EMPLOYEE ONSITE TASK ASSIGNMENT MANAGEMENT AND TRACKING APP

YEOH SHIN YUN

UNIVERSITI TUNKU ABDUL RAHMAN

EMPLOYEE ONSITE TASK ASSIGNMENT MANAGEMENT AND TRACKING APP

YEOH SHIN YUN

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

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

September 2023

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 : YEOH SHIN YUN

ID No. : 19UEB06581

Date : 08/09/2023

APPROVAL FOR SUBMISSION

I certify that this project report entitled **"EMPLOYEE ONSITE TASK ASSIGNMENT MANAGEMENT AND TRACKING APP"** was prepared by **YEOH SHIN YUN** has met the required standard for submission in partial fulfilment of the requirements for the award of Bachelor of Science (Honours) Software Engineering at Universiti Tunku Abdul Rahman.

Approved by,

Signature : D.gunavathi

Supervisor : Ms. Gunavathi a/p Duraisamy

Date : 06 Oct 2023

ACKNOWLEDGEMENTS

I would like to thank everyone who contributed to the successful completion of this project. I would like to thank my research supervisor, Ms Gunavathi a/p Duraisamy for her invaluable advice, guidance and enormous patience throughout the research development.

In addition, I would like to thank my family for their unwavering support and encouragement. Their confidence in my abilities has been a constant source of inspiration, and I am incredibly fortunate to have them by my side.

Lastly, I would like to acknowledge the countless others who have contributed in their own ways, whether large or minor, to the success of this endeavour. Your assistance and confidence in this endeavour have not gone unnoticed.

This accomplishment would not have been possible without the collective effort and support, and I sincerely appreciate your contributions.

ABSTRACT

The modern workplace faces challenges concerning employee retention, productivity, and engagement. A need has been identified to address these issues through a structured task assignment and monitoring solution. Comprehensive research was undertaken, encompassing the study of relevant articles on employee task assignment and monitoring. Insights were further enriched through surveys and consultations with experts to understand the workplace challenges employers and employees encounter. This research resulted in creating an application comprising three pivotal modules: task assignment and monitoring, location tracking and reporting. These modules empower employers with the capability to allocate tasks, oversee employee progress, and trace their onsite location, along with facilitating report generation. The choice of Laravel and Flutter for the application development was grounded in their renowned attributes: ease of use, efficiency, and adaptability across platforms. The envisaged impact of this application is a significant uplift in productivity, engagement, and employee retention, fostering a conducive and healthy work culture. The devised application using Laravel and Flutter is a potent solution to the prevailing workplace challenges of employee task management and monitoring. By facilitating task assignment, progress tracking, and location reporting, this tool redefines workplace culture, bolstering productivity and engagement to the mutual advantage of both employees and employers.

TABLE OF CONTENTS

CHAPTER

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS / ABBREVIATIONS

[This Section is Blank – Not Available]

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 General Introduction

This chapter intends to undertake a study on the task administration and supervision of onsite employees using a mobile application. This study's objective is to identify the prevalent practises and challenges in the field. The primary purpose of this study is to analyse and propose the optimal solution to these problems, as well as to assess the potential for development that can be realised as a result of this research.

The proposed plan will describe the project's scope as well as the solution that will be provided to resolve the identified problems. This plan will be presented in a clear and concise manner, serving as a road map for the remaining project report chapters. At the conclusion of this chapter, the reader will have a thorough comprehension of the project's goals and the approach that will be taken to attain them.

1.2 Project Background

According to Collins English Dictionary, onsite refers to the existance of someone or something in a particular location where people work, study, or reside. Onsite employees must be present at one specific spot as an illustration, a construction employee must get to the job site, and a part-time home cleaner moves to the client's location to clean the property.

In the current era of technology, various task management instruments have emerged to aid in labour management productivity. Therefore, it is recommended that organisations implement such systems in the hope of boosting employee productivity related to conventional task assignment management techniques. Using a task management system, supervisors may optimise the organisation's prioritising of activities, enhance collaboration among teams, and ultimately enable their staff members to operate at the most efficient and effective levels (Brikoshina et al., 1970). In the present state of management of tasks solutions, whereas there are several options accessible to address the management of tasks, relatively few systems pander to the particular job management and tracking requirements of onsite employees.

According to The Star, the Malaysian government's administration of the development of the nation's government capital has resulted in losses eclipsing RM800 million, predominantly due to the delay of its monorail project. Furthermore, it is pertinent to point out that as of the 31st of January in 2023, the Dewan Rakyat has announced that there are 429 "sick" housing developments involving 29,147 consumers nationwide. For unfamiliar people, "sick" housing projects have encountered more than 30% delays beyond their scheduled progress or those whose purchase and sale agreements (SPAs) are no longer valid. According to a study titled "Principal Factors Causing Delays in Large Construction Projects" issued by SSRN, experienced project managers and site supervisors who had worked on large construction projects for a minimum of ten years generally thought that delays of projects are triggered by "lack of timely delivery of design documents, late instructions, and undetermined and insufficient details on drawings." In summary, the broad issues that have been identified include misaligned objectives, tardiness, and others, but they are all due to the performance of team members.

In today's highly technological era, countless task management and tracking solutions have been produced to assist supervisors; however, there needs to be more systems created to assist onsite employees in handling and tracking their tasks. Although the general requirements for developing software to manage tasks may appear comparable for various work settings, the unique aspects of onsite work demanded a more comprehensive and particular approach to preparing and developing such a system. In contrast to the office or remote employees, employees onsite frequently require immediate form access to geographic location information, job prioritisation dependent on physical proximity, and extensive monitoring and reporting capabilities to guarantee that they comply with security regulations and performance requirements.

Consequently, an in-depth requirements assessment and collaboration with subject matter experts are required to create an efficient onsite software task management solution that suits this work context's particular goals and limitations.

The primary difference between workplace-based and onsite employees is that office-based employees are limited to carrying out their responsibilities within the confines of the workplace and their immediate environs. In contrast, staff members onsite are characterised by flexibility and often travel to various places to fulfil their responsibilities. In the case of onsite employees, the necessary equipment for completing the project may be prepared by the employees, whereas this possibility is significantly diminished for office-based employees.

1.3 Problem Statement

In this sub-chapter, the problems faced by both employers and employees is discussed. It is important to acknowledge that workplace issues can arise from both perspectives, and it's essential to address them.

Retention, productivity, and engagement are employer concerns. High turnover rates raise recruitment, hiring, and training expenses for firms. Poor productivity hurts a company's profits. Creating a healthy workplace culture requires strong employee engagement, which managers may struggle to sustain.

Yet, job unhappiness, burnout, and work-life balance difficulties may also occur. Lack of recognition, management, or advancement chances may lead to job unhappiness. Overwork and stress may cause burnout. Employees may struggle to manage their personal and work lives, causing stress and lower productivity.

1.3.1 Supervisors’ view

1.3.1.1 Lack of proper communication

As highlighted in Orechwa's (2021) article "Closing the communication gap with Field Workers," poor communication is among the most significant challenges in project management. At times, a project's planning stage requires multiple professionals' cooperation, requiring efficient communication of information for project success. However, interaction difficulties due to a divide between workplace and onsite employees are common, resulting in task-associated problems such as delays in meeting deadlines, ignorance of the amount of time needed to complete a task, and omission of essential tasks. In addition, teams may only recognise problems once it is too late.

Smartphone is one of the more common means for field employees and office workers to communicate. Therefore, onsite employees rely on cell phones to stay in touch with their supervisors, coworkers, and others. The telephone is a dependable and convenient method of communication, but it has limitations. Due to hectic schedules, a phone call can go unanswered, for instance. In addition, phone conversations require a visual component, preventing onsite employees from providing administrators with real-time information. In addition, phone conversations may occasionally require interpretation, leading to misunderstandings and incorrect assumptions.

As depicted in Figure 4.11, the survey titled "On-site Task Assignment and Monitoring Software: Opportunities and Challenges" reveals that the majority of task assignment occurs via messaging platforms. This approach may, however, result in ineffective communication, which has been identified by 45 percent of administrators as one of their most significant challenges. Messaging platforms are ineffectual as an all-inclusive solution for project management this is mainly attributed to the overwhelming amount of messages transmitted through multiple channels, which messaging platforms may not have the necessary tools to manage (O'Neill, 2018). Communication is an essential component of project administration. It ensures that team members agree on project objectives and expectations, facilitating a seamless and timely project completion (Joubert, 2021). Additionally, it promotes cooperation and trust, resulting in enhanced collaboration from project inception to completion.

1.3.1.2 Inadequate mechanism to accommodate real-time modifications.

The nature of a project is temporary and results in a new product or subject, not just limited to construction projects. Changes are inevitable, whether due to new designs or changed requirements (Hwang & Low, 2012). Additionally, real-time situations are unpredictable, such as when an employee falls sick or the weather turns rainy, causing drivers to slow down to avoid accidents. Such factors cannot be controlled since they originate externally to the organization. As a result, changes need to be accepted and adapted immediately to prevent delays.

To prevent delays, decisions must be made promptly without impacting progress. However, the current systems in place lack mechanisms to assist supervisors in providing accurate implementations, which can be challenging. Furthermore, supervisors may have to manage multiple projects simultaneously, leading to complex thinking and task rescheduling. Multitasking to handle several projects or tasks at the same time can result in lower productivity and errors that may incur costs for the company (Dey, 2021).

1.3.1.3 Poor Task Scheduling.

According to Stoop and Wiers (1996), the launch of these systems has uncovered multiple problems due to the complexity of scheduling. The scheduler frequently tweaks or deviates from these techniques-based schedules on the shop floor. Task assignment and scheduling are vital parts of project management. In order to complete a project on time, tasks have to be given to team workers with the required skills and knowledge and can be finished within the given time frame. Scheduling a project involves dividing it into more minor duties or sub-projects, analysing their interrelationships, and determining the time required to complete each task. This data is then utilised for creating a schedule for the project that outlines the order in which the tasks will be completed and the dates when they will commence and conclude.

Each team member is allotted specific duties or subprojects based on his or her skills, availability, and current burden. The process may be complicated, as it requires knowledge of every team member's assets and weaknesses and their current burden and availability. The assignment of work to an inappropriate person on a team can result in pauses, errors, and reduced efficiency.

1.3.2 Employees’ view

1.3.2.1 Overload of tasks assigned

According to research conducted by Kuroda & Yamamoto, overworking can have negative effects on an individual's mental health and increase the resignation rate of employees. This can have a significant impact on a company as they may lose valuable employees due to overworking. In some extreme cases, overworking can even result in the loss of life, as seen in the case mentioned by The Star which resulted in law investigations and compensations.

Overworking has always been a prevalent issue in the workplace, and it is important for employers to monitor their employees and assign tasks appropriately to balance their workload. This is crucial in ensuring a healthy work environment and maintaining the well-being of the employees, who are the assets of the company. The situation is particularly critical in sectors like the health industry, where there is a high demand for staff, but few employees are working long shifts that exceed an appropriate amount of working hours. This can lead to burnout and affect the quality of service provided (Arumugam, 2023). As seen in The Star's article, the community is holding the government accountable for this issue.

These issues are not only applicable to the health industry but can be seen in other companies that have not considered the consequences of overworking their employees. It is crucial for companies to prioritize their employees' health and well-being to prevent burnout and maintain a healthy and productive work environment.

1.3.2.2 Inefficient in handling project with complex request.

The nature of project briefings, which frequently contain a great deal of information, can present difficulties for those designated with administering the project. Typically, this information is shared during meetings or via communications platforms, making it difficult for employees to retain all the specifics. According to De Jong's article, it is more difficult to recollect materials with a large number of elements than those with fewer elements and negligible interactivity. This difficulty in recall can be especially pronounced when working on complex undertakings requiring consideration of numerous elements.

Due to the overwhelming quantity of information and duties that must be managed simultaneously, managing projects with multiple requests can be particularly difficult and may result in neglect. This issue is highlighted by a survey in which 30% of respondents reported experiencing it. Overall, it is evident that the complexity and information density of many projects can make it difficult for employees to retain and manage all the necessary details, which can have an adverse effect on project outcomes.

1.4 Project Goals

To design and develop a mobile application that can be used by supervisors and employees for onsite task management and assignment.

- To analyse the issues and obstacles encountered by user in managing and tracking onsite employees.
- To create a mobile application with the following features:
    - To create an onsite task assignment that documents the location, project details, project duration, and personnel requirements.
    - To readily assign employees to projects and to be able to reassign tasks at any time.

- To monitor the employee's daily location via GPS and record their check-in and check-out information.
- To monitor the task completion status and display it on a dashboard with graphs/pie charts.

1.5 Project Solution

As a potential remedy to the identified issues in this project, the creation of an Android mobile application was meticulously considered. The application's primary objective is to address the problem of employee monitoring and reduce project delays in onsite industries. In industries such as construction, where employees are frequently on the move, it can take time to maintain track of them. This issue can result in squandering time and resources that could have been allocated to more important duties.

As a potential remedy to the identified issues in this project, the creation of an Android mobile application was meticulously considered and developed. The primary objective of the application was to address the problem of employee monitoring and reduce project delays in onsite industries. In industries such as construction, where employees were frequently on the move, it often took time to keep track of them. This issue often resulted in squandering time and resources that could have been allocated to more crucial duties.

Figure 1.1: Android Logo. (Deniz, 2022)

To develop the onsite employee management and monitoring application accessible to the most significant number of users, it was developed for the Android platform. This choice was based on the affordability of Android phones compared to IOS phones, allowing most onsite employees to access the application without incurring additional costs. This approach not only made the application readily available but also ensured the project's budget was utilised effectively. By developing the application for Android, development is concentrated on delivering a streamlined application with all the required features. This application allowed supervisors to monitor employees' movements, assign duties, and measure their real-time progress. Additionally, it facilitated the administration of projects by providing a central repository for all project-related information.

Figure 1.2: Flutter widgets. (Suri, 2021)

The onsite work management and monitoring mobile app will be developed using Flutter for excellent reason. Flutter, a popular Google mobile app development framework, lets developers construct high-performance, aesthetically attractive, and natively built applications for Android and iOS with a single codebase. With customisable widgets and pre-built animations, Flutter lets developers build stunning, responsive user interfaces. Flutter's speed, flexibility, and simplicity of use make it ideal for constructing an onsite work management and monitoring mobile app.

Figure 1.3: What is Laravel? (Barnes, 2021)

For the backend development of this project, the Laravel framework is used. This allows the project to take advantage of Laravel's many powerful features and its efficient MVC architecture. The utilization of JWT, also known as JSON Web Tokens, is a significant step in improving the integration and safety features of our program. JSON Web Token (JWT) is an authentication system that offers both security and consistency. Not only does this ensure that our application has a strengthened security layer, but it also simplifies the process of integrating frontend technologies, most notably Flutter. Our implementation of JWT streamlines this process, which ensures a seamless and safe connection between our Laravel backend and the Flutter frontend. This is necessary in light of the fact that Flutter requires authentication methods that are both effective and secure. In addition, our backend is compatible with RESTful APIs, which are naturally adaptable with Flutter's HTTP package and so make it possible for users to exchange data in real time. The fact that Laravel is built on a modular architecture along with its support for stateful authentication and middleware further enhances the simplicity of integration, which in turn ensures that the Flutter frontend provides the best possible responsiveness and user experience.

In conclusion, developing an Android mobile application to address the problem of employee monitoring proved to be an innovative solution that significantly enhanced the efficiency of onsite operations. Choosing to develop the Android application ensured that a majority of employees had access to it while allowing the project team to focus on delivering a high-quality and robust application.

1.6 Project Approach

Rapid Application Development, an incremental paradigm, was selected as the development method for this project. The paradigm is a well-known approach to the software development life cycle (SDLC) that breaks a project into smaller and more manageable increments. The capability to practice incremental which breaks down the development into few versions is advantageous which promotes fast development but at the same time ensures that the application will meet up to the base requirements. This methodology produces an independent project that can be built upon the previous increment, resulting in an improved version with each delivery. It is known for its rapid approach to software development, which emphasises continuous improvement, quick prototyping, and tight collaboration among stakeholders, which Miss Gunavathi, the supervisor.

Through continual development cycles, RAD prioritises promptly producing functional software. Despite a tight development schedule for a final year project, this allows the developer to produce helpful software early on. In addition, RAD facilitates interaction among developers and stakeholders, which can help ensure that the final product meets the objectives and expectations of the stakeholder. (Chien, 2020).

The phased approach utilised by RAD can benefit a one-person group of developers, as it enables the setting of distinct, attainable objectives for every stage of the project (Egeonu, 2022). The phased method can also aid in identifying and resolving any challenges or roadblocks that may arise during the development phase (Chien, 2020).

Overall, the RAD methodology could be a good fit for one programmer engaged in a final-year project, especially when time is limited and close collaboration with stakeholders is required. By emphasising rapid prototyping, continual improvement, and cooperation, RAD can assist the programmer in producing a viable product quickly while ensuring that it meets the requirements and expectations of the stakeholders.

1.7 Project Scope

1.7.1 Target Users

The target audience for this project includes both current and former employees.

The primary focus is on individuals who have experience in supervising onsite employees and are responsible for managing tasks and assigning them to specific employees. The goal of the project is to provide an efficient and effective way for supervisors to manage their employees and tasks.

Apart from the supervisors, onsite employees are also important users who play a crucial role in the success of the project. These employees have a direct impact on the project and are therefore a key focus of the initiative.

1.7.2 Modules

1.7.2.1 Task assignment and monitoring module

The administrative superintendent is the central figure in the module for task management. As an administrator, he or she can create new assignments and enter pertinent information, such as due dates and required locations. Once the task has been generated appropriately, the supervisor can assign it to the appropriate personnel. Workers can then provide regular updates on the job's progress, including status updates and any required subtask updates. In addition, employees can estimate the time required to complete a task and adjust these estimates as necessary. This simplified procedure facilitates effective communication and collaboration between supervisors and staff, resulting in increased productivity and timely completion of assignments.

1.7.2.2 Location tracking module

The location monitoring module is a purpose-built device that permits businesses to monitor onsite personnel precisely. Historically, managers have had limited visibility into their employees' travels, but this module was created to address this issue. The module autonomously captures an employee's position when they log in at the beginning of their shift. When they arrive at a client's location, the module frequently logs their location and notifies the supervisor if they deviate significantly from the predicted location. In addition, the administrator can monitor an employee's location at any time during their shift, providing real-time visibility into their movements. This module automatically stops monitoring employees' location when they log out or take a lunch break, protecting their privacy and ensuring compliance with data protection regulations.

1.7.2.2.1 Reporting module

A reporting module is designed to generate reports based on data collected or generated. Employees could view metrics such as the number of daily completed tasks, critical tasks, the number of available employees, the number of active employees, the number of employees on vacation, and the total number of employees. Typically, the reporting module would have an interface that allows users to select the data to include in the report, configure the report's structure and layout, and generate the report. Once the report parameters have been specified, the reporting module would use a report-generating engine to create the report, possibly using charts, graphs, tables, and other visual aids to display the data. The reports generated by the reporting module can be used for various purposes, such as performance monitoring and business intelligence. In addition, the reporting module may include options for exporting reports to various formats, such as PDF and for scheduling the automatic creation of reports at predetermined intervals.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter aims to explore the studies and findings that have been conducted to develop a solution that fits the project's objectives. It encompasses various aspects such as the methodology adopted to carry out the project, algorithms to solve employee selection, and existing similar systems or solutions developed to address the issue.

The first part of the chapter focuses on the methodologies used in software development, including RAD phased development, Scrum agile, waterfall, and RAD throwaway prototype. Each methodology has been evaluated based on their strengths and limitations.

The second part of the chapter also chapter also delves into the existing systems such as Trello, Monday, and any.do, by comparing their features and capabilities. This section provides a deeper understanding of how these systems operate and their potential applications.

Lastly, the sub-chapter discusses algorithms that solve the issue of employee selection for specific tasks. This section compares and contrasts the greedy algorithm, generic algorithm, and Constraint Satisfaction Backtracking algorithm. By evaluating each algorithm's strengths and limitations, the most appropriate algorithm for the project can be easily identified.

The first paragraph in a subsection should align with the left margin. General alignment for texts in a paragraph should be “justified” (Warner, 2002).

2.2 Development methodologies

Software Development Life Cycle (SDLC), is a process that aims to develop and deliver a system that meets a company's requirements to its consumers.

While there are various software development methodologies, all of them involve four fundamental activities: specification, design and implementation, validation, and evolution. SDLC, in particular, comprises four primary phases: planning, analysis, design, and implementation(Altvater, 2023) . The planning phase includes project initiation, system request creation, and feasibility analysis, as well as project administration, work plan creation, personnel hiring, and project control. The analysis phase involves developing an analysis strategy, gathering requirements, and creating a system proposal. The design phase consists of creating a design strategy, developing the structure and screens, establishing databases and file requirements, and designing the software. Finally, the implementation phase entails building and setting up the system, providing training for customers, establishing a support strategy, and developing a new system and maintenance plan.

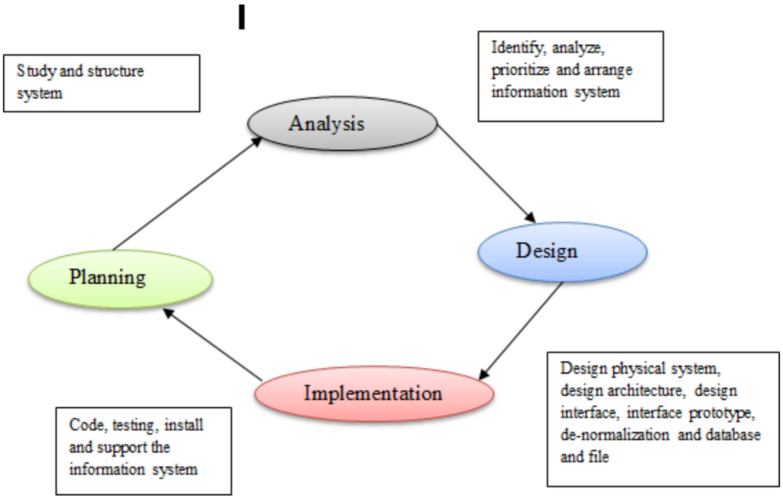

I
Study and structure system
Identify, analyze, prioritize and arrange information system
Analysis
Design
Planning
Implementation
Design physical system, design architecture, design interface, interface prototype, de-normalization and database and file
Code, testing, install and support the information system

Figure 2.1: SDLC Process. (Abubakar Shema)

These phases are crucial for the proper creation and implementation of software systems, ensuring that they meet the company's requirements and deliver value to its consumers.

2.2.1 Waterfall Methodology

The waterfall methodology is a sequential software development method in which every part is developed in a predetermined sequence. The phases within the waterfall paradigm are analysis, design, implementation, testing, and maintenance. During the Analysis stage, the project's viability is assessed, and its demands are outlined. In the Design phase, an operational concept is put together based on the decided requirements, responsibilities, and plans from the Analysis phase. In the Implementation phase, the software architecture devised in the Design phase is implemented by employing programming software, troubleshooting, and module testing. The application has been incorporated into the target system and assessed according to acceptance parameters defined in the Analysis stages. During maintenance, the software is made accessible for productive use, and any necessary improvements are implemented. This model focuses on recording the process steps; gained expertise is recorded in prerequisite or draught documents.

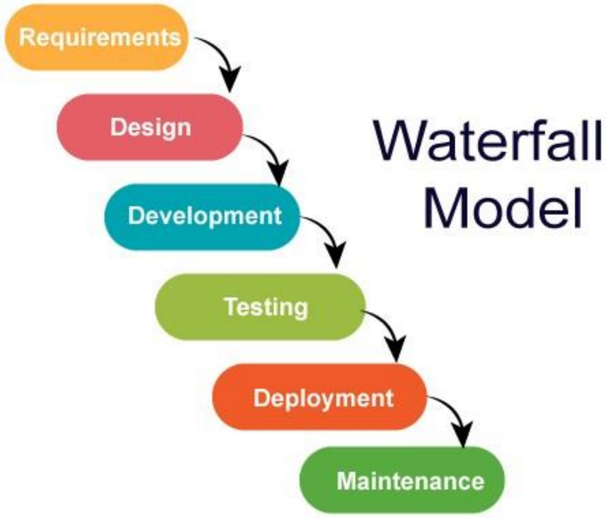


Figure 2.2: Waterfall Model. (Jayathilaka, 2020)

This model's distinct organisational structure is one of its advantages. Each project phase is distinct from the others, and the development process is straightforward to comprehend because it concludes with an intermediate result. This model also records acquired knowledge by documenting every process step (Lutkevich & Lewis, 2022). Theoretically, the waterfall model permits rapid and cost-effective project implementation through meticulous planning. However, this benefit is frequently contested because the model's rigorous linear sequence does not correspond to real-world conditions. In addition, large software development projects may be delayed long, and the results may become obsolete without regular adaptation to innovations.

The model's limitations can be regarded as a significant disadvantage. The model's linear sequence typically does not correspond to real-world situations, especially in complex software development initiatives where various application components may be in different development phases concurrently. Success under the cascade model may also necessitate a significant investment of time and resources in analysis and conception during the initial phases of the project (KUMAR PAL, 2023) . As no modifications to the waterfall model are intended, it may be inappropriate for projects where requirements and procedures are not precisely defined during the planning phase. In addition, the paradigm may not be appropriate for complex, multi-layered software projects because its sequential phases need to reflect reality.

The waterfall paradigm provides a transparent structure and documentation for software development. However, the rigid and linear sequence of the paradigm could be problematic for complex software development initiatives. This constraint led to the developing of alternative software development methods that provide greater flexibility. Before selecting a development methodology, it is essential to consider each software development project's unique conditions and requirements.

2.2.2 Rapid Application Development – Throwaway Methodology

Throwaway prototyping is a development strategy involving the construction of initial prototypes that are not intended to be incorporated into the final system. The prototypes are developed to collect feedback, demonstrate concepts, or conduct research. The primary benefit of this strategy is that it can expedite the development and reduce costs since prototypes can be deployment-ready (Martinez, 2023). This eliminates the need for related duties, freeing up resources to iterate through a greater diversity of prototypes. The intention is not to discard the prototypes but rather to construct them with the presumption that they will be discarded. The prototypes can be reviewed routinely with clients or utilised for concept validation or innovative problem-solving in internal tools or R&D contexts.

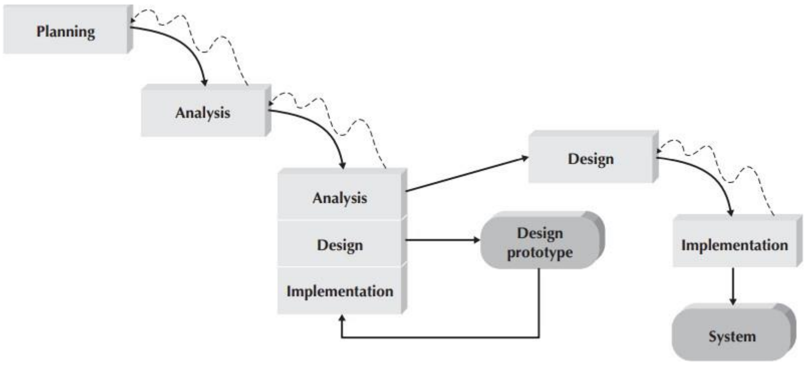


Figure 2.3: Throwaway Prototyping Model. (PadaKuu, 2023)

A significant benefit of disposable prototyping is that it enables a more iterative and adaptable development process. Developers can rapidly identify and address issues by developing multiple prototypes with various features and functionalities and refining their concepts (McQuillan, 2022). This approach is advantageous when requirements are only partially understood or technical obstacles must be overcome. In addition, discarded prototyping enables a more collaborative development process because prototypes can be shared with clients or stakeholders for feedback and input.

There are, however, disadvantages to disposable prototyping. The prototypes may need to accurately represent the final system, resulting in misunderstandings or misaligned expectations between developers and stakeholders. In addition, the emphasis on creating prototypes may divert resources from other essential duties, such as testing and documentation. Unmanaged, the requirement to continuously create and improve prototypes can result in scope expansion or project delays.

In conclusion, discarded prototyping is a development strategy involving the construction of initial prototypes that are not intended to be incorporated into the final system. This strategy can expedite development, permit iterative and flexible development processes, and facilitate stakeholder collaboration. However, if not properly managed, it can lead to misunderstandings, divert resources from essential duties, and result in scope expansion or project delays.

2.2.3 Agile Scrum Methodology

Scrum is a flexible and well-known methodology that rewards the implementation of agile principles within a context agreed upon by the entire team. It is a software development process founded on a defined set of practices and responsibilities that must be implemented during the development process. Sprints are temporary, periodic, and brief intervals in which the Scrum process is executed. Each iteration yields a complete outcome, a variation of the final product that can be delivered to the client with minimal effort upon request. The procedure begins with a list of the project plan's objectives and requirements. Prioritisation of these objectives by the customer determines iterations and deliveries. Scrum methodology is primarily used for software development, but other sectors are implementing it in their organisational models, including sales, marketing, human resources teams, etc (Peek, 2023).

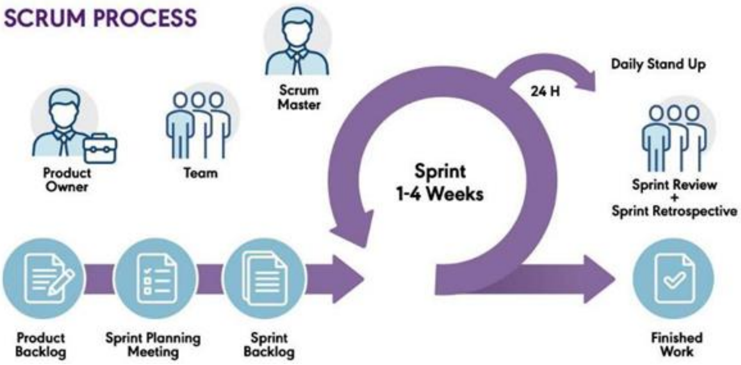


Figure 2.4: Agile Scrum Model. (PadaKuu, 2023)

Scrum is carried out by a team composed of three roles: Product Owner, Scrum Master, and Scrum Team. The Product Owner is accountable for the business aspect and represents the software's stakeholders and users . The Scrum Master directs the team to adhere to the rules and procedures of the methodology and administers the elimination of project roadblocks. The Scrum Team comprises professionals with the requisite technical expertise collaborating to develop the project (Atlassian).

Scrum events allow for modifying certain aspects of the process, the product, the progress, or the relationships. Sprint is the fundamental unit of work for a Scrum team, and Sprint Planning defines what will be accomplished and how it will be accomplished. Daily Scrum assesses progress, synchronises activities, and generates a 24-hour plan. Sprint Review indicates what work has been completed concerning the product backlog for future deliveries, whereas Sprint Retrospective serves to implement development process enhancements.

Scrum helps teams complete project deliverables quickly and effectively, ensures effective use of time and money, divides large projects into manageable sprints, works well for fast-moving development projects, and provides clear visibility to the team through Scrum meetings. Scrum is agile, so it incorporates customer and stakeholder feedback, and quick iterations make implementing changes based on feedback easier. During daily Scrum meetings, the individual effort of each team member is readily apparent. Scrum has some disadvantages, however. It frequently results in incomplete documentation, and the daily Scrum meeting can become ineffective. Moreover, a self-managing team can occasionally leave management feeling out of control.

Scrum is an agile methodology that assists teams in delivering project deliverables swiftly and effectively. Scrum facilitates efficient use of time and money, is adaptable and can be implemented in various industries, and is effective for rapid development initiatives. A crew of professionals with essential technical knowledge executes the methodology and develops the project collaboratively. Some disadvantages of the methodology include incomplete documentation and unproductive daily Scrum meetings. Scrum remains a popular methodology because it delivers fast and effective results while integrating customer and stakeholder feedback.

2.2.4 Rapid Application Development – Phased Methodology

Rapid Application Development (RAD) phased development is a software development strategy that emphasises rapid prototyping and incremental software delivery. The method involves dividing a project into phases, with each phase focusing on a distinct set of features or requirements. Typically, the process begins with a planning phase in which the requirements are identified, followed by a design phase in which the architecture and components of the software are defined. The subsequent phase is construction, where the actual coding occurs. Ultimately, the software's functionality is validated during the testing phase. The RAD methodology emphasises close collaboration between the development team and active end-user participation in the development process. This ensures that the software being developed fulfils the demands and requirements of the consumers. The methodology also encourages frequent feedback and iteration, allowing for rapid changes as the project develops (Schindler, 2010).

