enables to edit the information of a particular cargo Space by clicking on the “Pencil” icon which located at the “Cargoes Space” screen.

Test Scenario ID	EditCargoSpace-1	**Test Case ID**	EditCargoSpace -1A
Test Case Description	EditCargoSpace – Positive test case	**Test Priority**	High
Pre-Requisite	- Logged in with a valid user account - System already have cargoes space’ information in the database	**Post-Requisite**	N/A

Test Execution Steps:

S.No	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Test Comments**
1	1. Click on the “eye” icon which located at the “Action” column of each row of cargo space.	- Cargoes Space’ information from the firestore. - Inner Length: 2000	Display message “Cargo space content has been successfully save!”	“Cargo space content has been successfully save!”	Pass	-

	2. Modify the data. For example, user wish to change the Inner Length of the particular cargo space's record to 2000. 3. Click on the "Save" button.					

Test Scenario ID	EditCargoSpace -1	**Test Case ID**	EditCargoSpace -1B
Test Case Description	EditCargoSpace – Negative test case	**Test Priority**	High
Pre-Requisite	- Logged in with a valid user account - System already have cargoes space' information in the database	**Post-Requisite**	N/A

Test Execution Steps:

S.No	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Test Comments**
1	1. Click on the "eye" icon which located at the "Action" column of each row of the cargo space's record. 2. Modify the data. For example, user	- Cargoes space' information from the firestore. - Cargo Type: ""	Display error message "Cargo Type is require"	"Cargo Type is require"	Pass	-

	wish to change the Cargo Type of a particular cargo space's record. 3. Leave the Cargo Type empty. 4. Click on the "Save" button.					

6.1.3.4.4Delete Cargo Space Information

User is allows to delete the information of a particular cargo space by clicking on the "Trash" icon which located at the "Cargoes Space" screen.

Test Scenario ID	DeleteCargoSpace-1	**Test Case ID**	DeleteCargoSpace -1A
Test Case Description	DeleteCargoSpace – Positive test case	**Test Priority**	High
Pre-Requisite	- Logged in with a valid user account - System already have cargoes space' information in the database	**Post-Requisite**	N/A

Test Execution Steps:

S.No	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Test Comments**
1	1. Click on the "trash" icon which located at the "Action" column of each row of cargo space.	- Cargoes space' information from the firestore.	Display message "Cargo space content has been successfully removed!"	"Cargo space content has been successfully removed!"	Pass	-

	2. System prompt user to confirm the action. 3. Click on the "Confirm" button.					

6.1.3.5 Import Data with Excel format (.csv)

In order to provide flexibility and convenient to user, this system enable user to import their data in the Excel format instead of key in those data one-by-one.

6.1.3.5.1 Import D/O data with Excel format (.csv)

Test Scenario ID	ImportD/O-1	**Test Case ID**	ImportD/O -1A
Test Case Description	ImportD/O – Positive test case	**Test Priority**	High
Pre-Requisite	Logged in with a valid user account	**Post-Requisite**	N/A

Test Execution Steps:						
S.No	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Test Comments**
1	1. Click on the "Import Data" button which located at navigation bar. 2. Click on the "Browse" button which located under D/O Detail to upload your excel file. 3. Click on the "Save" button.	Excel file with D/O data filled.	Display message "All D/O content has been successfully created!"	"All D/O content has been successfully created!"	Pass	-

Test Scenario ID	ImportD/O-1	**Test Case ID**	ImportD/O -1B
Test Case Description	ImportD/O – Negative test case	**Test Priority**	High
Pre-Requisite	Logged in with a valid user account	**Post-Requisite**	N/A

Test Execution Steps:

S.No	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Test Comments**
1	1. Click on the "Import Data" button which located at navigation bar. 2. Click on the "Browse" button which located under D/O Detail to upload your excel file.	Excel file with D/O data filled but some part are missing.	Display error message "*There are some part missing in the excel file. Please check again.*"	"*There are some part missing in the excel file. Please check again.*"	Pass	-

	3. Click on the "Save" button.					
2	1. Click on the "Import Data" button which located at navigation bar. 2. Click on the "Browse" button which located under D/O Detail to upload your excel file. 3. Click on the "Save" button	Excel file with D/O data filled but some part of the data are filled with the wrong type.	Display error message "*Invalid data type. Please check again.*"	"*Invalid data type. Please check again.*"	Pass	-

2	1. Click on the "Import Data" button which located at navigation bar. 2. Click on the "Browse" button which located under D/O Detail to upload your excel file. 3. Click on the "Save" button	Import the wrong excel file.	Display error message "*The excel file you import is wrong. Please check again.*"	"*The excel file you import is wrong. Please check again.*"	Pass	-

6.1.3.5.2 Import Item data with Excel format (.csv)

Test Scenario ID	ImportItem-1	**Test Case ID**	ImportItem -1A
Test Case Description	ImportItem – Positive test case	**Test Priority**	High
Pre-Requisite	Logged in with a valid user account	**Post-Requisite**	N/A

Test Execution Steps:

S.No	Action	Inputs	Expected Output	Actual Output	Test Result	Test Comments
1	1. Click on the "Import Data" button which located at navigation bar. 2. Click on the "Browse" button which located under Item Detail	Excel file with Item data filled.	Display message "All Item content has been successfully created!"	"All Item content has been successfully created!"	Pass	-

	to upload your excel file. 3. Click on the "Save" button.					

Test Scenario ID	ImportItem-1	**Test Case ID**	ImportItem -1B
Test Case Description	ImportItem – Negative test case	**Test Priority**	High
Pre-Requisite	Logged in with a valid user account	**Post-Requisite**	N/A

Test Execution Steps:

S.No	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Test Comments**
1	1. Click on the "Import Data" button which	Excel file with Item data filled but some part are missing.	Display error message "*There are some part missing in the excel*	"*There are some part missing in the excel file. Please check again.*"	Pass	-

	located at navigation bar. 2. Click on the "Browse" button which located under Item Detail to upload your excel file. 3. Click on the "Save" button.		*file. Please check again.*"			
2	1. Click on the "Import Data" button which located at navigation bar.	Excel file with Item data filled but some part of the data are filled with the wrong type.	Display error message "*Invalid data type. Please check again.*"	"*Invalid data type. Please check again.*"	Pass	

	2. Click on the "Browse" button which located under Item Detail to upload your excel file. 3. Click on the "Save" button					
2	1. Click on the "Import Data" button which located at navigation bar. 2. Click on the "Browse" button which located	Import the wrong excel file.	Display error message "*The excel file you import is wrong. Please check again.*"	*"The excel file you import is wrong. Please check again."*	Pass	-

	under Item Detail to upload your excel file. 3. Click on the "Save" button					

6.1.3.5.3Import Customer data with Excel format (.csv)

Test Scenario ID	ImportCustomer-1		**Test Case ID**	ImportCustomer -1A	
Test Case Description	ImportCustomer – Positive test case		**Test Priority**	High	
Pre-Requisite	Logged in with a valid user account		**Post-Requisite**	N/A	
Test Execution Steps:					
S.No	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Test Comments**
1	1. Click on the "Import Data"	Excel file with Customer data filled.	Display message "All Customer content has	"All Customer content has been	Pass	-

	button which located at navigation bar. 2. Click on the “Browse” button which located under Customer Detail to upload your excel file. 3. Click on the “Save” button.		been successfully created!”	successfully created!”		

Test Scenario ID	ImportCustomer -1	**Test Case ID**	ImportCustomer -1B
Test Case Description	ImportCustomer – Negative test case	**Test Priority**	High
Pre-Requisite	Logged in with a valid user account	**Post-Requisite**	N/A

Test Execution Steps:

S.No	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Test Comments**
1	1. Click on the "Import Data" button which located at navigation bar. 2. Click on the "Browse" button which located under Customer Detail to upload your excel file.	Excel file with Customer data filled but some part are missing.	Display error message "*There are some part missing in the excel file. Please check again.*"	"*There are some part missing in the excel file. Please check again.*"	Pass	-

	3. Click on the "Save" button.					
2	1. Click on the "Import Data" button which located at navigation bar. 2. Click on the "Browse" button which located under Customer Detail to upload your excel file. 3. Click on the "Save" button	Excel file with Customer data filled but some part of the data are filled with the wrong type.	Display error message "*Invalid data type. Please check again.*"	"*Invalid data type. Please check again.*"	Pass	-

2	1. Click on the "Import Data" button which located at navigation bar. 2. Click on the "Browse" button which located under Customer Detail to upload your excel file. 3. Click on the "Save" button	Import the wrong excel file.	Display error message "*The excel file you import is wrong. Please check again.*"	"*The excel file you import is wrong. Please check again.*"	Pass	-

6.1.3.6 Configure the route setting

Before performing calculation, user must first configure the route setting else they are not allow to perform the calculation.

Test Scenario ID	ConfigureRoute-1	**Test Case ID**	ConfigureRoute -1A
Test Case Description	ConfigureRoute – Positive test case	**Test Priority**	High
Pre-Requisite	Logged in with a valid user account	**Post-Requisite**	N/A

Test Execution Steps:

S.No	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Test Comments**
1	1. Click on the "Home" button which located at navigation bar. 2. Click on the "Settings" button which located at the Home screen.	Starting Address: 22, Jalan SL 1/2, Bandar Sungai Long, 43000 Kajang, Selangor Departure Time: 9:05 AM	Display message "Route Settings has been successfully save!"	"Route Settings has been successfully save!"	Pass	-

	3. Filled in all the required data. 4. Click on the "Save" button.	Service Time: 30 Lunch Break: 60 Working Hours: 5 Respect Truck Weight: Optional Selected Vehicles: optional				

Test Scenario ID	ConfigureRoute -1	**Test Case ID**	ConfigureRoute -1B
Test Case Description	ConfigureRoute – Negative test case	**Test Priority**	High
Pre-Requisite	Logged in with a valid user account	**Post-Requisite**	N/A

Test Execution Steps:

S.No	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Test Comments**
1	1. Click on the "Home" button which located at navigation bar. 2. Click on the "Settings" button which located at the Home screen. 3. Some of the required input	Starting Address: "" Departure Time: 9:05 AM Service Time: 30 Lunch Break: 60 Working Hours:	Display error message "*Starting Address is require*"	"*Starting Address is require*"	Pass	-

	field remain empty For example: Starting Address is empty 4. Click on the "Save" button.	5 Respect Truck Weight: Optional Selected Vehicles: optional				

6.1.3.7 Display the D/O information which will be involved in later route calculation.

User is able to view all the D/Os' record which status are "Pending" on the "Smart D/O" tab which located under "Home" screen.

Test Scenario ID	ViewSelectedD/O-1	**Test Case ID**	ViewSelectedD/O-1A
Test Case Description	ViewSelectedD/O – Positive test case	**Test Priority**	High
Pre-Requisite	- Logged in with a valid user account	**Post-Requisite**	N/A

	- System already have D/Os information in the database					
Test Execution Steps:						
S.No	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Test Comments**
1	1. Click on the "Home" button at the navigation tab. 2. Click on the "Smart D/O" tab.	D/Os' information from the firestore.	Display a list of D/O record which status are "Pending".	List of D/O which status are "Pending" are displayed.	Pass	-
2	1. Click on the "Home" button at the navigation tab. 2. Click on the "Smart D/O" tab.	D/Os' information from the firestore.	Display a list of D/O record which status are "Pending" and the list of expire date of the D/O record displayed does not exist the date selected.	List of D/O which has been filter are displayed.	Pass	-

	3. Select a date at the top right corner.					
3	1. Click on the "Home" button at the navigation tab. 2. Click on the "Smart D/O" tab. 3. Click the "x" icon which located beside each D/O No.	D/Os' information from the firestore.	Display a list of D/O record which status are "Pending" without those D/O that already been removed.	D/O which has been removed is not exist in the screen.	Pass	-

Test Scenario ID	ViewSelectedD/O-1	**Test Case ID**	ViewSelectedD/O-1B
Test Case Description	ViewSelectedD/O – Negative test case	**Test Priority**	High
Pre-Requisite	- Logged in with a valid user account - System does not have any D/Os information in the database. Or All D/Os' had been delivered. Or No Pending's D/O between the particular date ranges that select by user.	**Post-Requisite**	N/A

Test Execution Steps:

S.No	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Test Comments**
1	1. Click on the "Home" button at the navigation tab.	empty	Display message "No Records Found."	"No Records Found."	Pass	-

	2. Click on the "Smart D/O" tab.					

6.1.3.8 Calculate and display suggested route

User is able to perform the calculation on the route to deliver after all information required had been full filled.

Test Scenario ID	CalculateRoute-1	**Test Case ID**	CalculateRoute -1A
Test Case Description	CalculateRoute – Positive test case	**Test Priority**	High
Pre-Requisite	- Logged in with a valid user account - System must have at least one D/O record which status is "Pending".	**Post-Requisite**	N/A

Test Execution Steps:

S.No	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Test Comments**
1	1. Click on the "Home" button at the navigation tab.	D/Os' information which status are "Pending".	Display the result of the calculation which include:	The calculated route has been displayed.	Pass	-

	2. Click on the "Smart D/O" tab. 3. Click on the "Start Calculation" button which located at the bottom right corner.		number of address which had been delivered, total distance travelled, total duration travelled, total volume of inventories involved, total mass of inventories involved, depart time, finish time, volume occupied, mass occupied, D/O which involved in that particular truck and also the direction to deliver.			

Test Scenario ID	CalculateRoute -1	**Test Case ID**	CalculateRoute -1B
Test Case Description	CalculateRoute – Negative test case	**Test Priority**	High
Pre-Requisite	- Logged in with a valid user account - System does not have any D/Os information in the database. Or All D/Os' had been delivered. Or No Pending's D/O between the particular date ranges that select by user.	**Post-Requisite**	N/A

Test Execution Steps:

S.No	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Test Comments**
1	1. Click on the "Home" button at the navigation tab.	empty	Display alert message "No D/O Records are Found."	"No D/O Records are Found."	Pass	-

	2. Click on the "Smart D/O" tab. 3. Click on the "Start Calculation" button.					

Test Scenario ID	CalculateRoute -1	**Test Case ID**	CalculateRoute -1C
Test Case Description	CalculateRoute – Negative test case	**Test Priority**	High
Pre-Requisite	- Logged in with a valid user account - System must have at least one D/O record which status is "Pending". - There are no vehicle selected at the "Vehicle Selection" section which located under the "Settings" tab.	**Post-Requisite**	N/A
Test Execution Steps:			

S.No	Action	Inputs	Expected Output	Actual Output	Test Result	Test Comments
1	1. Click on the "Home" button at the navigation tab. 2. Click on the "Smart D/O" tab. 3. Click on the "Start Calculation" button.	D/Os' information which status are "Pending".	Display alert message "There are no trucks found. Please select your truck at 'Settings' tab."	"There are no trucks found. Please select your truck at 'Settings' tab."	Pass	-

6.1.3.9 Visualization of the calculated route on the map

After the route had been calculated, system must be able to visualize the route with the sequence to deliver on the map.

Test Scenario ID	VisualizeRoute-1	**Test Case ID**	VisualizeRoute -1A
Test Case Description	VisualizeRoute – Positive test case	**Test Priority**	High
Pre-Requisite	- Logged in with a valid user account - User successfully perform the route calculation process.	**Post-Requisite**	N/A

Test Execution Steps:

S.No	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Test Comments**
1	1. Click on the "Home" button at the navigation tab. 2. Click on the "Smart D/O" tab.	The data of system suggested route after going through the route optimizer algorithm.	Display suggested route with the sequence to deliver on the map.	Suggested route to deliver was displayed.	Pass	-

	3. Click on the "Start Calculation" button which located at the bottom right corner.					

6.1.3.10 Manage Previous Planned Route Record

6.1.3.10.1 Create Previous Planned Route Record

Test Scenario ID	CreateRoute-1	**Test Case ID**	CreateRoute -1A
Test Case Description	CreateRoute – Positive test case	**Test Priority**	High
Pre-Requisite	- Logged in with a valid user account - User successfully perform the route calculation process.	**Post-Requisite**	N/A
Test Execution Steps:			

S.No	Action	Inputs	Expected Output	Actual Output	Test Result	Test Comments
1	1. Click on the "Home" button at the navigation tab. 2. Click on the "Smart D/O" tab. 3. Click on the "Start Calculation" button which located at the bottom right corner. 4.Click on the "Save" button	The optimize route's result after going through route optimizer algorithm.	The record is able to view on the "Route Record" screen.	Route record has been save.	Pass	-

	which located at the bottom right corner after suggested result has been displayed on the "Smart D/O" tab.					

6.1.3.10.2 View Previous Planned Route Record

Test Scenario ID	ViewRoute-1		**Test Case ID**	ViewRoute -1A	
Test Case Description	ViewRoute – Positive test case		**Test Priority**	High	
Pre-Requisite	- Logged in with a valid user account - System already have route information in the database		**Post-Requisite**	N/A	
Test Execution Steps:					
S.No	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Test Comments**

1	Click on the "Route Record" button at the navigation tab.	Routes' information from the firestore.	Display a list of Routes' information to user.	List of Routes' information were displayed.	Pass	-
2	Click on the "eye" icon which located at the 'Action" column of each row of route record.	Routes' information from the firestore.	Display the particular route information in PDF format.	Detail of a particular route information is displayed on the PDF.	Pass	-

Test Scenario ID	ViewRoute -1	**Test Case ID**	ViewRoute -1B
Test Case Description	ViewRoute – Negative test case	**Test Priority**	High
Pre-Requisite	- Logged in with a valid user account - System does not have any routes' information in the database.	**Post-Requisite**	N/A
Test Execution Steps:			

S.No	Action	Inputs	Expected Output	Actual Output	Test Result	Test Comments
1	Click on the "Route Record" button at the navigation tab.	empty	Display message "No Records Can Be Found."	"No Records Can Be Found."	Pass	-

6.1.3.10.3 **Delete Previous Planned Route Record**

Test Scenario ID	DeleteRoute -1			Test Case ID	DeleteRoute -1A	
Test Case Description	DeleteRoute – Positive test case			Test Priority	High	
Pre-Requisite	- Logged in with a valid user account - System already have cargoes space' information in the database			Post-Requisite	N/A	
Test Execution Steps:						
S.No	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Test Comments**

1	1. Click on the "trash" icon which located at the "Action" column of each row of route record. 2. System prompt user to confirm the action. 3. Click on the "Confirmed" button.	- Routes' information from the firestore.	Display message "Route content has been successfully removed!"	"Route content has been successfully removed!"	Pass	-

6.1.4 Unit Test Cases (Company Admin)

6.1.4.1 Manage Staff Information

6.1.4.1.1 Add and enable staff to access the system.

Test Scenario ID	CreateStaff-1	**Test Case ID**	CreateStaff -1A
Test Case Description	CreateStaff – Positive test case	**Test Priority**	High
Pre-Requisite	Logged in with company admin account	**Post-Requisite**	N/A

Test Execution Steps:

S.No	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Test Comments**
1	1. Click on the "Add People" icon which located beside the "Staff Detail" header.	Staff Id: St001 Staff Name: Tan Chee Kuan Staff Email:	Display message "The particular account had been successfully created!"	"The particular account had been successfully created!"	Pass	-

	2. Enter all the require input into the "Staff Detail" form. 3. Click on the "Create" button.	cheekuantan@1utar.my password: ******				

Test Scenario ID	CreateStaff -1	**Test Case ID**	CreateStaff -1B
Test Case Description	CreateStaff – Negative test case	**Test Priority**	High
Pre-Requisite	Logged in with company admin account	**Post-Requisite**	N/A

Test Execution Steps:

S.No	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Test Comments**
1	1. Enter an invalid email	Staff Id: St001	Display error message "*Invalid email address*."	"*Invalid email address*."	Pass	-

	address with all the require input. 2. Click on the "Create" button.	Staff Name: Tan Chee Kuan Staff Email: cheekuantan^email.com password: ******				
2	1. Enter a valid email address which already exist in the system with all the require input. 2. Click on the "Create" button.	Staff Id: St001 Staff Name: Tan Chee Kuan Staff Email: cheekuantan@1utar.my password:	Display error message "*The email address is already in use by another account.*"	"*The email address is already in use by another account.*"	Pass	-

		******				
3	1. Enter a valid email address with some of the input field remain empty. For example: StaffName is empty. 2. Click on the "Create" button.	Staff Id: St001 Staff Name: "" Staff Email: cheekuantan@1utar.my password: ******	Display error message "*StaffName is require.*"	"*StaffName is require.*"	Pass	-

6.1.4.1.2 View Staffs' Information

Company Admin is enables to read the information of the staff by clicking on the "Staff" which located at the navigation bar.

Test Scenario ID	ViewStaff-1	**Test Case ID**	ViewStaff -1A
Test Case Description	ViewStaff – Positive test case	**Test Priority**	High
Pre-Requisite	- Logged in with company admin account - System already have staffs' information in the database	**Post-Requisite**	N/A

Test Execution Steps:

S.No	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Test Comments**
1	Click on the "Staff" button at the navigation tab.	Staffs' information from the firestore.	Display a list of Staffs' information to user.	List of Staffs' information were displayed.	Pass	-

Test Scenario ID	ViewStaff -1	**Test Case ID**	ViewStaff -1B
Test Case Description	ViewStaff – Negative test case	**Test Priority**	High
Pre-Requisite	- Logged in with company admin account - System does not have any staffs' information in the database	**Post-Requisite**	N/A

Test Execution Steps:

S.No	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Test Comments**
1	Click on the "Staff" button at the navigation tab.	empty	Display message "No Records Can Be Found."	"No Records Can Be Found."	Pass	-

6.1.4.1.3 Change staff status

Company Admin is enables to change the status of the staff by clicking on the “Tick” icon or “Cross” icon which located at the status column of each staff record.

Test Scenario ID	ChangeStaffStatus-1	**Test Case ID**	ChangeStaffStatus -1A
Test Case Description	ChangeStaffStatus – Positive test case	**Test Priority**	High
Pre-Requisite	- Logged in with company admin account - System already have staffs’ information in the database	**Post-Requisite**	N/A

Test Execution Steps:

S.No	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Test Comments**
1	1. Click on the “Tick” icon or “Cross” icon which located at the status column of each staff’s record.	new value of the status (true/false)	Staff detail with status updated.	Staff status has updated.	Pass	-

6.2 Integration Testing

Integration testing will only be performed when all the unit testing had been done. Integration testing is performed by combining related units as a group. The purpose of integration testing is to make sure that there is no bugs or any default when two or more unit integrated with each other.

6.2.1 Integration Test Cases (for Logistic Company's staff and company Admin)

6.2.1.1 Manage D/O Information

Functionalities Involved	Test Case	Test Execution Steps	Expected Output	Actual Output	Status
- Create D/O Record - View D/O Information	User create new D/O record and the D/O record must be appear to the user.	1. User click on the "+" icon which located on the "D/O Information" page OR user go to the "Add D/O" tab which located at the "Home" Page. 2. User filled in all the required data.	1. Display Message 'D/O content has been successfully save!". 2. New D/O record had been successfully created and save into the firestore.	Created D/O record is shown in the D/O Information page.	Pass

		3. User clicked on the "Save" button. 4. User then search for the particular data that had been created. 5. User then click on the "eye" icon to view the detail of the created D/O record.	3. The created D/O record is shown in the D/O information page.		

Functionalities Involved	**Test Case**	**Test Execution Steps**	**Expected Output**	**Actual Output**	**Status**
- Update D/O Information	User edit the existing D/O record.	1. User click on the "D/O Information" button which located at the navigation bar.	1. Display message "D/O content has been successfully save!".	Updated D/O record with the changed value is shown in the D/O Information page.	Pass

- View D/O Information		2. User search for the particular D/O record he wish to modify. 3. User click on the “pencil” icon which located at the Action column of that particular record. 4. User then changes the desire data and click on the “Save” button.	2. The particular D/O record had been successfully update in the firestore. 3. The updated D/O record with the changed value is shown in the D/O information page.		

Functionalities Involved	**Test Case**	**Test Execution Steps**	**Expected Output**	**Actual Output**	**Status**
- Delete D/O Information - View D/O Information	User delete the existing D/O record.	1. User click on the "D/O Information" button which located at the navigation bar. 2. User search for the particular D/O record he wish to delete. 3. User click on the "trash" icon which located at the Action column of that particular record. 4. User then click on the "Confirm" button which	1. Display message "D/O content has been successfully removed!". 2. The particular D/O record had been successfully deleted from the firestore.	Deleted D/O record is not shown in the D/O Information page.	Pass

		located at the pop up confirmation dialog.			

6.2.1.2 Manage Item Information

Functionalities Involved	Test Case	Test Execution Steps	Expected Output	Actual Output	Status
- Create Item Record - View Item Information	User create new Item record and the Item record must be appear to the user.	1. User click on the "+" icon which located on the "Item Information" page. 2. User filled in all the required data. 3. User clicked on the "Save" button. 4. User then search for the particular data that had been created.	1. Display Message 'Item content has been successfully save!". 2. New Item record had been successfully created and save into the firestore. 3. The created Item record is shown in the Item information page.	Created Item record is shown in the Item Information page.	Pass

Functionalities Involved	**Test Case**	**Test Execution Steps**	**Expected Output**	**Actual Output**	**Status**
- Update Item Information - View Item Information	User edit the existing Item record.	1. User click on the "Item Information" button which located at the navigation bar. 2. User search for the particular Item record he wish to modify. 3. User click on the "pencil" icon which located at the Action column of that particular record. 4. User then changes the desire data and click on the "Save" button.	1. Display message "Item content has been successfully save!". 2. The particular Item record had been successfully update in the firestore. 3. The updated Item record with the changed value is shown in the Item information page.	Updated Item record with the changed value is shown in the Item Information page.	Pass

Functionalities Involved	**Test Case**	**Test Execution Steps**	**Expected Output**	**Actual Output**	**Status**
- Delete Item Information - View Item Information	User delete the existing Item record.	1. User click on the "Item Information" button which located at the navigation bar. 2. User search for the particular Item record he wish to delete. 3. User click on the "trash" icon which located at the Action column of that particular record. 4. User then click on the "Confirm" button which	1. Display message "Item content has been successfully removed!". 2. The particular Item record had been successfully deleted from the firestore.	Deleted Item record is not shown in the Item Information page.	Pass

		located at the pop up confirmation dialog.			

6.2.1.3 Manage Customer Information

Functionalities Involved	Test Case	Test Execution Steps	Expected Output	Actual Output	Status
- Create Customer Record - View Customer Information	User create new Customer record and the Customer record must be appear to the user.	1. User click on the "+" icon which located on the "Customer Information" page. 2. User filled in all the required data. 3. User clicked on the "Save" button.	1. Display Message 'Customer content has been successfully save!". 2. New Customer record had been successfully created and save into the firestore. 3. The created customer record is shown in the	Created customer record is shown in the Customer Information page.	Pass

		4. User then search for the particular data that had been created.	Customer information page.		

Functionalities Involved	**Test Case**	**Test Execution Steps**	**Expected Output**	**Actual Output**	**Status**
- Update Customer Information - View Customer Information	User edit the existing Customer record.	1. User click on the "Customer Information" button which located at the navigation bar. 2. User search for the particular Customer record he wish to modify. 3. User click on the "pencil" icon which located	1. Display message "Customer content has been successfully save!". 2. The particular Customer record had been successfully update in the firestore. 3. The updated Customer record with the changed value is shown in the	Updated Customer record with the changed value is shown in the Customer information page.	Pass

		at the Action column of that particular record. 4. User then changes the desire data and click on the "Save" button.	Customer information page.		

Functionalities Involved	**Test Case**	**Test Execution Steps**	**Expected Output**	**Actual Output**	**Status**
- Delete Customer Information - View Customer Information	User delete the existing Customer record.	1. User click on the "Customer Information" button which located at the navigation bar. 2. User search for the particular Customer record he wish to delete.	1. Display message "Customer content has been successfully removed!". 2. The particular Customer record had been successfully deleted from the firestore.	Deleted Customer record is not shown in the Customer Information page.	Pass

		3. User click on the “trash” icon which located at the Action column of that particular record. 4. User then click on the “Confirm” button which located at the pop up confirmation dialog.			

6.2.1.4 Manage Cargo Space Information

Functionalities Involved	Test Case	Test Execution Steps	Expected Output	Actual Output	Status
- Create Cargo Space Record	User create new Cargo Space record and the Cargo Space record must be	1. User click on the “+” icon which located on the “Cargoes Space” page.	1. Display Message ‘Cargo Space content has been successfully save!”.	Created cargo space record is shown in the Cargoes Space page.	Pass

- View Cargo Space Information	appear to the user.	2. User filled in all the required data. 3. User clicked on the "Save" button. 4. User then search for the particular data that had been created.	2. New Cargo Space record had been successfully created and save into the firestore. 3. The created Cargo record is shown in the Cargoes Space page.		

Functionalities Involved	**Test Case**	**Test Execution Steps**	**Expected Output**	**Actual Output**	**Status**
- Update Cargo Space Information	User edit the existing Cargo Space record.	1. User click on the "Cargoes Space" button which located at the navigation bar.	1. Display message "Cargo Space content has been successfully save!". 2. The particular Cargo Space record had been	Updated cargo space record with the changed value is shown in the Cargoes Space page.	Pass

- View Cargo Space Information		2. User search for the particular Cargo Space record he wish to modify. 3. User click on the "pencil" icon which located at the Action column of that particular record. 4. User then changes the desire data and click on the "Save" button.	successfully update in the firestore. 3. The updated Cargo record with the changed value is shown in the D/O information page.		

Functionalities Involved	**Test Case**	**Test Execution Steps**	**Expected Output**	**Actual Output**	**Status**
- Delete Cargo Space Information	User delete the existing Cargo space record.	1. User click on the "Cargoes Space" button	1. Display message "Cargo Space content has been successfully removed!".	Deleted cargo space record is not shown in the Cargoes Space page.	Pass

- View Cargo Space Information		which located at the navigation bar. 2. User search for the particular Cargo Space record he wish to delete. 3. User click on the “trash” icon which located at the Action column of that particular record. 4. User then click on the “Confirm” button which located at the pop up confirmation dialog.	2. The particular Cargo Space record had been successfully deleted from the firestore.		

6.2.1.5 Import Data

Functionalities Involved	Test Case	Test Execution Steps	Expected Output	Actual Output	Status
- Import Data with Excel format (.csv). - View D/O Information - View Customer Information - View Item Information	User wish to import a list of D/O/ Item/ Customer data	1. User click on the "Import Data" button which located at the navigation bar. 2. User click on the "Browse" button to upload the Excel file (D/O / Item / Customer). 3. User click on the "Save" button. 4. User click on "D/O Information" / "Item Information" / "Customer Information" button which	1. Display message "D/O / Item / Customer content had been save!". 2. New created records (D/O / Item / Customer) had been successfully save to the firestore. 3. Those created records are shown in their corresponding page.	Those created records are shown in their corresponding page.	Pass

		located at the navigation bar. 5. User then search for the created records.			

6.2.1.6 Route Calculation

Functionalities Involved	Test Case	Test Execution Steps	Expected Output	Actual Output	Status
- Configure the route setting - Display all the "Pending" D/O	User wish to calculate the route to deliver	1. User click on the "Settings" button which located at the "Home" page. 2. User filled up all the required information.	1. The optimized route result had been displayed on the "Smart D/O" screen. 2. The optimized route result had been visualized on the map with the	The created route record is shown in the Route Record Page.	Pass

- Calculate and display suggested route - Visualize the calculated route - Save system's planned route - View system's planned route		3. User select vehicles he wish to use for the calculation process. 4. User click on the "Save" button which located at the bottom right on the "Settings" screen. 5. User click on the "Smart D/O" button which located beside the "Settings" button. 6. User perform filter on the list of "Pending" D/O. 7. User then click on the "Start Calculation" button.	sequence of locations to deliver. 3. The calculated route record had been successfully store in the firestore. 4. The calculated route record is shown in the Route Record page.		

		8. User click on the “Save” button after the result had been shown by the system. 9. User click on the “Route Record” button which located at the navigation bar. 10. User search for the particular Route record. 11. User click on the “eye” icon to have a detail information of that particular Route record.			

6.2.1.7 Manage Route Record

Functionalities Involved	Test Case	Test Execution Steps	Expected Output	Actual Output	Status
- Delete Previous Route Record. - View Previous Route Information	User delete the existing Route record.	1. User click on the “Route Record” button which located at the navigation bar. 2. User search for the particular Route record he wish to delete. 3. User click on the “trash” icon. 4. User then click on the “Confirm” button which located at the pop up confirmation dialog.	1. Display message “Route content has been successfully removed!”. 2. The particular Route record had been successfully deleted from the firestore.	The particular route record is not shown in the Route Record screen.	Pass

6.2.2 Integration Test Cases (Company Admin)

6.2.2.1 Manage Staff Information

Functionalities Involved	Test Case	Test Execution Steps	Expected Output	Actual Output	Status
- Create/Add Staff - View Staff Information	Company Admin create/add an authenticated account for the staff to access the system with company information.	1. Company Admin click on the "Add people" icon which located on the "Staff" page. 2. Company Admin filled in all the required data. 3. Company Admin clicked on the "Create" button. 4. Company Admin then find for the particular data that had been created.	1. Display Message 'The particular account had been successfully created". 2. Staff has been created and treated as normal user which is allow to access the system and the staff detail had also stored in the firestore.	The particular staff is able to access the system using the created account and the staff record is displayed in the "Staff" page.	Pass

Functionalities Involved	**Test Case**	**Test Execution Steps**	**Expected Output**	**Actual Output**	**Status**
- Update the status of a particular staff. - View Staff Information	Company Admin change the status of the staff.	1. Company Admin click on the "Staff" button which located at the navigation bar. 2. Company Admin search for the particular Staff record he wish to activate or deactivate. 3. Company Admin click on the "Tick" icon or "Cross" icon which located at the Status column of that particular record.	1. The particular staff's status had been successfully update in the firestore.	The particular user is not able to login to the system.	Pass

6.3 Testing Results and Test Scripts

6.3.1 Firestore Mock Testing For Each CRUD Operation

```
beforeEach(() => {
  jest.resetModules();
  jest.clearAllMocks();
});

test('get all items', async () => {
  const firebase = require('firebase'); // or import firebase from 'firebase';
  const db = firebase.firestore();

  await db.collection('items').get().then((querySnapshot) => {
    expect(querySnapshot.forEach).toBeTruthy();
    expect(querySnapshot.docs.length).toBe(2);

    querySnapshot.forEach((doc) => {
      expect(doc.exists).toBe(true);
      expect(doc.data()).toBeTruthy();
    });
  })
});

test('get all customers', async () => {
  const firebase = require('firebase'); // or import firebase from 'firebase';
  const db = firebase.firestore();

  await db.collection('customers').get().then((querySnapshot) => {
    expect(querySnapshot.forEach).toBeTruthy();
    expect(querySnapshot.docs.length).toBe(2);

    querySnapshot.forEach((doc) => {
      expect(doc.exists).toBe(true);
      expect(doc.data()).toBeTruthy();
    });
  })
```