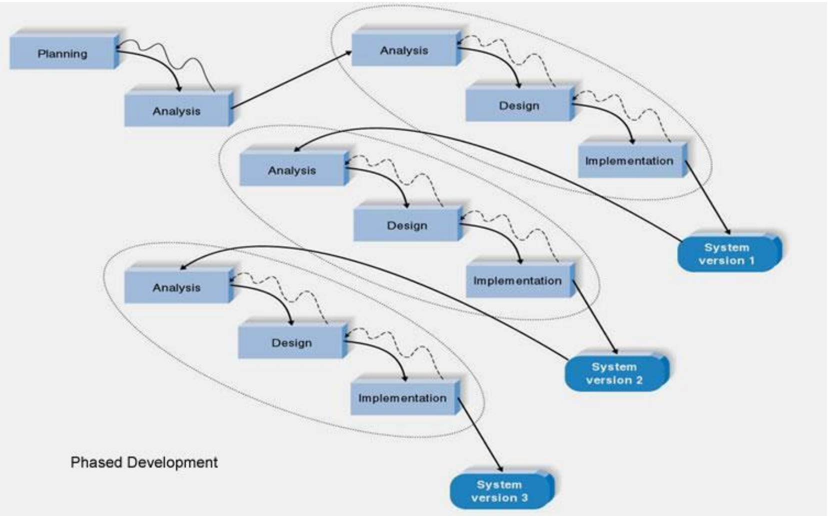


Figure 2.5: Rapid Application Development Phased Model. (PadaKuu, 2023)

Before deciding to implement the RAD phased development methodology on a project, its advantages and disadvantages must be weighed. The ability to rapidly develop and deploy software systems is one of the primary benefits of this methodology. The iterative nature of the process ensures that feedback can be received rapidly and adjustments can be made in response. This can reduce time-to-market and increase consumer satisfaction. It is also highly adaptable to altering needs, which is an additional advantage. As the project progresses and new requirements become apparent, they can be incorporated into the development process, allowing for greater flexibility and responsiveness (Francis, 2018). In addition, this methodology encourages collaboration and communication among team members, resulting in a greater comprehension of the project and its goals.

RAD phased development does have a number of disadvantages, however. The emphasis on quickness and agility can sometimes come at the expense of quality, which is a potential drawback. The rapid development process may lead to less stringent testing and quality control, which can result in errors and defects in the final product. Another potential disadvantage is that the iterative nature of the process can result in scope creep, in which the project becomes more complex than was originally intended. If not properly managed, this may result in delays and increased costs.

2.2.5 Summary

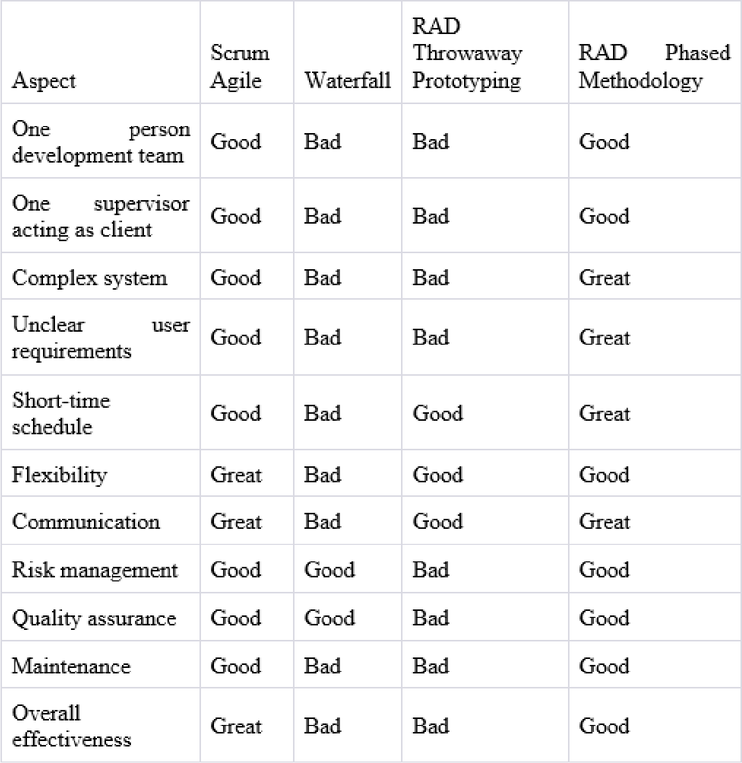
When it comes to a one-person development team, Scrum Agile and RAD Phased Methodology are good options as they encourage communication and collaboration among team members. Waterfall and RAD Throwaway Prototyping, on the other hand, may be more appropriate as they follow a structured approach with clearly defined roles and responsibilities. For a single administrator operating as a client, Scrum Agile and RAD Phased Methodology are ideal as they require constant feedback and collaboration between the client and the development team. However, Waterfall and RAD Throwaway Prototyping may not be suitable as they rely on an upfront requirements gathering phase that may not fully reflect the client's needs.

When managing complex systems, Scrum Agile is beneficial as it allows for flexibility and adaptability. Waterfall and RAD Throwaway Prototyping, on the other hand, may be better options as they follow a structured approach that may not be tailored to the system's complexity. The RAD Phased Methodology is beneficial because it allows for rapid prototyping and testing of various components, which can identify potential issues and optimize the design.

When dealing with unclear user requirements, Scrum Agile is beneficial as it allows for continuous collaboration and feedback to clarify requirements over time. Waterfall and RAD Throwaway Prototyping may be better options as they follow a more structured approach that is not suited for uncertain and fluid user requirements. The RAD Phased Methodology is useful because it allows for rapid prototyping and testing of various components to identify potential issues and optimize the design.

For rapid iteration and product delivery, Scrum Agile and RAD Phased Methodology are great options as they emphasize iterative development. Waterfall may not be suitable for a brief time frame, and RAD Throwaway Prototyping is beneficial as it permits rapid prototyping and testing of various designs. Scrum Agile is advantageous as it prioritizes adaptability and flexibility, allowing for alterations to the project scope over time. Once the project plan has been established, Waterfall may be more flexible and accommodate modifications. RAD Throwaway Prototyping is beneficial as it allows for rapid prototyping and testing of numerous designs.

Table 2.1: Summary of comparison between methodologies



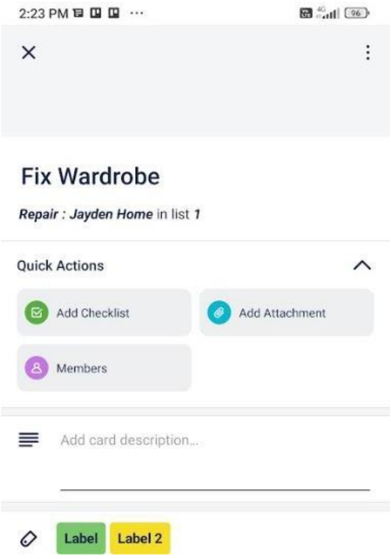
Aspect	Scrum Agile	Waterfall	RAD Throwaway Prototyping	RAD Phased Methodology
One person development team	Good	Bad	Bad	Good
One supervisor acting as client	Good	Bad	Bad	Good
Complex system	Good	Bad	Bad	Great
Unclear user requirements	Good	Bad	Bad	Great
Short-time schedule	Good	Bad	Good	Great
Flexibility	Great	Bad	Good	Good
Communication	Great	Bad	Good	Great
Risk management	Good	Good	Bad	Good
Quality assurance	Good	Good	Bad	Good
Maintenance	Good	Bad	Bad	Good
Overall effectiveness	Great	Bad	Bad	Good

2.3 System comparison

2.3.1 Trello

Figure 2.6: Trello Logo

Trello, a widely used project management application that facilitates effective team collaboration and project tracking. Trello mobile application is available for both Android and iOS devices and offers a very handy of customization set of features to users. Moreover, the Trello mobile application offers seamless synchronization with the desktop application, ensuring that users have access to the most recent project information regardless of their location.


2:23 PM

Fix Wardrobe

Repair : Jayden Home in list *1*

Add card description...

Figure 2.7: Trello’s Cards Interface.

Trello organizes through cards. It indicates a task, attempt, or other monitoring item. Cards may include due dates, attachments, comments, labels, and team members. As they advance, they may be moved across lists. Each task can be associated with a checklist which is a simple tick and complete concept. Members can be assigned through cards but when it comes to checklist it is not available. A card may also be labeled specifically to group them accordingly which is able to promote categorization.

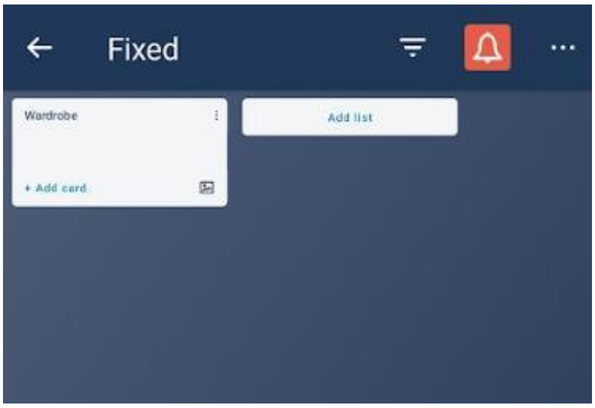


Figure 2.8: Trello’s Board Interface.

Trello lists include project category or stage cards. Lists help organize and prioritize cards by showing which tasks are done. Trello workspaces include team or project-related boards. Workspaces allow teams to work together in private. They show all project or team boards.

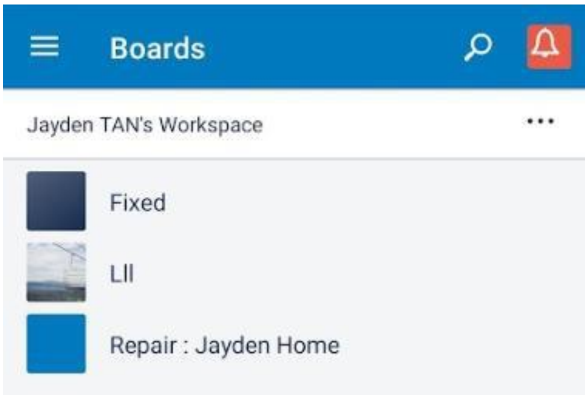
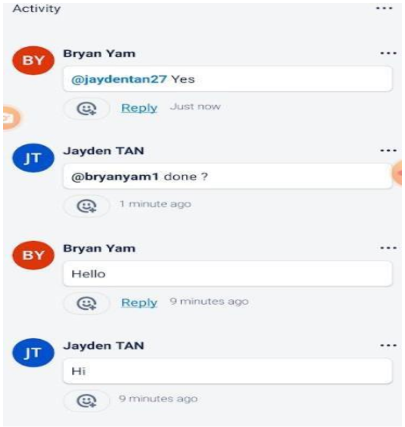


Figure 2.9: Trello’s Workspace Interface.

Trello boards include cards for a project, team, or process. It covers everything in a certain region. Each board may include several listings of project stages or labor categories.



Activity

Bryan Yam

@jaydentan27 Yes

Jayden TAN

@bryanyam1 done ?

Bryan Yam

Hello

Jayden TAN

Hi

Figure 2.10: Trello’s Communication Space Interface.

Members assigned to the specific task may also comment and communicate in the activity section where a few more features such like tagging a specific member or adding emojis which is able to express one feelings better

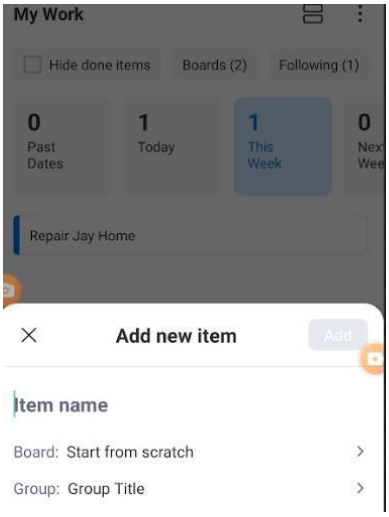
One of the primary benefits of Trello is its simplicity, as the user interface is straightforward and easy to use. This makes it an ideal tool for both managers and employees who may not have extensive technical expertise. Additionally, Trello allows users to organize their projects into boards, lists, and cards, providing a clear overview of all tasks and their status. In terms of task management, Trello allows users to create cards that represent specific tasks, which can be moved between lists as they progress through different stages. This helps managers to keep track of each task and ensure that it is completed on time. Trello also allows users to set deadlines, assign tasks to specific team members, and attach files or comments to each task, which improves collaboration and transparency.

Despite its many advantages, Trello also has some disadvantages. One potential issue is that Trello may not be suitable for complex projects that require detailed project management features. Another limitation is that Trello does not offer the ability to track time spent on tasks, which can make it difficult to accurately estimate. Trello is a very simple application that requires no high-expertise in technology where everyone can easily adapt to, but the main objective could be identified as managing tasks or project but does not aim to solve the painpoints that employees are experiencing.

2.3.2 Monday.com

Figure 2.11: Monday.com’s logo.

Monday.com is a cloud-based project management tool that allows teams to interact and organize their work. The platform has a mobile app for project and task management.



My Work

□ Hide done items | Boards (2) | Following (1)

0 Past Dates | 1 Today | 1 This Week | 0 Nex Wee

Repair Jay Home

Add new item | Add

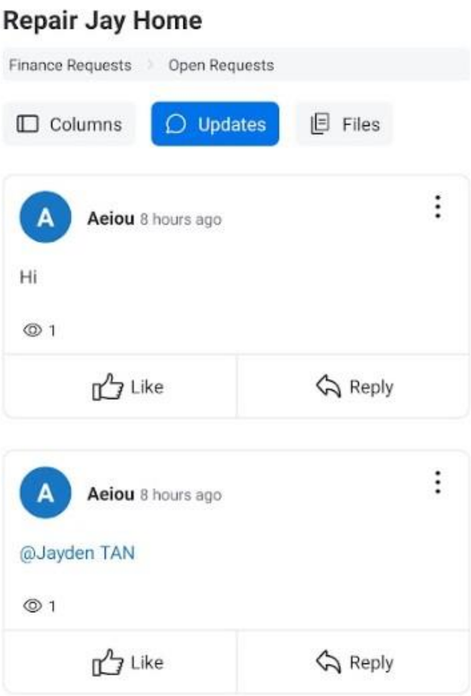
Item name

Board: Start from scratch

Group: Group Title

Figure 2.12: Monday.com’s Add Task Interface.

Monday.com mobile app lets users create and assign tasks, set due dates and reminders, and track progress.



Aeiou 8 hours ago

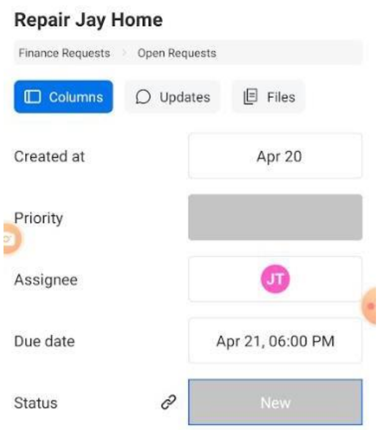
Hi

Aeiou 8 hours ago

@Jayden TAN

Figure 2.13: Monday.com’s Communication Space Interface.

For context and progress updates, users may add comments, files, and photos to tasks. Monday.com's mobile app has many communication channels for project management. Comments, @mentions, and direct messaging allow team members to discuss projects. Monday.com's mobile app for managing projects and on- site personnel is user-friendly, collaborative, and mobile. The platform's configurable dashboards also let customers view and measure their progress.



Created at Apr 20

Priority

Assignee JT

Due date Apr 21, 06:00 PM

Status New

Figure 2.14: Monday.com’s Task View Interface

The Monday.com mobile app may give tasks to staff. Project managers may assign tasks and track progress using this functionality. Managers may assign resources and finish projects on time using the platform's workload visibility.

Monday.com has several advantages when it comes to managing and monitoring tasks and on-site employees. Firstly, the platform offers a highly customizable and flexible workspace, allowing users to tailor their workflows and task management processes to their specific needs. This can greatly improve efficiency and productivity, as users can optimize their workflows to suit their unique requirements. Another key advantage of Monday.com is its ability to facilitate collaboration and communication between team members. The platform offers a range of communication tools, such as direct messaging and file sharing, which can help to streamline the collaboration process and ensure that everyone is on the same page. In addition, Monday.com provides a range of automation features that can help to reduce the amount of manual work required for certain tasks, such as task assignment and status updates. This can greatly improve efficiency and reduce the risk of errors or delays.

However, it has been reported that the Android version of the application has some issues with creating sub-tasks under the main task. This issue has caused frustration among Android users as they are unable to create sub-tasks. Additionally, when Android users try to comment or communicate through the application, the messages are unable to be sent out. These unsolved errors may lead to dissatisfaction among Android users. Eventhough Monday.com has a user-friendly UI, the application does not have the ability to handle real-time modifications to projects, which can be a major disadvantage. For instance, if an employee is sick and unable to complete their assigned tasks, there is no filter to filter out the tasks associated with the employee and reassign them to someone else. This lack of real-time management features can lead to a backlog of tasks and hinder the overall productivity of the team. Moreover, the application does not have features to handle complex project requests. This can pose a challenge when managing projects with complex requirements, and the application may not be suitable for such projects. The overload of tasks assigned to employees can also be a challenge when using Monday.com, and there is no efficient way to balance the workload of the employees.

2.3.3 Asana

Figure 2.15: Asana’s logo

Asana enables cloud-based task, project, and process management for teams. Its numerous features facilitate teamwork, communication, and organization.

The adaptable UI of Asana accommodates diverse processes and team configurations. It is utilized by teams of various sizes and industries due to its straightforward design and user-friendly interface.

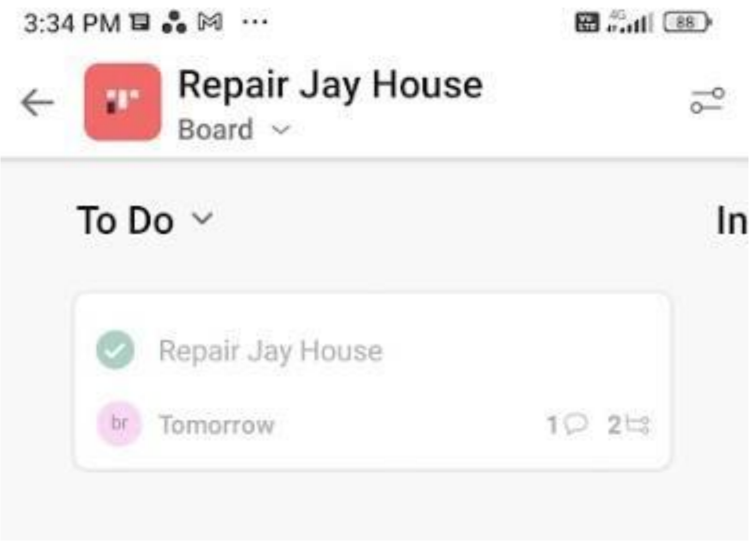
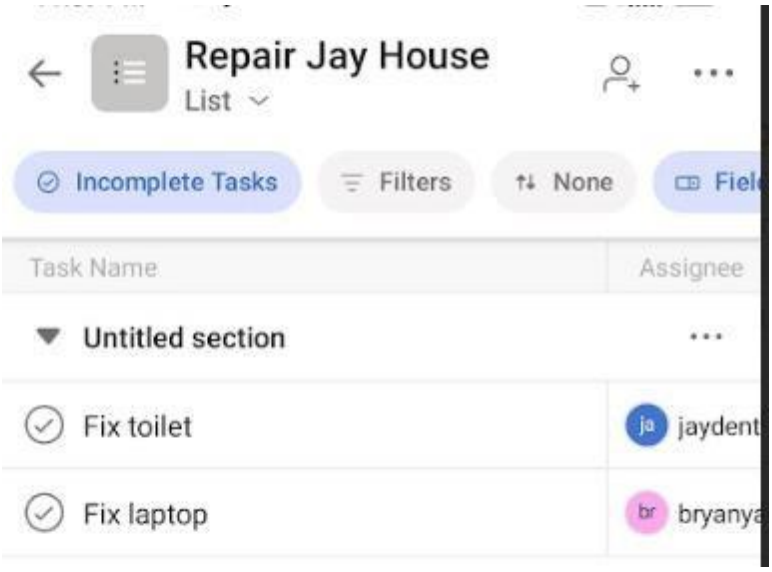


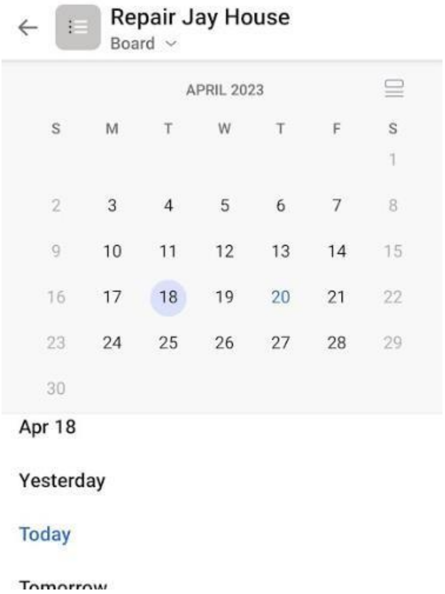
Figure 2.16: Asana’s Board Views



Repair Jay House
List

Task Name	Assignee
Untitled section	
Fix toilet	ja jaydent
Fix laptop	br bryanya

Figure 2.17: Asana’s List Views



Repair Jay House
Board

APRIL 2023

S	M	T	W	T	F	S
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30						

Apr 18

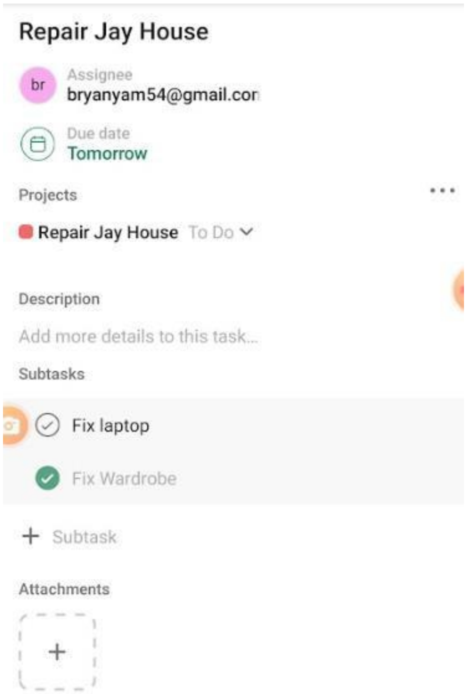
Yesterday

Today

Tomorrow

Figure 2.18: Asana’s Calendar Views

Asana enables users to create, assign, set due dates, attach files, and track progress. With its visual timeline, calendar, and configurable interface, users can monitor and manage their duties.



Repair Jay House

Assignee: bryanyam54@gmail.cor

Due date: Tomorrow

Projects: Repair Jay House To Do

Description: Add more details to this task...

Subtasks:
- Fix laptop
- ✓ Fix Wardrobe

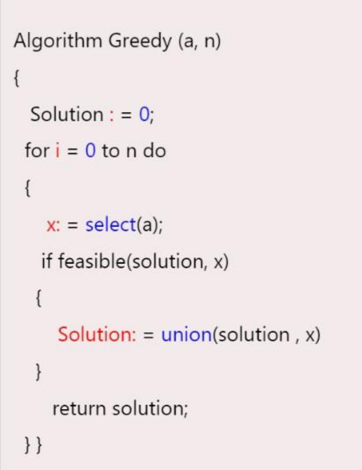
Figure 2.19: Asana’s Create Task

Asana has some disadvantages when it comes to managing on-site employees where more factors have to be considered. One major disadvantage is the lack of a dashboard that displays the current performance of each employee. This makes it difficult for managers to track the progress of their employees and ensure that they are meeting their targets. Additionally, Asana does not provide any solutions to aid in the re-assignment of tasks when an employee is unavailable or overloaded with work. This can lead to delays and inefficiencies in the project management process, ultimately affecting the project's success. Another issue is that Asana may not be suitable for managing complex projects with many interdependent tasks and timelines. In such cases, a more comprehensive project management software may be needed.

2.4 Algorithm

2.4.1 Greedy Algorithm

A greedy algorithm is a form of algorithmic strategy that entails making a series of locally optimal decisions at each stage in order to discover the global optimal solution. It operates by making the optimal decision at each stage based on the available information. The idea is to choose the option that provides the greatest immediate benefit, regardless of the prospective long-term consequences of that decision. In optimisation problems where the objective is to identify the optimal solution from a set of possible solutions, greedy algorithms are frequently employed.



```
Algorithm Greedy (a, n)
{
  Solution : = 0;
 for i = 0 to n do
 {
    x: = select(a);
   if feasible(solution, x)
   {
      Solution: = union(solution , x)
   }
     return solution;
 } }
```

Figure 2.20: Greedy Algorithm Pseudocode

The given pseudocode describes a greedy algorithm for a given problem. The input to the algorithm is an array 'a' of 'n' elements. The objective of the algorithm is to find a feasible solution that maximizes some given criteria. The algorithm starts by initializing a variable 'solution' to 0. Then, it iterates through the array 'a' from 0 to n. In each iteration, the algorithm selects an element 'x' from 'a' using a select function. The select function chooses an element based on some criteria, which varies depending on the problem. After selecting the element 'x', the algorithm checks if the solution is feasible by calling the 'feasible' function. If the selected element 'x' is feasible, the algorithm updates the solution by taking the union of the existing solution and the selected element 'x'. Finally, the algorithm returns the solution.

The algorithm can be modified as following:

```
Algorithm Greedy(a, n, task)
{
    EmployeeList := []; for i = 0 to n do
    {
        x := select(a);
        if feasible(task, x)
        {
            EmployeeList := union(EmployeeList, x)
        }
    }
    return EmployeeList;
}
```

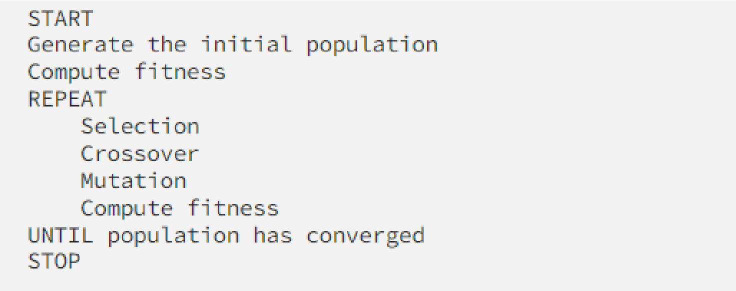
In this modification, the original "Solution" variable has been replaced with an "EmployeeList" list variable, which will store the suggested employees for the task. The "feasible" function has been replaced with a new function called "feasible(task, x)", which will check whether the employee x is suitable for the task based on their skillset, availability, schedule, and travel time. Considerations would be taken in if the employee has the required skillset for the task , available during the time frame required for the task, schedule allows for the required travel time to the task location. This modified algorithm will iterate through the list of employees and select those who meet the requirements of the task based on the feasibility function. The final result will be a list of suggested employees for the task.

The greedy algorithm has the advantages of being efficient and simple to implement. It can yield a reasonably decent solution rapidly, even if it is not the optimal solution. In the context of selecting employees for a task based on skillsets, availability, schedule, and travel time, a greedy algorithm can rapidly generate a list of candidates who fulfil the necessary criteria.

However, greedy algorithms have the disadvantage of not always producing the optimal solution. This is due to the fact that greedy algorithms make local optimal decisions at each stage without taking the bigger picture into account. Thus, the solution generated by a greedy algorithm might not be the most optimal or expedient solution conceivable.

2.4.2 Genetic algorithm

A genetic algorithm is a form of optimisation algorithm that draws inspiration from the natural selection and evolution processes. The algorithm generates new progeny by employing genetic operators, such as mutation and crossover, to a population of potential solutions (referred to as "individuals"). Individuals with higher fitness values are more likely to be chosen to contribute to the next generation based on how well they solve the problem at hand. The genetic algorithm is built around the concept of natural selection. The fittest individuals are chosen and recombined to produce progeny with the potential to become a further fitter. Over moment, the population develops and the individuals' fitness increases.



```
START
Generate the initial population
Compute fitness
REPEAT
    Selection
    Crossover
    Mutation
    Compute fitness
UNTIL population has converged
STOP
```

Figure 2.21: Genetic Algorithm Pseudocode (Mallawaarachchi, 2020)

The algorithm begins by generating a population consisting of a group of potential solutions to the problem at hand. Next, the fitness of each solution in the population is evaluated, where the fitness function determines how well each solution meets the given constraints. The algorithm then iterates through the processes of selection, crossover, and mutation. Selection is the process of selecting the most advantageous solutions from a population for transmission to the next generation. Crossover is the process of combining the genetic material of two or more parent solutions to produce new progeny solutions that may have improved fitness. Mutation introduces arbitrary modifications to the progeny solutions in order to investigate new regions of the solution space. After each iteration of selection, crossover, and mutation, the new population's fitness is reassessed. This process is repeated until the population converges or the maximal number of iterations has been attained. The algorithm terminates, and the most optimal solution in the final population is deemed the algorithm's output.

The algorithm can be modified as following:
starting with generating the initial population of potential employee assignments for the task.
Evaluate the fitness of each potential assignment based on factors such as skillset, availability, schedule, and traveling time.
Select a subset of the population to be used as parents for the next generation.
Use crossover and mutation operations to create a new generation of potential assignments.
Evaluate the fitness of each potential assignment in the new generation.
Repeat steps 3-5 until a satisfactory solution is reached or a predetermined number of iterations has been completed.

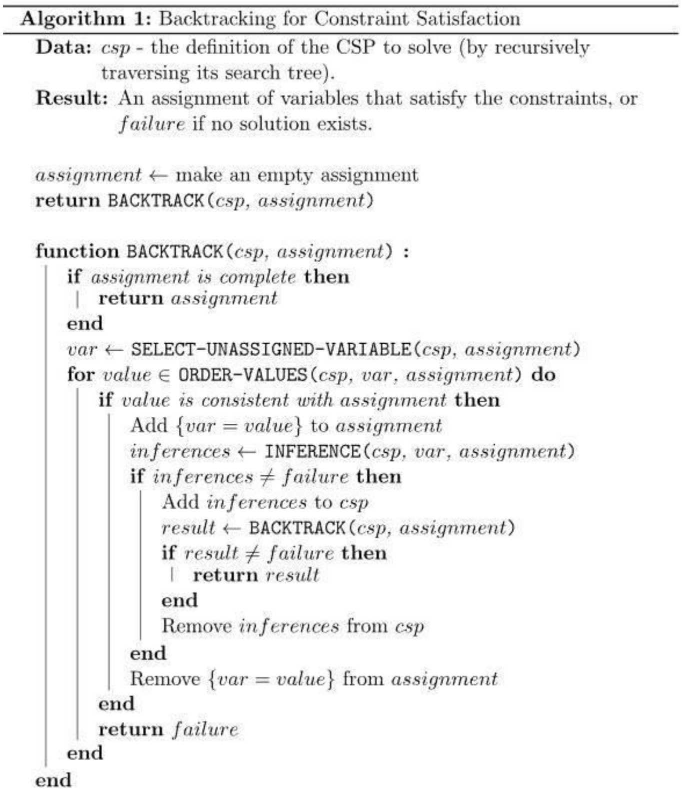
Individuals with higher fitness values have a greater chance of being chosen as parents during the selection procedure. Crossover is the process of combining genetic material from two individuals to produce a new individual. Mutation is the arbitrary alteration of the genetic composition of an organism to create a new organism. The fitness function evaluates each assignment based on a number of factors, including skillsets, availability, schedule, and travel time, to determine how well it meets the task's requirements. The modified genetic algorithm is a potent tool for generating a list of suggested employees for a task, taking into account factors such as skill sets, availability, schedule, and travel time.

One of the primary benefits of genetic algorithms is their ability to discover solutions that are not immediately evident. They are capable of handling problems that are challenging or impossible to solve with conventional methods. In addition, genetic algorithms can optimise solutions by simultaneously evaluating multiple criteria, including skill sets, availability, schedule, and travel time.

However, there are also disadvantages to genetic algorithms. They can be computationally expensive, necessitating a significant amount of time and resources to generate solutions. In addition, genetic algorithms may not always locate the optimal solution and may converge on a suboptimal one. Additionally, the results generated by genetic algorithms may be difficult to interpret and explain, making it difficult to determine why a particular solution was chosen.