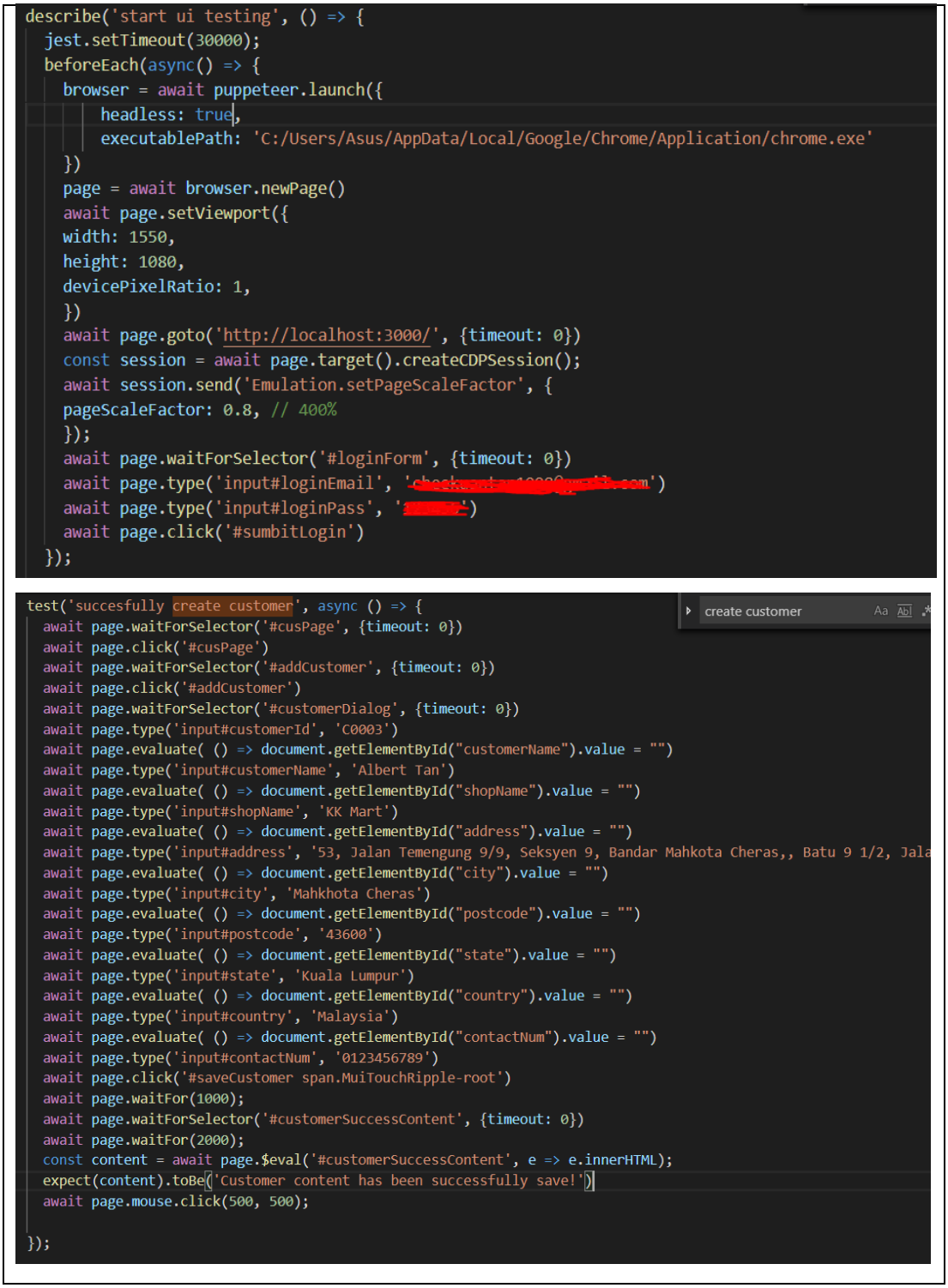
Figure 6.1 Several Sections of Code for Firestore Mock Testing

```
 PASS  src/App.test.js
  Firestore mock testing
    √ get all items (8ms)
    √ get all customers (2ms)
    √ get all cargoes (2ms)
    √ get all DOs (3ms)
    √ get only pending DO (2ms)
    √ create an item (2ms)
    √ create a new staff (2ms)
    √ create a new customer (2ms)
    √ create a new cargo (5ms)
    √ create a new D/O (2ms)
    √ updating an item (2ms)
    √ updating a customer (1ms)
    √ updating a cargo (1ms)
    √ updating a D/O (1ms)
    √ deleting a D/O (2ms)
    √ deleting an item (1ms)
    √ deleting a customer (1ms)
    √ deleting a cargo (1ms)

Test Suites: 1 passed, 1 total
Tests:       18 passed, 18 total
Snapshots:   0 total
Time:        3.62s
Ran all test suites matching /App.test.js/i.

Watch Usage: Press w to show more.
```

Figure 6.2 Results For Firestore Mock Testing

6.3.2 UI Testing On System's Logic

The testing tool that used to perform this UI testing is "Puppeteer". Puppeteer was made by the DevTools Team from the Google to help developer to automate the browser. When performing testing we can apply Puppeteer together with the Jest which is a Javascript testing framework to test on the logic of the system. By using this tool the unit testing for each function and also the integration testing can be performed. Puppeteer help to automate clicks on the button which will trigger a particular function, then the final output is then captured and compare with the expected output using the Jest function. Below are the examples of the code:



```
describe('start ui testing', () => {
  jest.setTimeout(30000);
  beforeEach(async() => {
    browser = await puppeteer.launch({
        headless: true,
        executablePath: 'C:/Users/Asus/AppData/Local/Google/Chrome/Application/chrome.exe'
    })
    page = await browser.newPage()
    await page.setViewport({
    width: 1550,
    height: 1080,
    devicePixelRatio: 1,
    })
    await page.goto('http://localhost:3000/', {timeout: 0})
    const session = await page.target().createCDPSession();
    await session.send('Emulation.setPageScaleFactor', {
    pageScaleFactor: 0.8, // 400%
    });
    await page.waitForSelector('#loginForm', {timeout: 0})
    await page.type('input#loginEmail', '[redacted]')
    await page.type('input#loginPass', '[redacted]')
    await page.click('#sumbitLogin')
  });
```

```
test('succesfully create customer', async () => {
  await page.waitForSelector('#cusPage', {timeout: 0})
  await page.click('#cusPage')
  await page.waitForSelector('#addCustomer', {timeout: 0})
  await page.click('#addCustomer')
  await page.waitForSelector('#customerDialog', {timeout: 0})
  await page.type('input#customerId', 'C0003')
  await page.evaluate( () => document.getElementById("customerName").value = "")
  await page.type('input#customerName', 'Albert Tan')
  await page.evaluate( () => document.getElementById("shopName").value = "")
  await page.type('input#shopName', 'KK Mart')
  await page.evaluate( () => document.getElementById("address").value = "")
  await page.type('input#address', '53, Jalan Temengung 9/9, Seksyen 9, Bandar Mahkota Cheras,, Batu 9 1/2, Jala
  await page.evaluate( () => document.getElementById("city").value = "")
  await page.type('input#city', 'Mahkhota Cheras')
  await page.evaluate( () => document.getElementById("postcode").value = "")
  await page.type('input#postcode', '43600')
  await page.evaluate( () => document.getElementById("state").value = "")
  await page.type('input#state', 'Kuala Lumpur')
  await page.evaluate( () => document.getElementById("country").value = "")
  await page.type('input#country', 'Malaysia')
  await page.evaluate( () => document.getElementById("contactNum").value = "")
  await page.type('input#contactNum', '0123456789')
  await page.click('#saveCustomer span.MuiTouchRipple-root')
  await page.waitFor(1000);
  await page.waitForSelector('#customerSuccessContent', {timeout: 0})
  await page.waitFor(2000);
  const content = await page.$eval('#customerSuccessContent', e => e.innerHTML);
  expect(content).toBe('Customer content has been successfully save!')
  await page.mouse.click(500, 500);

});
```

create customer

Figure 6.3 Several Sections of Code for UI Testing

Figure 6.4 Results For UI Testing

6.4 User Acceptance Testing

An user acceptance testing was conducted with manager of Hondail Trading Sdn. Bhd. The result of the UAT can be viewed at APPENDIX F.

6.4.1 User Acceptance Test Cases (for Logistic Company's staff and Company Admin)

6.4.1.1 Manage D/O Information

Name of Tester(s)			
Date Tested			
Testing Start Time		**Testing End Time**	
Test Functionality	**Test Description**	**Status**	**Comments**
Manage D/O Information	Able to create a new D/O record.		
	Able to add items detail into the D/O record.		
	Able to display error when the field required to create the data is empty.		
	Able to notify user that the particular D/O already exist when user enter the same D/O No.		
	Able to notify user when the D/O record has been successfully created.		

	Able to view a list of D/O record.		
	Able to filter the list of D/O record.		
	Able to view the detail of a particular D/O record.		
	Able to edit a particular D/O record.		
	Able to notify the user when the D/O record successfully updated.		
	Able to delete a particular D/O record.		
	Able to show the confirmation message before delete the D/O record.		
	Able to notify user that the particular D/O record had been removed.		
Sign-off by: _____ **Name:** **Date:**			

6.4.1.2 Manage Item Information

Name of Tester(s)			
Date Tested			
Testing Start Time		**Testing End Time**	
Test Functionality	**Test Description**	**Status**	**Comments**
Manage Item Information	Able to create a new Item record.		
	Able to display error when the field required to create the Item is empty.		
	Able to notify user that the particular Item already exist when user enter the same Item Id.		
	Able to notify user when the Item record has been successfully created.		
	Able to view a list of Item record.		
	Able to filter the list of Item record.		
	Able to edit a particular Item record.		
	Able to notify the user when the Item record successfully updated.		
	Able to delete a particular Item record.		
	Able to show the confirmation message before delete the Item record.		

	Able to notify user that the particular Item record had been removed.		
Sign-off by: _____ **Name:** **Date:**			

6.4.1.3 Manage Customer Information

Name of Tester(s)				
Date Tested				
Testing Start Time		**Testing End Time**		
Test Functionality	**Test Description**		**Status**	**Comments**
Manage Customer Information	Able to create a new Customer record.			
	Able to display error when the field required to create the Customer is empty.			

	Able to notify user that the particular Customer record already exist when user enter the same Customer Id.		
	Able to notify user when the Customer record has been successfully created.		
	Able to view a list of Customer record.		
	Able to filter the list of Customer record.		
	Able to edit a particular Customer record.		
	Able to notify the user when the Customer record successfully updated.		
	Able to delete a particular Customer record.		
	Able to show the confirmation message before delete the Customer record.		
	Able to notify user that the particular Customer record had been removed.		
Sign-off by: _____ **Name:** **Date:**			

6.4.1.4 Manage Cargo Space Information

Name of Tester(s)			
Date Tested			
Testing Start Time		**Testing End Time**	
Test Functionality	**Test Description**	**Status**	**Comments**
Manage Cargo Space Information	Able to create a new Cargo Space record.		
	Able to display error when the field required to create the Cargo Space is empty.		
	Able to notify user that the particular Cargo Space record already exist when user enter the same Cargo Id.		
	Able to notify user when the Cargo Space record has been successfully created.		
	Able to view a list of Cargo Space record.		
	Able to filter the list of Cargo Space record.		
	Able to edit a particular Cargo Space record.		

	Able to notify the user when the Cargo Space record successfully updated.		
	Able to delete a particular Cargo Space record.		
	Able to show the confirmation message before delete the Cargo Space record.		
	Able to notify user that the particular Cargo Space record had been removed.		
Sign-off by: **_____** **Name:** **Date:**			

6.4.1.5 Import Data with the Excel File (.csv)

Name of Tester(s)			
Date Tested			
Testing Start Time		**Testing End Time**	
Test Functionality	**Test Description**	**Status**	**Comments**
Import Data with the Excel File (.csv)	Able to create a list of Item records at one time.		
	Able to create a list of Customer records at one time.		
	Able to create a list of D/O records at one time.		
	Able to notify user when those records had been successfully created.		
	Able to display error when there are missing field in the excel file.		
	Able to display error when user import the wrong excel file.		
	Able to display error when the data type is invalid.		
Sign-off by: _____ **Name:** **Date:**			

6.4.1.6 Route Setting Process

Name of Tester(s)			
Date Tested			
Testing Start Time		**Testing End Time**	
Test Functionality	**Test Description**	**Status**	**Comments**
Route Setting Process	Able to display vehicle on the "Vehicle Selection" section after user change the status of the vehicle to become a tick icon which located at the "Cargoes Space" screen.		
	Able to select the vehicle for calculation purpose.		
	Able to display error when the field required for the Route Setting is empty.		
	Able to save the Route Settings data.		
	Able to notify user when the data has been successfully save.		
Sign-off by: _____ **Name:** **Date:**			

6.4.1.7 Route Calculation Process

Name of Tester(s)			
Date Tested			
Testing Start Time		**Testing End Time**	
Test Functionality	**Test Description**	**Status**	**Comments**
Route Calculation process	Able to display list of D/Os which status are still "Pending".		
	Able to filter the list of D/Os after a particular date had been selected.		
	Able to remove a particular D/O from the list of D/Os		
	Able to prohibit the calculation process and notify user that there are no "Pending" D/O record.		
	Able to prohibit the calculation process and notify user that there are no truck selected.		
	Able to combine all the D/O which have the same address.		
	Able to cluster those addresses into several groups which corresponding to the number of truck selected.		

	Able to calculate the shortest route to deliver for each cluster.		
	Able to display the result of the calculated route for each cluster on the “Smart D/O” tab.		
	Able to visualize the calculated route with the sequence to deliver on the map for each cluster.		
	Able to displayed those removed D/O		
Sign-off by: _____ **Name:** **Date:**			

6.4.1.8 Manage Previous Planned Route Record

Name of Tester(s)			
Date Tested			
Testing Start Time		**Testing End Time**	
Test Functionality	**Test Description**	**Status**	**Comments**
Manage Previous Planned Route	Able to save the calculated route record.		
	Able to view a list of previous calculated route record.		
	Able to filter the list of calculated route record after a particular date had been selected.		
	Able to view the detail of a particular previous calculated route record.		
	Able to delete a particular previous calculated route record.		
	Able to show the confirmation message before delete a particular previous calculated route record.		
	Able to notify user that the particular previous calculated route record had been removed.		
Sign-off by:			

_____ **Name:** **Date:**

6.4.2 User Acceptance Test Case (for Company Admin)

6.4.2.1 Manage Staff Information

Name of Tester(s)				
Date Tested				
Testing Start Time		**Testing End Time**		
Test Functionality	**Test Description**		**Status**	**Comments**
Manage Staff Information	Able to create a new Staff to access the system and also the company's information.			
	Able to display error when the field required to create the Staff is empty.			

	Able to notify user that the particular account for the staff already exist when user enter the same staff id.		
	Able to notify user when the staff's account has been successfully created.		
	Able to view a list of Staff record.		
	Able to filter the list of Staff record.		
	Able to change the status of a particular staff.		
Sign-off by: _____ **Name:** **Date:**			

CHAPTER 7
CONCLUSIONS AND RECOMMENDATIONS

This chapter will discuss on how well this Route Optimization System helps Logistic Company to solve their daily route planning issues. Besides, limitations of this system and also the recommendations to improve it will also be described under this chapter.

7.1 Conclusion

This system has been developed with a total period of six months, starting from the planning phase, analysis phase, design phase, implementation phase until the testing phase in the SDLC.

At the beginning of this project, an interview had been conducted to gather the information regarding the overall workflow of the route planning process and also the cargoes loading process. The result collected from the interviewee is then finalized and transformed to an algorithm. In this iteration of the SDLC, only route optimizer algorithm will be involved while the container loading algorithm will be implemented in the future.

After all required information had been collected, the development life cycle continues with the design phase. In this phase, several designs which consist of use case diagram, system architecture diagram, ERD diagram, data flow diagram and also the activity diagram had been generated to have a better understanding on the overall flow of the system. By referencing on those designs and requirements, a web-based route optimization system had been implemented.

The implemented system is to helps logistic company to arrange their route to deliver based on the number of truck/cargo space they owned. This system undergoes several steps to suggest the logistic company's staff an autonomous planned routes after the staff had key in all the required data. This system first cluster those addresses based on the distance between each location. The number of clusters that will be formed is depending on the number of the trucks that the staff selected. After that, system will start to arrange on the sequence of the locations to deliver in order to find the most optimal (shortest) route for each cluster. The main feature of this system is to

arrange the sequence of locations to deliver and suggest the optimal route for logistic company, therefore they no need to struggle on the route planning task.

The completed system for this iteration is then sent for testing. The system must pass all those testing which include unit testing, integration testing and also user acceptance testing to make sure that all the requirements had been met.

In conclusion, a web-based route optimization system had been delivered at the end of the SDLC and all the objectives have been full filled:

- To perform literature reviews on the algorithm for travelling salesman problem, vehicle routing problem and container loading problem.
- To develop an algorithm which is able to optimize the route to deliver compare with the previous route that manually planned by manager.
- To develop a system that is able to show the calculated route on the SMART D/O based on defined rules above.

7.2 Limitations and Recommendations for Future Improvement

Although all objectives which had been set for this project had been achieved, this route optimization system still consists of several limitations due to the time constraint. The following table shows a list of limitations of the system along with the recommendation for future improvement.

Table 7.1 Limitations and Recommendations

No	Limitation	Recommendation
1.	The current system is not able to fully utilize the volume of the truck/cargo space.	Enhance on the currently developed algorithm to include the container loading algorithm which is able to fully utilize the truck and provide visualization of the orientation of the placement of those cargoes in the truck.to the user.
2.	The route optimizing algorithm for this current system is not involving any real-life traffic pattern.	- Perform literature review on the route planning algorithm which involved historical pattern of the real-life traffic.

		- Apply machine learning on the daily traffic hence system is able to suggest a route which enables driver to avoid traffic congestion.
3.	The current system has restricted user in the way that user is not able to make changes on the route after the result had been displayed. User may need to cancel the current result and filter the D/O again to perform a new calculation on the route.	- Enable user to set priority on each D/O. System will make sure that D/O with the higher priority will be considered to deliver first comparing to others D/O which have the lower priority. - Enable user to add/remove a particular D/O on the planned route. After an action (add/remove) been trigger, the system shall be able to re-calculate the route instead of cancelling the current result.
4.	Current system does not provide a feedback feature for the user.	Develop a feedback channel for user to comment on the problem they face when using this system.
5.	The current clustering algorithm is not wise enough to evenly cluster those D/O that exist.	Perform literature review on other clustering methods and make improvements on the current clustering algorithm.
6.	The route optimizer algorithm used for this system is genetic algorithm and it only can provide an approximation solution that is closer to the actual result. And sometime the result may not be accurate due to the characteristic of genetic algorithm – randomness.	Perform literature review on other route optimize and make improvement on the current one or add the new algorithm as an alternative to the route calculation process.
7.	Limitation on the usage of Google Maps API.	Design the algorithm in a way that is able to optimize the usage of the API or upgrade the usage limit of the particular API from Google services.

REFERENCES

Agency for Healthcare Research and Quality, 2013. *Mixed Methods: Integrating Quantitative and Qualitative Data Collection and Analysis While Studying Patient-Centered Medical Home Models*, [online] Available at: <https://www.coursehero.com/file/32765951/MixedMethods-032513comppdf/> [Accessed 17 June 2019].

Ahn, C.W. and Ramakrishna, R.S., 2002. A genetic algorithm for shortest path routing problem and the sizing of populations. *IEEE Transactions on Evolutionary Computation*, [online] Available at: <https://ieeexplore.ieee.org/abstract/document/1134124> [Accessed 23 June 2019].

Balaji, S. and Murugaiyan, M.S., 2012. Waterfall Model Vs Agile: A Comparative Study On SDLC. *International Journal of Information Technology and Business Management*, [online] Available at: <www.jitbm.com> [Accessed 13 June 2019].
s

Carter, R.A., Antón, A.I., Dagnino, A. and Williams, L., 2001. Evolving beyond requirements creep: A risk-based evolutionary prototyping model. *Proceedings of the IEEE International Conference on Requirements Engineering*, [online] Available at: <https://ieeexplore.ieee.org/document/948548> [Accessed 15 June 2019].

Carwalo, T., Thankappan, J. and Patil, V., 2017. A Variant Fisher and Jaikuamr Algorithm to Solve Capacitated Vehicle Routing Problem Tanzima. *2017 2nd International Conference on Communication Systems, Computing and IT Applications, CSCITA 2017 - Proceedings*, [online] Available at: <https://ieeexplore.ieee.org/abstract/document/8079932> [Accessed 3 July 2019].

CHAOS, 2014. *Chaos Summary 2014*. [online] Available at: <https://www.projectsmart.co.uk/white-papers/chaos-report.pdf> [Accessed 22 June 2019].

Clement, J., 2019. *Introduction to Hierarchical Clustering*. [online] Available at: < https://towardsdatascience.com/introduction-hierarchical-clustering-d3066c6b560e > [Accessed 15 March 2020].

Easwaramoorthy, M. and Zarinpoush, F., 2011. Interviewing For Research. *APAP 2011 - Proceedings: 2011 International Conference on Advanced Power System Automation and Protection*, [online] Available at: < http://sectorsource.ca/sites/default/files/resources/files/tipsheet6_interviewing_for_research_en_0.pdf > [Accessed 17 June 2019].

EasyCargo, n.d.. *Container loading and truck loading software EasyCargo*. [online] Available at: <https://www.easycargo3d.com/en/> [Accessed 25 June 2019].

Fan, W., 2006. Optimal Transit Route Network Design Problem with Variable Transit Demand: Genetic Algorithm Approach. *Journal of Transportation Engineering*, [online] Available at: <https://www.researchgate.net/publication/248571503_Optimal_Transit_Route_Network_Design_Problem_with_Variable_Transit_Demand_Genetic_Algorithm_Appro

ach> [Accessed 28 June 2019].

Frechtling, J., Hood, S., Frierson, H. and Hughes, G., 2002. The 2002 User-Friendly Handbook for Project Evaluation for Project Evaluation. *Communication*, [online] Available at: <http://eric.ed.gov/ERICWebPortal/recordDetail?accno=ED468812> [Accessed 13 June 2019].

Geambasu, C.V., Jianu, I., Jianu, I. and Gavrila, A., 2011. Influence Factors for the Choice of a Software Development Methodology. *Contabilitate şi Informatică de Gestiune*, [online] Available at: <https://www.researchgate.net/publication/227367934_Influence_Factors_for_the_Choice_of_a_Software_Development_Methodology> [Accessed 14 June 2019].

Gordon, V.S. and Bieman, J.M., 1995. Rapid Prototyping: Lessons Learned. *IEEE Software*, [online] Available at: <https://ieeexplore.ieee.org/document/363162> [Accessed 16 June 2019].

Grover, V., 2015. *RESEARCH APPROACH: AN OVERVIEW*. [online] Available at: <https://www.researchgate.net/publication/273352276_RESEARCH_APPROACH_AN_OVERVIEW> [Accessed 16 June 2019].

Khalil, M., Li, J.P., Wang, Y. and Khan, A., 2017. Algorithm to solve travel salesman problem efficently. *2016 13th International Computer Conference on Wavelet Active Media Technology and Information Processing, ICCWAMTIP 2017*, [online] Available at: <https://ieeexplore-ieee-org.libezp2.utar.edu.my/stamp/stamp.jsp?tp=&arnumber=8079819> [Accessed 28 June 2019].

Langkos, A., 2014. *CHAPTER3 - RESEARCH METHODOLOGY: Data collection method and Research tools*. [online] Available at: <https://www.researchgate.net/publication/270956555_CHAPTER_3_-_RESEARCH_METHODOLOGY_Data_collection_method_and_Research_tools> [Accessed 19 June 2019].

Mak, K.S., Hafit, H., Kasim, S., Fudzee, M.F. et al, 2017. Web Based Watermarking System. *Acta Informatica Malaysia*, [online] Available at: <https://pdfs.semanticscholar.org/1083/14c4a6f7679d0179c8d89f1bc26d53437e3b.pdf> [Accessed 13 June 2019].

Martínez, D.A., Alvarez-Valdes, R. and Parreño, F., 2015. a Grasp Algorithm for the Container Loading Problem With Multi-Drop Constraints. *Pesquisa Operacional*, [online] Available at: <http://www.scielo.br/pdf/pope/v35n1/0101-7438-pope-35-01-00001.pdf> [Accessed 15 July 2019].

Maule, A.J, 2010. Can Computers Help Overcome Limitations in Human Decision Making? *International Journal of Human-Compouter Interaction.* [Online] Available at: < https://www.tandfonline.com/doi/full/10.1080/10447310903498684?scroll=top&needAccess=true > [Accessed 1June 2019].

McCusker, K. and Gunaydin, S., 2015. Research using qualitative, quantitative or mixed methods and choice based on the research. *Perfusion (United Kingdom)*, [online] Available at: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.968.2338&rep=rep1&type=pdf> [Accessed 17 June 2019].

My Route Online, 2010. *Multiple Stop Route Planner*. [online] Available at: <https://www.myrouteonline.com/?gclid=Cj0KCQjwu6fzBRC6ARIsAJUwa2So_HsZ5GwOazgUEFw0S50wcxMHcYHlniTgNUaVdOAcvrtpUsY4ywYaAroPEALw_wcB> [Accessed 16 June 2019].

SEARATES, 2005. *Find the best freight quote*. [online] Available at: < https://www.searates.com/reference/stuffing/> [Accessed 27 June 2019].

Sharma, S., Sarkar, D. and Gupta, D., 2012. Agile Processes and Methodologies : A Conceptual Study. *International Journal on computer science and Engineering*, [online] Available at: < https://www.yashada.org/yash/egovcii/static_pgs/TC/IJCSE12-04-05-186.pdf> [Accessed 16 June 2019].

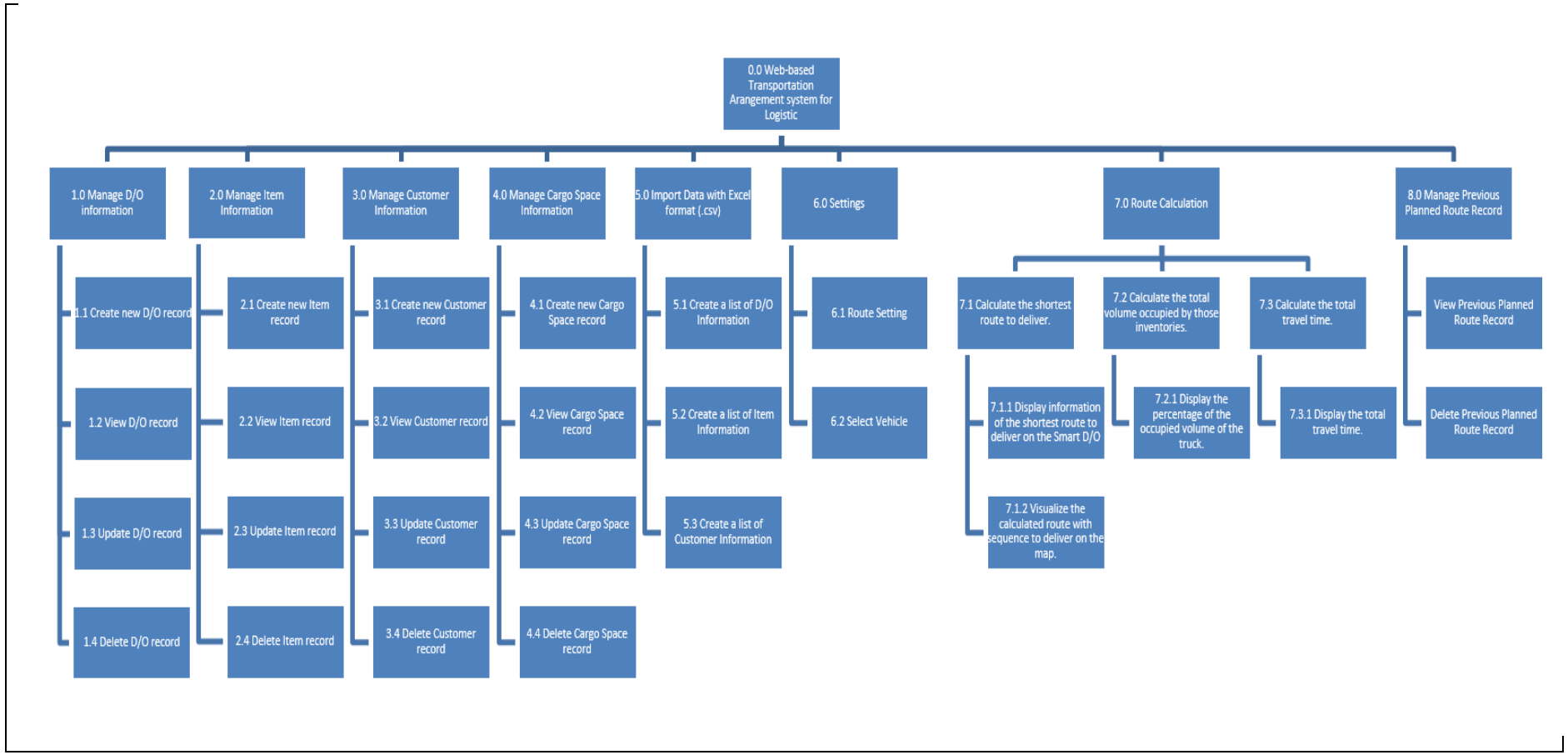
Software Testing Fundamentals, 2020. *Unit Testing - Software Testing Fundamentals*. [online] Available at: <http://softwaretestingfundamentals.com/unit-testing/> [Accessed 1 April 2020].

Tipping, A. and Kauschke, P., 2016. Shifting patterns. *Nursing standard (Royal College of Nursing (Great Britain) : 1987)*, [online] Available at: <https://www.pwc.com/sg/en/publications/assets/future-of-the-logistics-industry.pdf> [Accessed 5 July 2019].

Zuraimi, A.A., Rafi, Y.M., Dahlan, I.M. and Fadiah, M.Z., 2012. A Study of Logistics Development In The Malaysia Eastern Region: A Descriptive Analysis. *International Journal of Business and Social Research (IJBSR)*, [online] Available at: <http://thejournalofbusiness.org/index.php/site/article/view/170> [Accessed 1 June 2019].

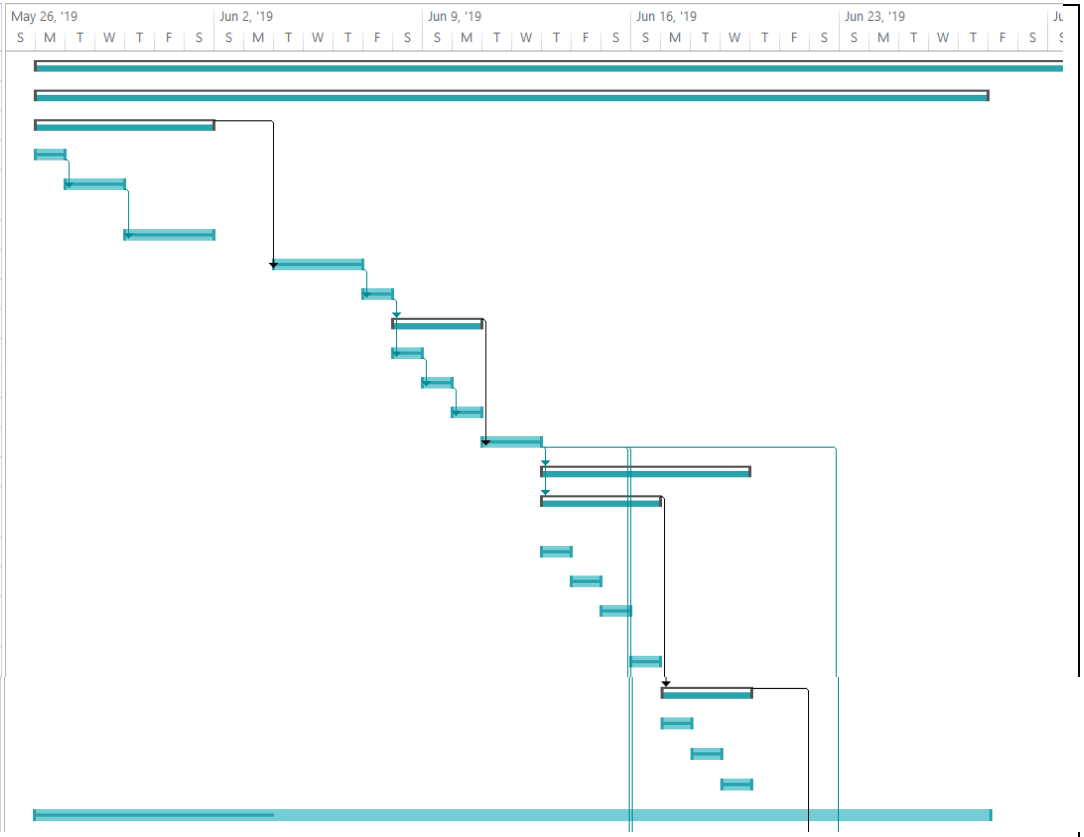
APPENDICES

APPENDIX A: Work Breakdown Structure for System

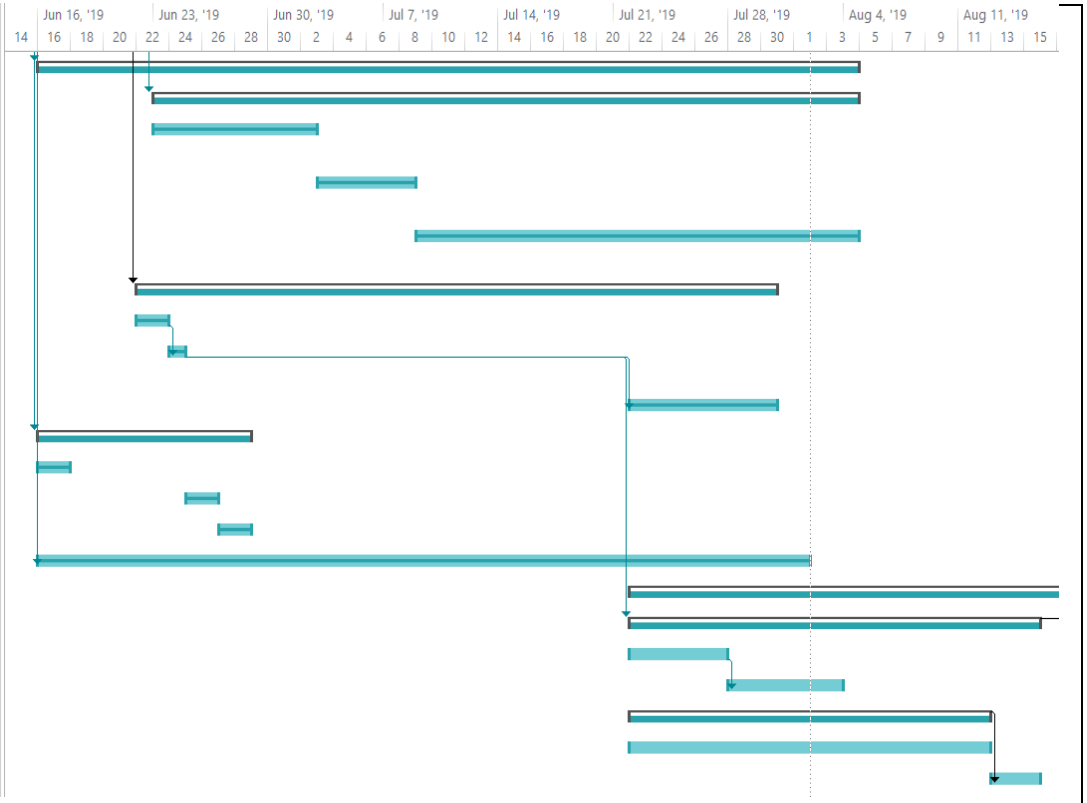

0.0 Web-based Transportation Arangement system for Logistic
1.0 Manage D/O information
1.1 Create new D/O record
1.2 View D/O record
1.3 Update D/O record
1.4 Delete D/O record
2.0 Manage Item Information
2.1 Create new Item record
2.2 View Item record
2.3 Update Item record
2.4 Delete Item record
3.0 Manage Customer Information
3.1 Create new Customer record
3.2 View Customer record
3.3 Update Customer record
3.4 Delete Customer record
4.0 Manage Cargo Space Information
4.1 Create new Cargo Space record
4.2 View Cargo Space record
4.3 Update Cargo Space record
4.4 Delete Cargo Space record
5.0 Import Data with Excel format (.csv)
5.1 Create a list of D/O Information
5.2 Create a list of Item Information
5.3 Create a list of Customer Information
6.0 Settings
6.1 Route Setting
6.2 Select Vehicle
7.0 Route Calculation
7.1 Calculate the shortest route to deliver.
7.1.1 Display information of the shortest route to deliver on the Smart D/O
7.1.2 Visualize the calculated route with sequence to deliver on the map.
7.2 Calculate the total volume occupied by those inventories.
7.2.1 Display the percentage of the occupied volume of the truck.
7.3 Calculate the total travel time.
7.3.1 Display the total travel time.
8.0 Manage Previous Planned Route Record
View Previous Planned Route Record
Delete Previous Planned Route Record