2.4.3 Constraint Satisfaction Backtracking

A Restriction Satisfaction The backtracking algorithm is an algorithm used to tackle constraint satisfaction issues. It operates by recursively attempting various values for each variable in the problem and backtracking when a constraint is violated until a solution is found or it is determined that none is possible. The algorithm begins by designating a variable a value that satisfies the constraints. If a constraint is broken, the algorithm returns to the prior variable and attempts a new value. This procedure is repeated until all variables have been assigned values that satisfy the constraints or until it is determined that no solution exists.



```
Algorithm 1: Backtracking for Constraint Satisfaction
  Data: csp - the definition of the CSP to solve (by recursively
        traversing its search tree).
  Result: An assignment of variables that satisfy the constraints, or
          failure if no solution exists.

  assignment ← make an empty assignment
  return BACKTRACK(csp, assignment)

  function BACKTRACK(csp, assignment) :
    if assignment is complete then
      return assignment
    end
    var ← SELECT-UNASSIGNED-VARIABLE(csp, assignment)
    for value ∈ ORDER-VALUES(csp, var, assignment) do
      if value is consistent with assignment then
        Add {var = value} to assignment
        inferences ← INFERENCE(csp, var, assignment)
        if inferences ≠ failure then
          Add inferences to csp
          result ← BACKTRACK(csp, assignment)
          if result ≠ failure then
            return result
          end
          Remove inferences from csp
        end
        Remove {var = value} from assignment
      end
      return failure
    end
  end
```

Figure 2.22: Constraint Satisfaction - Backtracking Algorithm Pseudocode. (E. Ortiz, 2006)

The algorithm works recursively by traversing the search tree of the CSP. It starts by making an empty assignment and passes it to the BACKTRACK function. If the assignment is complete, the algorithm returns the assignment. Otherwise, it selects an unassigned variable and orders its values. For each value, if it is consistent with the current assignment, it adds the variable-value pair to the assignment. The INFERENCE function is then called to perform any necessary forward checking or constraint propagation to reduce the search space. If the inference results in a failure, the algorithm backtracks to the previous state and tries the next value for the variable. If the inference is successful, the algorithm recursively calls the BACKTRACK function with the updated assignment. If the result of the recursive call is not a failure, it returns the assignment. Otherwise, the algorithm removes the inference and the variable-value pair from the assignment and continues to the next value.

The algorithm can be modified as below:

Define the CSP (constraint satisfaction problem) in terms of the factors to be considered for the task, such as skillsets, availability, schedule, and travelling time.

Instead of "variables" and "values", define the set of employees as the domain of the CSP and their skillsets, availability, and schedule as the attributes.

Define the constraints between the attributes of employees, such as their availability and schedule, and the travelling time required to reach the task location.

Modify the SELECT-UNASSIGNED-VARIABLE function to select an employee who satisfies the constraints of the task based on their attributes.

Modify the ORDER-VALUES function to order the employees based on their skillsets, availability, and proximity to the task location.

Modify the INFERENCE function to check if assigning an employee to the task violates any of the constraints, such as their availability and travelling time.

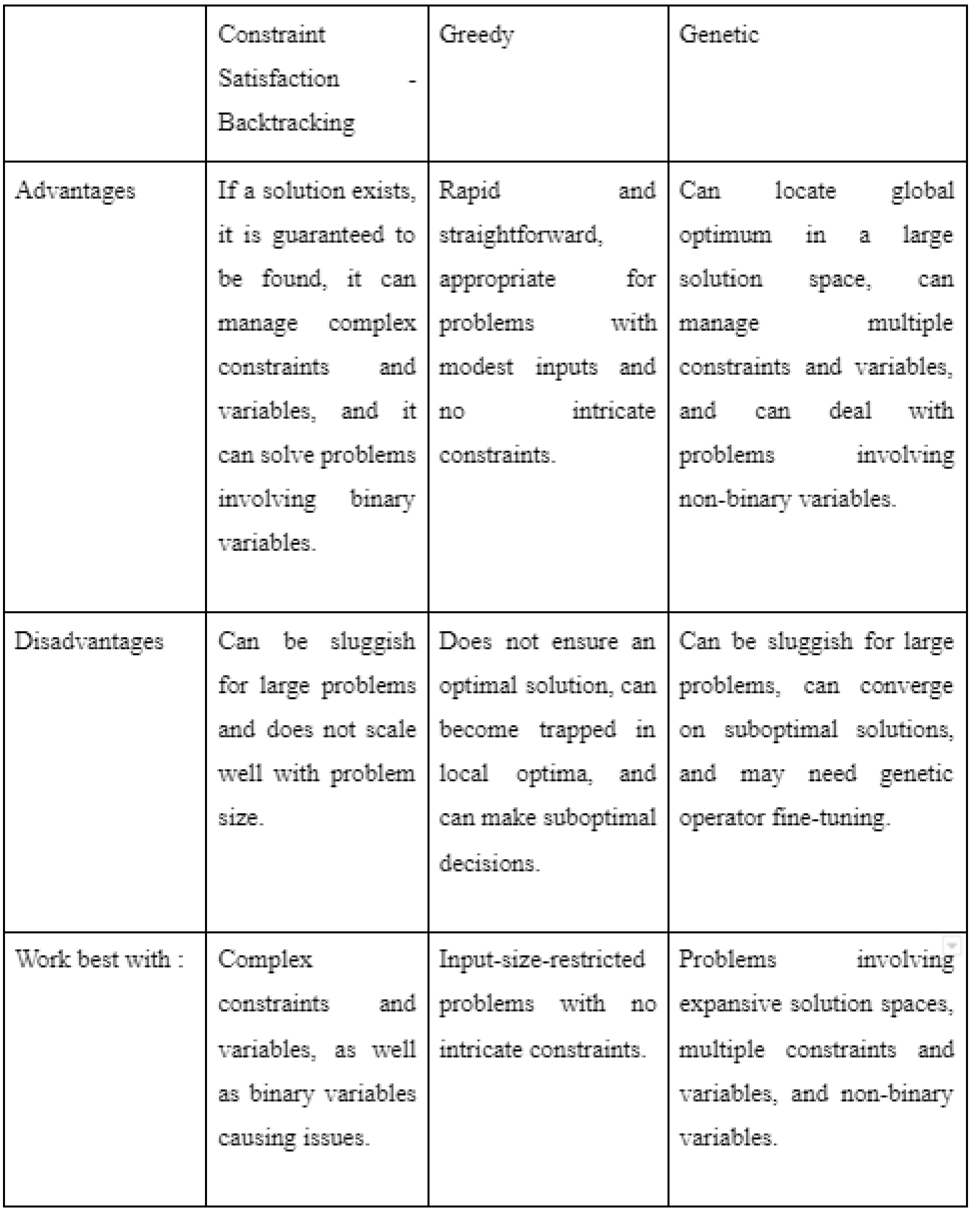
If the solution is found, return the list of suggested employees that satisfy the constraints of the task. If no solution is found, return failure.

Backtracking algorithm for Constraint Satisfaction is advantageous for solving problems with a large number of constraints, for which brute force methods are impractical. It can be utilised to efficiently seek through a large solution space and locate the best solution. Additionally, the algorithm is simple to comprehend and implement.

However, using backtracking to solve the employee assignment problem has some disadvantages. Particularly when the search space is expansive, it can be computationally expensive. In addition, the algorithm does not ensure the discovery of an optimal solution, only a feasible one. Additionally, it can be difficult to determine the order in which to designate variables and values, which can impact the algorithm's performance. Lastly, the algorithm may need a substantial quantity of memory to store the search tree, which can be a limitation on some applications.

2.4.4 Summary

Table 2.2: Summary of algorithms



	Constraint Satisfaction - Backtracking	Greedy	Genetic
Advantages	If a solution exists, it is guaranteed to be found, it can manage complex constraints and variables, and it can solve problems involving binary variables.	Rapid and straightforward, appropriate for problems with modest inputs and no intricate constraints.	Can locate global optimum in a large solution space, can manage multiple constraints and variables, and can deal with problems involving non-binary variables.
Disadvantages	Can be sluggish for large problems and does not scale well with problem size.	Does not ensure an optimal solution, can become trapped in local optima, and can make suboptimal decisions.	Can be sluggish for large problems, can converge on suboptimal solutions, and may need genetic operator fine-tuning.
Work best with :	Complex constraints and variables, as well as binary variables causing issues.	Input-size-restricted problems with no intricate constraints.	Problems involving expansive solution spaces, multiple constraints and variables, and non-binary variables.

CHAPTER 3

METHODOLOGY AND WORK PLAN

3.1 Introduction

In this chapter, a comprehensive description of the RAD phased development methodology will be presented, with a detailed explanation of each phase in relation to the project at hand.

Additionally, the planning, design, and development tools utilized in this project will be clearly defined.

Finally, this chapter will conclude with the creation of a work breakdown structure and Gantt chart, providing a comprehensive overview of the project's scope and schedule planning.

3.2 RAD Phaed Development

The RAD phased development methodology is a useful approach for quickly delivering a functional system to users. In this project, the methodology will be used to develop an employee onsite task assignment management and tracking app. As the only developer on the project, the phased development approach will enable one to focus on developing a minimal viable product (MVP) that can provide immediate business value to the client.

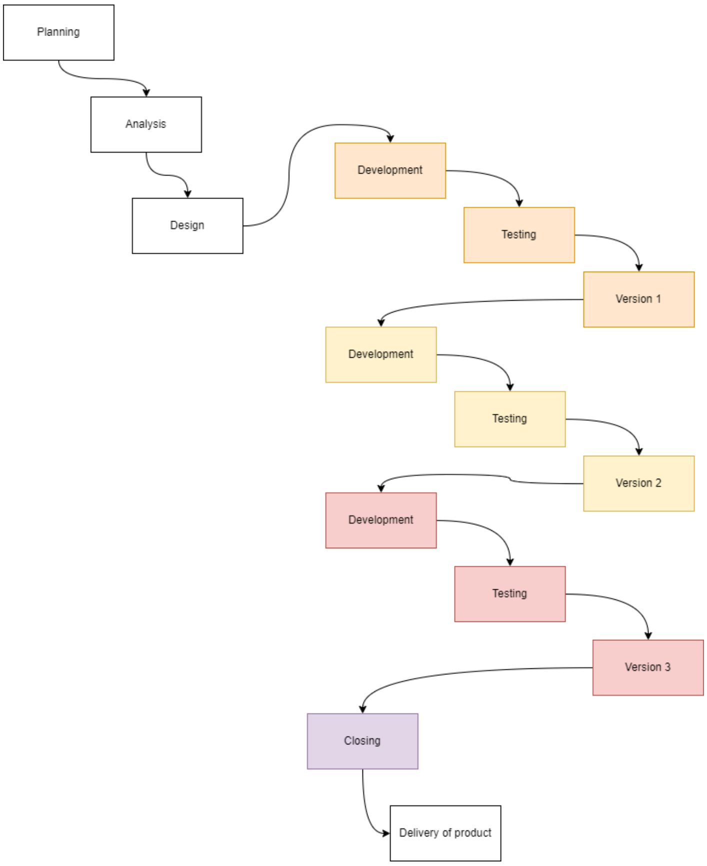


Figure 3.1: RAD Phased Methodology Implemented

3.2.1 Planning

In the Software Development Life Cycle (SDLC), the initial stage is typically the planning phase, where a project plan is developed. This plan outlines the project scope, goals, timelines, and resources required for successful completion.

3.2.1.1 Project initiation

In the first stage of the Software Development Life Cycle (SDLC), planning is divided into two main parts: project initiation and project management. The initial part of project initiation involves developing a system request, which focuses on collecting and documenting information regarding the project's objective, scope, and requirements which is to identify the business needs, problems, and opportunities that the project aims to address, resulting in the creation of a problem statement.

Speaking to Miss Tan Su Yee, a team supervisor, based on her accumulated experience over the years in the specific field, has contributed to the main problem for this project. This was done to better understand the most pressing issues in the field. To gain a deeper understanding of the current situation in the field, several articles and journals were studied. Once the problem statement is generated, the project objectives, aim, and solution are further developed by researching algorithms and methodologies that could be applied. The next step is to define what will be covered in the project scope to narrow down the goals and the system's importance.

Furthermore, studying the existing applications is a valuable approach to gain further insight into the field. By registering two accounts, one as the admin and another as the staff, the to evaluate the usability of the applications from the perspective of both parties. This has contributed to identify additional requirements, areas for improvement, and potential drawbacks that could be addressed in this project. By utilizing the applications and correlating the responses from the survey on how they can be improved, identification of both functional and non-functional requirements for the system.

3.2.1.2 Project Management

To ensure effective control and direction of the project, the next step in the planning stage is to develop a work plan. This involves creating a work breakdown structure (WBS) that breaks the project into smaller, more manageable parts. Once the WBS is completed, a Gantt chart is generated using Microsoft Project to depict the timeline and schedule of the project. The Gantt chart is for visualizing project timelines and schedules. It displays the start and end dates of all tasks and subtasks, as well as the dependencies between them. It allows for the allocation of resources and time, provides a graphical representation of the project's progress, and helps to identify potential bottlenecks or delays that ensures the project is completed on time, within budget, and meets the project goals and objectives.

3.2.2 Analysis

During the analysis stage, the focus is on producing a system proposal that involves the development of requirements and analysis strategy. This stage builds upon the partial work done during the planning stage in accordance with the SDLC.

3.2.2.1 Develop analysis strategy and gather of requirements

In the development analysis strategy part of the analysis stage, the main topic of managing and monitoring onsite employees was already studied, and some issues were identified through planning stage. In addition to speaking to Miss Tan Su Yee, a conversation with Mr. Loy was arranged over the phone to gain a better understanding of current practices. Mr. Loy Chai Meng from SUPERCOOL CREATIVE CULTURE SDN BHD, a company that specializes in furniture roofing system, metal structural work, timber work, and house renovation, provided relevant data based on his 10 years of experience in the onsite field sector. Although Mr. Loy did not cover all aspects of the topic which is the onsite supervisors, the major and critical issues were understood, and the expectations of the system's users were identified. The main objective of this phone call was to identify the problems and improvements required so that the concerns could be taken into consideration in the project's scope.

Supercool
走入自然，
创造理想家居。
LET'S GET NATURALYOUR HOME
Our Products 我们的服务
木制凉亭・铁制凉蓬・透明凉蓬・
玻璃凉亭・泥瓦凉亭・木制地板・木制围栏
Pergola • skylight • Polycarbonate roofing •
Laminated Glass Roofing • Timber Decking •
Timber Gate • Timber Fencing • Tiles Roofing
荣获2014组别
卓越荣誉
Honesty
Enterprise
Keris
Award 2014
Supercool Roofing Sdn Bhd 1142173-U
Supercool Creative Culture Sdn Bhd Co. No. 874137-M
Website: www.supercool.com.my / www.supercool2u.com.my Email: sales@supercool.com.my
Finished with water resistant and ultra violet protection
采用抗腐热带硬木并经加工防水防晒等保护
KL HQ : No. 12 & 14, Jalan Permai 4, Taman Perindustrian Puchong Permai, 47150 Selangor.
Tel: 03-5885 1180 Fax: 03-5885 1198
JB Branch: No.5, Jalan Ekopemiagaan1/19, Taman Ekopemiagaan 81100 JohorBahru, Johor. (ECO BUSSINESS PARK I)
Tel: 07-595 2928 Fax: 07-595 2023
Seremban Branch: 181-1st Floor, Jln S2 B3, Seremban Two, 70300 Negeri Sembilan D.K.
Tel: 06-601 2502 Fax: 06-601 1410
We also accept
VISA
0% Interest
*Terms and conditions apply

Figure 3.2: Supercool Creative Culture Sdn Bhd

In addition to speaking with an authorized person with deep experience in the field, surveys were conducted to collect data relevant to the topic. Google Forms is utilized as the survey tool due to its user-friendly interface and ease of use. It is widely recognized and accessible, with many users familiar with its operation. Real-time data collection is another benefit of Google Forms, allowing for immediate analysis and summary of survey responses. This feature enables valuable insights to be extracted efficiently without the need for manual calculations or data processing. The survey questions were designed to collect data on user demographics, such as age and occupation, user behavior, such as how they interact with the current system, user preferences, such as what they like and dislike about the system, and any other feedback they may have on their experiences with the system and how it could be improved. The main objective of the survey was to gain a more comprehensive understanding of the user experience with the current practice and to identify potential improvements that could be made to enhance the user's experience fully. The survey process typically takes approximately three weeks to collect responses from the selected personas. Participants are selected by distributing the survey to known individuals and requesting that they share it with those who may be related to the field or title.

3.2.3 Design

After wrapping up the analysis phase, the project advances to the design phase, where the product's technical specifications are defined. During the design phase of the software development life cycle, it is essential to complete a number of significant tasks, including the establishment of the use case design, entity-relationship diagram (ERD), flowchart, and prototype. These responsibilities are essential for satisfying the requirements of the users while developing the ultimate product.

Designing the use case entails identifying the system's various use cases and describing the interactions between the actors and the system. This task is essential for comprehending the system requirements and ensuring that all required functionality is implemented. The result is a thorough use case diagram and use case descriptions that describe and illustrate the use cases and their relationships.

The entity-relationship diagram (ERD), a graphical representation of the system's data model, is then constructed. This model includes the numerous entities, their attributes, and their connections. The ERD is a crucial instrument for developers, as it ensures the system's data integrity and consistency. This task yields an exhaustive ERD diagram that outlines the entities and their relationships.

Flowcharts are also created during the design phase to illustrate the system's process flow. This task identifies potential system bottlenecks and inefficiencies and provides an opportunity to optimise the flow. This task will result in a detailed flowchart that illustrates the system's various phases and decision points.

To provide the supervisor with an illustration of the system, a prototype is created as the final step. This is a crucial step for ensuring that the final product satisfies the requirements of the target audience and identifying potential problems early in the development process. This task will result in a drafted working prototype that can be utilised for reference during the development stage.

In conclusion, in this stage the outcome would be the SRS that includes the use case diagram and descriptions, flow chart diagram of the system, ERD diagram, as well as the functional and non-functional requirements of the product.

3.2.4 Implementation

In the implementation stage, where the system is developed that usually is the part developers spend most efforts on it to develop something new or to make a change to ease a usual practice. From the few previous stages ,the base project details are already clearly defined clearly and to be developed accordingly. In the implementation stage of SDLC , the stage will be broken down into 2 main phases which are development and testing to tailorise for the needs and requirements of this project.

During the initial stage of the project, the foundation will be established. The first step is to establish the database and establish connections. Following that, the code architecture will be set up, including models with corresponding get and set methods, as well as basic controllers that include CRUD operations.

Additionally, Git will be utilized to keep a backup of the code after each segment is completed and to maintain previous versions. This is useful for reverting changes in case of issues or to make previous modifications.

3.2.4.1 Iteration 1

In the first iteration of the RAD phased development, several modules will be developed such as login and register pages, authentications, and employee's basic functions. Additionally, the project will also develop the employee, task, admin/supervisor model, and company model. The task management module would also be created with features such as computing deadlines, sending notifications to both employee and admin, and the ability to create tasks. To ensure that these modules are functioning correctly, unit testing will be conducted. The testing process will involve testing the modules one by one, which is the most major task in the first stage to ensure any issues can be detected and resolved early, preventing them from becoming major problems later on in the development process.

3.2.4.2 Iteration 2

In the second iteration of the RAD phased development, the employee management module will be developed. This will include implementing the selected algorithms to assign tasks to the most suitable employees based on their skill set and availability. The module will also prompt suggested employees when a task is created, and send reminders to both the supervisor and the employee about the task by arranging the tasks in respective dates. Unit testing will be conducted on the employees assignment module to ensure it functions as intended, and testing will also be done on the reminders sent to ensure they are received by both the supervisor and the employee.

3.2.4.3 Iteration 3

In the third iteration of RAD Phased development, the focus is on developing location tracking and reporting module. The location tracking module will locate the current location of the employee and arrange the route for employees to leave for the next appointment. The reporting module will contain the dashboard of daily tasks completed and progress, including information on how many employees are on leave. The system will also send a daily report on the task progress, indicating whether the expected base tasks were completed. The testing will involve unit testing of the calling of the API for estimated time to leave and location tracking. The purpose of testing is to ensure that the system is working as expected and that it meets the requirements specified in the project scope.

3.2.4.4 Closing stage

In the final stage of the RAD phased development, the system will undergo User Acceptance Testing (UAT) to ensure it aligns with the expectations and needs of the end-users. This critical testing phase will validate that the system is user-friendly and resonates with the community's requirements. In addition to UAT, we will re-perform unit testing to ascertain that individual components of the system function correctly. The combined insights from these tests will guide any necessary refinements and ensure that the system is robust and ready for the final release.

3.3 Development tools

In the realm of software development, tools are indispensable for increasing efficiency and effectiveness. Software, hardware, and platforms that facilitate the creation of software applications are referred to as development tools. These tools are intended to expedite and automate the coding, testing, and debugging processes, allowing developers to work more efficiently and effectively.

There are numerous development tools available on the market today, each with its own capabilities and features. From Integrated Development Environments (IDEs) to version control systems, every software development project has an appropriate instrument. With the correct set of development tools, developers can concentrate on creating high-quality software that fulfils the requirements of their consumers, while simultaneously increasing their productivity and efficiency.

This subsection will examine the features and benefits of some of the most widely used development tools in the industry today.

3.3.1.1 Android Studio Emulator

Android Studio Emulator is a software tool used for developing mobile applications. It is useful for developers who want to test their applications without using a physical device. The emulator simulates the behavior of a real device, allowing developers to test different aspects of their application, such as user interface, performance, and compatibility with different versions of Android. The emulator also allows developers to test their application on different screen sizes, resolutions, and configurations, which can help them identify and fix issues early in the development process. In addition, Android Studio Emulator provides a platform for developers to test their application on different Android versions and devices without having to purchase them. This helps to reduce the cost and time associated with testing and development.

3.3.1.2 Visual studio code

Visual Studio Code is a lightweight and versatile code editor that is helpful for developing mobile applications. It supports multiple programming languages and has a rich set of extensions and plugins, making it a popular choice among developers. With its powerful debugging tools and code completion features, it helps to reduce errors and improve the quality of the code. It also has built-in support for version control systems like Git, allowing for easy collaboration between team members. Visual Studio Code's integrated terminal makes it easy to run commands and perform tasks, and its customizable user interface allows developers to tailor their development environment to their specific needs. All these features make Visual Studio Code a valuable tool for mobile application development.

3.3.1.3 Flutter

Flutter is a mobile app development framework that offers an extensive collection of customizable widgets and tools for developing high-performance mobile applications for both Android and iOS platforms. It allows developers to build natively compiled applications using a single codebase, making it faster and more efficient than other frameworks. Flutter offers features like hot reload, which allows developers to see the changes in real-time as they make them, and easy integration with popular backend technologies. Additionally, Flutter offers a wide range of UI widgets and animations, making it easier to build visually appealing and responsive applications. Its support for Material Design and Cupertino widgets makes it easy to create pixel-perfect apps that follow both Google's and Apple's design guidelines. These features make Flutter a popular choice for developers looking to build high-quality mobile applications with minimal effort and reduced development time.

3.3.1.4 Laravel

Laravel is a renowned web application framework that has become a cornerstone for modern web development. Designed with elegance, simplicity, and readability in mind, it offers a comprehensive solution for developing web applications. Laravel boasts features that make it a standout choice for developers, including its expressive ORM (Eloquent), easy-to-use routing, powerful templating engine (Blade), and built-in security measures. Its MVC (Model-View-Controller) architecture ensures a clean separation of logic and presentation, making the codebase organized and maintainable. Laravel's integration with tools like Composer ensures seamless package management, allowing developers to tap into a vast ecosystem of libraries and packages. The framework's emphasis on clean code, scalability, and rapid development ensures that web applications built with Laravel are not only high-performing but also future-ready.

3.4 Workbreakdown Structure

1. Planning
    - 1.1. Project Initiation
        - 1.1.1. Study the subject matter
            - 1.1.1.1. Study On -Site Meaning
            - 1.1.1.2. Study On-Site Employees
        - 1.1.2. Identify the problem background
            - 1.1.2.1. Research on the problems
            - 1.1.2.2. Speak to related authorities
            - 1.1.2.3. Analyze current practices
        - 1.1.3. Define project objectives
            - 1.1.3.1. Identify potential risks
            - 1.1.3.2. Set goals of deliverable
        - 1.1.4. Generate project solution and approach
            - 1.1.4.1. Study existing solutions
            - 1.1.4.2. Define the approach
        - 1.1.5. Create project proposal
            - 1.1.5.1. Identify the scope and constraints of the project
    - 1.2. Project Management
        - 1.2.1. Generate a work breakdown structure
        - 1.2.2. Generate Gantt CharT

2. Analysis

2.1. Requirement gathering

2.1.1. Conduct survey

2.1.2. Conduct interviews

2.1.3. Gather feedback and opinions

2.1.4. Review and refine requirements

2.2. Literature review

2.2.1. Comparison of algorithms

2.2.2. Comparison of existing systems

2.2.3. Comparison of methodologies

3. Design

3.1. Use Case Diagram

3.2. Use Case Description

3.3. Entity Relationship Diagram

3.4. Flowchart

3.5. Prototypes

4. Implementation

4.1. Pre-Project

4.1.1. Setup Database Connection

4.1.2. Generate Basic Architecture

4.1.3. Setup Github for Version control purposes

4.1.4. Unit Testing

4.2. Version 1

4.2.1. Login/Logout Module

4.2.2. Task Management Module

4.2.3. Unit Testing

4.3. Version 2

4.3.1. Employee Management Module

4.3.2. Unit Testing

4.4. Version 3

4.4.1. Location Tracking Module

4.4.2. Reporting Module

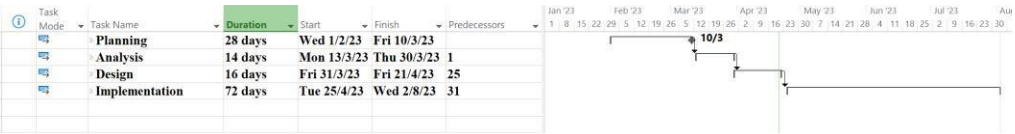
4.4.3. Unit testing

4.5. Closing

4.5.1. User Acceptance Test

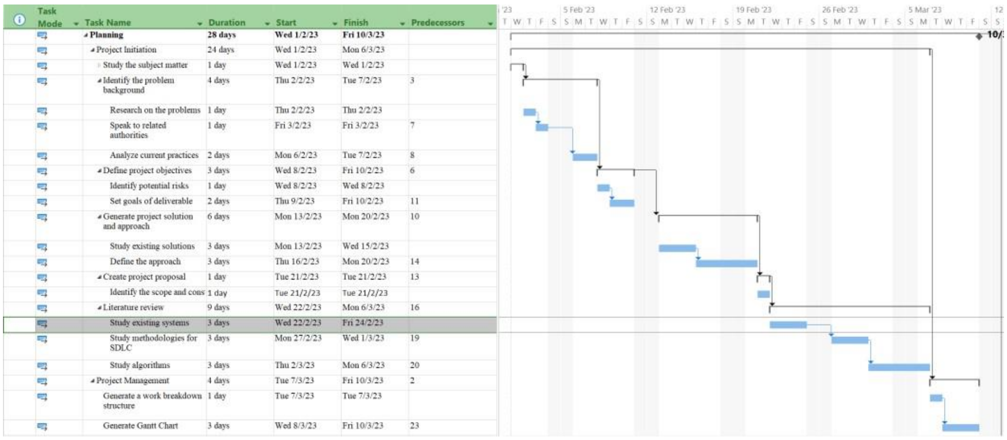
4.5.2. Unit Testing

3.5 Gantt Chart



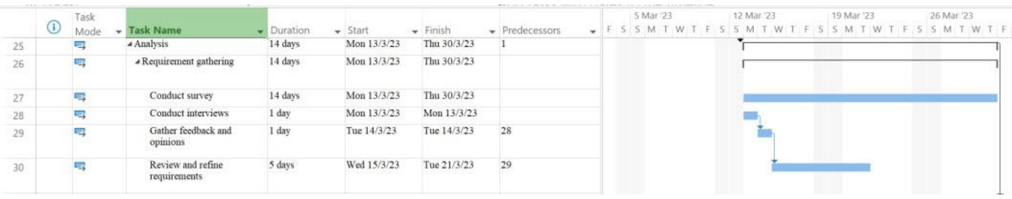
Task Mode	Task Name	Duration	Start	Finish	Predecessors
	Planning	**28 days**	**Wed 1/2/23**	**Fri 10/3/23**	
	Analysis	**14 days**	**Mon 13/3/23**	**Thu 30/3/23**	**1**
	Design	**16 days**	**Fri 31/3/23**	**Fri 21/4/23**	**25**
	Implementation	**72 days**	**Tue 25/4/23**	**Wed 2/8/23**	**31**

Figure 3.3: Overview of Project schedule



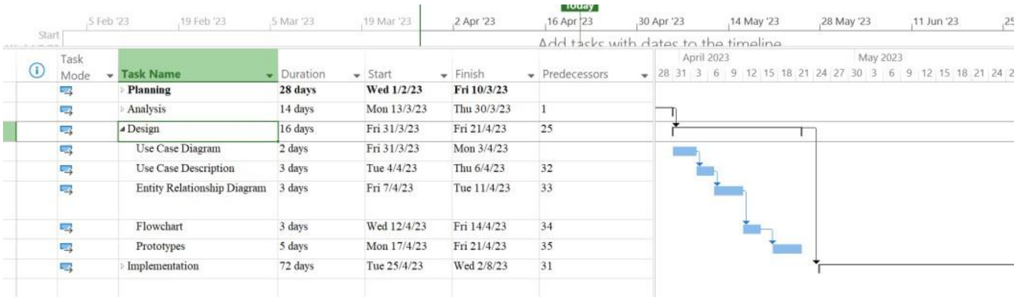
Task Mode	Task Name	Duration	Start	Finish	Predecessors
	Planning	**28 days**	**Wed 1/2/23**	**Fri 10/3/23**	
	Project Initiation	24 days	Wed 1/2/23	Mon 6/3/23	
	Study the subject matter	1 day	Wed 1/2/23	Wed 1/2/23	
	Identify the problem background	4 days	Thu 2/2/23	Tue 7/2/23	3
	Research on the problems	1 day	Thu 2/2/23	Thu 2/2/23	
	Speak to related authorities	1 day	Fri 3/2/23	Fri 3/2/23	7
	Analyze current practices	2 days	Mon 6/2/23	Tue 7/2/23	8
	Define project objectives	3 days	Wed 8/2/23	Fri 10/2/23	6
	Identify potential risks	1 day	Wed 8/2/23	Wed 8/2/23	
	Set goals of deliverable	2 days	Thu 9/2/23	Fri 10/2/23	11
	Generate project solution and approach	6 days	Mon 13/2/23	Mon 20/2/23	10
	Study existing solutions	3 days	Mon 13/2/23	Wed 15/2/23	
	Define the approach	3 days	Thu 16/2/23	Mon 20/2/23	14
	Create project proposal	1 day	Tue 21/2/23	Tue 21/2/23	13
	Identify the scope and cons	1 day	Tue 21/2/23	Tue 21/2/23	
	Literature review	9 days	Wed 22/2/23	Mon 6/3/23	16
	Study existing systems	3 days	Wed 22/2/23	Fri 24/2/23	
	Study methodologies for SDLC	3 days	Mon 27/2/23	Wed 1/3/23	19
	Study algorithms	3 days	Thu 2/3/23	Mon 6/3/23	20
	Project Management	4 days	Tue 7/3/23	Fri 10/3/23	2
	Generate a work breakdown structure	1 day	Tue 7/3/23	Tue 7/3/23	
	Generate Gantt Chart	3 days	Wed 8/3/23	Fri 10/3/23	23

Figure 3.4: Planning phase



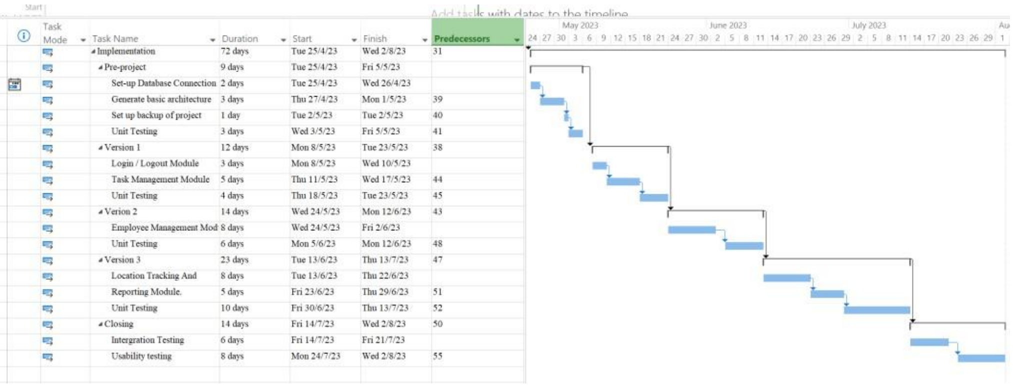
	Task Mode	Task Name	Duration	Start	Finish	Predecessors
25		Analysis	14 days	Mon 13/3/23	Thu 30/3/23	1
26		Requirement gathering	14 days	Mon 13/3/23	Thu 30/3/23	
27		Conduct survey	14 days	Mon 13/3/23	Thu 30/3/23	
28		Conduct interviews	1 day	Mon 13/3/23	Mon 13/3/23	
29		Gather feedback and opinions	1 day	Tue 14/3/23	Tue 14/3/23	28
30		Review and refine requirements	5 days	Wed 15/3/23	Tue 21/3/23	29

Figure 3.5: Analysis Phase



Task Mode	Task Name	Duration	Start	Finish	Predecessors
	Planning	**28 days**	**Wed 1/2/23**	**Fri 10/3/23**	
	Analysis	14 days	Mon 13/3/23	Thu 30/3/23	1
	Design	16 days	Fri 31/3/23	Fri 21/4/23	25
	Use Case Diagram	2 days	Fri 31/3/23	Mon 3/4/23	
	Use Case Description	3 days	Tue 4/4/23	Thu 6/4/23	32
	Entity Relationship Diagram	3 days	Fri 7/4/23	Tue 11/4/23	33
	Flowchart	3 days	Wed 12/4/23	Fri 14/4/23	34
	Prototypes	5 days	Mon 17/4/23	Fri 21/4/23	35
	Implementation	72 days	Tue 25/4/23	Wed 2/8/23	31

Figure 3.6: Design Phase



	Task Mode	Task Name	Duration	Start	Finish	Predecessors
		◢ Implementation	72 days	Tue 25/4/23	Wed 2/8/23	31
		◢ Pre-project	9 days	Tue 25/4/23	Fri 5/5/23	
		Set-up Database Connection	2 days	Tue 25/4/23	Wed 26/4/23	
		Generate basic architecture	3 days	Thu 27/4/23	Mon 1/5/23	39
		Set up backup of project	1 day	Tue 2/5/23	Tue 2/5/23	40
		Unit Testing	3 days	Wed 3/5/23	Fri 5/5/23	41
		◢ Version 1	12 days	Mon 8/5/23	Tue 23/5/23	38
		Login / Logout Module	3 days	Mon 8/5/23	Wed 10/5/23	
		Task Management Module	5 days	Thu 11/5/23	Wed 17/5/23	44
		Unit Testing	4 days	Thu 18/5/23	Tue 23/5/23	45
		◢ Verion 2	14 days	Wed 24/5/23	Mon 12/6/23	43
		Employee Management Mod	8 days	Wed 24/5/23	Fri 2/6/23	
		Unit Testing	6 days	Mon 5/6/23	Mon 12/6/23	48
		◢ Version 3	23 days	Tue 13/6/23	Thu 13/7/23	47
		Location Tracking And	8 days	Tue 13/6/23	Thu 22/6/23	
		Reporting Module.	5 days	Fri 23/6/23	Thu 29/6/23	51
		Unit Testing	10 days	Fri 30/6/23	Thu 13/7/23	52
		◢ Closing	14 days	Fri 14/7/23	Wed 2/8/23	50
		Intergration Testing	6 days	Fri 14/7/23	Fri 21/7/23	
		Usability testing	8 days	Mon 24/7/23	Wed 2/8/23	55

Figure 3.7: Implementation phase

CHAPTER 4

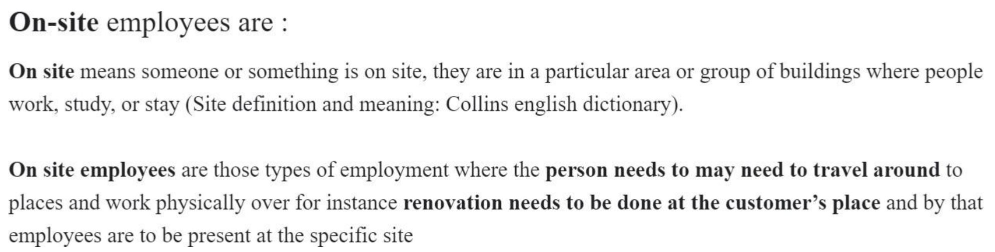
PROJECT SPECIFICATION

4.1 Introduction

A survey was conducted online to gain insights from individuals with onsite work experience. The survey was distributed among a select group of participants. Thanks to Google Forms, a feature was used that directs participants to specific sections of the survey based on their previous answers. The survey was divided into five sections, with the first section being the introduction and declaration. The second section required participants to provide their personal details, while the third section was aimed at assessing their familiarity with digital devices.