APPENDIX B: Gantt Chart

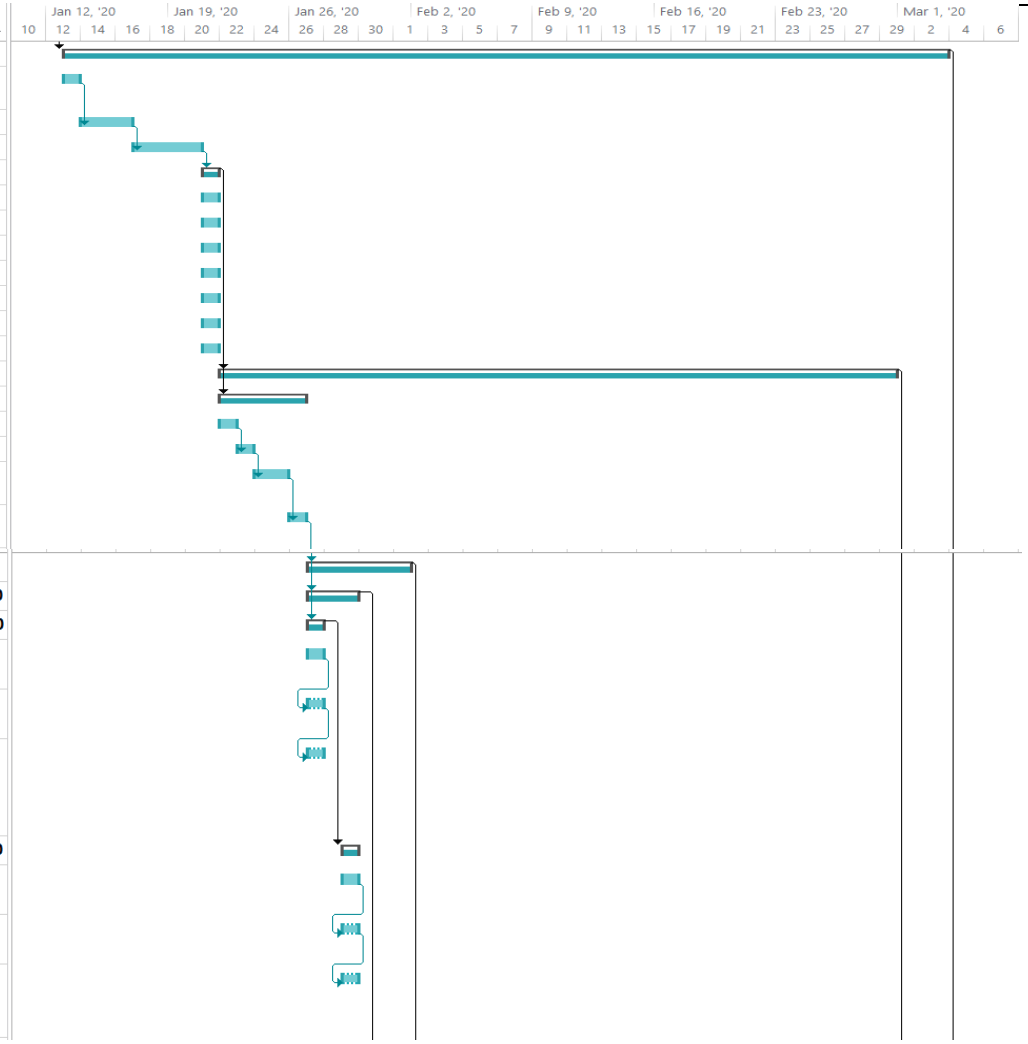
	Task Name	Duration	Start	Finish
1	**Web-Based Route Optimization System For Logistic**	**314 days**	**Mon 5/27/19**	**Sat 4/4/20**
2	**1.0 Planning Phase**	**32 days**	**Mon 5/27/19**	**Thu 6/27/19**
3	**1.1 Develop Work Plan**	**6 days**	**Mon 5/27/19**	**Sat 6/1/19**
4	1.1.1 Determine Project Milestones	1 day	Mon 5/27/19	Mon 5/27/19
5	1.1.2 Develop Project's Work Breakdown Structure	2 days	Tue 5/28/19	Wed 5/29/19
6	1.1.3 Develop Project's Gantt Chart	3 days	Thu 5/30/19	Sat 6/1/19
7	1.2 Determine Problem Statement	3 days	Tue 6/4/19	Thu 6/6/19
8	1.3 Determine Project's Objectives	1 day	Fri 6/7/19	Fri 6/7/19
9	**1.4 Determine Project's Scopes**	**3 days**	**Sat 6/8/19**	**Mon 6/10/19**
10	1.4.1 Identify Scope Covered	1 day	Sat 6/8/19	Sat 6/8/19
11	1.4.2 Identify Topic Out of Scope	1 day	Sun 6/9/19	Sun 6/9/19
12	1.4.3 Identify Assumption of Scope	1 day	Mon 6/10/19	Mon 6/10/19
13	1.5 Determine Project's Proposed Solution	2 days	Tue 6/11/19	Wed 6/12/19
14	**1.6 Determine Project's Proposed Approach**	**7 days**	**Thu 6/13/19**	**Wed 6/19/19**
15	**1.6.1 Determine Software Development Methodology**	**4 days**	**Thu 6/13/19**	**Sun 6/16/19**
16	1.6.1.1 Research on Waterfall Methodology	1 day	Thu 6/13/19	Thu 6/13/19
17	1.6.1.2 Research on V-Model Methodology	1 day	Fri 6/14/19	Fri 6/14/19
18	1.6.1.3 Research on Prototyping Methodology	1 day	Sat 6/15/19	Sat 6/15/19
19	1.6.1.4 Research on Agile Methodology	1 day	Sun 6/16/19	Sun 6/16/19
20	**1.6.2 Determine Research Methodology**	**3 days**	**Mon 6/17/19**	**Wed 6/19/19**
21	1.6.2.1 Research on Qualitative Approach	1 day	Mon 6/17/19	Mon 6/17/19
22	1.6.2.2 Research on Quantitative Approach	1 day	Tue 6/18/19	Tue 6/18/19
23	1.6.2.3 Research on Mixed Approach	1 day	Wed 6/19/19	Wed 6/19/19
24	1.7 Preliminary Report Writing	32 days	Mon 5/27/19	Thu 6/27/19



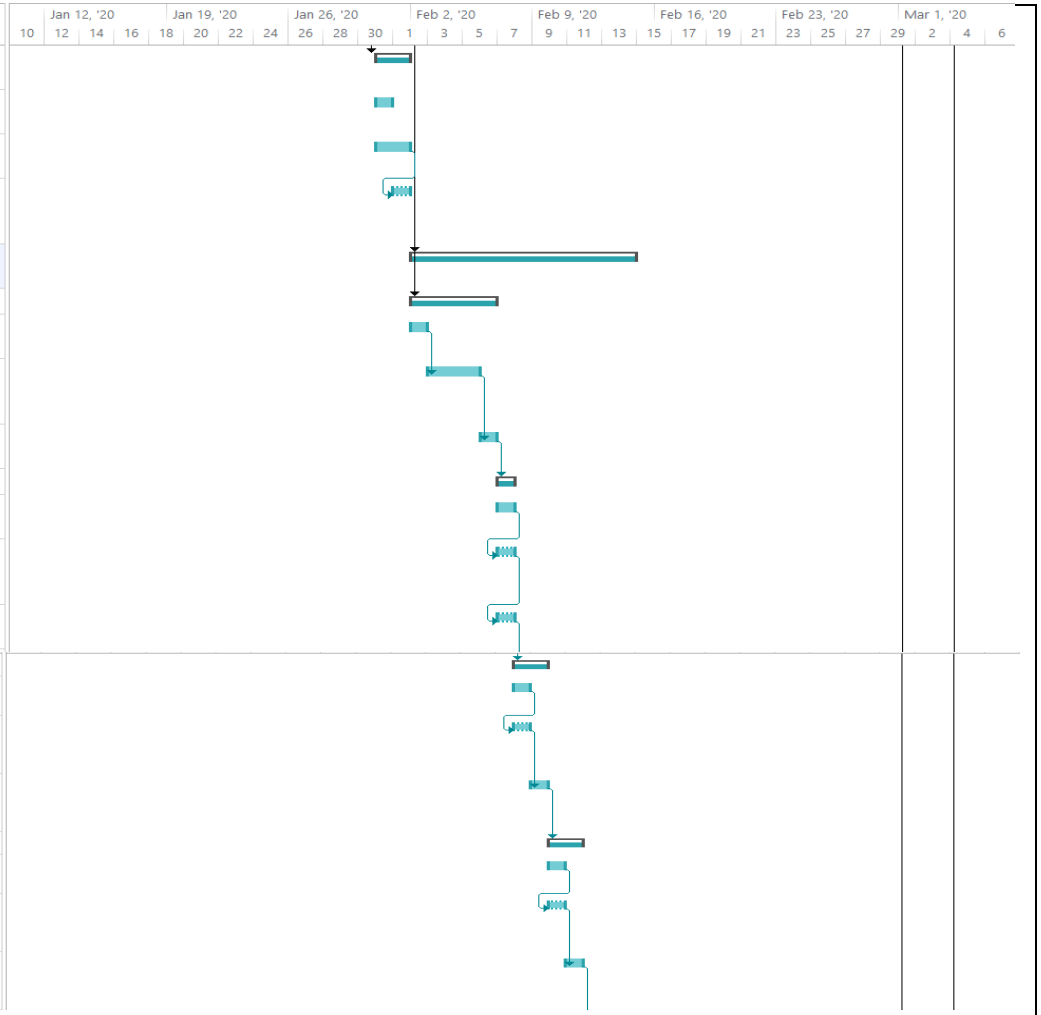
	Task Name	Duration	Start	Finish
25	**2.0 Requirement Gathering Phase**	**50 days**	**Sun 6/16/19**	**Sun 8/4/19**
26	**2.1 Perform Literature Review**	**43 days**	**Sun 6/23/19**	**Sun 8/4/19**
27	2.1.1 Research on Travelling Salesman Problem and its algorithm	10 days	Sun 6/23/19	Tue 7/2/19
28	2.1.2 Research on Vehicle Routing Problem and its algorithm	6 days	Wed 7/3/19	Mon 7/8/19
29	2.1.3 Research on Bin/Truck Packing Problem and its algorithm	27 days	Tue 7/9/19	Sun 8/4/19
30	**2.2 Conduct an Interview**	**39 days**	**Sat 6/22/19**	**Tue 7/30/19**
31	2.2.1 Generation of Interview Questions	2 days	Sat 6/22/19	Sun 6/23/19
32	2.2.2 Conduct Interview with manager of Hondail Trading Sdn Bhd	1 day	Mon 6/24/19	Mon 6/24/19
33	2.2.3 Analysis of findings	9 days	Mon 7/22/19	Tue 7/30/19
34	**2.3 Review on Existing Application**	**13 days**	**Sun 6/16/19**	**Fri 6/28/19**
35	2.3.1 Review on My Route Online	2 days	Sun 6/16/19	Mon 6/17/19
36	2.3.2 Review on Easy Cargo	2 days	Tue 6/25/19	Wed 6/26/19
37	2.3.3 Review on LoadCalculator	2 days	Thu 6/27/19	Fri 6/28/19
38	2.4 Writing Literature Review	47 days	Sun 6/16/19	Thu 8/1/19
39	**3.0 Prototype Development Phase**	**237 days**	**Mon 7/22/19**	**Sat 3/14/20**
40	**3.1 First Iteration**	**25 days**	**Mon 7/22/19**	**Thu 8/15/19**
41	3.1.1 Determine Use Case Diagram	6 days	Mon 7/22/19	Sat 7/27/19
42	3.1.2 Determine Use Case Descriptions	7 days	Sun 7/28/19	Sat 8/3/19
43	**3.1.3 Build Prototype**	**22 days**	**Mon 7/22/19**	**Mon 8/12/19**
44	3.1.3.1 Design User Interface	22 days	Mon 7/22/19	Mon 8/12/19
45	3.1.4 Collect User Feedback	3 days	Tue 8/13/19	Thu 8/15/19



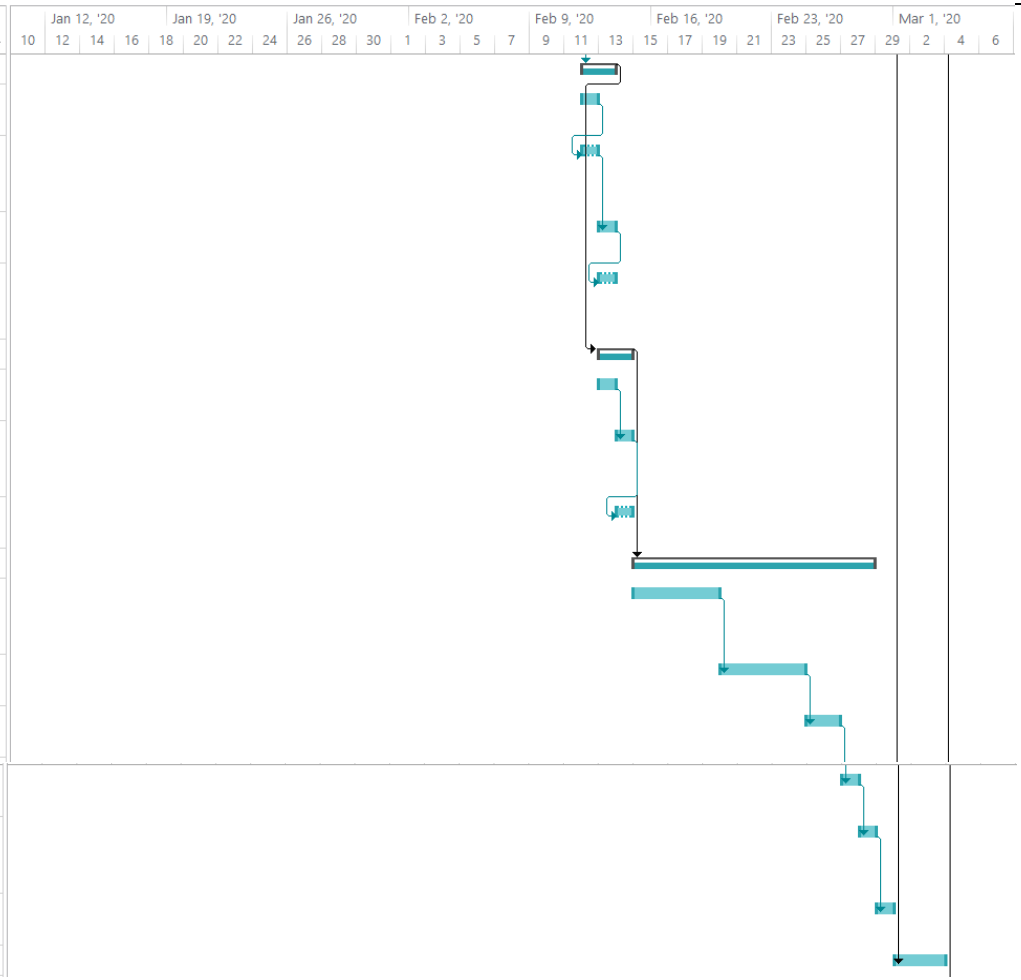
	Task Name	Duration	Start	Finish
46	**3.2 Second Iteration**	**51 days**	**Mon 1/13/20**	**Tue 3/3/20**
47	3.2.1 Improve on use case and use case description	1 day	Mon 1/13/20	Mon 1/13/20
48	3.2.2 Draw Activity Diagram	3 days	Tue 1/14/20	Thu 1/16/20
49	3.2.3 Draw Data Flow Diagram	4 days	Fri 1/17/20	Mon 1/20/20
50	**3.2.4 Draw ERD Diagram**	**1 day**	**Tue 1/21/20**	**Tue 1/21/20**
51	3.2.4.1 User Collection	1 day	Tue 1/21/20	Tue 1/21/20
52	3.2.4.2 Customer Collection	1 day	Tue 1/21/20	Tue 1/21/20
53	3.2.4.3 Item Collection	1 day	Tue 1/21/20	Tue 1/21/20
54	3.2.4.4 D/O Collection	1 day	Tue 1/21/20	Tue 1/21/20
55	3.2.4.5 Cargoes Collection	1 day	Tue 1/21/20	Tue 1/21/20
56	3.2.4.6 Route Setting Collection	1 day	Tue 1/21/20	Tue 1/21/20
57	3.2.4.7 Route Record Collection	1 day	Tue 1/21/20	Tue 1/21/20
58	**3.2.5 Build Prototype**	**39 days**	**Wed 1/22/20**	**Sat 2/29/20**
59	**3.2.5.1 Login and SignUp Feature**	**5 days**	**Wed 1/22/20**	**Sun 1/26/20**
60	3.2.5.1.1 Create Login screen	1 day	Wed 1/22/20	Wed 1/22/20
61	3.2.5.1.2 Create Sign Up screen	1 day	Thu 1/23/20	Thu 1/23/20
62	3.2.5.1.3 Connect with Firebase Authentication	2 days	Fri 1/24/20	Sat 1/25/20
63	3.2.5.1.4 Connect and store user data into Cloud Firestore	1 day	Sun 1/26/20	Sun 1/26/20
64	**3.2.5.2 Import Data Feature**	**6 days**	**Mon 1/27/20**	**Sat 2/1/20**
65	**3.2.5.2.1 Create Home Screen**	**3 days**	**Mon 1/27/20**	**Wed 1/29/20**
66	**3.2.5.2.1.1 "Add D/O" tab**	**1 day**	**Mon 1/27/20**	**Mon 1/27/20**
67	3.2.5.2.1.1.1 Create UI for Add D/O view	1 day	Mon 1/27/20	Mon 1/27/20
68	3.2.5.2.1.1.2 Enable user to create new D/O and store data to firestore	1 day	Mon 1/27/20	Mon 1/27/20
69	3.2.5.2.1.1.3 Enable system to retrieve customer and item data for selection purpose when creating new D/O	1 day	Mon 1/27/20	Mon 1/27/20
70	**3.2.5.2.1.2 "Settings" tab**	**1 day?**	**Wed 1/29/20**	**Wed 1/29/20**
71	3.2.5.2.1.2.1 Create UI for Settings view	1 day	Wed 1/29/20	Wed 1/29/20
72	3.2.5.2.1.2.1 Store route setting information into the firestore	1 day	Wed 1/29/20	Wed 1/29/20
73	3.2.5.2.1.2.3 Retrieve cargo information and display its on the Vehicle Selection area	1 day	Wed 1/29/20	Wed 1/29/20