Both supervisors and employees were required to complete sections one, two, and three. However, the last question in section three directed participants to the appropriate section, either section four for supervisors or section five for employees. Undertaking this survey was of paramount importance as it enabled us to delve deeply into the existing practices, viewpoints, and onsite work experiences of both supervisors and employees. The ultimate objective of the survey was to obtain a comprehensive understanding of the target demographic and their encounters.

4.2 Section 1 – Introduction



On-site employees are :

On site means someone or something is on site, they are in a particular area or group of buildings where people work, study, or stay (Site definition and meaning: Collins english dictionary).

On site employees are those types of employment where the **person needs to may need to travel around** to places and work physically over for instance **renovation needs to be done at the customer's place** and by that employees are to be present at the specific site

Figure 4.1: Survey Introduction

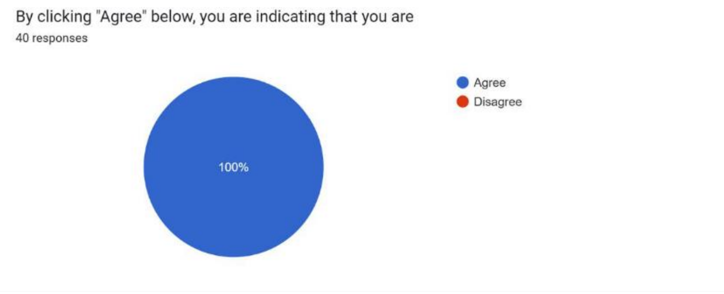
Example of on-site employees: Site employees

Example of on-site employees : Part time home cleaners

Example of on-site employees : Plumbers

Figure 4.2: Survey Introduction

The first section of the questionnaire provided a concise overview and explanation of onsite work, as well as several examples for respondents to recognize and relate to onsite employees. This was done to ensure that participants accurately comprehended the query and could convey their opinions in an appropriate manner. By delineating and illustrating the concept of onsite work, the survey intended to provide all respondents with a thorough understanding of the topic.



By clicking "Agree" below, you are indicating that you are

40 responses

Figure 4.3: Survey declaration

A crucial question in the survey requires respondents to designate whether they agree or disagree with a set of terms and conditions. By consenting, participants indicated they satisfy the required qualifications, which include being at least 18 years old, currently or formerly employed, and possessing onsite work experience. This query is necessary to guarantee that the survey results are accurate and pertinent. It functions as a filter to eliminate responses from individuals who do not satisfy the required criteria, which can distort the data and produce inaccurate results. By requiring respondents to consent to the terms and conditions, the survey is able to collect data from individuals pertinent to the study, resulting in insightful insights and conclusions. This query is crucial to ensuring the accuracy and dependability of the survey.

4.3 Section 2 - personal details

The second section of the report is aimed at collecting personal details of the respondents, which can help to understand any specific impact or influences on the work experiences and perspectives of the participants.

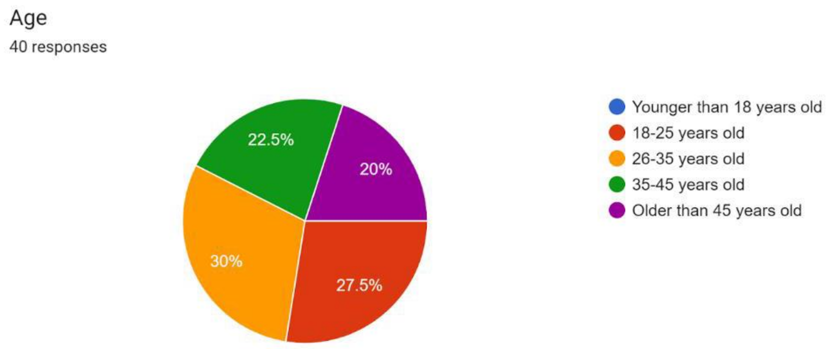


Figure 4.4: Age

Based on the figure above, the majority of the respondents are within the age range of 26-35 years old, comprising 30% or 12 out of 40 respondents. The minor age group is younger than 18 years old, which has no representation in the survey. The second-largest age group is between 18-25 years old, accounting for 27.5% or 11 out of 40 respondents. The third-largest age group is between 35-45 years old, with 22.5% or 9 out of 40 respondents. The last age group is older than 45 years old, with 20% or 8 out of 40 respondents. The findings suggest that the majority of the respondents are within the working-age range of 26-35 years old. This age group may have more experience and knowledge in the workforce, and their opinions and feedback may be valuable in evaluating the product or service.

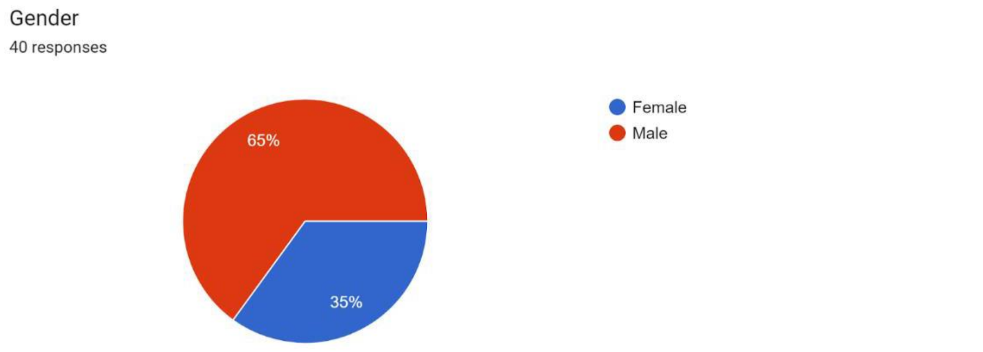


Figure 4.5: Gender

Based on the figure above, out of the 40 respondents, 35% or 14 were female and 65% or 26 were male. Therefore, the majority of the respondents were male while the minors were female.

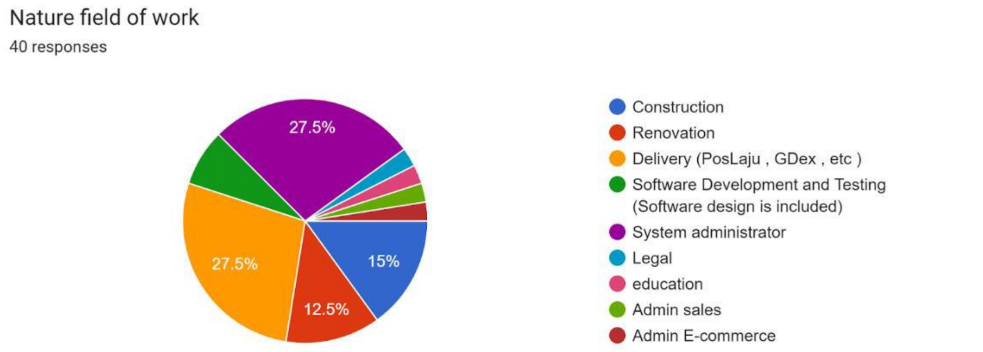


Figure 4.6: Nature field of work

Based on the figure 4.6 above, the majority of respondents work in delivery (PosLaju, GDex, etc.) with 11 respondents out of 40, which accounts for 27.5% of the total respondents. The minor group of respondents are in the legal field, with only one respondent out of 40, accounting for 2.5% of the total respondents. Out of the 40 respondents, 6 respondents work in construction (15%), 5 respondents work in renovation (12.5%), 11 respondents work in delivery (27.5%), 3 respondents work in software development and testing (7.5%), 11 respondents work in system administration (27.5%), and 1 respondent each work in legal and education fields (2.5% each). The findings indicate that delivery and system administration are the most common fields of work among the respondents, while legal and education fields are the least common. Therefore, these findings suggest that both the delivery and system administration sectors have potential implications for the study. The nature of work in these sectors is unique and requires specific approaches to handle the work-related issues effectively.

4.4 Section 3 - Familiarity with digital technology

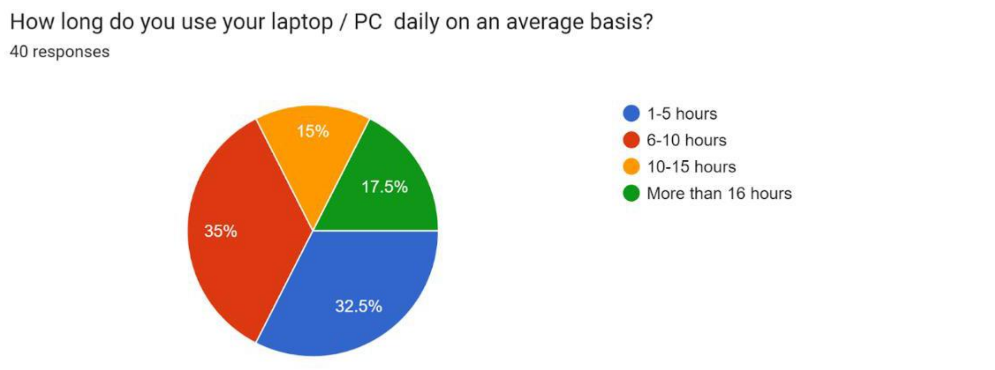


Figure 4.7: How long do you use your laptop / PC daily on an average basis?

Based on the figure above, the majority of respondents, comprising 35% or 14 out of 40 people, reported using their laptop or PC for 6-10 hours on an average basis. The next most popular option was 1-5 hours, chosen by 32.5% or 13 out of 40 respondents. The minority options were 10-15 hours, selected by 15% or 6 out of 40 respondents, and more than 16 hours, chosen by 17.5% or 7 out of 40 respondents. In conclusion, the survey results show that the majority of respondents use their laptop or PC for 6-10 hours daily, while roughly one-third use it for 1-5 hours. This information can be useful for understanding the needs and habits of individuals in terms of computer usage, which could be relevant in various contexts, such as technology design and marketing.

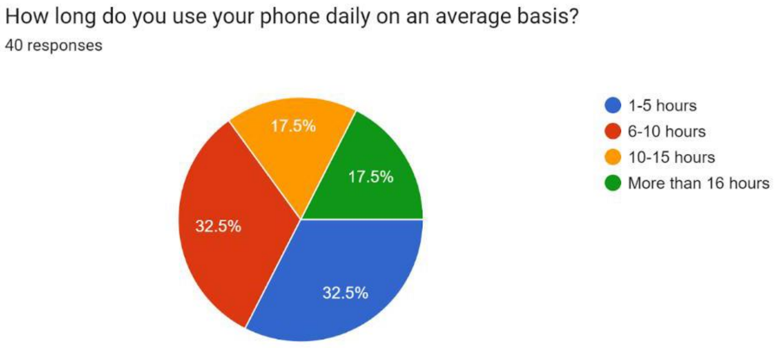


Figure 4.8: How long do you use your phone daily on an average basis?

Based on the figure above, the majority of respondents (32.5% or 13 out of 40) reported using their phone for 6-10 hours on an average basis. The second most popular option was using the phone for 10-15 hours, with 32.5% (13 out of 40) of respondents choosing this option as well. The least common options were using the phone for 1-5 hours, selected by 17.5% (7 out of 40) of respondents, and using the phone for more than 16 hours, chosen by another 17.5% (7 out of 40) of respondents. Based on the data presented, it seems that a significant portion of respondents use their phone for extended periods of time, with 50% reporting usage of 10 or more hours per day. Overall, this survey provides insight into the average daily phone usage of the respondents, with the majority indicating usage between 6-15 hours. This information has resulted in the phone applications being more of a preferred choice than laptops.

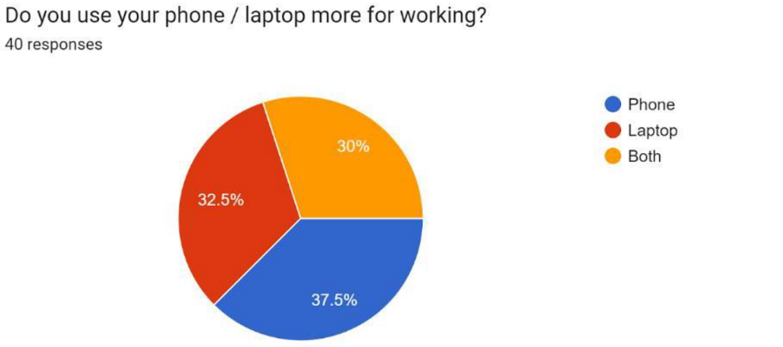


Figure 4.9: Do you use your phone / laptop more for working?

Based on the figure above, it can be observed that 37.5% or 15 out of 40 respondents use their phone more for working, while 32.5% or 13 out of 40 respondents use their laptop more for working. The third most common response was using both equally, with 30% or 12 out of 40 respondents selecting this option. From these findings, it can be inferred that a significant proportion of the respondents use both their phone and laptop for work, with a slightly higher percentage of respondents preferring to use their phone.

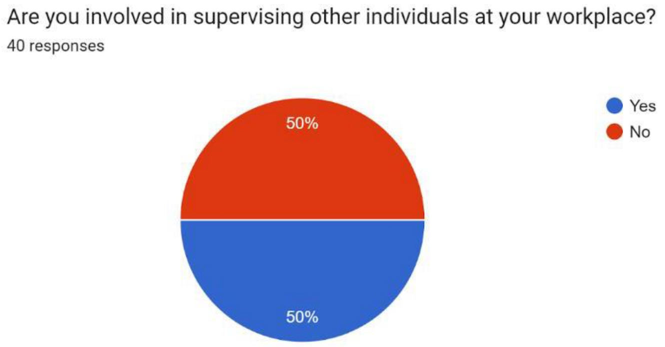


Figure 4.10: Are you involved in supervising other individuals at your workplace?

Based on the figure above, half of the 40 respondents, or 50%, indicated that they are involved in supervising other individuals at their workplace. On the other hand, the other half, or 50% of respondents, indicated that they are not involved in supervising others and are being supervised by someone else. The findings suggest that a significant proportion of the respondents are in supervisory positions at their workplaces, which may have implications for their responsibilities and job dutiesRespondents who have indicated that they are involved in supervising others at their workplace, which accounts for half or 50% of the total respondents, will be directed to proceed with section 4 of the survey to complete the remaining questions and submit their responses. The other half of the respondents, who selected no and therefore are employees being supervised, will be directed to section 5 of the survey, which is the final section for both supervisors and employees to complete. This division ensures that the questions in the survey are relevant and applicable to the respective roles and responsibilities of the respondents. It also allows for more targeted data analysis and interpretation in order to draw meaningful conclusions and insights for the final year project report.

4.5 Section 4 – Current Practice (Supervisors)

Section 4 aims to gather insights on the preferences and limitations to be considered during development, with a focus on identifying the most critical requirements that could determine high priority areas. Through this section, the project seeks to gain a comprehensive understanding of the factors that must be taken into account to ensure optimal performance and user satisfaction. The findings obtained from this section can serve as a crucial foundation for the subsequent stages of development and testing, allowing for a more targeted and effective approach to product design and improvement.

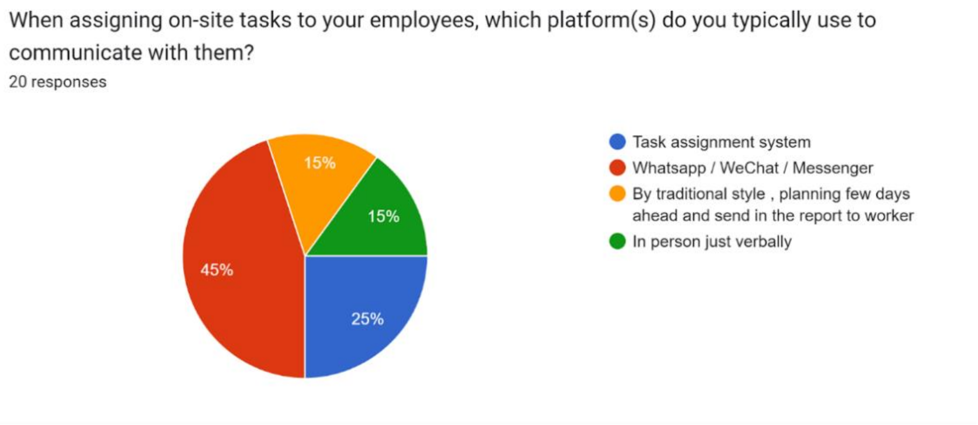


Figure 4.11: When assigning on-site tasks to your employees, which platform(s) do you typically use to communicate with them?

Based on the figure above, the most common platform used for assigning onsite tasks to employees is through Whatsapp/WeChat/Messenger, selected by 45% or 18 out of 40 respondents. The second most common option was the task assignment system, chosen by 25% or 10 out of 40 respondents. Meanwhile, 25% or 10 out of 40 respondents reported using the traditional method of planning a few days ahead and sending reports to workers, and only 15% or 6 out of 40 respondents preferred to communicate in person just verbally. From these findings, it can be concluded that a significant portion of supervisors prefer to use messaging platforms for assigning tasks to their employees.

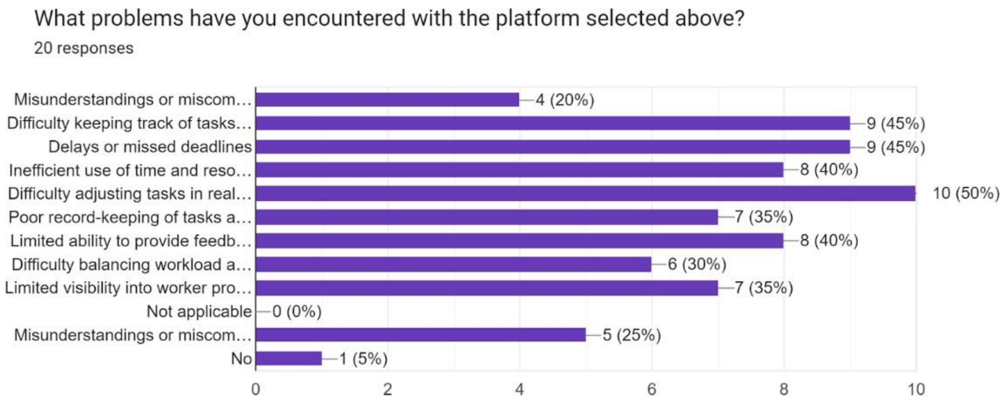


Figure 4.12: What problems have you encountered with the platform selected above?

Based on the figure above, the majority of respondents encountered difficulty adjusting tasks in real-time and reported a percentage of 50% or 10 out of 20 respondents. Meanwhile, the minority options were "Not applicable" with a percentage of 5% or 1 out of 20 respondents. Additionally, the percentages and whole numbers for the other options were as follows: Misunderstandings or miscommunications 45% or 9 out of 20 respondents, Difficulty keeping track of tasks and progress 45% or 9 out of 20 respondents, Delays or missed deadlines 45% or 9 out of 20 respondents, Inefficient use of time and resources 40% or 8 out of 20 respondents, Poor record-keeping of tasks and progress 35% or 7 out of 20 respondents, Limited ability to provide feedback 40% or 8 out of 20 respondents, Difficulty balancing workload among team members 30% or 6 out of 20 respondents, and Limited visibility into worker productivity or performance 35% or 7 out of 20 respondents. It is clear from the data that the majority of respondents encountered difficulty adjusting tasks in real-time while using the selected platform. This could indicate a need for a platform with better real-time communication and collaboration features. Additionally, the high percentages for misunderstandings or miscommunications, difficulty keeping track of tasks and progress, and delays or missed deadlines suggest that there is room for improvement in the platform's task management and communication features. Overall, these findings suggest that a more robust and efficient task management and communication platform may be needed to address the issues faced by the respondents.

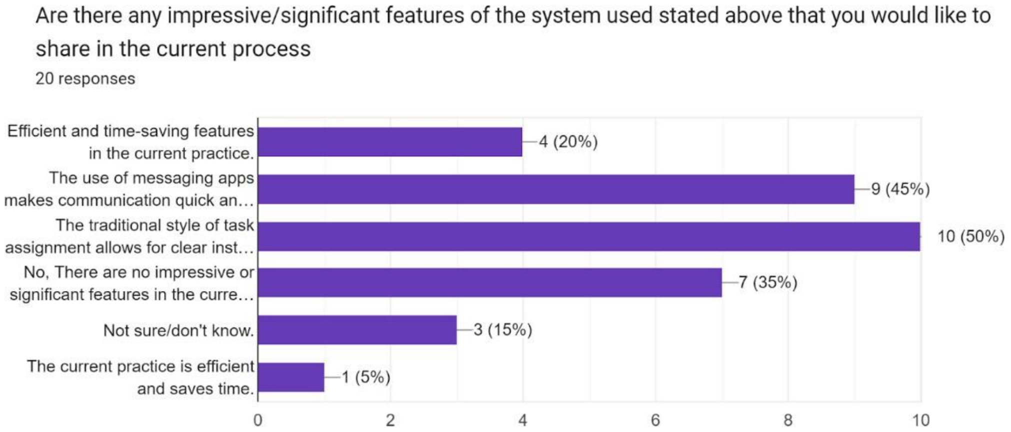


Figure 4.13: Are there any impressive/significant features of the system used stated above that you would like to share in the current process

Based on the figure above, the majority of respondents, with 50% or 10 out of 20, indicated that the traditional style of task assignment allowed for clear instructions and expectations. The second most common response was the use of messaging apps to make communication quick and easy, with 45% or 9 out of 20 respondents selecting this option. A quarter of the respondents, or 5 out of 20, noted efficient and time-saving features in the current practice. In contrast, 35% or 7 out of 20 respondents indicated that there were no impressive or significant features in the current practice, and 15% or 3 out of 20 were unsure or did not know. No respondents selected options that were not explicitly stated in the survey. In conclusion, the majority of respondents in this survey preferred the traditional style of task assignment, while the use of messaging apps for communication was also viewed favorably. The survey showed that only a few respondents felt that the current practice had significant features, and some were unsure or did not know. The findings suggest that it is essential to consider employees' preferences and needs when selecting and implementing systems to enhance work processes.

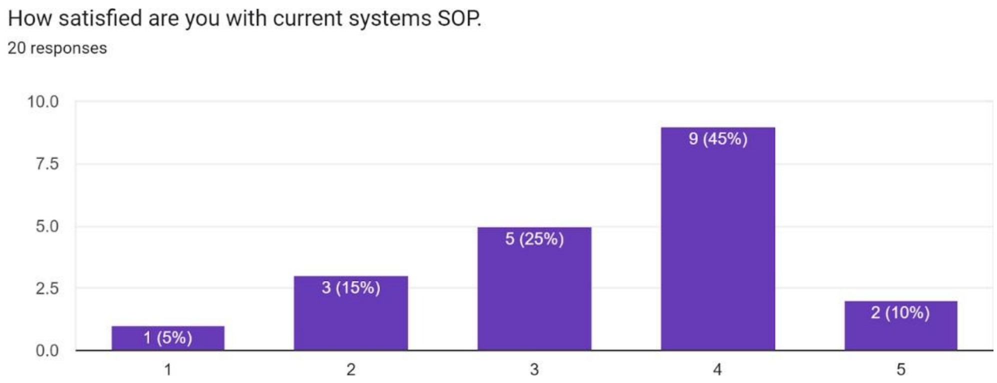
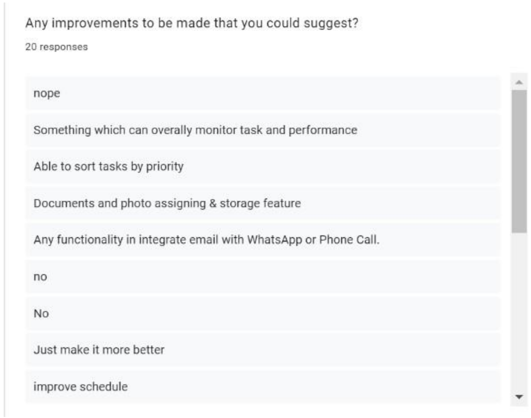


Figure 4.14: How satisfied are you with current systems SOP.

Based on the figure above, out of the 20 respondents, the majority of 9 respondents (45%) rated their satisfaction with the current systems SOP as 4, while the minority of 1 respondent (5%) rated it as 1. Additionally, 3 respondents (15%) rated it as 2, 5 respondents (25%) rated it as 3, and only 2 respondents (10%) rated it as 5. From the data presented, it can be concluded that the majority of the 20 respondents were generally satisfied with the current systems SOP, with almost half of them rating it as 4. However, a considerable percentage of respondents rated their satisfaction as 2 or 3, which may suggest some areas for improvement. The low percentage of respondents rating their satisfaction as 1 or 5 indicates that there are not significant issues or improvements that are outstandingly necessary. Overall, this data could be useful for the company to identify areas of the SOP that need improvement and areas that are performing well.



Any improvements to be made that you could suggest?

20 responses

nope

Something which can overally monitor task and performance

Able to sort tasks by priority

Documents and photo assigning & storage feature

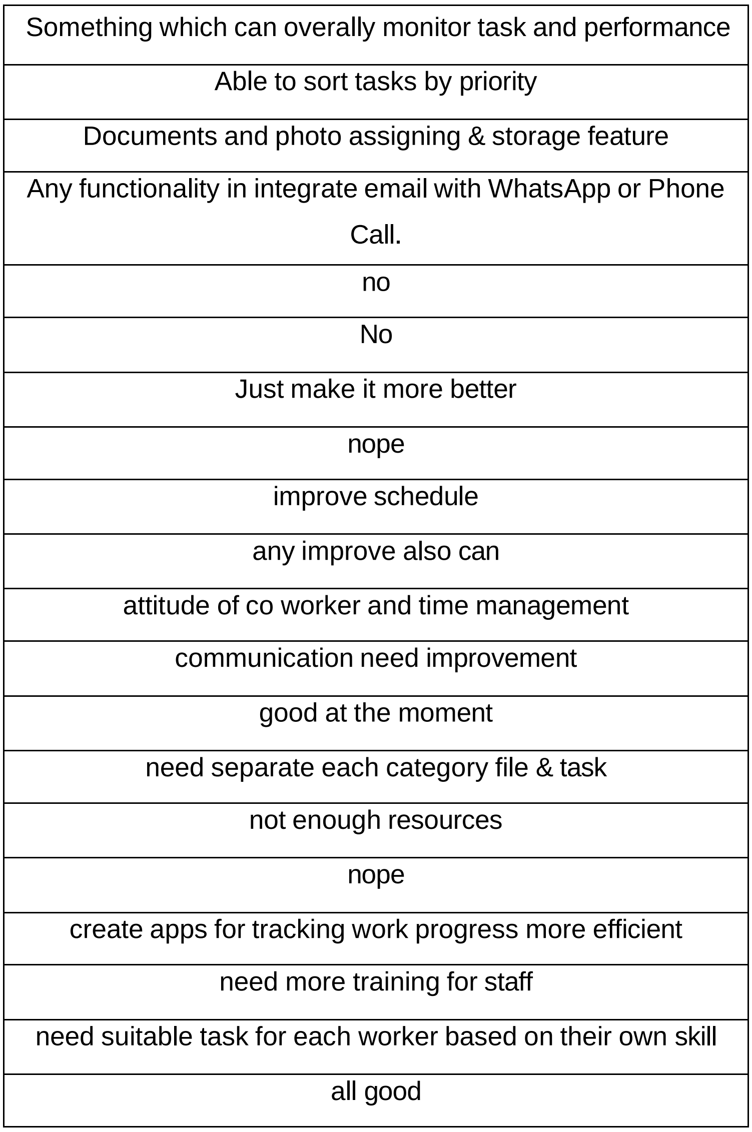
Any functionality in integrate email with WhatsApp or Phone Call.

no

No

Just make it more better

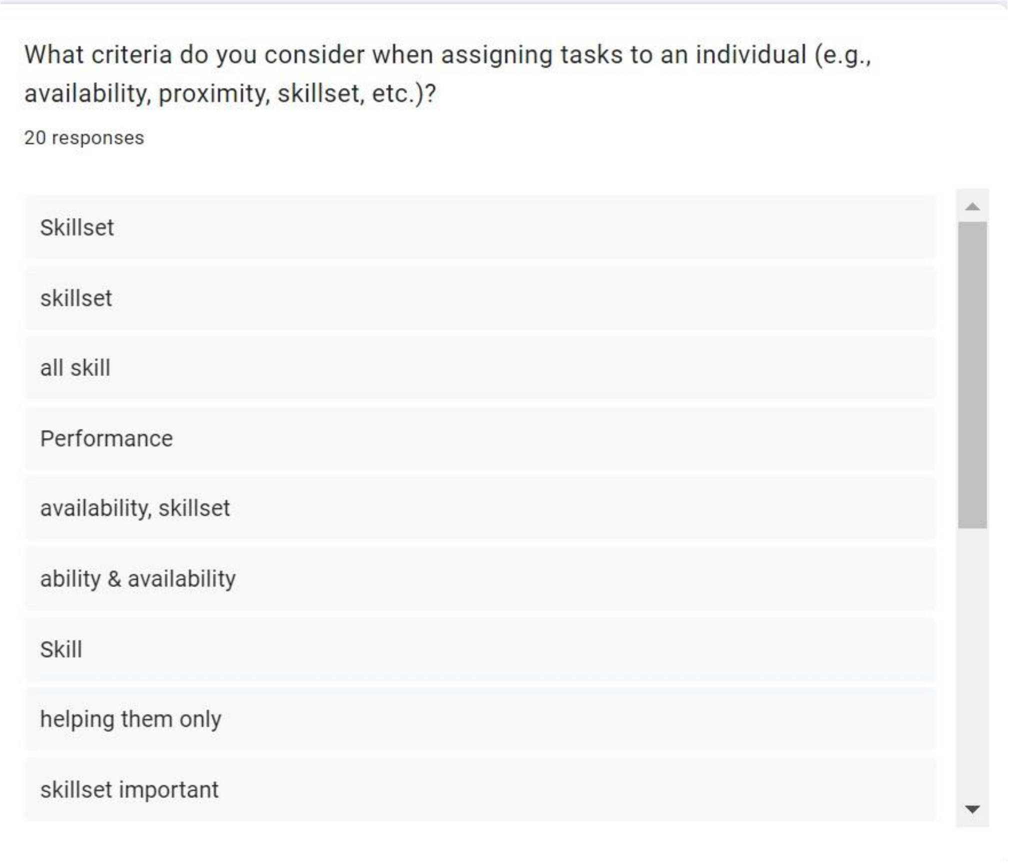
improve schedule



Something which can overally monitor task and performance
Able to sort tasks by priority
Documents and photo assigning & storage feature
Any functionality in integrate email with WhatsApp or Phone Call.
no
No
Just make it more better
nope
improve schedule
any improve also can
attitude of co worker and time management
communication need improvement
good at the moment
need separate each category file & task
not enough resources
nope
create apps for tracking work progress more efficient
need more training for staff
need suitable task for each worker based on their own skill
all good

Figure 4.15: Any improvements to be made that you could suggest?

Based on the figure 4.15 above, the majority of respondents, 30%, did not provide any suggestions for improvement to the current system used. Among those who did suggest improvements, the most common suggestion was the need for better task and performance monitoring, which was mentioned by 10% of respondents. Other suggestions included the ability to sort tasks by priority, document and photo assigning and storage features, integration of email with messaging apps like WhatsApp or phone call, and improving the scheduling function. Some respondents also mentioned the need for better communication, resources, and training for staff. A few respondents suggested creating an app for more efficient work progress tracking, and some suggested assigning tasks based on workers' skills. Overall, it appears that there is room for improvement in the current system, especially in the areas of task and performance monitoring, communication, and resource allocation.



What criteria do you consider when assigning tasks to an individual (e.g., availability, proximity, skillset, etc.)?

20 responses

Skillset

skillset

all skill

Performance

availability, skillset

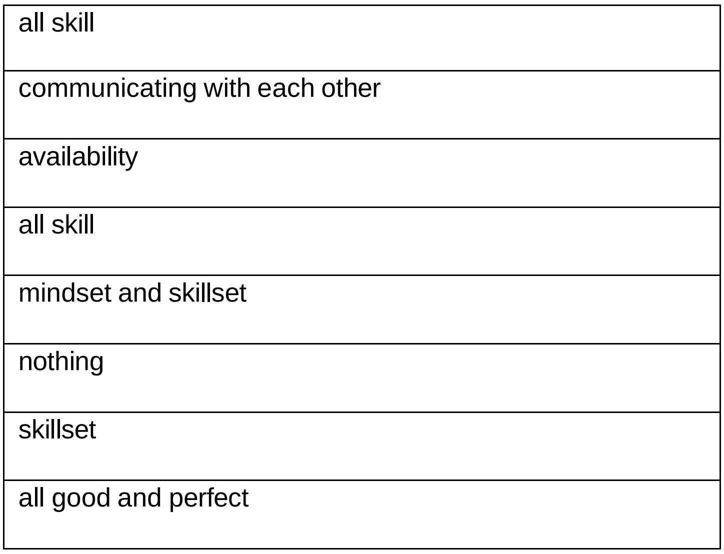
ability & availability

Skill

helping them only

skillset important

Performance
availability, skillset
ability & availability
Skillset
Skill
helping them only
Skillset
skillset
skillset important
not sure. maybe all criteria
time management
communication and skillset



all skill
communicating with each other
availability
all skill
mindset and skillset
nothing
skillset
all good and perfect

Figure 4.16: What criteria do you consider when assigning tasks to an individual (e.g., availability, proximity, skillset, etc.)?

Based on the data above, the responses to the question of what criteria individuals consider when assigning tasks can be summarized as follows: the majority of respondents prioritize skillset, with a total of 8 out of 20 respondents indicating this as a key factor. Availability was the second most common factor, with a total of 4 respondents stating it as an important consideration. Other criteria mentioned by respondents include performance, proximity, time management, communication, and mindset. One respondent stated that they consider all criteria when assigning tasks. Overall, the data suggests that skillset is the most important factor when assigning tasks, followed by availability. However, it is important to consider all relevant criteria when assigning tasks, as each individual task may require a different set of skills or considerations.

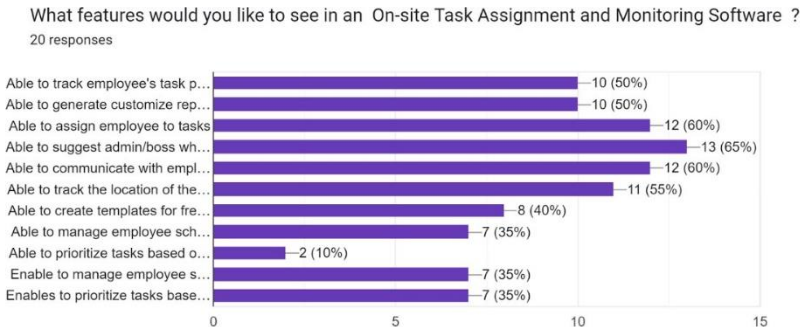


Figure 4.17: What features would you like to see in an On-site Task Assignment and Monitoring Software ?

Based on the figure above, out of 20 respondents who were asked about the features they would like to see in an onsite task assignment and monitoring software, the majority chose the ability to manage employee schedules and workload, with 14 respondents selecting this option (70%). The ability to assign employees to tasks was also a popular choice, with 12 respondents selecting this option (60%). The least popular options were the ability to create templates for frequently assigned tasks, with only 8 respondents selecting this option (40%), and the ability to prioritize tasks based on urgency or importance, with 9 respondents selecting this option (45%). Additionally, 10 respondents (50%) chose both the ability to track employee's task progress and the ability to generate customized reports, while 11 respondents (55%) selected the ability to track the location of the employee. It's worth noting that the survey did not include all possible features, and some respondents may have preferred features that were not listed. Overall, the data suggests that employees value software that can help them manage employee schedules and workload, assign employees to tasks, and communicate with employees in the software. supervisors should take note of these preferences when considering which software to implement for task assignment and monitoring.

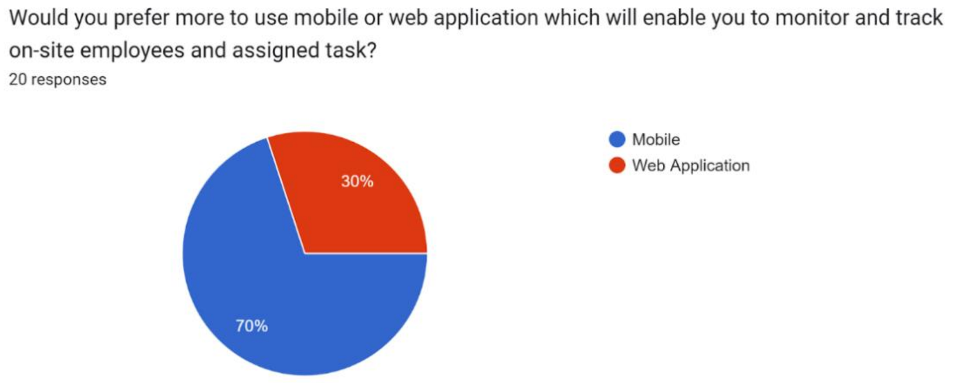


Figure 4.18: Would you prefer more to use mobile or web application which will enable you to monitor and track on-site employees and assigned task?

Based on the figure above, the majority of respondents, which is 70% or 14 individuals, prefer using a mobile application to monitor and track onsite employees and assigned tasks. The minority of respondents, which is 30% or 6 individuals, would rather use a web application. It is also worth noting that no respondents selected an option other than the two given choices. From these findings, it can be concluded that a significant proportion of respondents prefer using mobile applications for onsite employee monitoring and task tracking. This information is important for businesses or organizations developing such software as it can help guide them in deciding which platform to prioritize or develop more features for.

4.6 Section 5 - Current Practice (Supervisors)

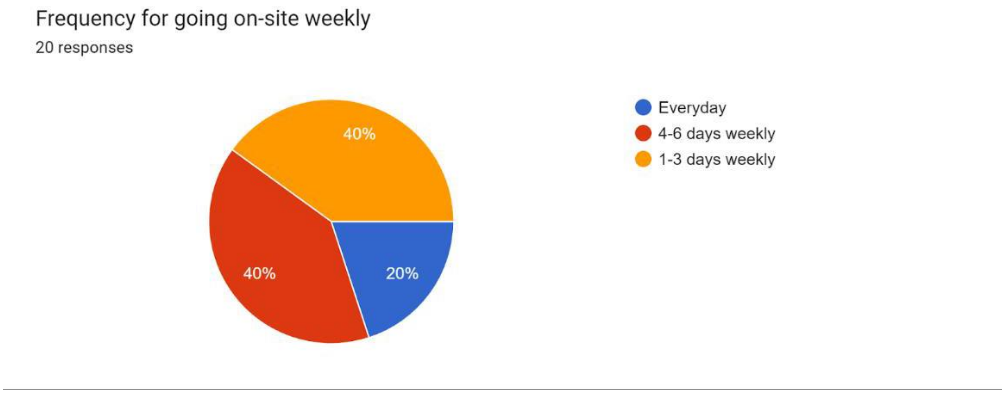


Figure 4.19: Frequency for going on-site weekly

Based on the figure 4.19 above, the majority of respondents, 40% or 8 individuals, indicated that they go onsite for work 4-6 days a week. An equal percentage of respondents, also 40% or 8 individuals, stated that they go onsite 1-3 days per week. The minority of respondents, 20% or 4 individuals, said that they go onsite every day. It is important to note that none of the respondents selected the option for "never" going onsite.

The data suggests that a significant portion of the respondents still need to go onsite for work at least once a week. With the current global situation, it is crucial to ensure the safety and well-being of these individuals while they are onsite. This highlights the importance of implementing proper safety protocols and monitoring systems to help mitigate the risk of exposure to the virus. Additionally, it may be useful to provide support and resources for remote work for those who can work from home, in order to limit the number of individuals who need to be onsite.

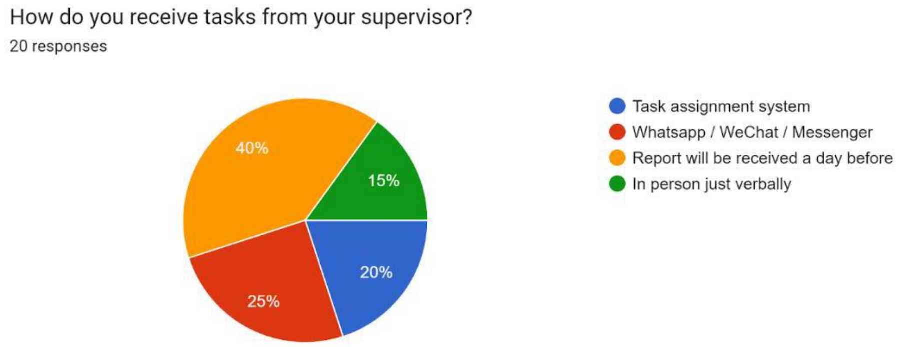


Figure 4.20: How do you receive tasks from your supervisor?

Based on the figure above, the most popular way that respondents receive tasks from their supervisor is by receiving the report a day before, with 40% of respondents choosing this option, which corresponds to 8 respondents. The next most popular method is through instant messaging platforms such as Whatsapp, WeChat, or Messenger, with 25% of respondents, which translates to 5 respondents. Task assignment systems were chosen by 20% of respondents (4 respondents), while receiving tasks in person just verbally was the least popular choice, selected by only 15% of respondents (3 respondents). The data shows that a significant proportion of respondents prefer to receive their tasks through receiving a report a day before. This method provides employees with time to review the tasks and prepare accordingly. It is also worth noting that instant messaging platforms are also popular among respondents, which may indicate the importance of communication between supervisors and employees. Overall, this data suggests that providing clear and timely communication, as well as allowing employees enough time to prepare for their assigned tasks, are important factors in managing employee tasks and productivity.

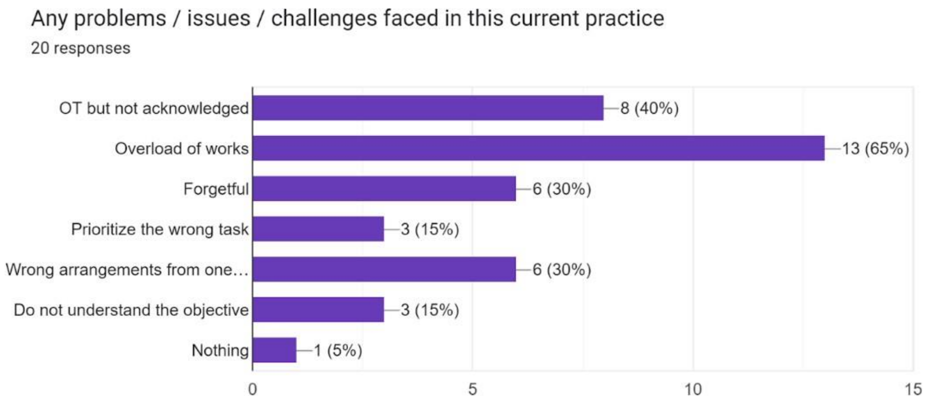


Figure 4.21: Any problems / issues / challenges faced in this current practice

Based on the figure above, the majority of respondents (65% or 13 out of 20) reported facing an overload of work as the biggest problem or challenge in the current task management practice. The second most common problem was forgetting tasks, which was reported by 30% (6 out of 20) of respondents. The other problems reported were prioritizing the wrong task (15% or 3 out of 20), wrong arrangements of one location to another (30% or 6 out of 20), and not understanding the objective (15% or 3 out of 20). Interestingly, 5% (1 out of 20) of respondents reported facing no problems or challenges. In addition, no respondents chose the option of "Other" or provided any responses that were not explicitly stated in the survey.

In conclusion, the data suggests that the biggest challenge faced in the current task management practice is an overload of work, followed by forgetting tasks. These findings highlight the need for more effective and efficient task management practices to alleviate the burden of an overload of work and minimize the incidence of forgetting tasks. This could involve the use of technology, such as task management software or mobile applications, or the implementation of clearer task assignment and prioritization processes to enhance overall productivity and job satisfaction among employees.

Are there any impressive/significant features of the system used stated above that you would like to share

20 responses

sometimes become driver , sometimes become warehouse staff

I love my job.. stress but it can improve my skill.. work hard & earn teamwork

don't have

no idea

report issues not properly handle by SV

Tired with multitask

Task need to improve for all aspect

wrong task and asignment

Nope

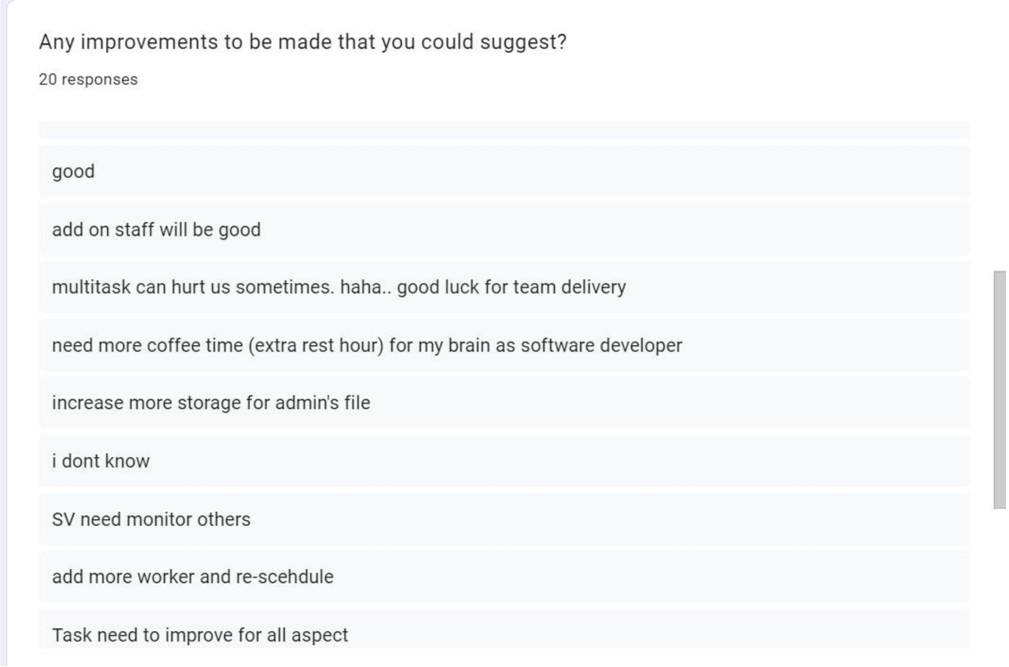
all good for us

All good
i dont know
sometimes multitask
tired but worth for salary
Wrong task, sometimes need to ot
super duper overload works need to finish before end of shift / day
good
alot of work
sometimes become driver , sometimes become warehouse staff
I love my job.. stress but it can improve my skill.. work hard & earn teamwork
don't have
no idea
report issues not properly handle by SV
Tired with multitask
Task need to improve for all aspect
wrong task and asignment
Nope
all good for us
for me, this company provide all we need
WORK OVERLOAD BUT SUPERVISOR VERY HELPFUL

Figure 4.22: Are there any impressive/significant features of the system used stated above that you would like to share

Based on the figure above, the responses to the question about any impressive/significant features of the system used show that there is no clear majority or consensus among the respondents. Out of the 20 responses, there were a variety of answers, ranging from "all good" and "I love my job" to "report issues not properly handled by SV" and "wrong task and assignment." Interestingly, 5% of respondents indicated that they had "no idea" about any impressive or significant features of the system. There were no responses that explicitly mentioned any impressive or significant features of the system.

From the data presented, it can be concluded that the respondents did not perceive any outstanding or remarkable features of the system used. Instead, the majority of the responses were focused on the challenges and issues that they faced in their current practice, such as work overload, forgetting tasks, and prioritizing the wrong task. This suggests that there may be room for improvement in the current system or process used, and that addressing these challenges and issues could potentially lead to increased job satisfaction and productivity among the respondents.



Any improvements to be made that you could suggest?

20 responses

good

add on staff will be good

multitask can hurt us sometimes. haha.. good luck for team delivery

need more coffee time (extra rest hour) for my brain as software developer

increase more storage for admin's file

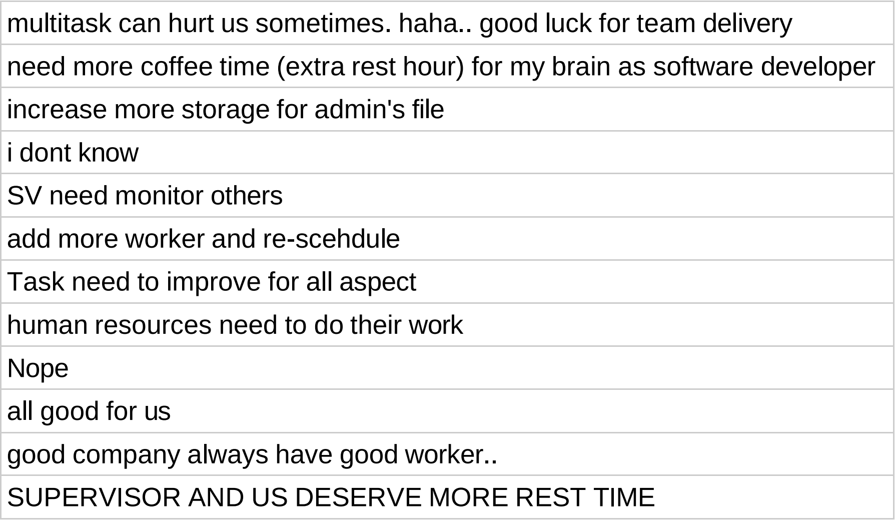
i dont know

SV need monitor others

add more worker and re-scehdule

Task need to improve for all aspect

No Improvement need
all suggestion for improvement
just focus 1 task for each person
Nope
improve schedule!!!
Please add more worker
good
add on staff will be good



multitask can hurt us sometimes. haha.. good luck for team delivery
need more coffee time (extra rest hour) for my brain as software developer
increase more storage for admin's file
i dont know
SV need monitor others
add more worker and re-scehdule
Task need to improve for all aspect
human resources need to do their work
Nope
all good for us
good company always have good worker..
SUPERVISOR AND US DESERVE MORE REST TIME

Figure 4.23: Any improvements to be made that you could suggest?

Based on the figure 4.23 above, the majority of respondents (40%) stated that no improvements are needed, while the minority (5%) expressed that human resources should do their work. Other responses include all suggestions for improvement (10%), focusing on one task for each person (5%), improving the schedule (5%), adding more workers (15%), monitoring others (5%), and improving tasks for all aspects (10%). It is also noteworthy that 5% of the respondents did not provide any suggestion for improvement. In terms of whole numbers, out of the 20 respondents, 8 stated that no improvements are needed, 2 did not provide any suggestion, and the remaining 10 respondents provided various suggestions for improvement. In conclusion, the data indicates that the majority of respondents are satisfied with the current system and do not see any need for improvements. However, there are still some who suggest adding more workers, improving tasks and schedules, and monitoring others. It is essential for the management to carefully consider these suggestions and make the necessary changes to improve the system and overall employee satisfaction.

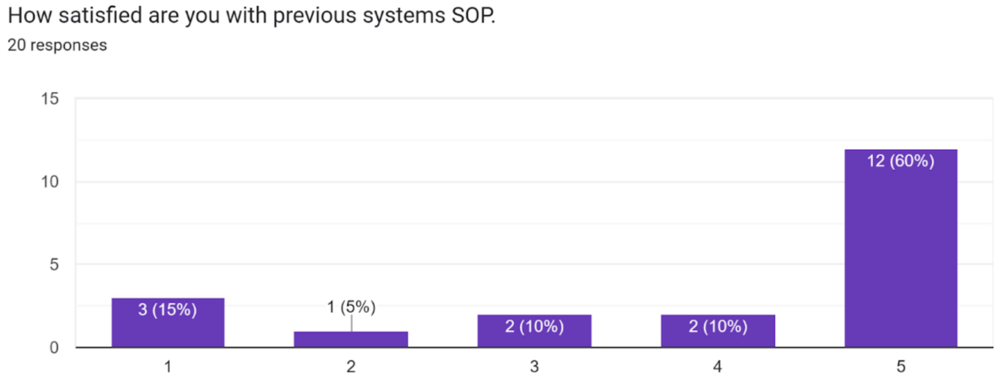


Figure 4.24: How satisfied are you with previous systems SOP.

Based on the figure above, the majority of respondents (60%) were "very satisfied" with the previous systems SOP, with a whole number of 12 respondents out of 20. The minority of respondents (5%) were "very dissatisfied" with the previous systems SOP, with a whole number of 1 respondent out of 20. Additionally, 10% of the respondents gave a rating of 4, while another 10% gave a rating of 3. Only 5% of respondents gave a rating of 2. It is also worth noting that 15% of respondents chose not to provide a rating. Overall, the data suggests that the majority of respondents were satisfied with the previous systems SOP, with a significant number of respondents giving the highest rating possible. However, a small minority expressed their dissatisfaction, which could suggest room for improvement in the system. It is also important to note the percentage of respondents who chose not to provide a rating, as it could be an indication of apathy or lack of interest in the topic.

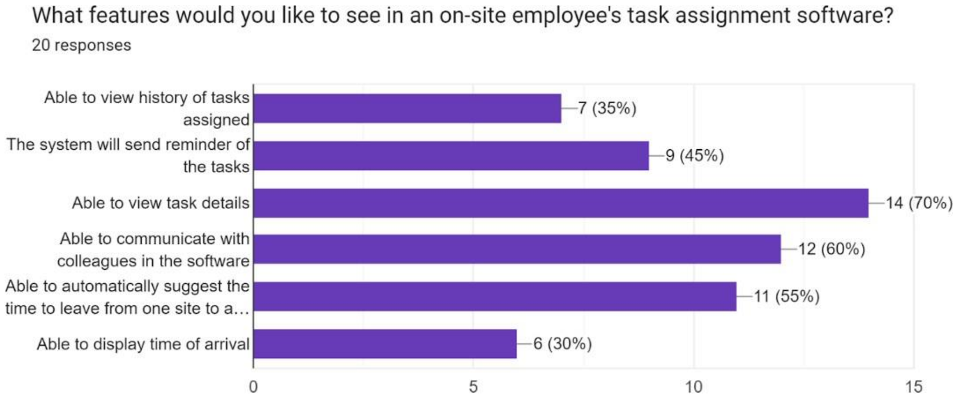


Figure 4.25: What features would you like to see in an on-site employee's task assignment software?

Based on the figure above, out of 20 respondents asked about what features they would like to see in an onsite employee's task assignment software, the majority (70% or 14 respondents) said they would like to be able to view task details, while the minority (30% or 6 respondents) said they would like to be able to display time of arrival. Additionally, 35% or 7 respondents said they would like to be able to view the history of tasks assigned, 45% or 9 respondents said they would like the system to send reminders of the tasks, 60% or 12 respondents said they would like to be able to communicate with colleagues in the software, and 55% or 11 respondents said they would like the software to automatically suggest the time to leave from one site to another. Overall, the data shows that a majority of respondents would like to be able to view task details in an onsite employee's task assignment software, while other desired features include the ability to view task history, receive task reminders, communicate with colleagues in the software, and receive suggestions for travel time between sites. These findings suggest that employees value having a comprehensive and efficient task management system in place, and that incorporating these features in a software could lead to increased satisfaction and productivity among employees.

4.7 Requirement Specification

4.7.1 Functional Requirements

General

1. The application shall allow users to login and logout
2. The application shall allow users to register
3. The application shall allow both employees and employers to communicate in the specific task.

Supervisors

1. The application shall allow supervisors to invite employees into the workspace
2. The application shall allow supervisors to attach skillset and/or labels onto employees.
3. The application shall allow supervisors to create workspace and task.

4. The application shall allow supervisors to assign employees to the task.
5. The application shall suggest whom to assign on to the project based on best case
6. The application shall able to track the employee's location as long as they are in their working hours and have checked in the application.
7. The application shall generate performance report which computes the completed task, delayed task, still in progress task at 1155pm daily

Employees

1. The application shall allow employee to view the tasks assigned.
2. The application shall request employees to let clients sign once the employees update the task to completed status.
3. The application shall request to upload a picture of the completed sub-task once employees update the task to completed status.
4. The application shall send a notification to employees to remind them the next upcoming task.

4.7.2 Non functional Requirements

Availability

1. The application shall be available during 5am to 12am
2. The application shall update and maintain during 12am to 5am

Security

1. The application shall authenticate and prevent unauthorised user access The application shall ensure the roles of the user
2. The application shall verify that the user has an account

Usability

1. The application shall be easily understood after 3 attempts to explore all features
2. The application shall be easy to use.

Compatibility

1. The application shall be compatible with android version 4.5 and above

Performance

1. The application shall load request within 3 seconds
2. The application shall be responsive within 1 minute of loading more than 1GB of data

4.8 Use case diagram

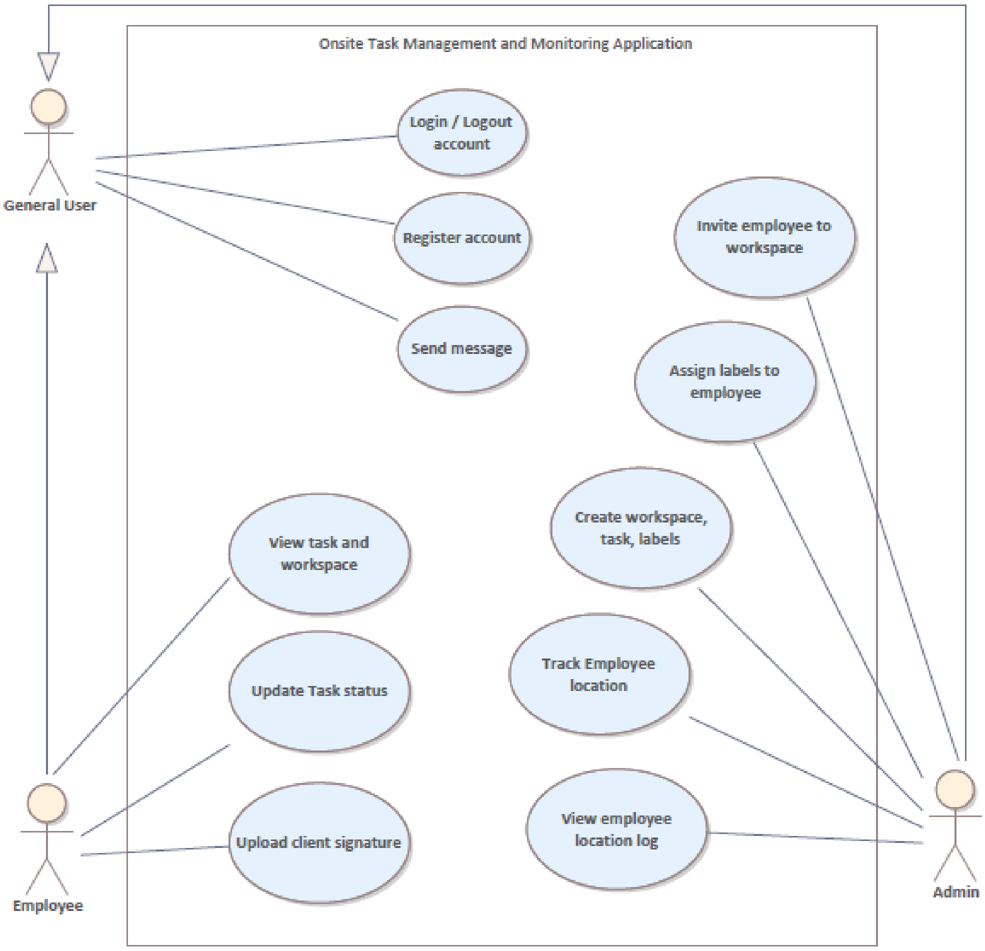


Figure 4.26: Use case diagram

4.8.1 Use case description

Table 4.1: Login / Logout Use Case Description

Use Case Name: Login /Logout	ID: G1	**Importance Level: High**
Primary Actor: User	Use Case Type: Detail and Real	
Stakeholders and Interests: User - has the intention to login into account to access the application or logout from account		
Brief Description: This use case describes the process of allowing users to login and logout of the application.		
Trigger: The user wants to access their account information and perform their desired actions on the application.		
Relationships: Association :User Include : Extend : Generalization:		
Normal Flow of Events: 1. The user accesses the application 2. The user navigate to login page 3. The user enters their username and password. 4. The system verifies the entered credentials and grants access to the user's account. 5. The system redirects the user to the main landing page of the application (base of admin / employee). 6. The user performs their desired actions on the application. 7. The user clicks on the "Logout" button. 8. The system securely closes the session and redirects the user to the login screen.		

Sub-flows:
Alternate/Exceptional Flows: 2a. If the user enters incorrect credentials, the system displays an error message and prompts the user to enter their credentials again. 2b. If the user forgets their password, the system provides a password reset option, which sends an email to the user with a link to reset their password.

Table 4.2: Register Account Use Case Description

Use Case Name: Register account	ID: G2	**Importance Level: High**
Primary Actor: User	Use Case Type: Detail and Real	
Stakeholders and Interests: User - register an account and use the services		
Brief Description: This use case describes how users can register with the application.		
Trigger: User wants to register an account and use the services		
Relationships: Association : Include : Extend : Generalization:		

Normal Flow of Events: 1. The system display a registration form for the user to fill. 2. The user fills in the required information such as name, email, and password. 3. The user submits the form to the system. 4. The system validates the user information to ensure that all required fields are filled and the email is in the correct format.. 5. The system then presents a success message to the user.
Sub-flows:
Alternate/Exceptional Flows: 5a. If the user submits the form with missing or invalid information, the system displays an error message indicating the missing/invalid fields and prompts the user to fill in the missing information. 5b. If the user enters an email address that is already registered with the application, the system displays an error message and direct to login.

Table 4.3: Send message into task

Use Case Name: Send messages	ID: G3	**Importance Level: High**
Primary Actor: Supervisors and Employees	Use Case Type: Detail and Real	
Stakeholders and Interests: Employees and employers - need to communicate with each other.		
Brief Description: This use case describes the steps involved in allowing employees and employers to communicate with each other within a specific task in the application.		
Trigger: An employee or an employer wants to communicate with each other regarding a specific task.		

Relationships: Association : Include : Extend : Generalization:
Normal Flow of Events: 1. The user selects the workspace 2. The user selects the tasks 3. The user selects the communication section 4. The user types the message and send 5. The message is sent into the task chat
Sub-flows:
Alternate/Exceptional Flows: 2a. If the user does not have access to the task, they will be prompted to contact their supervisor.

Table 4.4: Invite employees into workspace

Use Case Name: Invite employee into workspace	ID: S1	**Importance Level: High**
Primary Actor: Supervisors	Use Case Type: Detail and Real	
Stakeholders and Interests: Supervisor - to invite employees into the workspace		
Brief Description: This use case describes the process of how supervisors can invite employees to join a workspace within the application.		