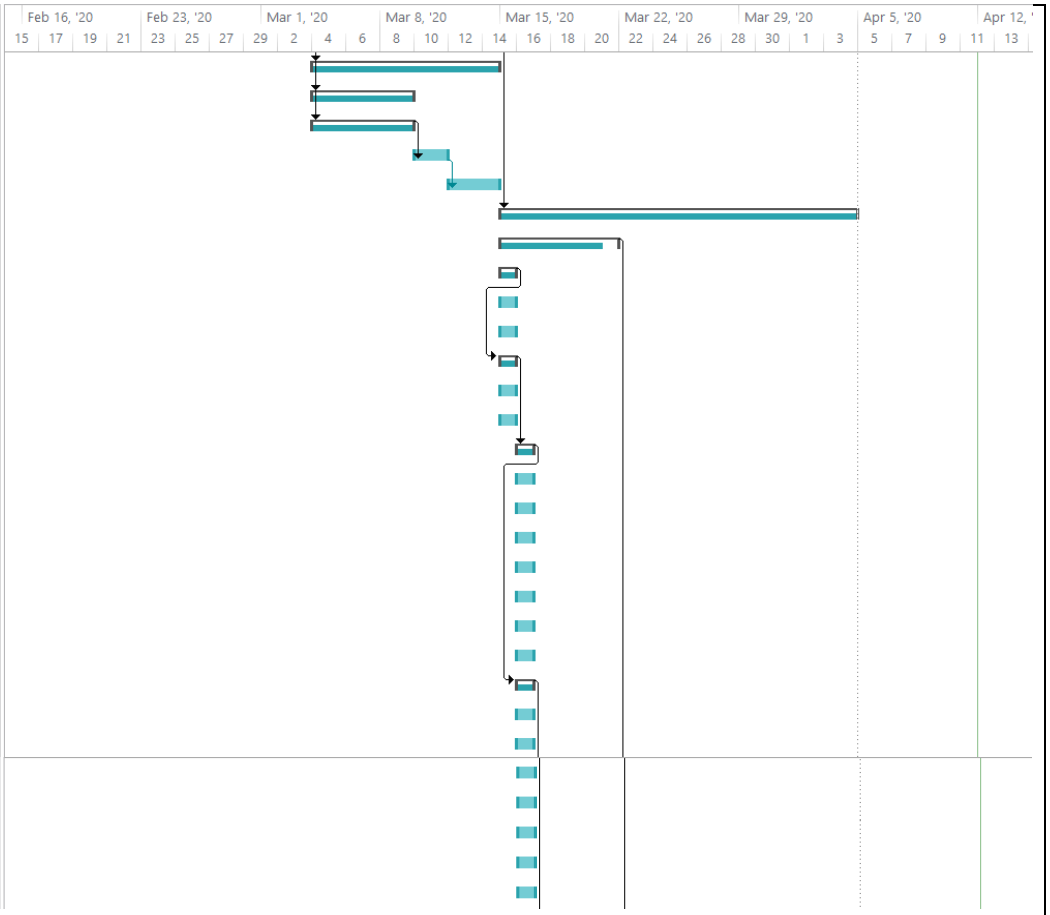
	Task Name	Duration	Start	Finish
74	**3.2.5.2.2 Import data from excel file (.csv) format**	**2 days**	**Fri 1/31/20**	**Sat 2/1/20**
75	3.2.5.2.2.1 Create UI for Import Data Screen	1 day	Fri 1/31/20	Fri 1/31/20
76	3.2.5.2.2.2 Research and implement react-excel-renderer on this screen	2 days	Fri 1/31/20	Sat 2/1/20
77	3.2.5.2.2.3 Enable the system to store data (customer, d/o and item) into the firestore	1 day	Sat 2/1/20	Sat 2/1/20
78	**3.2.5.3 Display and Update record for the system**	**13 days**	**Sun 2/2/20**	**Fri 2/14/20**
79	**3.2.5.3.1 Smart D/O screen**	**5 days**	**Sun 2/2/20**	**Thu 2/6/20**
80	3.2.5.3.1.1 Create UI for Smart D/O tab view	1 day	Sun 2/2/20	Sun 2/2/20
81	3.2.5.3.1.2 Enable to display the information of D/Os based on the selected date range	3 days	Mon 2/3/20	Wed 2/5/20
82	3.2.5.3.1.3 Remove a particular D/O from the displayed list	1 day	Thu 2/6/20	Thu 2/6/20
83	**3.2.5.3.2 Cargoes Space screen**	**1 day**	**Fri 2/7/20**	**Fri 2/7/20**
84	3.2.5.3.2.1 Create UI for Cargoes Space screen	1 day	Fri 2/7/20	Fri 2/7/20
85	3.2.5.3.2.2 Enable system to retrieve cargoes data from firestore and display them on screen	1 day	Fri 2/7/20	Fri 2/7/20
86	3.2.5.3.2.3 Enable user to update the satus of a particular cargo	1 day	Fri 2/7/20	Fri 2/7/20
87	**3.2.5.3.3 Customer Information screen**	**2 days**	**Sat 2/8/20**	**Sun 2/9/20**
88	3.2.5.3.3.1 Create UI for Customer Informationi screen	1 day	Sat 2/8/20	Sat 2/8/20
89	3.2.5.3.3.2 Enable system to retrieve Customer data from firestore and display them on the screen	1 day	Sat 2/8/20	Sat 2/8/20
90	3.2.5.3.3.3 Enable user to edit a particular Customer then update the data to firestore	1 day	Sun 2/9/20	Sun 2/9/20
91	**3.2.5.3.4 Item Information screen**	**2 days**	**Mon 2/10/20**	**Tue 2/11/20**
92	3.2.5.3.4.1 Create UI for Item Informationi screen	1 day	Mon 2/10/20	Mon 2/10/20
93	3.2.5.3.4.2 Enable system to retrieve Item data from firestore and display them on the screen	1 day	Mon 2/10/20	Mon 2/10/20
94	3.2.5.3.4.3 Enable user to edit a particular Item then update the data to firestore	1 day	Tue 2/11/20	Tue 2/11/20



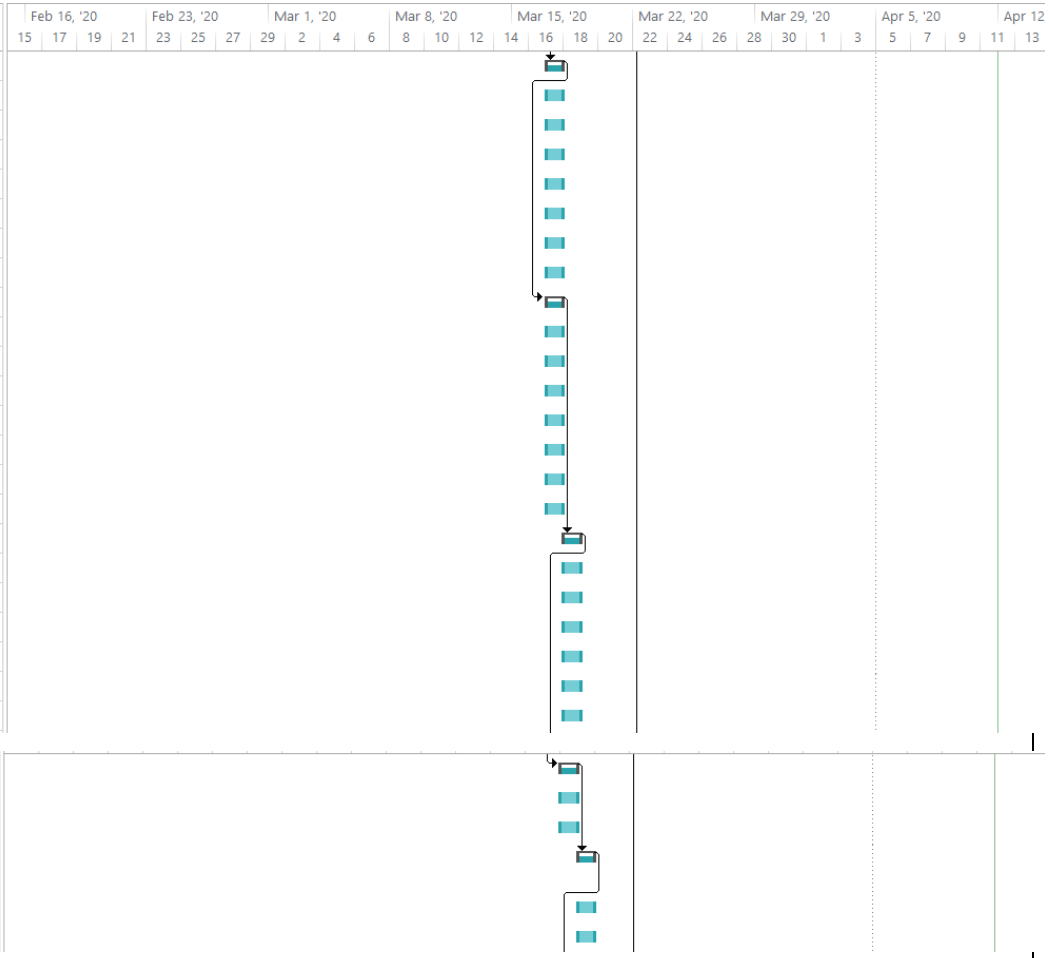
	Task Name	Duration	Start	Finish
95	**3.2.5.3.5 D/O Information screen**	**2 days**	**Wed 2/12/20**	**Thu 2/13/20**
96	3.2.5.3.5.1 Create UI for D/O Informationi screen	1 day	Wed 2/12/20	Wed 2/12/20
97	3.2.5.3.5.2 Enable system to retrieve D/O data from firestore and display them on the screen	1 day	Wed 2/12/20	Wed 2/12/20
98	3.2.5.3.5.3 Display the detail of the D/O in PDF format	1 day	Thu 2/13/20	Thu 2/13/20
99	3.2.5.3.5.4 Enable user to edit a particular D/O then update the data to firestore	1 day	Thu 2/13/20	Thu 2/13/20
100	**3.2.5.3.6 Route Record Screen**	**2 days**	**Thu 2/13/20**	**Fri 2/14/20**
101	3.2.5.3.6.1 Create UI for Route Record screen	1 day	Thu 2/13/20	Thu 2/13/20
102	3.2.5.3.6.2 Enable system to retrieve route record from firestore and display them on the screen	1 day	Fri 2/14/20	Fri 2/14/20
103	3.2.5.3.6.3 Display the detail of the route record in PDF format	1 day	Fri 2/14/20	Fri 2/14/20
104	**3.2.5.4 Implementation on algorithm**	**14 days**	**Sat 2/15/20**	**Fri 2/28/20**
105	3.2.5.4.1 Implement algorithm to cluster those customers' location based on the number of trucks	5 days	Sat 2/15/20	Wed 2/19/20
106	3.2.5.4.2 Implement Algorithm to find the shortes path (Genetic Algorithm)	5 days	Thu 2/20/20	Mon 2/24/20
107	3.2.5.4.3 Explore and apply Google Map Api in the algorithm	2 days	Tue 2/25/20	Wed 2/26/20
108	3.2.5.4.4 Implement algorithm to count on the total occupied space in the truck	1 day	Thu 2/27/20	Thu 2/27/20
109	3.2.5.4.5 Implement algorithm to count on the total travelled distance and total travel duration	1 day	Fri 2/28/20	Fri 2/28/20
110	3.2.5.5 Visualize the route on the Google Map	1 day	Sat 2/29/20	Sat 2/29/20
111	3.2.6 Collect User Feedback	3 days	Sun 3/1/20	Tue 3/3/20



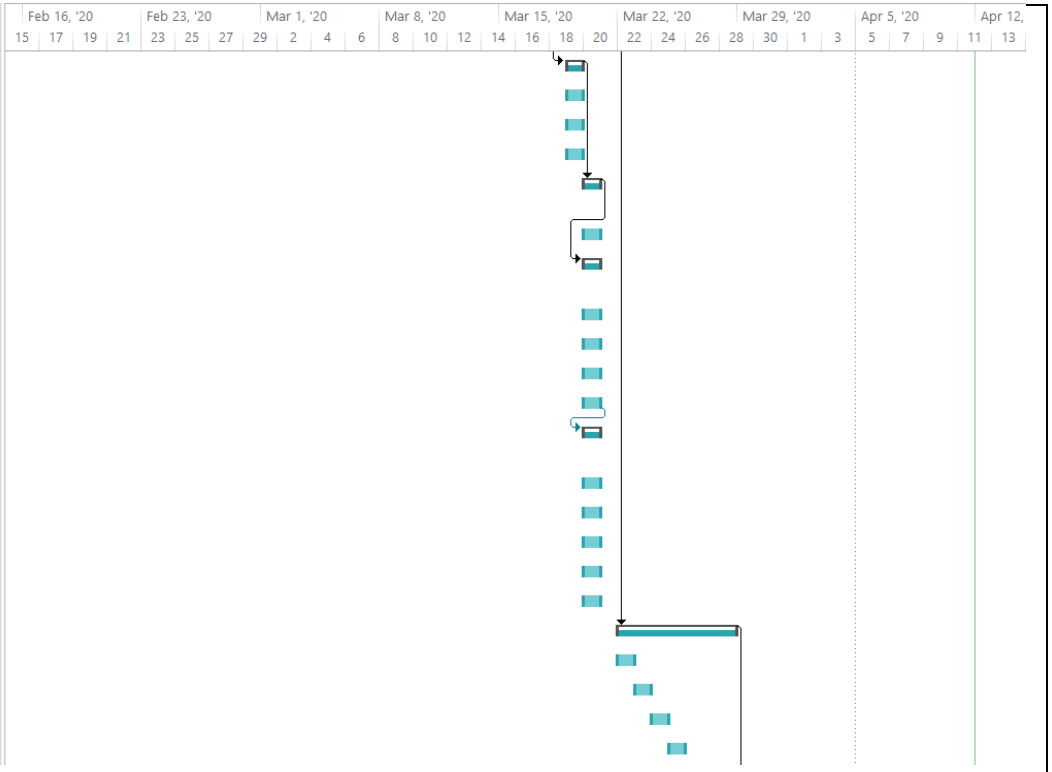
	Task Name	Duration	Start	Finish
112	**3.3 Third Iteration**	**11 days**	**Wed 3/4/20**	**Sat 3/14/20**
113	**3.3.1 Build Prototype**	**6 days**	**Wed 3/4/20**	**Mon 3/9/20**
114	**3.3.1.1 Add on missing requirement**	**6 days**	**Wed 3/4/20**	**Mon 3/9/20**
129	Collect User Feedback	2 days	Tue 3/10/20	Wed 3/11/20
130	Integrate prototype to final product	3 days	Thu 3/12/20	Sat 3/14/20
131	**4.0 Testing Phase**	**21 days**	**Sun 3/15/20**	**Sat 4/4/20**
132	**4.1 Unit Testing**	**7 days**	**Sun 3/15/20**	**Sat 3/21/20**
133	**4.1.1 Sign In feature**	**1 day**	**Sun 3/15/20**	**Sun 3/15/20**
134	4.1.1.1 Test Case ID: Login-1A	1 day	Sun 3/15/20	Sun 3/15/20
135	4.1.1.2 Test Case ID: Login-1B	1 day	Sun 3/15/20	Sun 3/15/20
136	**4.1.2 Sign Up feature**	**1 day**	**Sun 3/15/20**	**Sun 3/15/20**
137	4.1.2.1 Test Case ID: SignUp-1A	1 day	Sun 3/15/20	Sun 3/15/20
138	4.1.2.2 Test Case ID: SignUp-1B	1 day	Sun 3/15/20	Sun 3/15/20
139	**4.1.3 Manage D/O Information**	**1 day**	**Mon 3/16/20**	**Mon 3/16/20**
140	4.1.3.1 Test Case ID: CreateD/O-1A	1 day	Mon 3/16/20	Mon 3/16/20
141	4.1.3.2 Test Case ID: CreateD/O-1B	1 day	Mon 3/16/20	Mon 3/16/20
142	4.1.3.3 Test Case ID: ViewD/O-1A	1 day	Mon 3/16/20	Mon 3/16/20
143	4.1.3.4 Test Case ID: ViewD/O-1B	1 day	Mon 3/16/20	Mon 3/16/20
144	4.1.3.5 Test Case ID: EditD/O-1A	1 day	Mon 3/16/20	Mon 3/16/20
145	4.1.3.6 Test Case ID: EditD/O-1B	1 day	Mon 3/16/20	Mon 3/16/20
146	4.1.3.7 Test Case ID: DeleteD/O-1A	1 day	Mon 3/16/20	Mon 3/16/20
147	**4.1.4 Manage Item Information**	**1 day**	**Mon 3/16/20**	**Mon 3/16/20**
148	4.1.4.1 Test Case ID: CreateItem-1A	1 day	Mon 3/16/20	Mon 3/16/20
149	4.1.4.2 Test Case ID: CreateItem-1B	1 day	Mon 3/16/20	Mon 3/16/20
150	4.1.4.3 Test Case ID: ViewItem-1A	1 day	Mon 3/16/20	Mon 3/16/20
151	4.1.4.4 Test Case ID: ViewItem-1B	1 day	Mon 3/16/20	Mon 3/16/20
152	4.1.4.5 Test Case ID: EditItem-1A	1 day	Mon 3/16/20	Mon 3/16/20
153	4.1.4.6 Test Case ID: EditItem-1B	1 day	Mon 3/16/20	Mon 3/16/20
154	4.1.4.7 Test Case ID: DeleteItem-1A	1 day	Mon 3/16/20	Mon 3/16/20