Trigger: Supervisors wants to invite employee to a workspace .
Relationships: Association : Include : Extend : Generalization:
Normal Flow of Events: 1. Supervisor navigate to the staff list page 2. Supervisor key in the email and search 3. System display the user details 4. Supervisor select the workspace 5. Supervisor assign the employee into workspace
Sub-flows:
Alternate/Exceptional Flows: 3a. If the email address entered by the supervisor is invalid or does not match any user account, the system does not produce any search result.

Table 4.5: Assign labels to employee

Use Case Name: Assign labels to employee	ID: S2	**Importance Level: High**
Primary Actor: Supervisors	Use Case Type: Detail and Real	
Stakeholders and Interests: Supervisor- wants to categorise employees based on skillset and labels		

Brief Description: This use case describes the process of attaching skillsets and/or labels to employees by supervisors.
Trigger: Supervisor wants to assign tasks to employees based on their specific skills and/or expertise.
Relationships: Association : Include : Extend : Generalization:
Normal Flow of Events: 1. Supervisor selects the employee form staff list 2. The supervisor selects the skillset and/or label they want to attach. 3. The application saves the skillset and/or label to the employee's profile.
Sub-flows:
Alternate/Exceptional Flows:

Table 4.6: Create workspace and task Use Case Description

Use Case Name: Create workspace and task	ID: S3	**Importance Level: High**
Primary Actor: Supervisors	Use Case Type: Detail and Real	
Stakeholders and Interests: Supervisors - have the intention to create a new workspace or task or subtask		

Brief Description: This use case describes the process of creating workspaces, tasks, and subtasks by supervisors within the application.

Trigger: The supervisor needs to create a new workspace, task or subtask within an existing workspace.

Relationships:

Association :

Include :

Extend :

Generalization:

Normal Flow of Events:

1. The supervisor navigates to base page
2. Supervisor selects the create
3. Supervisor selects create a workspace
4. Supervisor writes the workspace name
5. Supervisor selects create task
6. Supervisor fills up task details

Sub-flows:

Alternate/Exceptional Flows:

4a / 6a. If the supervisor tries to create a workspace or task without entering the required information, an error message will be displayed, and the supervisor will need to enter the missing information before proceeding.

Table 4.7: Assign labels to employee Use Case Description

Use Case Name: Assign labels to employee	ID: S4	**Importance Level: High**
Primary Actor: Supervisors	Use Case Type: Detail and Real	
Stakeholders and Interests: Supervisor - intention to assign tasks to employee		
Brief Description: This use case describes the process by which a supervisor can assign labels to employees.		
Trigger: The supervisor selects a task or subtask that needs to be completed and assign to employee		
Relationships: Association : Include : Extend : Generalization:		
Normal Flow of Events: 1. Supervisor navigate to staff list and selects a staff 2. Supervisor selects thelabel to assign him to 3. The system displays a list of available employees who have the necessary skillset and is available. 4. The supervisor selects the desired employee(s) to be assigned to the task/subtask. 5. The system updates the task/subtask with the selected employee(s).		
Sub-flows:		

Alternate/Exceptional Flows:

Table 4.8: Track employee location Use Case Description

Use Case Name: Track employee location	ID: S7	**Importance Level: High**
Primary Actor: Supervisors	Use Case Type: Detail and Real	
Stakeholders and Interests: Supervisor - wants to know where are the employees located at		
Brief Description: This use case describes the functionality of the application to track an employee's location when they are within their working hours and have checked in to the application.		
Trigger: The trigger for this use case is a supervisor has the intention to know whether employee is at the accurate work location		
Relationships: Association : Include : Extend : Generalization:		
Normal Flow of Events: 1. The supervisor select the employee 2. The supervisor navigate to location section 3. The supervisor selects the map 4. System will display the current location of employees		

Sub-flows:
Alternate/Exceptional Flows:

Table 4.9: View employee location log Use Case Description

Use Case Name: View Employee Location Log	ID: S8	**Importance Level: High**
Primary Actor: Supervisors	Use Case Type: Detail and Real	
Stakeholders and Interests: Supervisor - has the intention to view location log		
Brief Description: This use case describes the process of view employee’s location log		
Trigger: The trigger for this use case is the system clock reaching 11:55 PM daily.		
Relationships: Association : Include : Extend : Generalization:		

Normal Flow of Events: 1. The supervisor select the employee 2. The supervisor navigate to location section 3. System will display the location log of employees
Sub-flows:
Alternate/Exceptional Flows:

Table 4.10: View workspace and task Use Case Description

Use Case Name: View workspace and task	ID: E1	**Importance Level: High**
Primary Actor: Employee	Use Case Type: Detail and Real	
Stakeholders and Interests: Employees - to view task assigned to them		
Brief Description: This use case describes how an employee can view the tasks assigned to them in the application.		
Trigger:The trigger for this use case is the employee logging into the application and navigating to the tasks section.		

Relationships: Association : Include : Extend : Generalization:
Normal Flow of Events: 1. The employee navigates to the workspace list section. 2. The application displays a list of tasks assigned to the employee. 3. The employee can click on a task to view its details. 4. The employee can mark a task as complete if it has been finished.
Sub-flows:
Alternate/Exceptional Flows: 2a. If there are no tasks assigned to the employee, the application displays a message indicating that there are no tasks currently assigned to them.

Table 4.11: Upload client signature

Use Case Name: Upload client signature	ID: E2	**Importance Level: High**
Primary Actor: Employee	Use Case Type: Detail and Real	
Stakeholders and Interests: Employee - ensure clients acknowledged the completion of task		

Brief Description: This use case describes the process of requesting clients to sign off on completed tasks. The application shall enable employees to request clients to sign off on a completed task once they have updated its status to "completed."
Trigger: The trigger for this use case is an employee updating a task's status to "completed."
Relationships: Association : Include : Extend : Generalization:
Normal Flow of Events: 1. The employee updates a task's status to "completed." 2. The application sends a reminder for signature request to the employee associated with the completed task. 3. The employee navigate to the signature section and let clients sign
Sub-flows:
Alternate/Exceptional Flows: 3a. This action is fully optional depending whether employers require the signature of acknowledgement

Use Case Name: Update task status	ID: E3	**Importance Level: High**
Primary Actor: Employee	Use Case Type: Detail and Real	

Stakeholders and Interests: Employee - intention in submitting a picture of the completed sub-task.
Brief Description: This use case describes how staff update the task status
Trigger:
Relationships: Association : Include : Extend : Generalization:
Normal Flow of Events: 1. The employee updates the task to intended status
Sub-flows:
Alternate/Exceptional Flows: 1a. I staff uploads employee signature it can also update the tsk directly to completed

CHAPTER 5

SYSTEM DESIGN

5.1 Introduction

This chapter will discuss the system architecture, database architecture, data flow architecture, high fidelity prototype, and interface flow diagram of the system.

The system architecture describes the overall structure of the system, including the front-end and back-end components. The front-end is the part of the system that users interact with, while the back-end is the part of the system that performs the actual processing and data storage. The database architecture describes how the data in the system is organized and stored. This includes the entity-relationship diagram (ERD) and data dictionary. The ERD is a visual representation of the relationships between the different entities in the system, while the data dictionary provides detailed information about each entity, such as its attributes and data types. The data flow architecture describes how data moves through the system. This is typically represented using data flow diagrams (DFDs). DFDs show the different processes in the system, as well as the data that flows between them. The high fidelity prototype is a working model of the system that can be used to test and validate the design. It should be as close to the final product as possible, both in terms of functionality and appearance. The interface flow diagram shows the flow of data between the different components of the system. This includes both the front-end and back-end components, as well as any external systems that the system interacts with.

This chapter will provide a detailed overview of each of these topics, and explain how they all fit together to create a complete and functional system.

5.2 System Architecture

The system architecture is divided into three distinct layers, each of which plays a specific role in the application's overall operation. Flutter, which is supported by Dart, is initially the front-end layer. Within this layer, the BLoC (Business Logic Component) pattern ensures a distinct separation between presentation and business logic. This architectural decision improves the user interface's maintainability and scalability. Laravel, a comprehensive PHP framework, orchestrates the server-side operations at the back-end layer. MySQL, a well-known relational database system, manages the storage and administration of data. This architecture is notable for its ability to communicate to external APIs over the internet. Figure 5.1 may visualise the overview of the system workings as below.

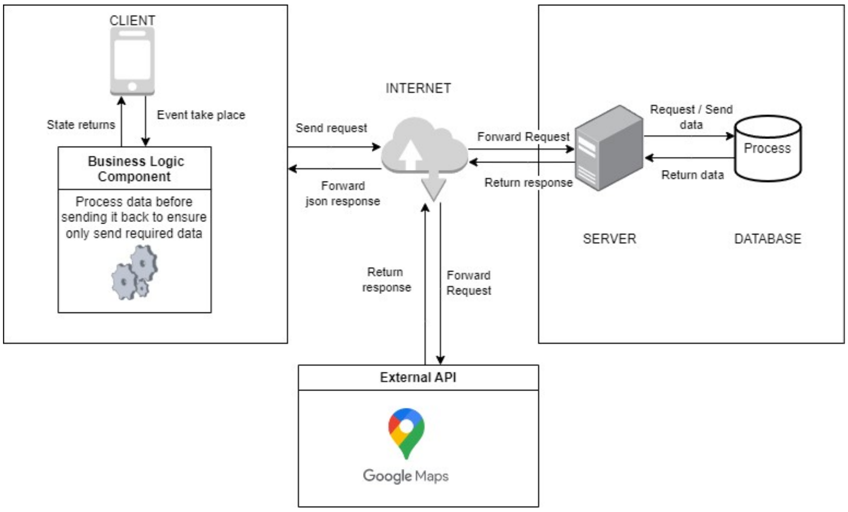
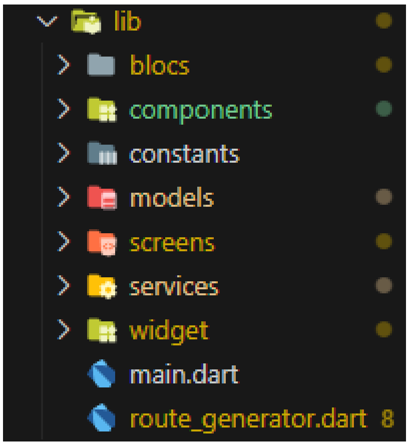


Figure 5.1: System Architecture

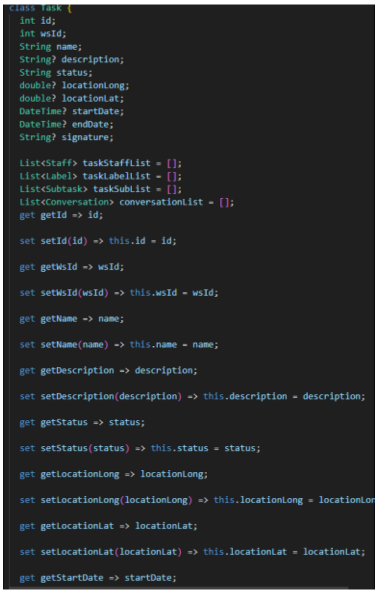
5.2.1 Front end architecture



```
lib
  blocs
  components
  constants
  models
  screens
  services
  widget
  main.dart
  route_generator.dart  8
```

Figure 5.2: Front End Directory

Flutter is an open-source user interface toolkit that is well-known for its ability to develop cross-platform apps from a single codebase. It is used as the front-end of the system. Flutter's core programming language is Dart, which combines the flexibility of quick UI prototyping with the robustness of a strongly typed, object-oriented language. This makes Dart an ideal choice for the development of Flutter.



```
class Task {
  int id;
  int wsId;
  String name;
  String? description;
  String status;
  double? locationLong;
  double? locationLat;
  DateTime? startDate;
  DateTime? endDate;
  String? signature;

  List<Staff> taskStaffList = [];
  List<Label> taskLabelList = [];
  List<Subtask> taskSubList = [];
  List<Conversation> conversationList = [];
  get getId => id;

  set setId(id) => this.id = id;

  get getWsId => wsId;

  set setWsId(wsId) => this.wsId = wsId;

  get getName => name;

  set setName(name) => this.name = name;

  get getDescription => description;

  set setDescription(description) => this.description = description;

  get getStatus => status;

  set setStatus(status) => this.status = status;

  get getLocationLong => locationLong;

  set setLocationLong(locationLong) => this.locationLong = locationLon

  get getLocationLat => locationLat;

  set setLocationLat(locationLat) => this.locationLat = locationLat;

  get getStartDate => startDate;
```

Figure 5.3: Object Oriented Structured

The Business Logic Component (BLoC) pattern, which can be found in the blocs directory, is an integral element of this front-end architecture. BLoC is more than a design pattern; it introduces a method for separating the presentation layer from the business logic of an application. Not only does this ensure a clean codebase, but it also makes it easier for developers to identify and modify specific logic sections without disrupting the UI layer. BLoC is in charge of state management within the Flutter ecosystem, ensuring a consistent and predictable user experience across the application.

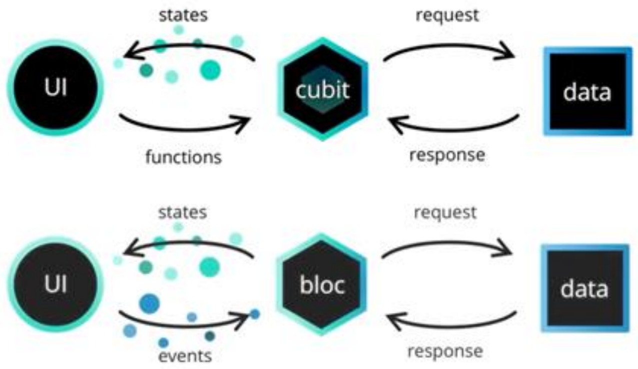


Figure 5.4: Getting Started with Flutter Bloc Pattern (Paramitha, 2021)

Interaction with multiple APIs is a fundamental capability of the system. These interactions are separated meticulously between component and service directories. Components are designed to manage interactions with external third-party data sources, ensuring that any data or functionality originating from the system's periphery is processed here. The latter, services, is solely concerned with interacting with the Laravel infrastructure. This distinction is crucial for both clarity and potential scalability, ensuring that the system's foundations remain stable as it grows.

The screens directory contains the primary user interfaces within the user interface. These are the tangible interfaces with which the end-users interact, equivalent to full-fledged pages or views within the application. Moreover, the widgets directory is a repository of reusable UI components, ensuring a DRY (Don't Repeat Yourself) approach to UI development (Haoyu & Haili, 2012). Here resides every component that is used across multiple displays, promoting code reuse and a consistent user experience.

The list of constants that are fundamental to a variety of applications. Serving as the central reservoir for values used ubiquitously throughout the application, it simplifies maintenance by providing a unified reference point. In parallel, the models directory delineates Dart classes to demystify the application's data structures. This abstraction is necessary for consistent data administration, particularly when interacting with multiple APIs.

route_generator.dart is responsible for navigation, the cornerstone of any user experience. This script provides a unified map for navigation from screen to screen. Centralising routing optimises navigation logic and provides a centralised reference point for any future modifications.

This front-end architecture based on Flutter exemplifies contemporary software design principles. It echoes modularity, clarity, and a sharp separation of concerns, ensuring that each system segment is not only identifiable but also modular and future-proof.

5.2.2 Back End Architecture

The backend architecture of this system is powered by Laravel, which is part of the architecture. Laravel, as is customary, adheres to the Model-View-Controller (MVC) architectural pattern. MVC is a design pattern that is frequently utilised in the development of online applications in order to compartmentalise the application's responsibilities. The interaction with the programme is broken up into three different parts that are all connected to one another by this pattern.

The Model is responsible for managing the data, the View is in charge of displaying it, and the Controller is in charge of the logic that integrates the two. This is the primary objective of the MVC design. However, in this design, the frontend functionality is handled by an external framework called Flutter. As a consequence of this design choice, the 'View' component of Laravel is not required. As a direct result of this, the primary duty of the Laravel backend is now to perform the duties associated with operating as an API-centric service. Instead of being responsible for supplying views, the backend is responsible for processing and serving data, most of the time in the JSON format. This division means that duties are clearly demarcated, with Flutter managing user interactions and visual presentations and Laravel focusing on data management and business logic respectively.

Controllers continue to be an essential component of the design, as they are tasked with controlling incoming HTTP requests. JSON Web Tokens (JWT) are functionalities that are encapsulated within these controllers, particularly the "Auth" controller. JSON Web Tokens serve as a token format that is both concise and URL-safe, making it easier for claims to be exchanged between two parties. A JWT is produced whenever user authentication is completed effectively. This provides a mechanism for further request validation.

Middleware provides an additional layer of processing for incoming HTTP requests. This processing layer enables particular actions or checks to be carried out prior to the request reaching its intended destination. Middleware can be thought of as an intermediary between the client and the server. Because JWT is now part of the middleware, you can rest assured that any request coming from the Flutter frontend will have a valid token attached to it. The validity of the token is determined throughout the process of validation. Should it be determined that the token is legitimate, the request will be processed; otherwise, an error response will be sent. A technique of this type would bring in an authentication layer, ensuring that system resources would continue to be accessible only to requests that have been authenticated.

This architecture, which places an emphasis on security, includes safeguards that instantly reject any efforts to gain unauthorised access. Any inbound request that either does not have a valid JWT or that attempts to tap into resources for which it does not have the requisite permissions will be immediately denied. By virtue of its structural architecture, the system is protected from any unauthorised entry that might be attempted, hence enhancing its security apparatus.

In most cases, the process will start when a user interacts with the Flutter frontend, which will cause a request to be made. This request is being sent to the backend of the Laravel framework. In the event that the request requires authentication, JWT is brought into action. The token that is connected to the request is checked for validity by the middleware. In the event that a valid token is present, the request will be routed to the relevant controller, which will then interface with the model in order to retrieve or modify the data. The data that was generated as a consequence is then delivered back to the front end in a manner that is structured and is prepared for presentation.

In conclusion, the backend architecture of this system, which is supported by Laravel, has been specifically designed to meet the requirements of a decoupled frontend framework called Flutter. The backend ensures that operations are efficient, secure, and scalable by placing an emphasis on structured data exchange, enforcing security using JWT, and establishing unambiguous communication paths.

5.3 Database Architecture

5.3.1 Entity Relationship Diagram

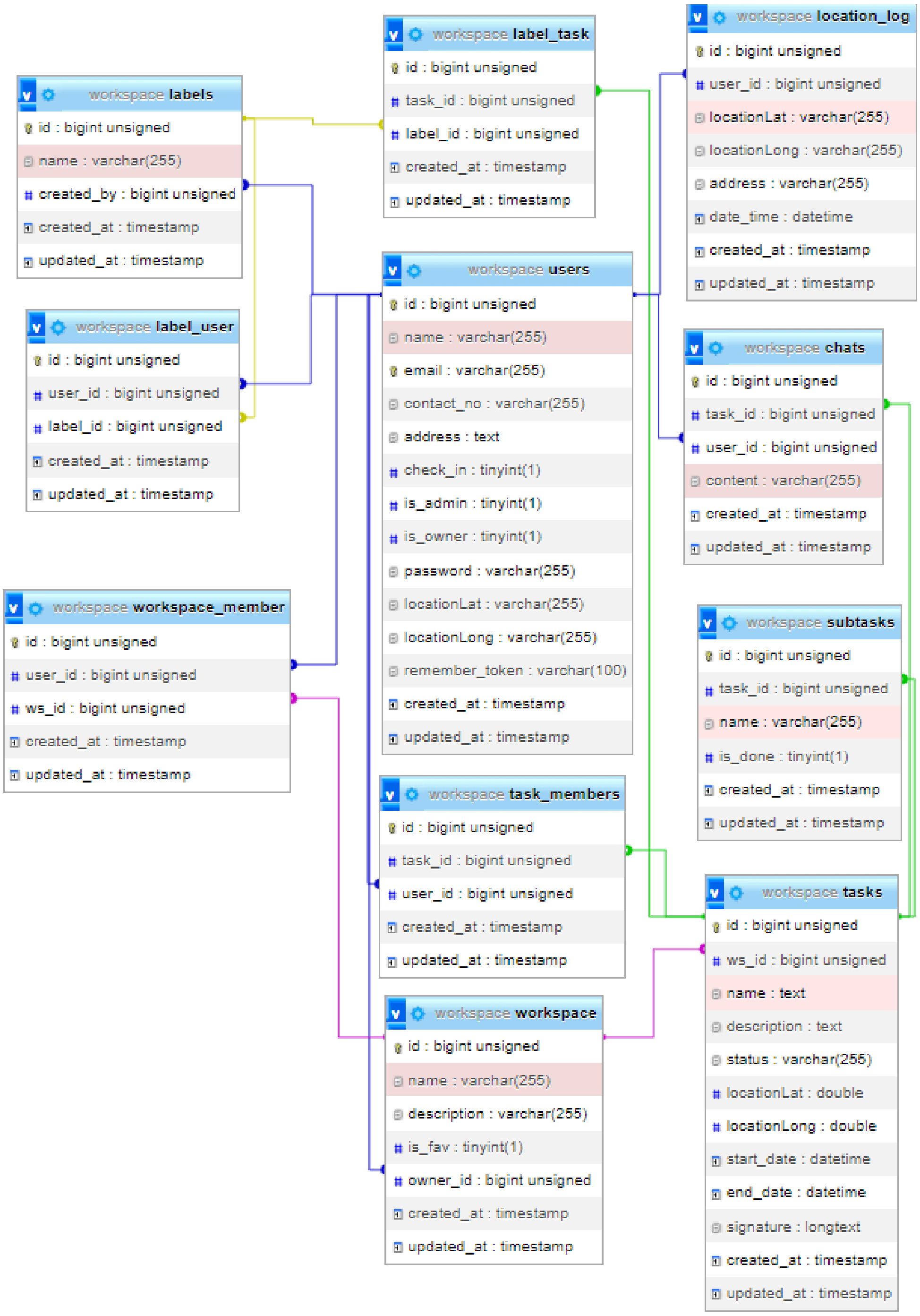
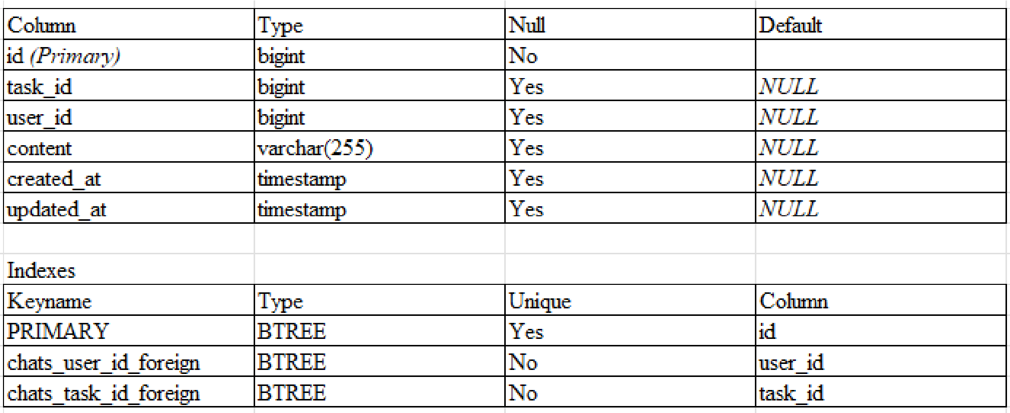


Figure 5.5: Physical Erd

5.3.2 Data Dictionary

Table name: chats

Table 5.1: Chat Data Dictionary Table



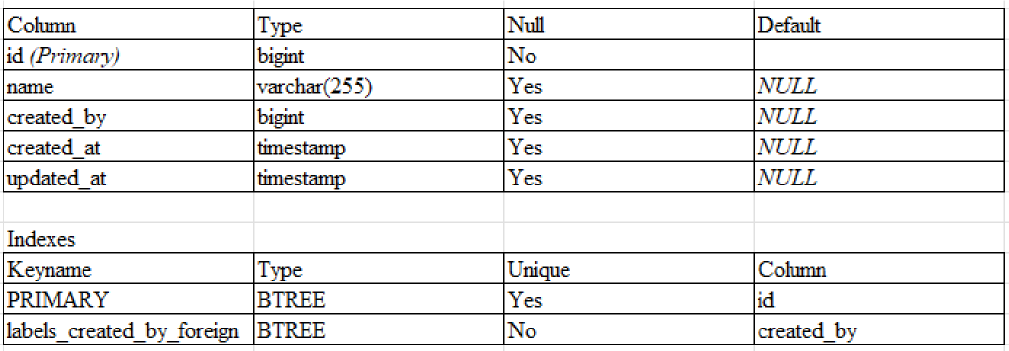
Column	Type	Null	Default
id *(Primary)*	bigint	No	
task_id	bigint	Yes	*NULL*
user_id	bigint	Yes	*NULL*
content	varchar(255)	Yes	*NULL*
created_at	timestamp	Yes	*NULL*
updated_at	timestamp	Yes	*NULL*

Indexes

Keyname	Type	Unique	Column
PRIMARY	BTREE	Yes	id
chats_user_id_foreign	BTREE	No	user_id
chats_task_id_foreign	BTREE	No	task_id

Table name : labels

Table 5.2: Labels Data Dictionary Table



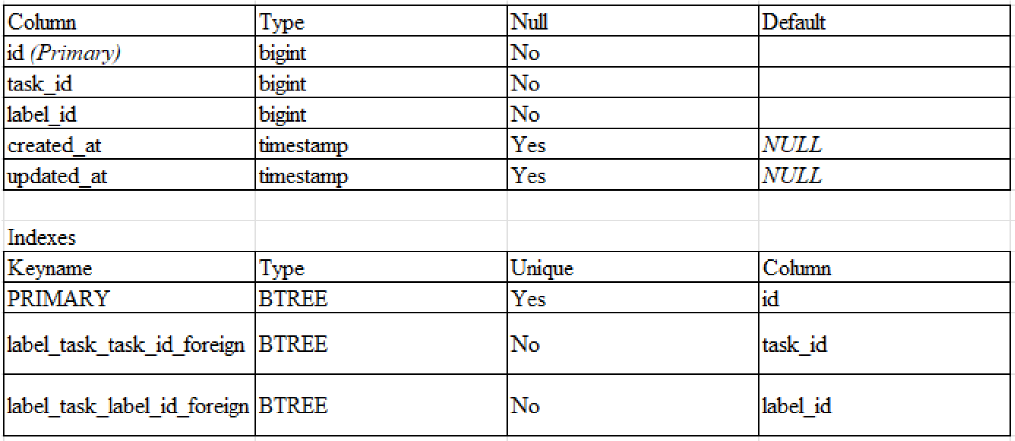
Column	Type	Null	Default
id *(Primary)*	bigint	No	
name	varchar(255)	Yes	*NULL*
created_by	bigint	Yes	*NULL*
created_at	timestamp	Yes	*NULL*
updated_at	timestamp	Yes	*NULL*

Indexes

Keyname	Type	Unique	Column
PRIMARY	BTREE	Yes	id
labels_created_by_foreign	BTREE	No	created_by

Table name: label_task

Table 5.3: Label-Task Data Dictionary Table



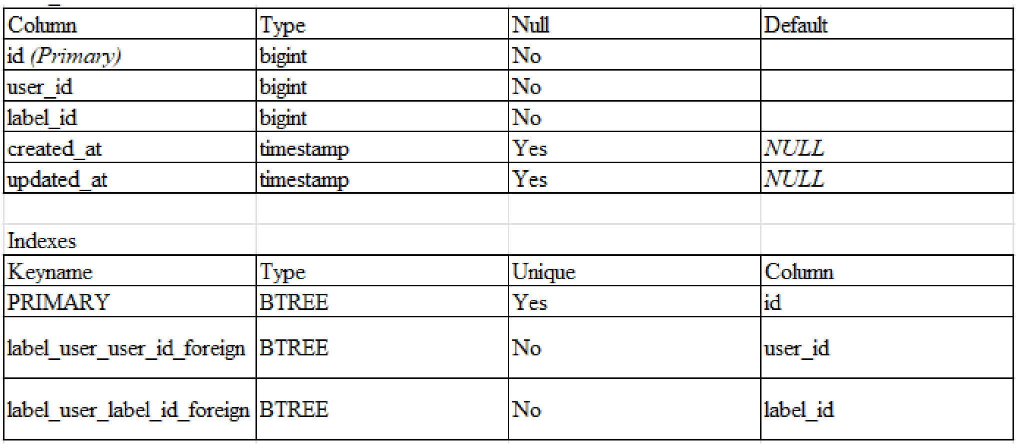
Column	Type	Null	Default
id *(Primary)*	bigint	No	
task_id	bigint	No	
label_id	bigint	No	
created_at	timestamp	Yes	*NULL*
updated_at	timestamp	Yes	*NULL*

Indexes

Keyname	Type	Unique	Column
PRIMARY	BTREE	Yes	id
label_task_task_id_foreign	BTREE	No	task_id
label_task_label_id_foreign	BTREE	No	label_id

Table name: label_user

Table 5.4: Label-User Data Dictionary Table



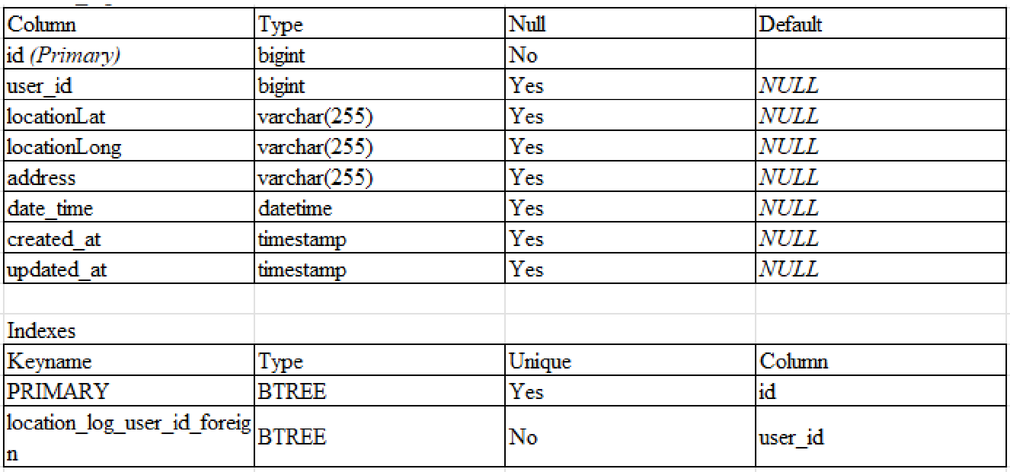
Column	Type	Null	Default
id *(Primary)*	bigint	No	
user_id	bigint	No	
label_id	bigint	No	
created_at	timestamp	Yes	*NULL*
updated_at	timestamp	Yes	*NULL*

Indexes

Keyname	Type	Unique	Column
PRIMARY	BTREE	Yes	id
label_user_user_id_foreign	BTREE	No	user_id
label_user_label_id_foreign	BTREE	No	label_id

Table name: location_log

Table 5.5: Location Log Data Dictionary Table



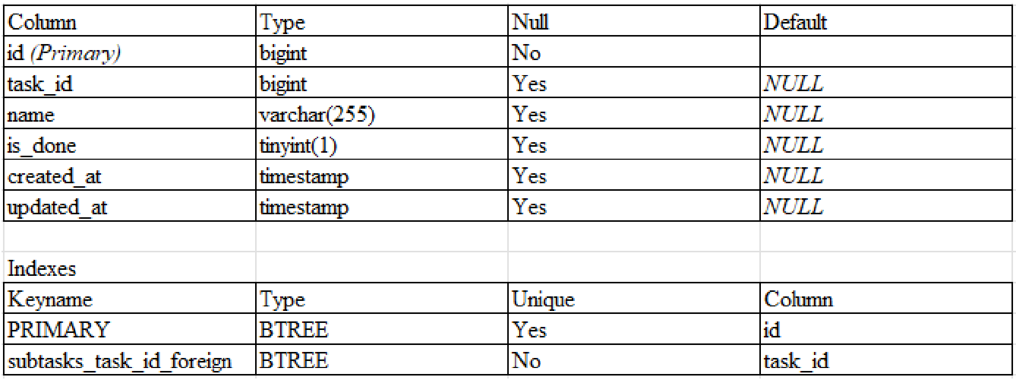
Column	Type	Null	Default
id *(Primary)*	bigint	No	
user_id	bigint	Yes	*NULL*
locationLat	varchar(255)	Yes	*NULL*
locationLong	varchar(255)	Yes	*NULL*
address	varchar(255)	Yes	*NULL*
date_time	datetime	Yes	*NULL*
created_at	timestamp	Yes	*NULL*
updated_at	timestamp	Yes	*NULL*

Indexes

Keyname	Type	Unique	Column
PRIMARY	BTREE	Yes	id
location_log_user_id_foreign	BTREE	No	user_id

Table name: subtasks

Table 5.6: Subtask Data Dictionary Table



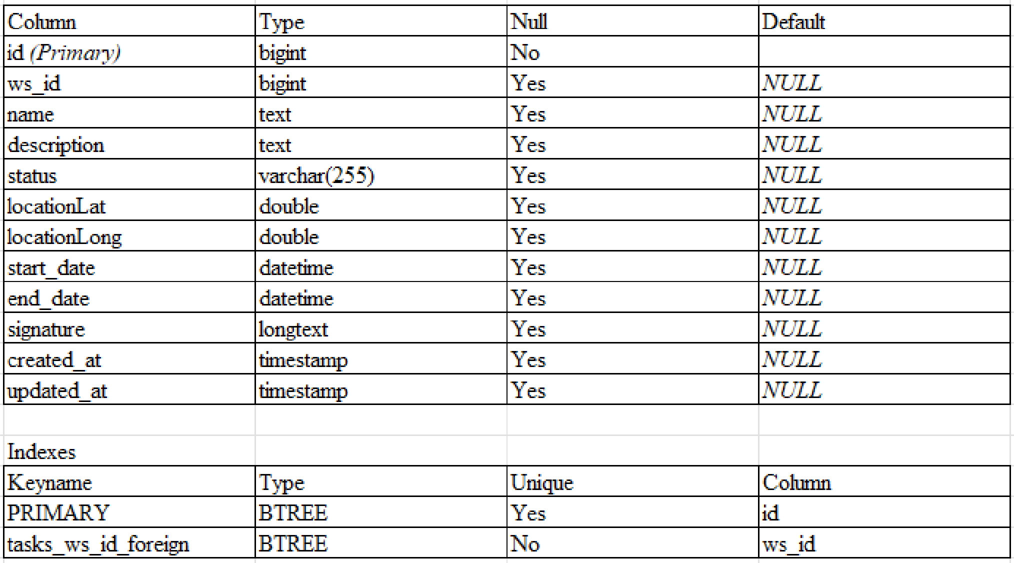
Column	Type	Null	Default
id *(Primary)*	bigint	No	
task_id	bigint	Yes	*NULL*
name	varchar(255)	Yes	*NULL*
is_done	tinyint(1)	Yes	*NULL*
created_at	timestamp	Yes	*NULL*
updated_at	timestamp	Yes	*NULL*

Indexes

Keyname	Type	Unique	Column
PRIMARY	BTREE	Yes	id
subtasks_task_id_foreign	BTREE	No	task_id

Table name: tasks

Table 5.7: Task Data Dictionary Table



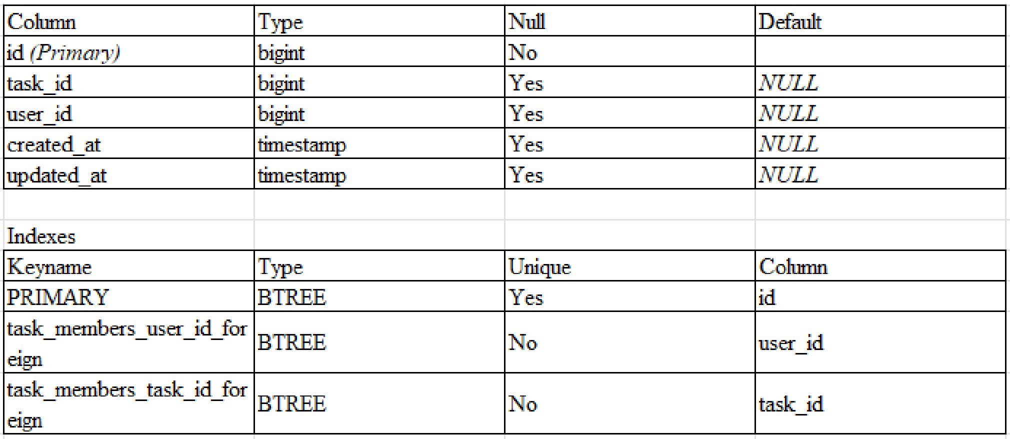
Column	Type	Null	Default
id *(Primary)*	bigint	No	
ws_id	bigint	Yes	*NULL*
name	text	Yes	*NULL*
description	text	Yes	*NULL*
status	varchar(255)	Yes	*NULL*
locationLat	double	Yes	*NULL*
locationLong	double	Yes	*NULL*
start_date	datetime	Yes	*NULL*
end_date	datetime	Yes	*NULL*
signature	longtext	Yes	*NULL*
created_at	timestamp	Yes	*NULL*
updated_at	timestamp	Yes	*NULL*

Indexes

Keyname	Type	Unique	Column
PRIMARY	BTREE	Yes	id
tasks_ws_id_foreign	BTREE	No	ws_id

Table name: task_members

Table 5.8: Task-Members Data Dictionary Table



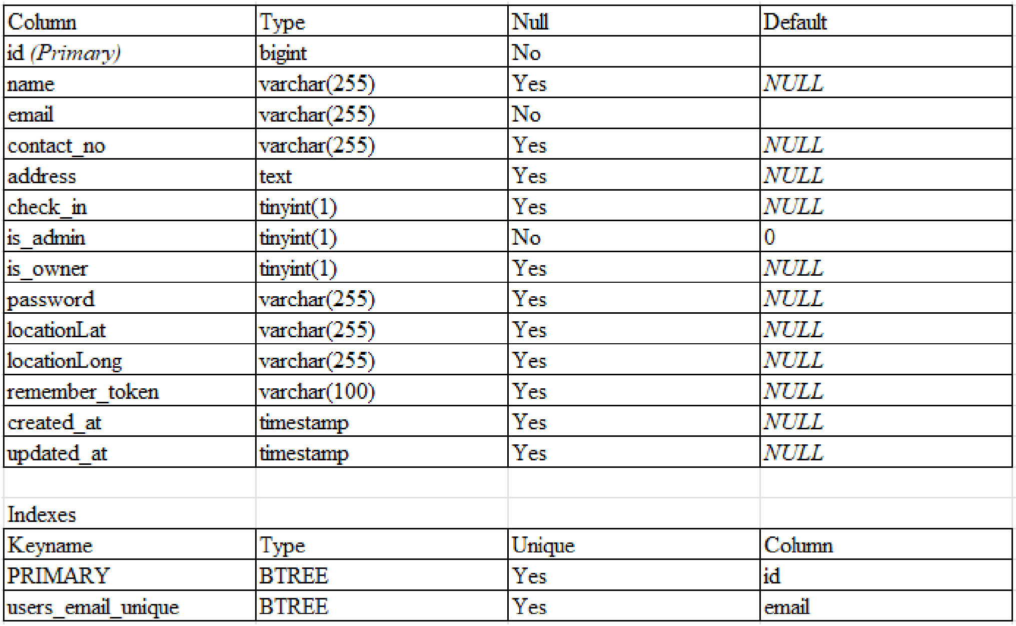
Column	Type	Null	Default
id *(Primary)*	bigint	No	
task_id	bigint	Yes	*NULL*
user_id	bigint	Yes	*NULL*
created_at	timestamp	Yes	*NULL*
updated_at	timestamp	Yes	*NULL*

Indexes

Keyname	Type	Unique	Column
PRIMARY	BTREE	Yes	id
task_members_user_id_foreign	BTREE	No	user_id
task_members_task_id_foreign	BTREE	No	task_id

Table name: users

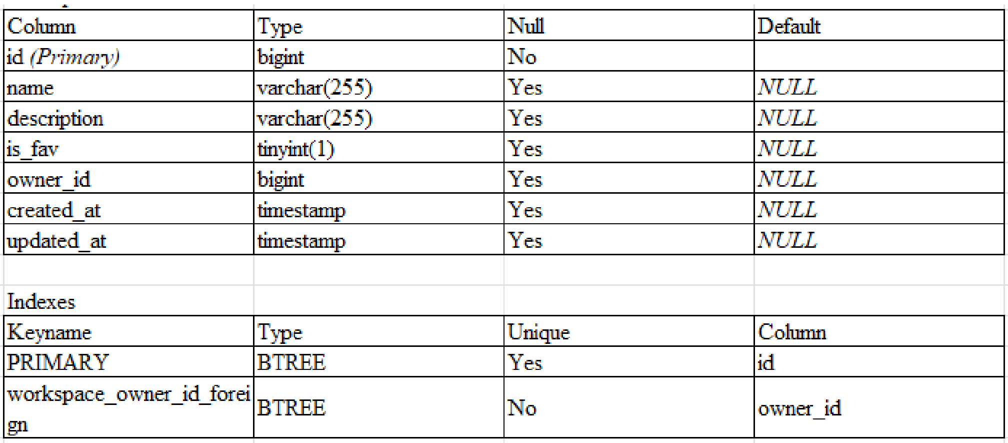
Table 5.9: Users Data Dictionary Table



Column	Type	Null	Default
id *(Primary)*	bigint	No	
name	varchar(255)	Yes	*NULL*
email	varchar(255)	No	
contact_no	varchar(255)	Yes	*NULL*
address	text	Yes	*NULL*
check_in	tinyint(1)	Yes	*NULL*
is_admin	tinyint(1)	No	0
is_owner	tinyint(1)	Yes	*NULL*
password	varchar(255)	Yes	*NULL*
locationLat	varchar(255)	Yes	*NULL*
locationLong	varchar(255)	Yes	*NULL*
remember_token	varchar(100)	Yes	*NULL*
created_at	timestamp	Yes	*NULL*
updated_at	timestamp	Yes	*NULL*
Indexes			
Keyname	Type	Unique	Column
PRIMARY	BTREE	Yes	id
users_email_unique	BTREE	Yes	email

Table name: workspace

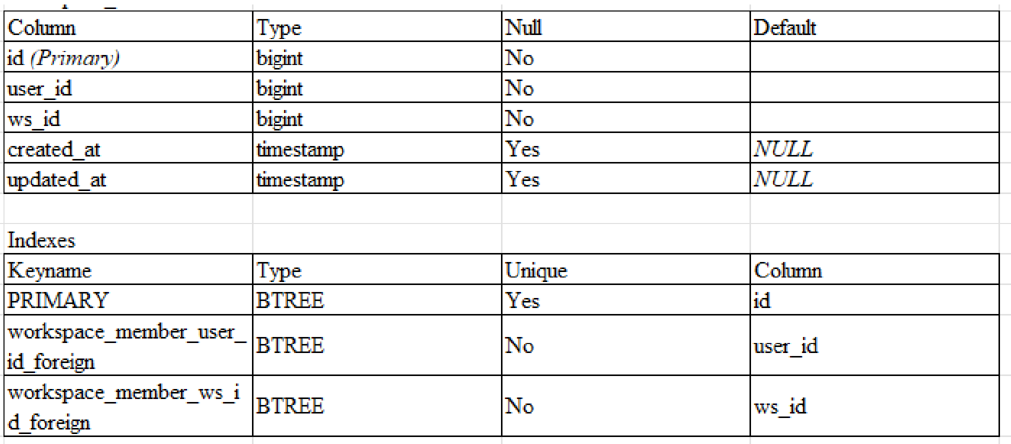
Table 5.10: Workspace Data Dictionary Table



Column	Type	Null	Default
id *(Primary)*	bigint	No	
name	varchar(255)	Yes	*NULL*
description	varchar(255)	Yes	*NULL*
is_fav	tinyint(1)	Yes	*NULL*
owner_id	bigint	Yes	*NULL*
created_at	timestamp	Yes	*NULL*
updated_at	timestamp	Yes	*NULL*
Indexes			
Keyname	Type	Unique	Column
PRIMARY	BTREE	Yes	id
workspace_owner_id_foreign	BTREE	No	owner_id

Table name: workspace_member

Table 5.11: Workspace-Member Data Dictionary Table



Column	Type	Null	Default
id *(Primary)*	bigint	No	
user_id	bigint	No	
ws_id	bigint	No	
created_at	timestamp	Yes	*NULL*
updated_at	timestamp	Yes	*NULL*

Indexes

Keyname	Type	Unique	Column
PRIMARY	BTREE	Yes	id
workspace_member_user_id_foreign	BTREE	No	user_id
workspace_member_ws_id_foreign	BTREE	No	ws_id

5.4 Data Flow Architecture

5.4.1 Context Diagram

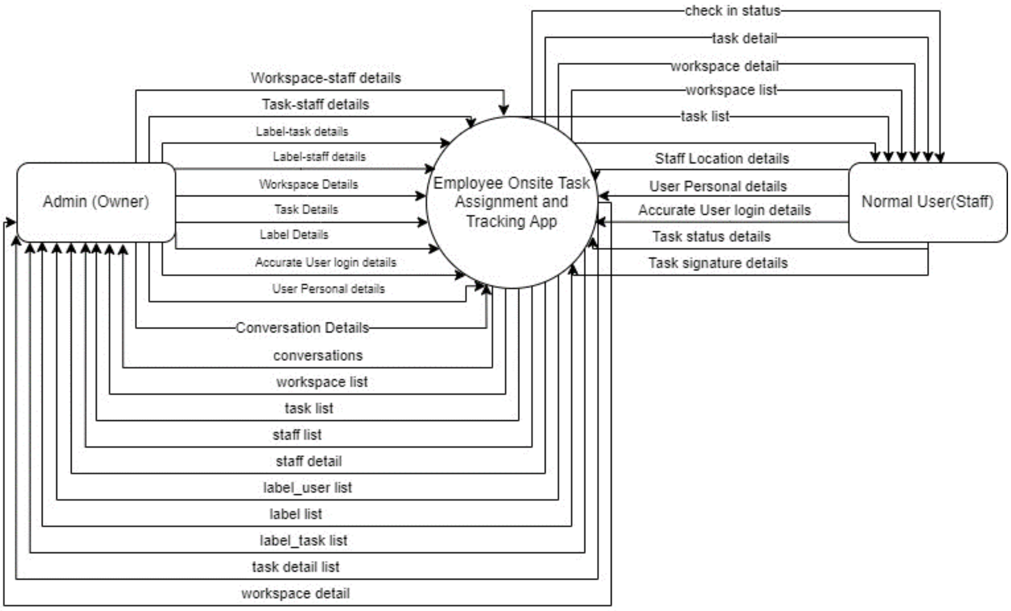


Figure 5.6: Context Diagram

5.4.2 Data Flow Diagram Level 0

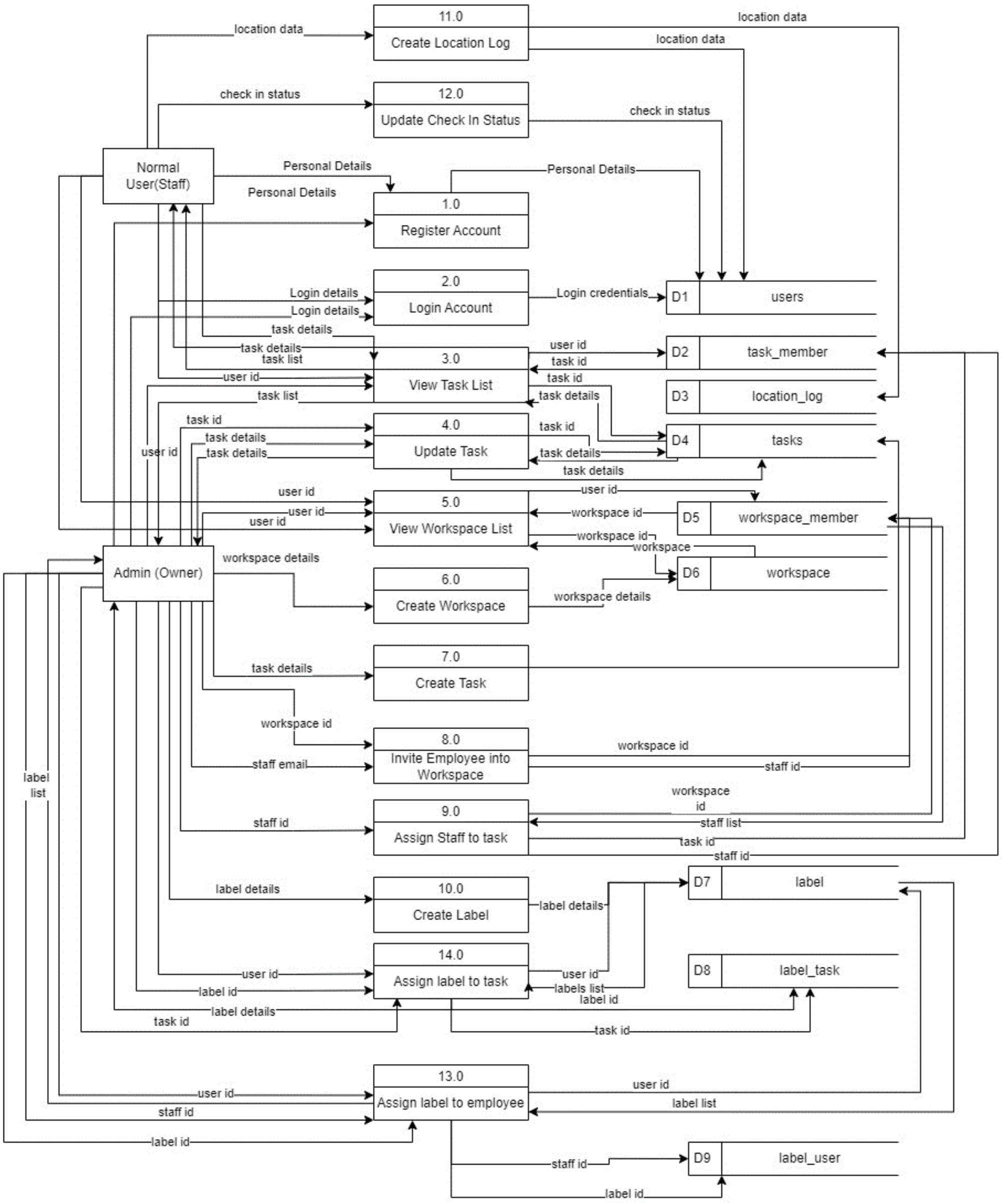


Figure 5.7: Data Flow Diagram Level 0

5.4.3 Data Flow Diagram Level 1

1.0 Register Account

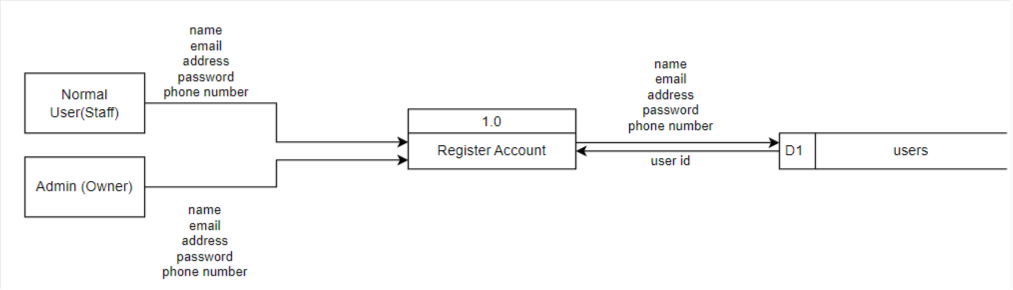


Figure 5.8: Register Account DFD Level 1

2.0 Login Account

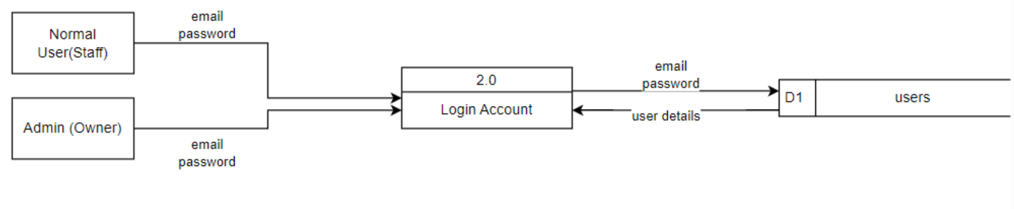


Figure 5.9: Login Account DFD Level 1

3.0 View Task

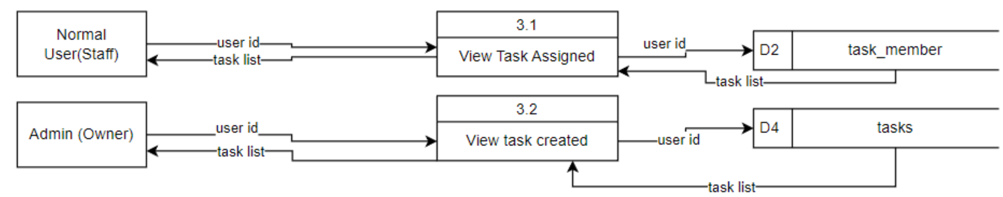


Figure 5.10: View Task DFD Level 1

4.0 Update Task Details

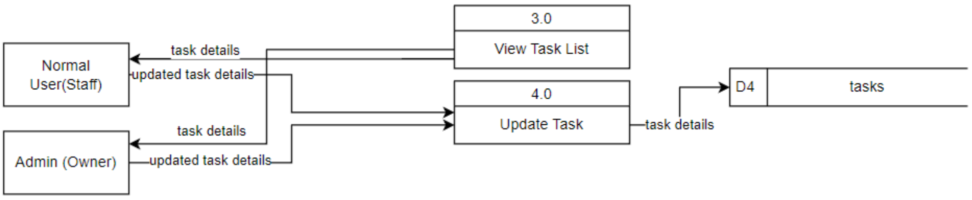


Figure 5.11: Update Task Details DFD Level 1

5.0 View Workspace

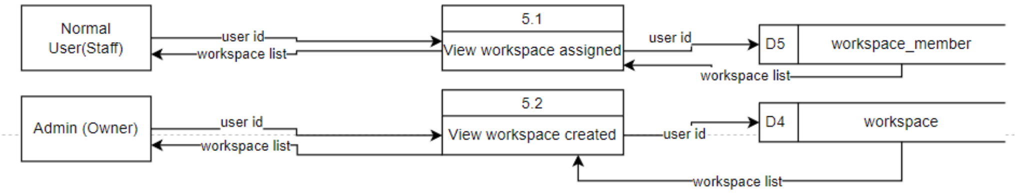


Figure 5.12: View Workspace DFD Level 1

5.0 Create Workspace

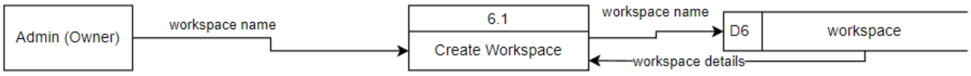


Figure 5.13: Create Workspace DFD Level 1

7.0 Create Task

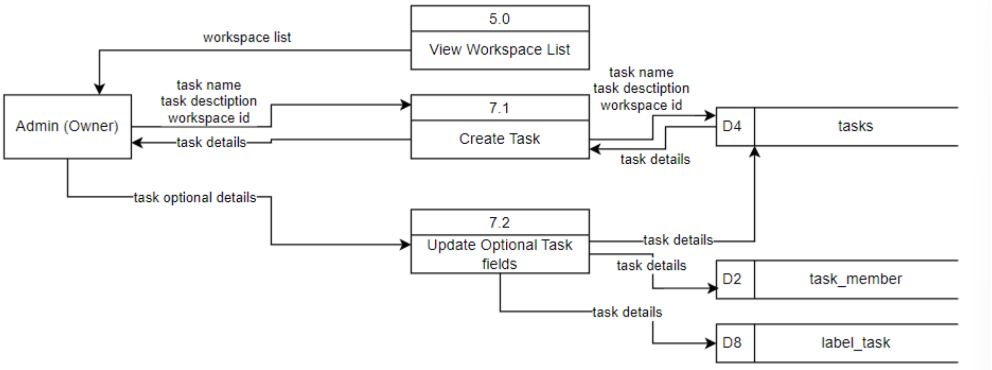


Figure 5.14: Create Task DFD Level 1

8.0 Assign staff to workspace

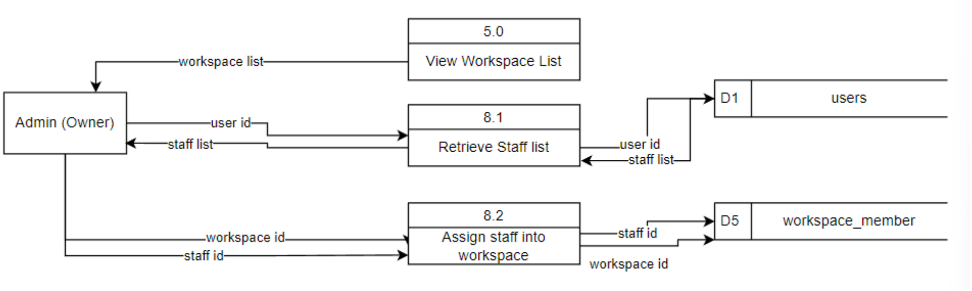


Figure 5.15: Assign Staff to workspace DFD Level 1

9.0 Assign staff to task

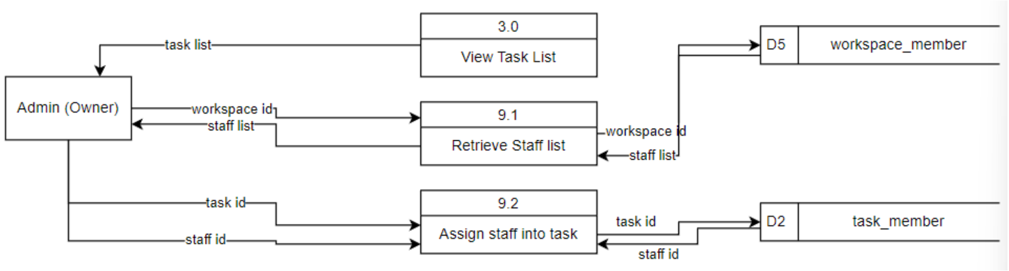


Figure 5.16: Assign staff to task DFD Level 1

10.0 Create label

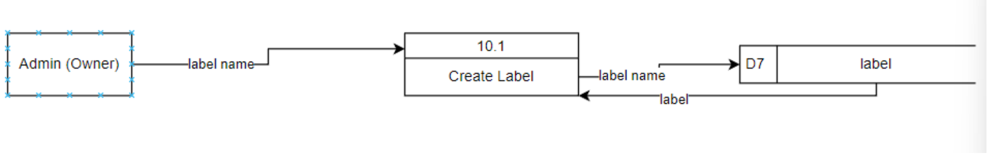


Figure 5.17: Create label DFD Level 1

11.0 Update / Create Location Log

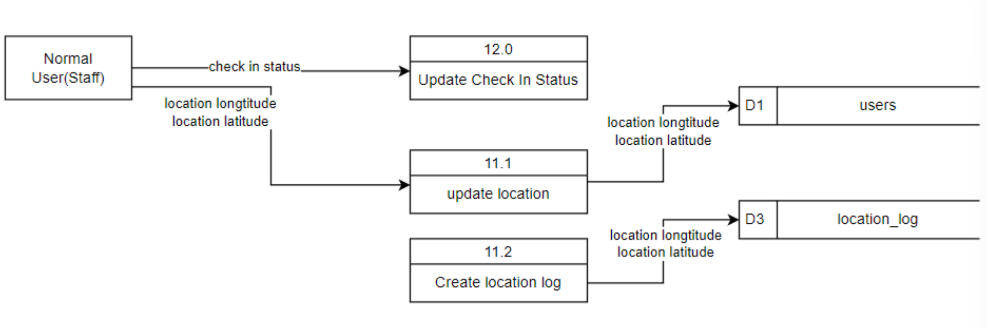


Figure 5.18: Update/Create Location Log

12.0 Update check in status

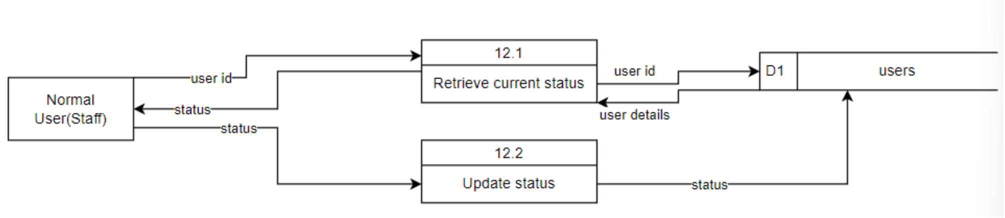


Figure 5.19: Update check in status DFD Level 1

13.0 Assign Label to Staff

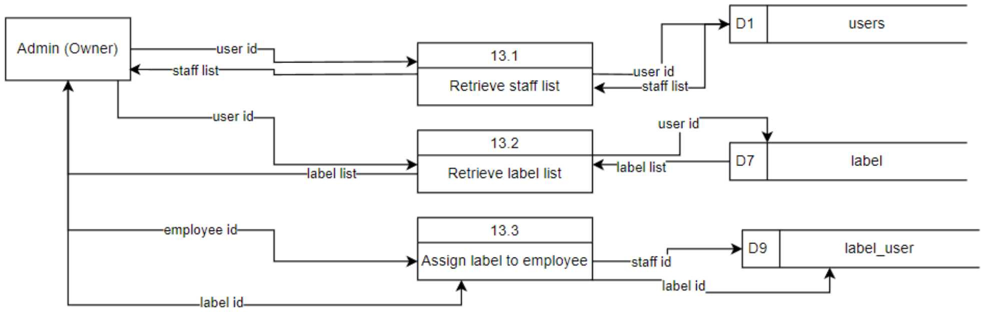


Figure 5.20: Assign Label to Staff DFD Level 1

14.0 Assign Label to Task

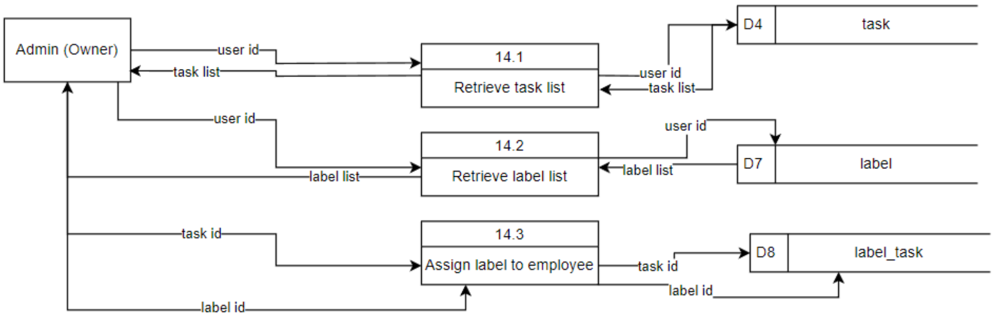


Figure 5.21: Assign Label to Task DFD Level 1

5.5 High Fidelity Prototype Design

5.5.1 General UI Design

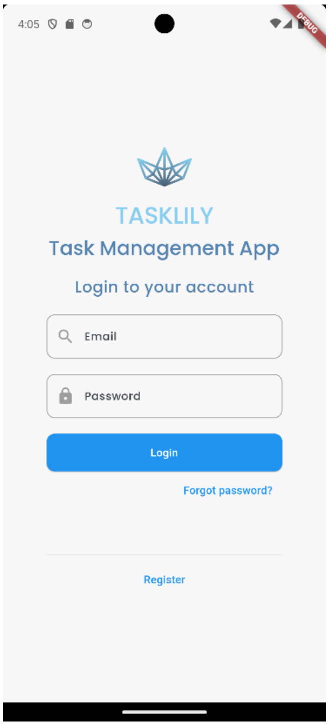


Figure 5.22: Login Page

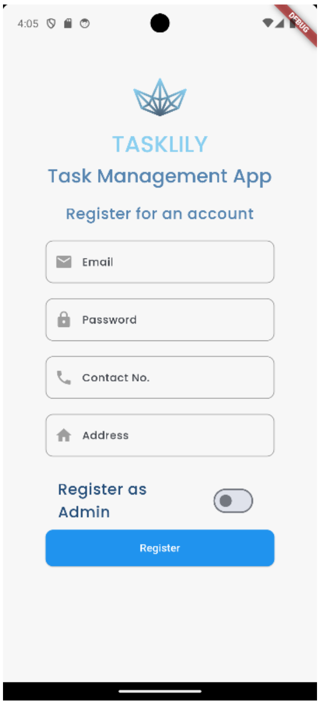


Figure 5.23: Register Page

Login Page and Register Page is always accessible by anyone without the need of logging in

5.5.2 Staff UI Design

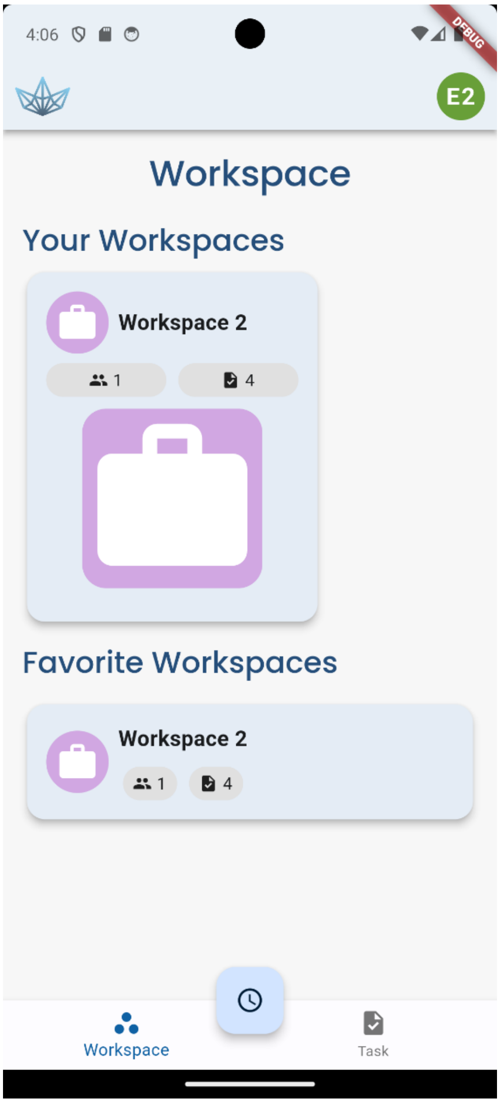
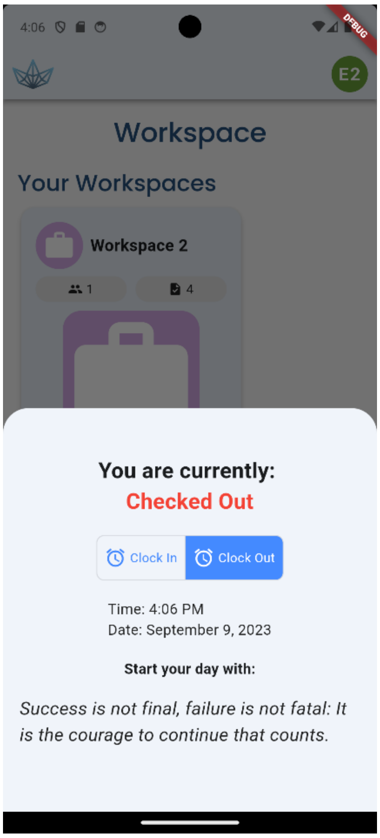


Figure 5.24: Staff’s Workspace Page



You are currently:
Checked Out

Time: 4:06 PM
Date: September 9, 2023

Start your day with:

Success is not final, failure is not fatal: It is the courage to continue that counts.