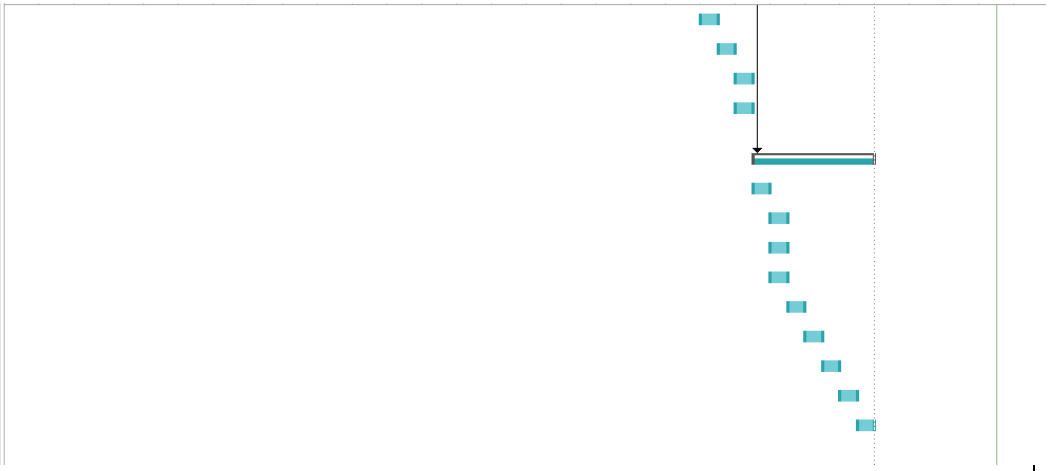
	Task Name	Duration	Start	Finish
155	**4.1.5 Manage Customer Information**	**1 day**	**Tue 3/17/20**	**Tue 3/17/20**
156	4.1.5.1 Test Case ID: CreateCustomer-1A	1 day	Tue 3/17/20	Tue 3/17/20
157	4.1.5.2 Test Case ID: CreateCustomer-1B	1 day	Tue 3/17/20	Tue 3/17/20
158	4.1.5.3 Test Case ID: ViewCustomer-1A	1 day	Tue 3/17/20	Tue 3/17/20
159	4.1.5.4 Test Case ID: ViewCustomer-1B	1 day	Tue 3/17/20	Tue 3/17/20
160	4.1.5.5 Test Case ID: EditCustomer-1A	1 day	Tue 3/17/20	Tue 3/17/20
161	4.1.5.6 Test Case ID: EditCustomer-1B	1 day	Tue 3/17/20	Tue 3/17/20
162	4.1.5.7 Test Case ID: DeleteCustomer-1A	1 day	Tue 3/17/20	Tue 3/17/20
163	**4.1.6 Manage Cargo Information**	**1 day**	**Tue 3/17/20**	**Tue 3/17/20**
164	4.1.6.1 Test Case ID: CreateCargo-1A	1 day	Tue 3/17/20	Tue 3/17/20
165	4.1.6.2 Test Case ID: CreateCargo-1B	1 day	Tue 3/17/20	Tue 3/17/20
166	4.1.6.3 Test Case ID: ViewCargo-1A	1 day	Tue 3/17/20	Tue 3/17/20
167	4.1.6.4 Test Case ID: ViewCargo-1B	1 day	Tue 3/17/20	Tue 3/17/20
168	4.1.6.5 Test Case ID: EditCargo-1A	1 day	Tue 3/17/20	Tue 3/17/20
169	4.1.6.6 Test Case ID: EditCargo-1B	1 day	Tue 3/17/20	Tue 3/17/20
170	4.1.6.7 Test Case ID: DeleteCargo-1A	1 day	Tue 3/17/20	Tue 3/17/20
171	**4.1.7 Import Data with Excel format (.csv)**	**1 day**	**Wed 3/18/20**	**Wed 3/18/20**
172	4.1.7.1 Test Case ID: ImportD/O-1A	1 day	Wed 3/18/20	Wed 3/18/20
173	4.1.7.2 Test Case ID: ImportD/O-1B	1 day	Wed 3/18/20	Wed 3/18/20
174	4.1.7.3 Test Case ID: ImportItem-1A	1 day	Wed 3/18/20	Wed 3/18/20
175	4.1.7.4 Test Case ID: ImportItem-1B	1 day	Wed 3/18/20	Wed 3/18/20
176	4.1.7.5 Test Case ID: ImportCustomer-1A	1 day	Wed 3/18/20	Wed 3/18/20
177	4.1.7.6 Test Case ID: ImportCustomer-1B	1 day	Wed 3/18/20	Wed 3/18/20
178	**4.1.8 Configure the route Setting**	**1 day**	**Wed 3/18/20**	**Wed 3/18/20**
179	4.1.8.1 Test Case ID: ConfigureRoute-1A	1 day	Wed 3/18/20	Wed 3/18/20
180	4.1.8.2 Test Case ID: ConfigureRoute-1B	1 day	Wed 3/18/20	Wed 3/18/20
181	**4.1.9 Display the D/O information which will be involved in later route calculation.**	**1 day**	**Thu 3/19/20**	**Thu 3/19/20**
182	4.1.9.1 Test Case ID: ViewSelectedD/O-1A	1 day	Thu 3/19/20	Thu 3/19/20
183	4.1.9.2 Test Case ID: ViewSelectedD/O-1B	1 day	Thu 3/19/20	Thu 3/19/20



	Task Name	Duration	Start	Finish
184	**4.1.10 Calculate and display suggested route**	**1 day**	**Thu 3/19/20**	**Thu 3/19/20**
185	4.1.10.1 Test Case ID: CalculateRoute-1A	1 day	Thu 3/19/20	Thu 3/19/20
186	4.1.10.2 Test Case ID: CalculateRoute-1B	1 day	Thu 3/19/20	Thu 3/19/20
187	4.1.10.3 Test Case ID: CalculateRoute-1A	1 day	Thu 3/19/20	Thu 3/19/20
188	**4.1.11 Visualization of the calculated route on the map.**	**1 day**	**Fri 3/20/20**	**Fri 3/20/20**
189	4.1.11.1 Test Case ID: VisualizeRoute-1A	1 day	Fri 3/20/20	Fri 3/20/20
190	**4.1.12 Manage Previous Planned Route Record**	**1 day**	**Fri 3/20/20**	**Fri 3/20/20**
191	4.1.12.1 Test Case ID: CreateRoute-1A	1 day	Fri 3/20/20	Fri 3/20/20
192	4.1.12.2 Test Case ID: ViewRoute-1A	1 day	Fri 3/20/20	Fri 3/20/20
193	4.1.12.3 Test Case ID: ViewRoute-1B	1 day	Fri 3/20/20	Fri 3/20/20
194	4.1.12.4 Test Case ID: DeleteRoute-1A	1 day	Fri 3/20/20	Fri 3/20/20
195	**4.1.13 Manage Staff Information (Company Admin)**	**1 day**	**Fri 3/20/20**	**Fri 3/20/20**
196	4.1.13.1 Test Case ID: CreateStaff-1A	1 day	Fri 3/20/20	Fri 3/20/20
197	4.1.13.2 Test Case ID: CreateStaff-1B	1 day	Fri 3/20/20	Fri 3/20/20
198	4.1.13.3 Test Case ID: ViewStaff-1A	1 day	Fri 3/20/20	Fri 3/20/20
199	4.1.13.4 Test Case ID: ViewStaff-1B	1 day	Fri 3/20/20	Fri 3/20/20
200	4.1.13.5 Test Case ID: ChangeStaffStatus-1A	1 day	Fri 3/20/20	Fri 3/20/20
201	**4.2 Integration Testing**	**7 days**	**Sun 3/22/20**	**Sat 3/28/20**
202	4.2.1 Manage D/O Information	1 day	Sun 3/22/20	Sun 3/22/20
203	4.2.2 Manage Item Information	1 day	Mon 3/23/20	Mon 3/23/20
204	4.2.3 Manage Customer Information	1 day	Tue 3/24/20	Tue 3/24/20
205	4.2.4 Manage Cargo Information	1 day	Wed 3/25/20	Wed 3/25/20



206	4.2.5 Import Data	1 day	Thu 3/26/20	Thu 3/26/20
207	4.2.6 Route Calculation	1 day	Fri 3/27/20	Fri 3/27/20
208	4.2.7 Manage Route Record	1 day	Sat 3/28/20	Sat 3/28/20
209	4.2.8 Manage Staff Information (Company Admin)	1 day	Sat 3/28/20	Sat 3/28/20
210	**4.3 User Acceptance Testing**	**7 days**	**Sun 3/29/20**	**Sat 4/4/20**
211	4.3.1 Manage D/O Information	1 day	Sun 3/29/20	Sun 3/29/20
212	4.3.2 Manage Item Information	1 day	Mon 3/30/20	Mon 3/30/20
213	4.3.3 Manage Customer Information	1 day	Mon 3/30/20	Mon 3/30/20
214	4.3.4 Manage Cargo Information	1 day	Mon 3/30/20	Mon 3/30/20
215	4.3.5 Import Data with the Excel Format (.csv)	1 day	Tue 3/31/20	Tue 3/31/20
216	4.3.6 Route Setting Process	1 day	Wed 4/1/20	Wed 4/1/20
217	4.3.7 Route Calculation Process	1 day	Thu 4/2/20	Thu 4/2/20
218	4.3.8 Manage Previous Planned Route Record	1 day	Fri 4/3/20	Fri 4/3/20
219	4.3.9 Manage Staff Information (Company Admin)	1 day	Sat 4/4/20	Sat 4/4/20



APPENDIX C: Interview Questions

Interview Questions:

Purpose:

I am software engineering student from UTAR Sungai Long, currently conducting an interview for my final year project titled "Web-Based Route Optimization System For Logistic". The purpose of this interview is to for me to understand and analyze the problems that had been faced by most of the logistic company. Your responses are voluntary and will be confidential. It is solely for academic purpose. Thank you for your cooperation in this interview.

Please answer all the questions below:

1. Based on your experience, is there any factors that need to be consider when arranging the route to deliver? For example, distance between each customers' location, order of the invoice to be sent, or maybe time taken for you drivers to travel.

   _____
   _____

2. Are you familiar with the location and direction of your customer's address without the help of Google Map? If no, do you think that Google Map is helpful for you to find out the exact location of your customer?

   _____
   _____

3. What will you do when there is certain address you are not familiar with during arranging the delivery orders (D/Os)? Google Map will not be included for this question.

   _____
   _____

4. Do you believe that there will be alternative routes which are shorter compare to what you have been planned? If yes, continue to question 5.

_____
_____

5. Have you try to find out this route? If yes, what mechanism or software you have been used to find out those routes? If no, why?

_____
_____

6. Based on your opinion, do you think that finding the shortest route to deliver is able to reduce fuel cost and travelling cost in the long-term? Other than that is there any other benefits that your company will achieved? For example, satisfaction of your customer, because those invoice is able to delivery to them before they close up their shop.

_____
_____

7. If we are going to develop a system which are able to find the shortest route to deliver among your customer's location what suggestion or advise that you can provide?

☐ time taken to calculate the shortest route must be short

☐ able to detect the daily traffic to avoid the congestion

☐ able to automatically select (D/Os) for calculation based on the deadline

☐ at least able to reduce the total distance to travel by 5 to 10%

☐ others (please list below)

_____

_____

8. Do you face problem like yours worker are unable to fit one or two (D/Os) into the truck? If yes, what is the root cause of this?

_____
_____

9. As a manager, what will you do to address the problem in question 8?

_____
_____

10. Based on your experience, how many (D/Os) can be fix into the truck at one time.

_____
_____

11. Based on your experience, how many percentage you will be given in term of utilization the space in the truck?

☐ 51% - 60%
☐ 61% - 70%
☐ 71% - 80%
☐ 81% - 90%
☐ above 90%

12. What factors that you will be consider in order to utilize the available space when loading inventories into the truck? For example:
- the order of the D/O to be sent
- volume of the inventories
- arrangement of the inventories

_____
_____

13. Do you think that having system which are able to calculate and arrange the order of placement of inventories in order to fully utilize the space in the truck is helpful for you? If yes, how? For example:
- able to smooth the overall process of loading inventories into the truck.
- minimize the overall number of trip to deliver

_____
_____

APPENDIX D: Interview Report

Interview Report

Location: Timez Business Hotel

Date: 7 July 2019

An interview was conducted on 7 July 2019 at Timez Business Hotel which located at No. 21, Jalan C180/1, Dataran C180, Kuala Lumpur. The respondent that I conducted interview is the manager of Hondail Trading Sdn Bhd named Tan Hooi Theam. Hondail Trading Sdn Bhd is a company which involving transportation and delivery services in their daily operation. Mr. Tan work with Hondail Trading Sdn Bhd since 2007 years which has a 12 years' experience on planning the route to deliver and supervise on the cargoes loading process. The goals of this interview are to have a better understanding on the overall route planning and cargoes loading process. Besides, this interview also help us to identify what is the factors that will affect the performance of both route planning and cargoes loading process without the help of any existing software. Last, by conducting this interview we can also gathered related requirements which will be implemented in the system to meet the objectives of this project.

Question 1

Based on your experience, is there any factors that need to be consider when arranging the route to deliver? For example, distance between each customers' location, order of the invoice to be sent, or maybe time taken for you drivers to travel.

Basically we will select those D/O which has a close expire date. After those D/Os had been selected, we will look at the address of those D/Os and cluster those D/O based on their area. The number of cluster will be based on the number of vehicles they own. From each cluster of D/Os we will try to plan the shortest route to deliver the cargoes to our customers based on their experience. The order of the route to deliver is based on the location which has the shortest distance. Which mean if they felt that location B has the shortest distance from the starting point compare to others' location then location B will becoming the first address to be deliver. This steps is repeat until all

the D/Os has been arranged into a route. Besides, we will also try to estimate the volume of the cargoes that will loaded into the truck. If the particular truck is not able to fit into the original planned truck then we will look for others truck which still have the available space. The original route may get affected therefore we may need to re-cluster and re-arrange the route. Another factors that will be considered will be the travel time of a particular route. We will try to plan for the route which is able to complete within the working hours.

After got the respond from Mr. Tan I found out that all of the factors that had mentioned above is based on their personal experience and their own estimation which including the location which has the shortest distance, the volume of the cargoes, and the time to deliver. All of them are not reliable because they are not the exact value that calculated from a mathematical statement. Although experience sometime may give us a good solution but sometime we may also have the possibility to select a high cost route. Other than that, I found out that the process for Mr. Tan to plan the route is quite logic hence some part of this process will be used as the algorithm of the system which is going to implement. In the system those estimation factors will replace by mathematical statement with the help of API and algorithm. Therefore a more reliable result will be displayed. In conclusion, the distance between those locations, the total volume of the cargoes and the time to deliver cargoes will be the factors that may need to consider when implementing the system.

Question 2

Are you familiar with the location and direction of your customer's address without the help of Google Map? If no, do you think that Google Map is helpful for you to find out the exact location of your customer?

Not really, I may know some of the address, but some of them I may not exactly know their location. Yes, Google Map is very helpful for us to check for a specific location before planning the route to deliver.

This question prove that with the help of Google Map, Mr. Tan is able to find out the location of a particular address. Therefore they can have a better planning on

their route. In conclusion, a Google Map is able to help them understand where the exact location of a particular address is.

Question 3

What will you do when there is certain address you are not familiar with during arranging the delivery orders (D/Os)?

Here are some examples which provided by Mr. Tan.

- Having a phone call to ask for the detail of the location
- Ask description on the location from the salesman who responsible for this particular customer
- Request customer to send their current location through social media
- Searching for a particular location on the Google Map.

From the respond, we can clearly saw that the solutions that had been used by Mr. Tan to clarify the address which he is not familiar with. Having a phone call or ask for the description from the salesman is not effective, this is because both solution may not exactly show out the location of the address and there may has the possibility that the description provided by the salesman may be wrong. There is also have a high chances that the customer did not answer to your phone call when you are planning the route. While asking for current location through social media likes WhatsApp and direct search on Google Map will be more effective due to the exact location has been shown to the people who responsible in arranging the route. Although Google Map is able to view the exact location, but it is still ineffective when planning the route. People may need to search a particular address one by one on the Google Map before planning for the route and it may use up huge amount of time if the address which are going to search is many. In conclusion, although Google Map is able to show the exact location of a particular address but it is unable to speed up the planning process. Therefore, a system which is able to check the locations of the customer and can handle the route planning process will be a great asset for those company who involving daily operation which is similar with Hondai Trading Sdn Bhd.

Question 4

Do you believe that there will be alternative routes which are more shorter compare to what you have been planned? If yes, continue to question 5.

Yes, I believed that there may have alternative route which is shorter than what I had planned.

Question 5

Have you try to find out this route? If yes, what mechanism or software you have been used to find out those routes? If no, why?

Yes, through daily experience by discovering those alternative route during driving on the route and using apps like Waze to travel through the route.

Question 6

Based on your opinion, do you think that finding the shortest route to deliver is able to reduce fuel cost and travelling cost in the long-term? Other than that is there any other benefits that your company will achieved? For example, satisfaction of your customer, because those invoice is able to delivery to them before they close up their shop.

Yes. The other benefits are may have the chance to continue for second trip, increase on customer satisfaction because customer is able to receive their inventories before they close up their shop. From the answer given by Mr. Tan we know that finding the shortest path to deliver can bring several benefits such as reduce fuel cost and travelling cost, increase customer satisfaction and able to increase company profit due to there has the chance to planned for second route trip. In conclusion, system which is able to calculate for the shortest route to deliver will brings benefit to those transportation and delivery services company.

Questions 7

If we are going to develop a system which are able to find the shortest route to deliver among your customer's location what suggestion or advise that you can provide?

- **time taken to calculate the shortest route must be short**
- **able to detect the daily traffic to avoid the congestion**
- **able to automatically select (D/Os) for calculation based on the deadline**
- **at least able to reduce the total distance to travel by 5 to 10%**
- **others (please list below)**

Mr. Tan select all of the features which has been listed on the questions and provide extra suggestions like the system should be able to clustering the location of the customer based on the area they stay. Besides, the system should be able to plan the route to deliver based on the working hour of their customers and it is good if the system is able to keep track of the particular truck and let the customers and company's manager to know where is the current location of the truck and when will the truck arrive the next stopping point. Due to time constraint several features may be included in this project while others like detect the daily traffic to avoid the congestion, plan the route based on the working hour of customers and keep track of a particular truck in order to increase the transparency of the route will be considered on the future plan. In conclusion, features like time taken to calculate the route should fall within a reasonable time, able to planned the route based on the D/O's expire date, reduce the total distance to travel by 5% to 10% compare to the one that planned by people based on their experience and clustering those locations based on the area their shop is located will be implemented in this web-based route optimization system to achieve the objective of this project.

Question 8

Do you face problem like yours worker are unable to fit one or two (D/Os) into the truck? If yes, what is the root cause of this?

Yes. The main cause are due to a wrong estimation on the total volume of the cargoes that can be fit into the particular truck and the orientation of the cargoes that loaded into the truck is unable to fully utilize the space of the truck.

Based on the respond given by Mr. Tan we are able to know that human estimation may give a wrong instruction when performing a task and human is unable to solve those NP-hard problem (bin packing problem in this situation) if we are just relied on our personal experience on performing a task. Human is unable to perform complex visualization and calculation due to limitation of memory and processing power of our brain. In conclusion, to optimize the available space in the truck, we may need to perform a complex calculation by comparing on different combination of the orientation of the cargoes in the truck and at the same time keep track on the total volume of the cargoes and volume of the truck, human brain is not able to solve this problem hence a software system with predefined rules will be used to replace human in performing the optimization task on cargoes loading process.

Question 9

As a manager, what will you do to address the problem in question 8?

First we will try to partially unload the cargoes then rearrange the orientation of the cargoes in the truck, if those inventories still could not fit into the truck then we will place the particular D/O to another available truck, if there is no other available truck then D/O which is not that urgent will be remove from the original route and place on the next day's schedule.

Based on the above responds we are able to know that the solution to solve the problem in question 8 is quite troublesome and it is ineffective due to the process may spent a lot of time to trial and error on different solutions given. How if we are able to know how many D/O can be fit into the truck and also how if we are able to know all the orientation of the placement of the cargoes in the truck before we start to load inventories into the vehicle. It is effective due to we are able to know whether inventories of the D/O in a particular route that had been planned can fit into the vehicle or not. In conclusion, a system which is able to handle the optimization of the available space of the truck is able to smooth and speed up the cargo loading process.

Question 10

Based on your experience, how many (D/Os) can be fix into the truck at one time.

Depend on the quantity of the cargoes in the D/O. Basically each truck will be fit with max 20 D/O.

Question 11

Based on your experience, how many percentage you will be given in term of utilization the space in the truck?

□ 51% - 60%
□ 61% - 70%
□ 71% - 80%
■ 81% - 90%
□ above 90%

Question 12

What factors that you will be consider in order to utilize the available space when loading inventories into the truck? For example:

- the order of the D/O to be sent

- volume of the inventories

- arrangement of the inventories

Besides from the above three factors which has been listed, Mr. Tan provided another three factors that may need to be consider which are the total weight of the cargoes that can be fit into the truck, the volume of the particular truck and the weight of the particular truck. In conclusion, those six factors will be consider in the algorithm in order to maximize the available space of the truck.

Question 13

Do you think that having system which are able to calculate and arrange the order of placement of inventories in order to fully utilize the space in the truck is helpful for you? If yes, how? For example:

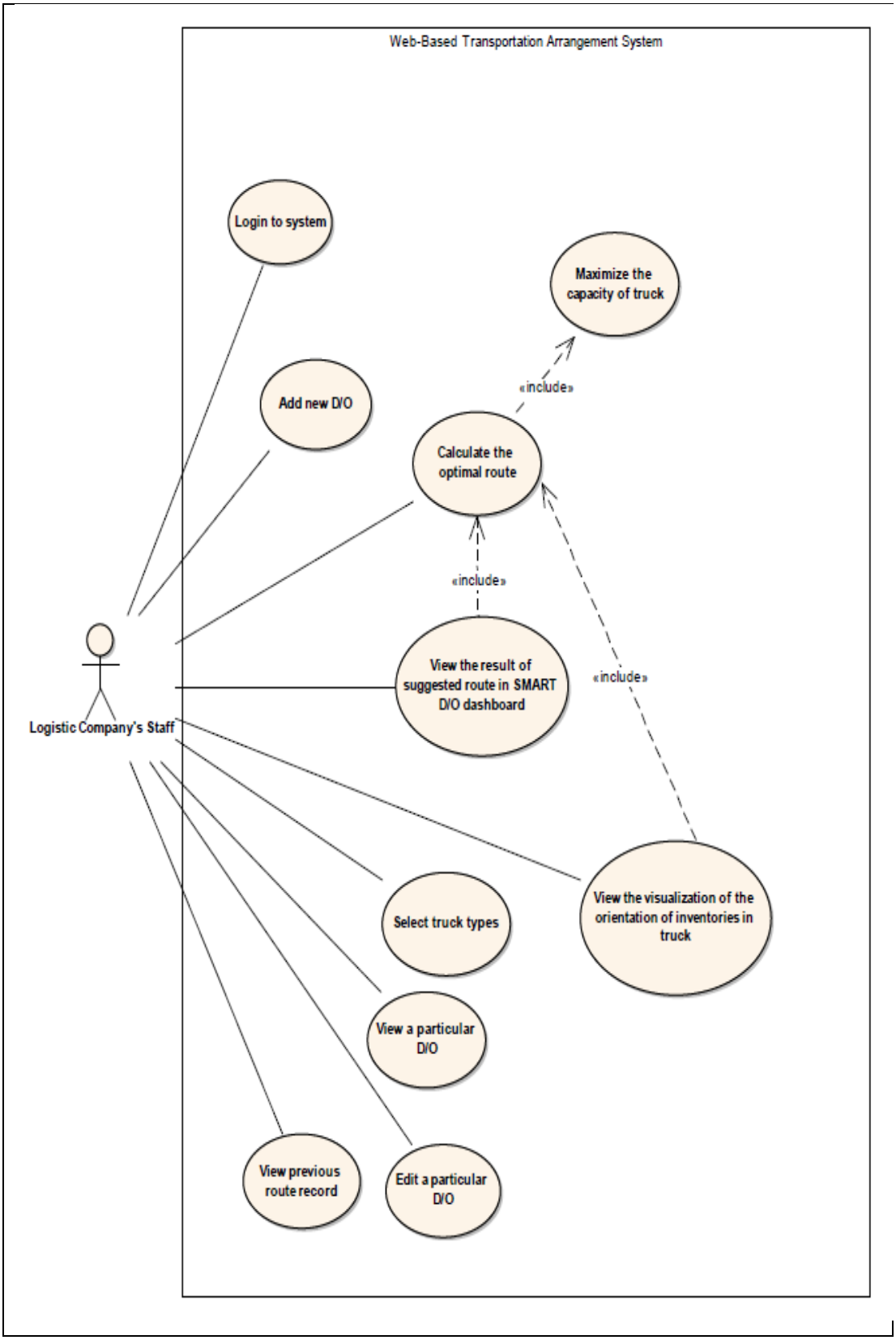
- able to smooth the overall process of loading inventories into the truck.

- minimize the overall number of trip to deliver

Yes. The system can help us to smooth the overall process of cargoes loading, minimize the overall number of trip, inexperience worker is able to load the cargoes into the truck by following the orientation of cargoes which has been calculated by the system hence they will not worry about whether the cargoes can be fit into the truck or not. Last, it is able to increase the profit per truck due to each truck is fully maximize. From the above respond, we can make a conclusion that system which is able to

maximize the available space in the truck will bring benefits to those shipment related company.

Appendix E: Use Case Diagram and Use Case Description before revised.



Use Case Name: Maximize the capacity of truck	**ID:** 004	**Importance Level:** High
Primary Actor: Logistic Company's Staff	**Use Case Type:** Detailed, Essential	
Stakeholders and Interests: Logistic Company's Staff – Staff wants to find the optimal route to deliver cargoes and at the same time maximize the capacity in the truck.		
Brief Description: This use case describe how company's staff maximize the capacity of the truck.		
Trigger: Staff click on the "Start Calculate" button which located at the Result tab bar.		
Relationships: **Association:** Logistic Company's Staff **Include:** - **Extend:** - **Generalization:** -		
Normal Flow of Events: 1. Perform use case "Calculate the optimal route". 2. System search for the combination of orientation of cargoes which is able to maximize the capacity of truck.		
SubFlows: Not Applicable.		
Alternate/Exceptional Flows: Not Applicable		

Use Case Name: View the visualization of the orientation of inventories in truck	**ID:** 006	**Importance Level:** High
Primary Actor: Logistic Company's Staff	**Use Case Type:** Detailed, Essential	
Stakeholders and Interests: Logistic Company's Staff – Staff wants to view the result of truck optimization.		
Brief Description: This use case describe how the system show the result of visualization of the orientation of inventories in the truck.		
Trigger: Staff click on the "Start Calculate" button on the "Result" tab.		
Relationships: **Association:** Logistic Company's Staff **Include:** Calculate the optimal route. **Extend:** - **Generalization:** -		
Normal Flow of Events: 1. Perform use case "Calculate the optimal route". 2. Click on the "truck" icon which located at the bottom right corner of the map. 3. System show the orientation of the cargoes in the truck.		
SubFlows: Not Applicable.		
Alternate/Exceptional Flows: Not Applicable		

Appendix F: Results of User Acceptance Testing

Name of Tester(s)	Tan Hooi Theam (Manager of Hondail Trading Sdn Bhd)			
Date Tested	20 April 2020			
Testing Start Time	3pm	**Testing End Time**	4pm	
Test Functionality	**Test Description**		**Status**	**Comments**
Manage D/O Information	Able to create a new D/O record.		Pass	-
	Able to add items detail into the D/O record.		Pass	-
	Able to display error when the field required to create the data is empty.		Pass	-
	Able to notify user that the particular D/O already exist when user enter the same D/O No.		Pass	-
	Able to notify user when the D/O record has been successfully created.		Pass	-
	Able to view a list of D/O record.		Pass	-
	Able to filter the list of D/O record.		Pass	-
	Able to view the detail of a particular D/O record.		Pass	-
	Able to edit a particular D/O record.		Pass	-
	Able to notify the user when the D/O record successfully updated.		Pass	-
	Able to delete a particular D/O record.		Pass	-
	Able to show the confirmation message before delete the D/O record.		Pass	-
	Able to notify user that the particular D/O record had been removed.		Pass	-
Sign-off by: **Name: Tan Hooi Theam** **Date: 20 April 2020**				

鋒泰貿易有限公司
HONDAIL TRADING SDN. BHD.
(755589-K)
228, Jalan Batu Maung,
11960 Bayan Lepas, Penang.
Tel: 04-6263239 Fax: 04-6264753

Name of Tester(s)	Tan Hooi Theam (Manager of Hondail Trading Sdn Bhd)			
Date Tested	20 April 2020			
Testing Start Time	3pm	**Testing End Time**	4pm	
Test Functionality	**Test Description**		**Status**	**Comments**
Manage Item Information	Able to create a new Item record.		Pass	-
	Able to display error when the field required to create the Item is empty.		Pass	-
	Able to notify user that the particular Item already exist when user enter the same Item Id.		Pass	-
	Able to notify user when the Item record has been successfully created.		Pass	-
	Able to view a list of Item record.		Pass	-
	Able to filter the list of Item record.		Pass	-
	Able to edit a particular Item record.		Pass	-
	Able to notify the user when the Item record successfully updated.		Pass	-
	Able to delete a particular Item record.		Pass	-
	Able to show the confirmation message before delete the Item record.		Pass	-
	Able to notify user that the particular Item record had been removed.		Pass	-

Sign-off by:

Name: Tan Hooi Theam

Date: 20 April 2020

鋒泰貿易有限公司
HONDAIL TRADING SDN. BHD.
(755589-K)
228, Jalan Batu Maung,
11960 Bayan Lepas, Penang.
Tel: 04-6263239 Fax: 04-6264753

Name of Tester(s)	Tan Hooi Theam (Manager of Hondail Trading Sdn Bhd)		
Date Tested	20 April 2020		
Testing Start Time	3pm	**Testing End Time**	4pm

Test Functionality	**Test Description**	**Status**	**Comments**
Manage Customer Information	Able to create a new Customer record.	Pass	-
	Able to display error when the field required to create the Customer is empty.	Pass	-
	Able to notify user that the particular Customer record already exist when user enter the same Customer Id.	Pass	-
	Able to notify user when the Customer record has been successfully created.	Pass	-
	Able to view a list of Customer record.	Pass	-
	Able to filter the list of Customer record.	Pass	-
	Able to edit a particular Customer record.	Pass	-
	Able to notify the user when the Customer record successfully updated.	Pass	-
	Able to delete a particular Customer record.	Pass	-
	Able to show the confirmation message before delete the Customer record.	Pass	-
	Able to notify user that the particular Customer record had been removed.	Pass	-

Sign-off by:

Name: Tan Hooi Theam
Date: 20 April 2020

鋒泰貿易有限公司
HONDAIL TRADING SDN. BHD.
(755589-K)
228, Jalan Batu Maung,
11960 Bayan Lepas, Penang.
Tel: 04-6263239 Fax: 04-6234703

Name of Tester(s)	Tan Hooi Theam (Manager of Hondail Trading Sdn Bhd)			
Date Tested	20 April 2020			
Testing Start Time	3pm	**Testing End Time**	4pm	
Test Functionality	**Test Description**		**Status**	**Comments**
Manage Cargo Space Information	Able to create a new Cargo Space record.		Pass	-
	Able to display error when the field required to create the Cargo Space is empty.		Pass	-
	Able to notify user that the particular Cargo Space record already exist when user enter the same Cargo Id.		Pass	-
	Able to notify user when the Cargo Space record has been successfully created.		Pass	-
	Able to view a list of Cargo Space record.		Pass	-
	Able to filter the list of Cargo Space record.		Pass	-
	Able to edit a particular Cargo Space record.		Pass	-
	Able to notify the user when the Cargo Space record successfully updated.		Pass	-
	Able to delete a particular Cargo Space record.		Pass	-
	Able to show the confirmation message before delete the Cargo Space record.		Pass	-
	Able to notify user that the particular Cargo Space record had been removed.		Pass	-

Sign-off by:

Name: Tan Hooi Theam

Date: 20 April 2020

鋒泰貿易有限公司
HONDAIL TRADING SDN. BHD.
(755589-K)
228, Jalan Batu Maung,
11960 Bayan Lepas, Penang.
Tel: 04-6263239 Fax: 04-6264753

Name of Tester(s)	Tan Hooi Theam (Manager of Hondail Trading Sdn Bhd)		
Date Tested	20 April 2020		
Testing Start Time	3pm	Testing End Time	4pm

Test Functionality	Test Description	Status	Comments
Import Data with the Excel File (.csv)	Able to create a list of Item records at one time.	Pass	-
	Able to create a list of Customer records at one time.	Pass	-
	Able to create a list of D/O records at one time.	Pass	-
	Able to notify user when those records had been successfully created.	Pass	-
	Able to display error when there are missing field in the excel file.	Pass	-
	Able to display error when user import the wrong excel file.	Pass	-
	Able to display error when the data type is invalid.	Pass	-

Sign-off by:

Name: Tan Hooi Theam

Date: 20 April 2020

鋒泰貿易有限公司
HONDAIL TRADING SDN. BHD.
(755589-K)
228, Jalan Batu Maung,
11960 Bayan Lepas, Penang.
Tel: 04-6263239 Fax: 04-6264753

Name of Tester(s)	Tan Hooi Theam (Manager of Hondail Trading Sdn Bhd)			
Date Tested	20 April 2020			
Testing Start Time	3pm	Testing End Time	4pm	
Test Functionality	**Test Description**		**Status**	**Comments**
Route Setting Process	Able to display vehicle on the "Vehicle Selection" section after user change the status of the vehicle to become a tick icon which located at the "Cargoes Space" screen.		Pass	Hard to understand at when first time login. Prefer to have a guideline on this feature or re-design the interface in a more user friendly method.
	Able to select the vehicle for calculation purpose.		Pass	-
	Able to display error when the field required for the Route Setting is empty.		Pass	-
	Able to save the Route Settings data.		Pass	-
	Able to notify user when the data has been successfully save.		Pass	-

Sign-off by:

Name: Tan Hooi Theam

Date: 20 April 2020

鋒泰貿易有限公司
HONDAIL TRADING SDN. BHD.
(755589-K)
228, Jalan Batu Maung,
11960 Bayan Lepas, Penang.
Tel: 04-6263239 Fax: 04-6264753

Name of Tester(s)	Tan Hooi Theam (Manager of Hondail Trading Sdn Bhd)		
Date Tested	20 April 2020		
Testing Start Time	3pm	**Testing End Time**	4pm

Test Functionality	**Test Description**	**Status**	**Comments**
Route Calculation process	Able to display list of D/Os which status are still "Pending".	Pass	-
	Able to filter the list of D/Os after a particular date had been selected.	Pass	-
	Able to remove a particular D/O from the list of D/Os	Pass	-
	Able to prohibit the calculation process and notify user when there are no "Pending" D/O record.	Pass	-
	Able to prohibit the calculation process and notify user when there are no truck selected.	Pass	-
	Able to combine all the D/O which have the same address.	Pass	-
	Able to cluster those addresses into several groups which corresponding to the number of truck selected.	Pass	Not wise enough to evenly cluster those locations.
	Able to calculate the shortest route to deliver for each cluster.	Pass	-
	Able to display the result of the calculated route for each cluster on the "Smart D/O" tab.	Pass	-
	Able to visualize the calculated route with the sequence to deliver on the map for each cluster.	Pass	-
	Able to displayed those removed D/O	Pass	- Prefer to add on severity level for each D/O. - Not able to add back the removed D/O into the particular routes. (prefer to have this feature in the future)

Sign-off by:

Name: Tan Hooi Theam

Date: 20 April 2020

鋒泰貿易有限公司
HONDAIL TRADING SDN. BHD.
(755589-K)
228, Jalan Batu Maung,
11960 Bayan Lepas, Penang.
Tel: 04-6263239 Fax: 04-6264753

Name of Tester(s)	Tan Hooi Theam (Manager of Hondail Trading Sdn Bhd)		
Date Tested	20 April 2020		
Testing Start Time	3pm	**Testing End Time**	4pm

Test Functionality	**Test Description**	**Status**	**Comments**
Manage Previous Planned Route	Able to save the calculated route record.	Pass	-
	Able to view a list of previous calculated route record.	Pass	-
	Able to filter the list of calculated route record after a particular date had been selected.	Pass	-
	Able to view the detail of a particular previous calculated route record.	Pass	-
	Able to delete a particular previous calculated route record.	Pass	-
	Able to show the confirmation message before delete a particular previous calculated route record.	Pass	-
	Able to notify user that the particular previous calculated route record had been removed.	Pass	-

Sign-off by:

Name: Tan Hooi Theam

Date: 20 April 2020

鋒泰貿易有限公司
HONDAIL TRADING SDN. BHD.
(755589-K)
228, Jalan Batu Maung,
11960 Bayan Lepas, Penang.
Tel: 04-6263239 Fax: 04-6264753

Name of Tester(s)	Tan Hooi Theam (Manager of Hondail Trading Sdn Bhd)		
Date Tested	20 April 2020		
Testing Start Time	3pm	**Testing End Time**	4pm

Test Functionality	**Test Description**	**Status**	**Comments**
Manage Staff Information	Able to create a new Staff to access the system and also the company's information.	Pass	-
	Able to display error when the field required to create the Staff is empty.	Pass	-
	Able to notify user that the particular account for the staff already exist when user enter the same staff id.	Pass	-
	Able to notify user when the staff's account has been successfully created.	Pass	-
	Able to view a list of Staff record.	Pass	-
	Able to filter the list of Staff record.	Pass	-
	Able to change the status of a particular staff.	Pass	-

Sign-off by:

Name: Tan Hooi Theam

Date: 20 April 2020

鋒泰貿易有限公司
HONDAIL TRADING SDN. BHD.
(755589-K)
228, Jalan Batu Maung,
11960 Bayan Lepas, Penang.
Tel: 04-6263239 Fax: 04-6264753