Figure 5.25: Staff Check In Modal

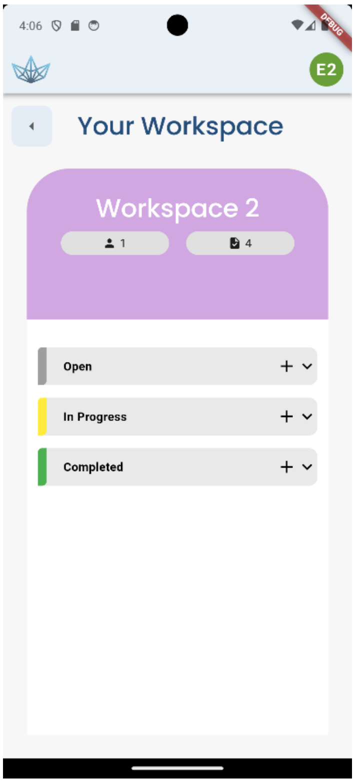
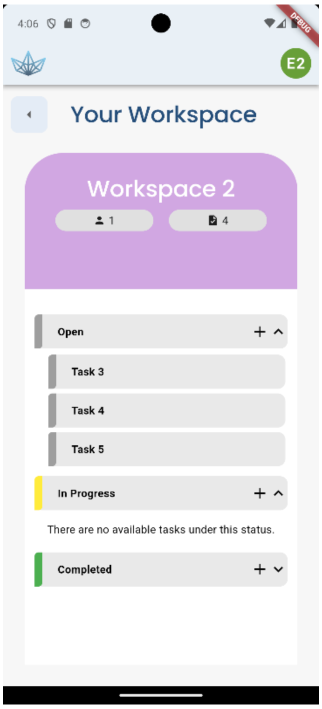


Figure 5.26: Workspace Detailed Page



4:06

E2

Your Workspace

Workspace 2

1 4

Open

Task 3

Task 4

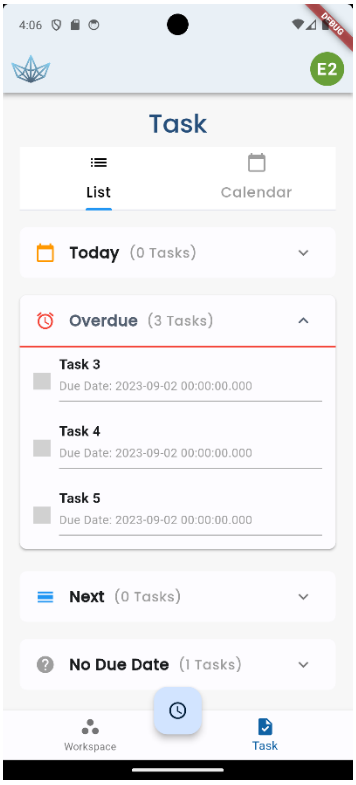
Task 5

In Progress

There are no available tasks under this status.

Completed

Figure 5.27: Workspace Task Display Page



4:06

E2

Task

List Calendar

Today (0 Tasks)

Overdue (3 Tasks)

Task 3
Due Date: 2023-09-02 00:00:00.000

Task 4
Due Date: 2023-09-02 00:00:00.000

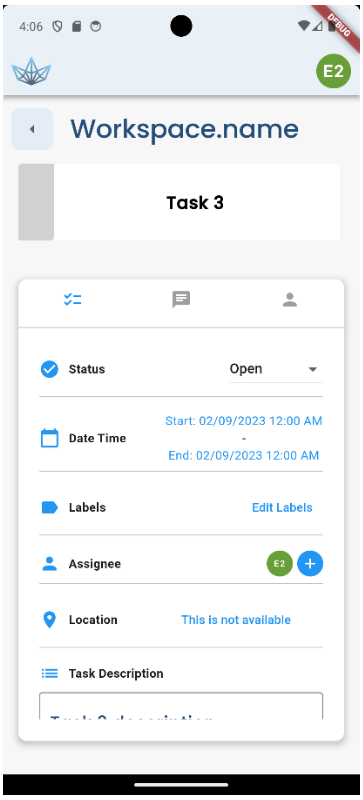
Task 5
Due Date: 2023-09-02 00:00:00.000

Next (0 Tasks)

No Due Date (1 Tasks)

Workspace Task

Figure 5.28: Task List Page



Status Open

Date Time Start: 02/09/2023 12:00 AM - End: 02/09/2023 12:00 AM

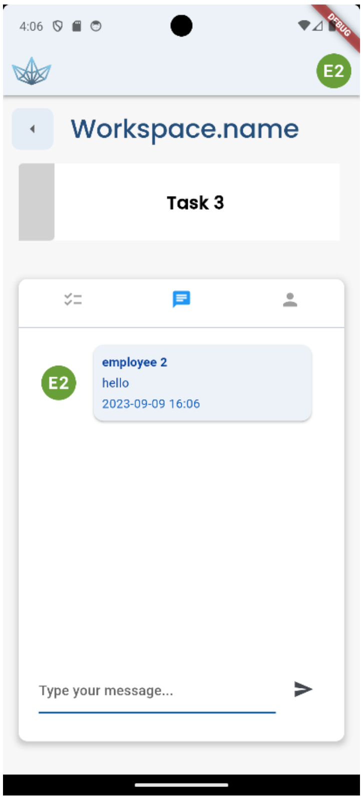
Labels

Assignee

Location This is not available

Task Description

Figure 5.29: Task Detail Tab Page



employee 2
hello
2023-09-09 16:06

Figure 5.30: Task Chat Tab Page

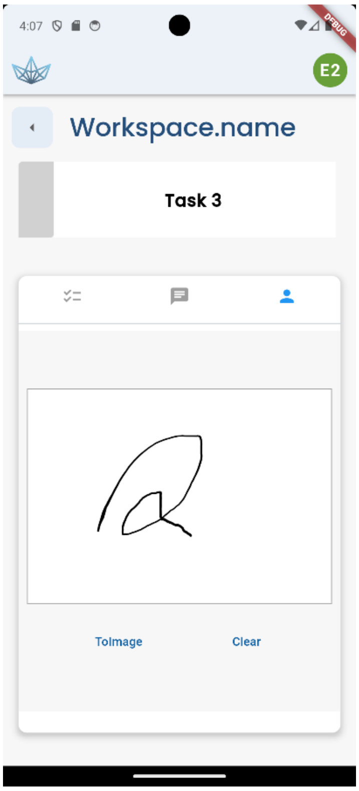
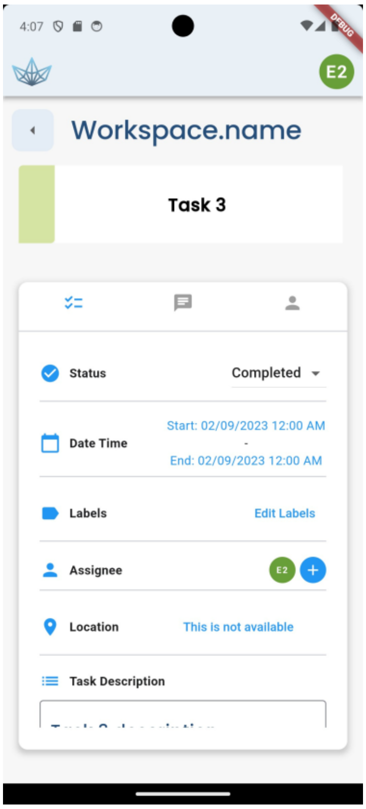


Figure 5.31: Task Client Review Tab Page



Status Completed

Date Time Start: 02/09/2023 12:00 AM - End: 02/09/2023 12:00 AM

Labels Edit Labels

Assignee E2 +

Location This is not available

Task Description

Figure 5.32: On Signed The status of Task Will be Updated to Completed

5.5.3 Admin UI Design

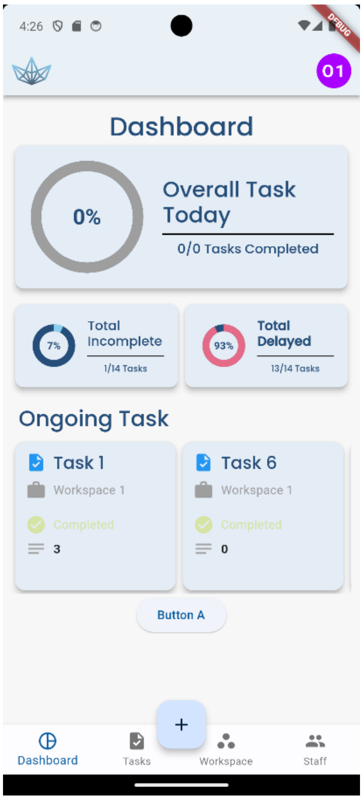


Figure 5.33: Admin Dashboard Page

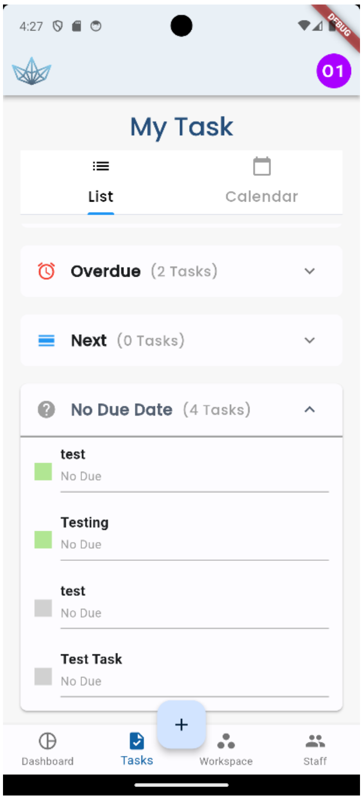


Figure 5.34: Admin MyTask Page

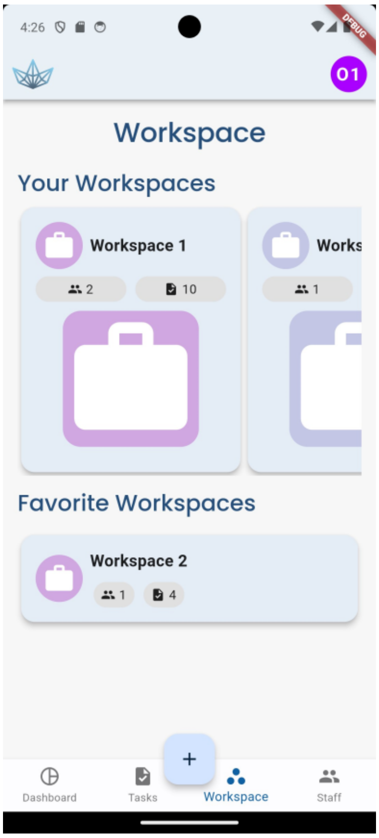


Figure 5.35: Workspace List Page

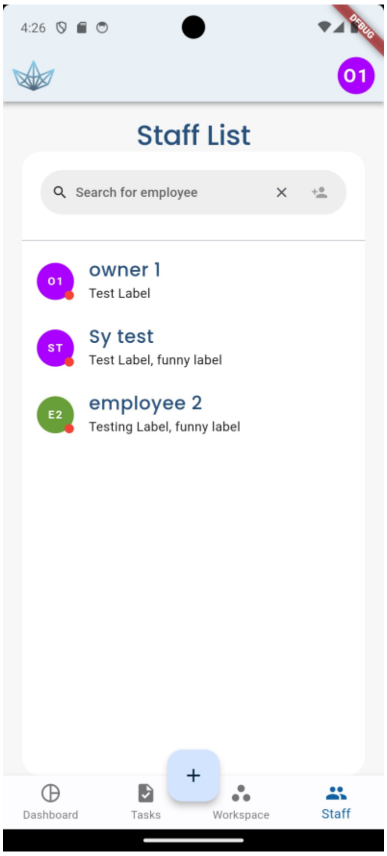
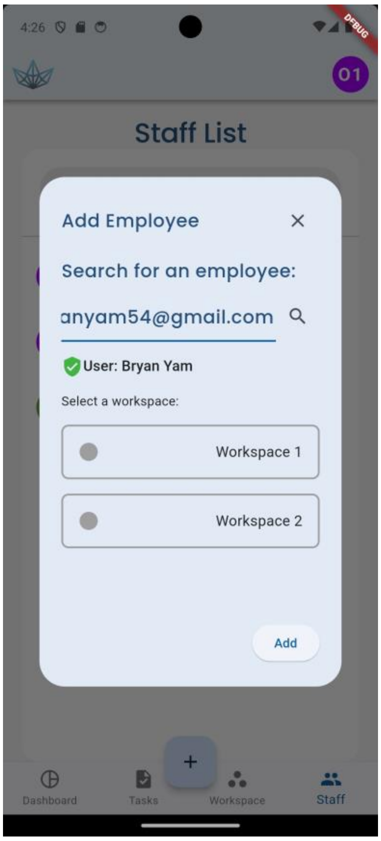


Figure 5.36: Staff List Page



4:26

01

Staff List

Add Employee

Search for an employee:

anyam54@gmail.com

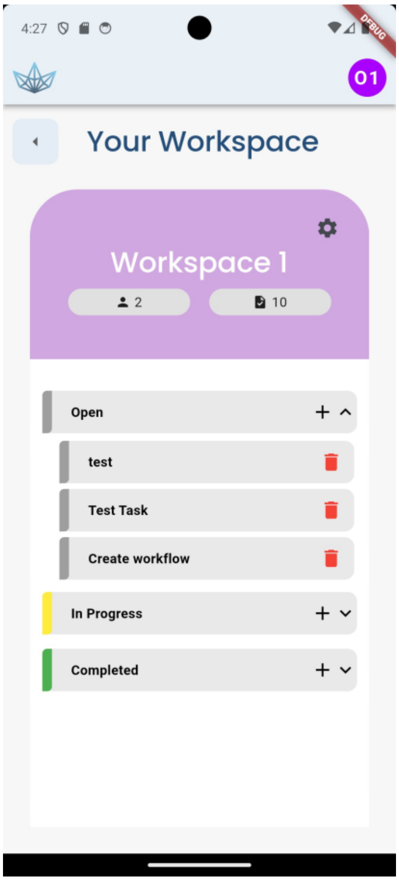
User: Bryan Yam

Select a workspace:

○ Workspace 1

○ Workspace 2

Figure 5.37: Invite Employee To Workspace Modal Page



4:27

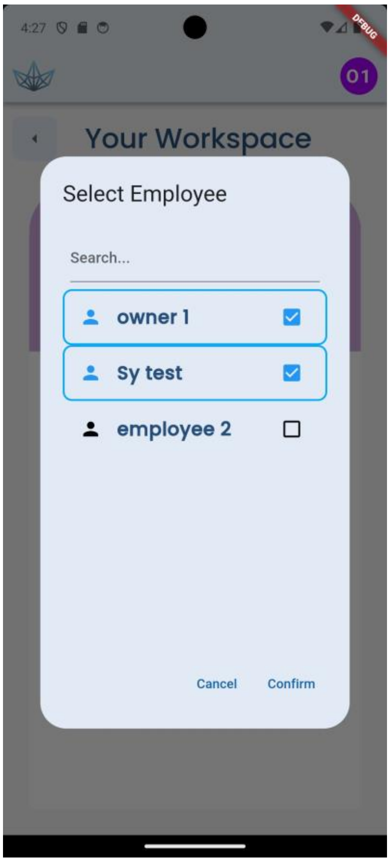
01

Your Workspace

Workspace 1

2 10

Figure 5.38: Workspace Detailed Task Page



4:27

DEBUG

01

Your Workspace

Select Employee

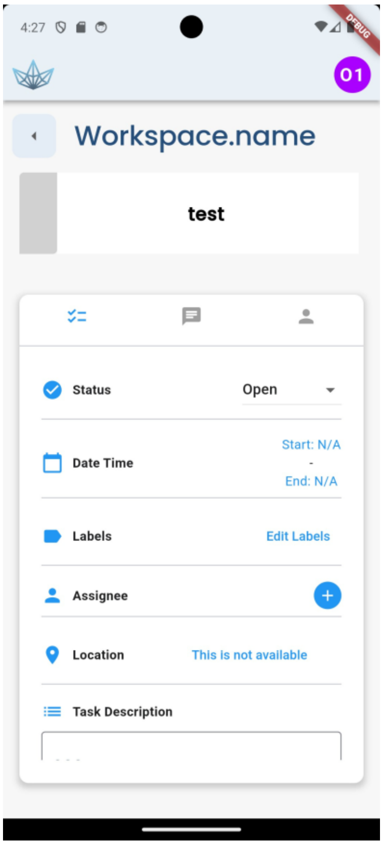
Search...

owner 1 ☑

Sy test ☑

employee 2 □

Figure 5.39: Workspace Invite Employee Modal Page



4:27

DEBUG

01

Workspace.name

test

Status: Open

Date Time: Start: N/A - End: N/A

Labels: Edit Labels

Assignee

Location: This is not available

Task Description

Figure 5.40: Task Detail Tab Page

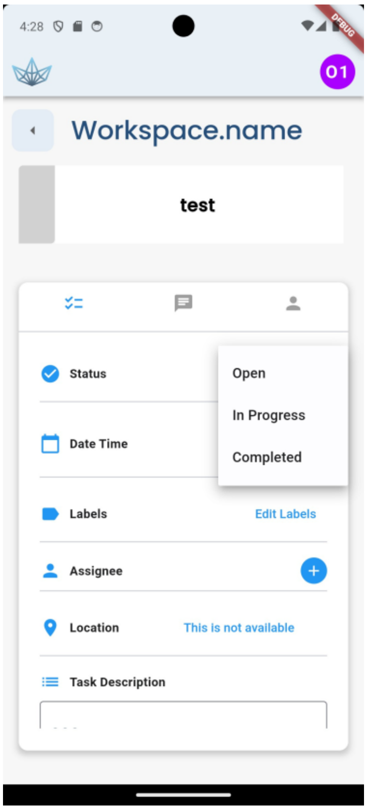


Figure 5.41: Set Status

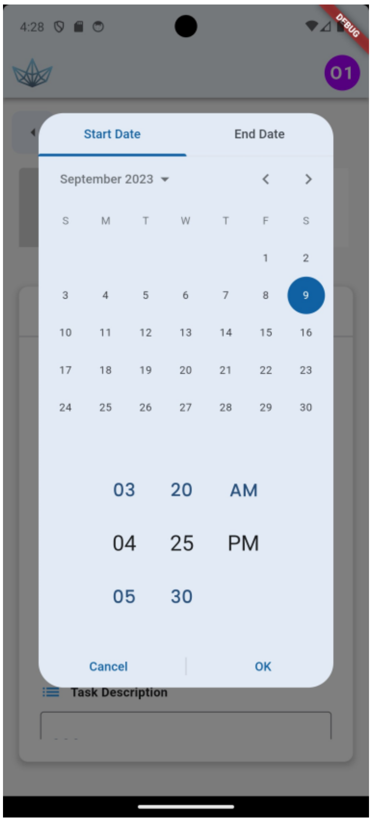
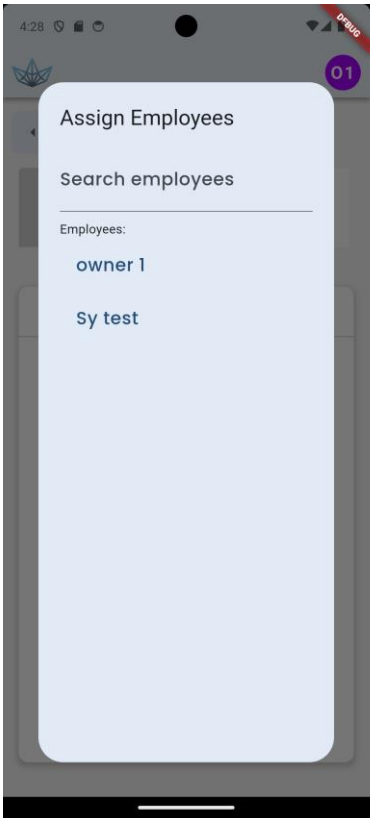


Figure 5.42: Set start date end date



Assign Employees

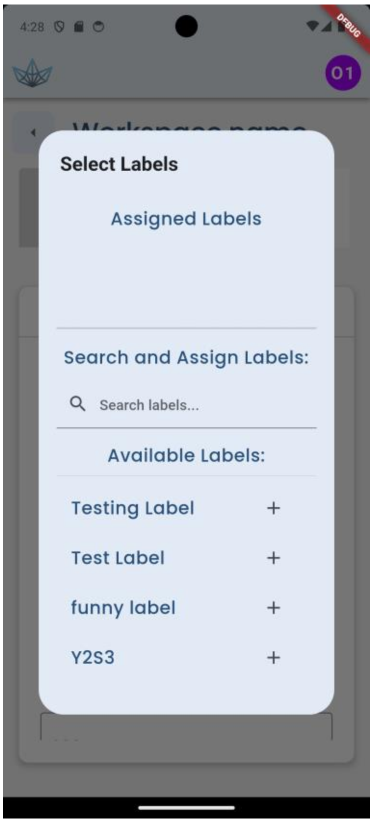
Search employees

Employees:

owner 1

Sy test

Figure 5.43: Assign staff to task



Select Labels

Assigned Labels

Search and Assign Labels:

Search labels...

Available Labels:

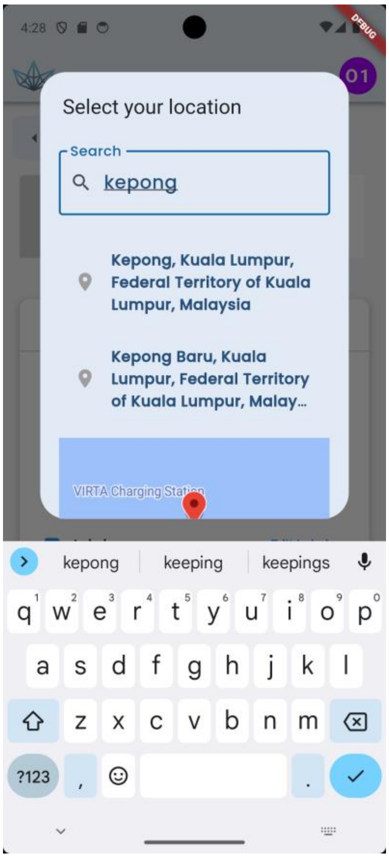
Testing Label +

Test Label +

funny label +

Y2S3 +

Figure 5.44: Assign Label to task



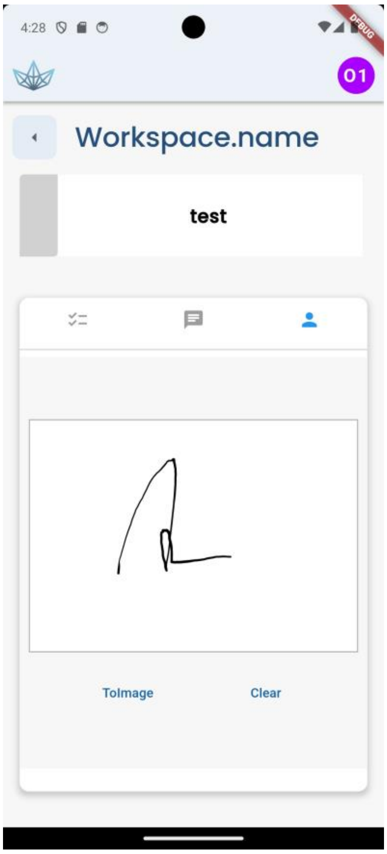
Select your location

Search: kepong

Kepong, Kuala Lumpur, Federal Territory of Kuala Lumpur, Malaysia

Kepong Baru, Kuala Lumpur, Federal Territory of Kuala Lumpur, Malay...

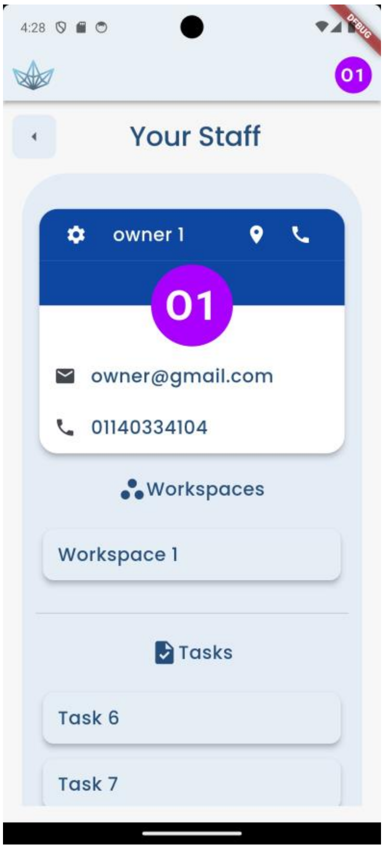
Figure 5.45: Set location Page



Workspace.name

test

Figure 5.46: Client Review Tab Page



owner 1

01

owner@gmail.com

01140334104

Figure 5.47: Staff detailed page (P1)

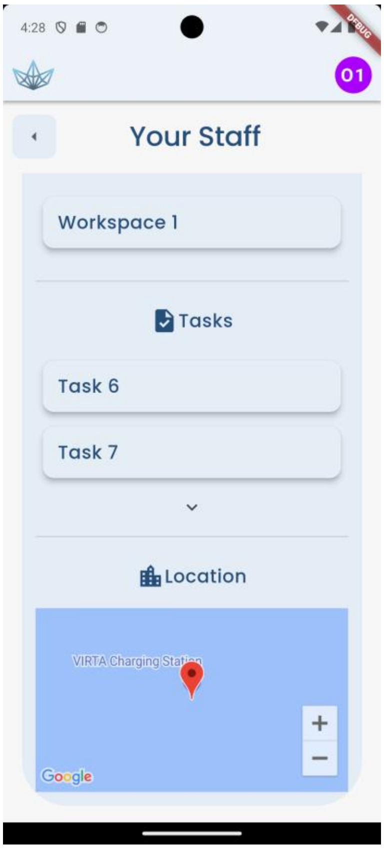
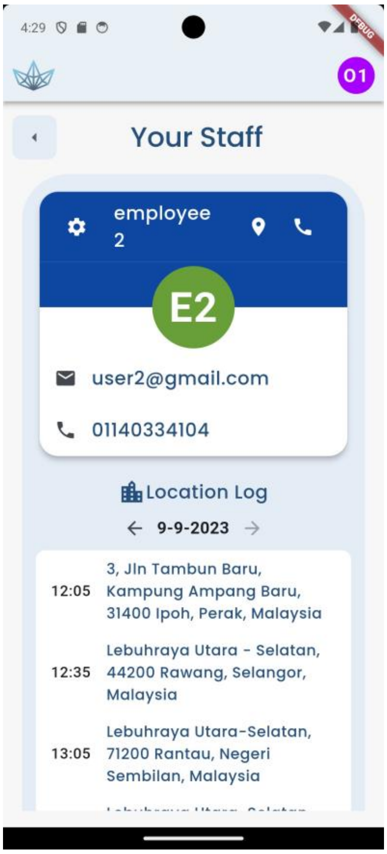


Figure 5.48: Staff Detailed Page (P2)



employee 2

E2

user2@gmail.com

01140334104

Location Log

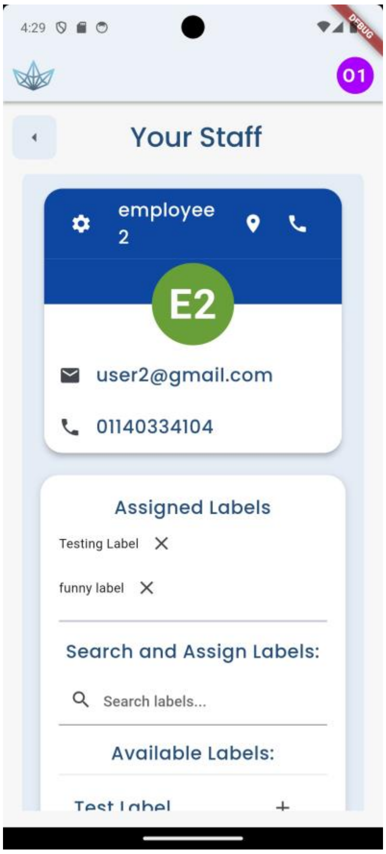
9-9-2023

12:05 3, Jln Tambun Baru, Kampung Ampang Baru, 31400 Ipoh, Perak, Malaysia

12:35 Lebuhraya Utara - Selatan, 44200 Rawang, Selangor, Malaysia

13:05 Lebuhraya Utara-Selatan, 71200 Rantau, Negeri Sembilan, Malaysia

Figure 5.49: Staff Location Page



employee 2

E2

user2@gmail.com

01140334104

Assigned Labels

Testing Label ✕

funny label ✕

Search and Assign Labels:

Search labels...

Available Labels:

Test Label +

Figure 5.50: Staff Assign Label Page

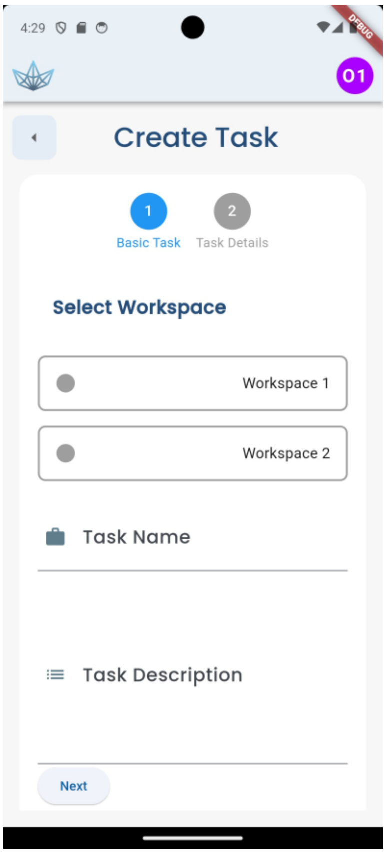


Figure 5.51: Create Task Page

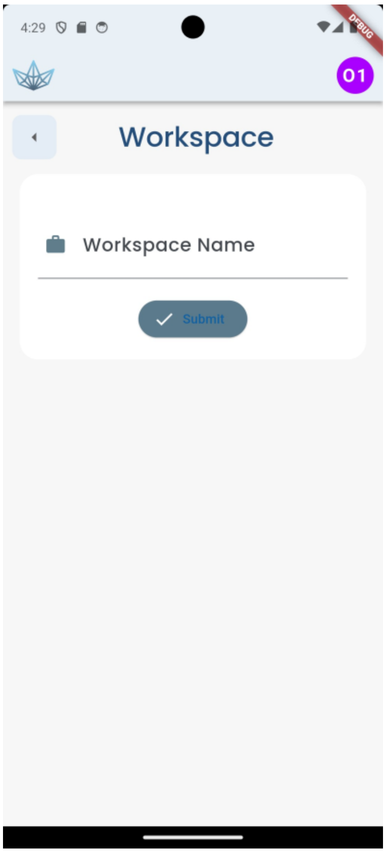


Figure 5.52: Create Workspace Page

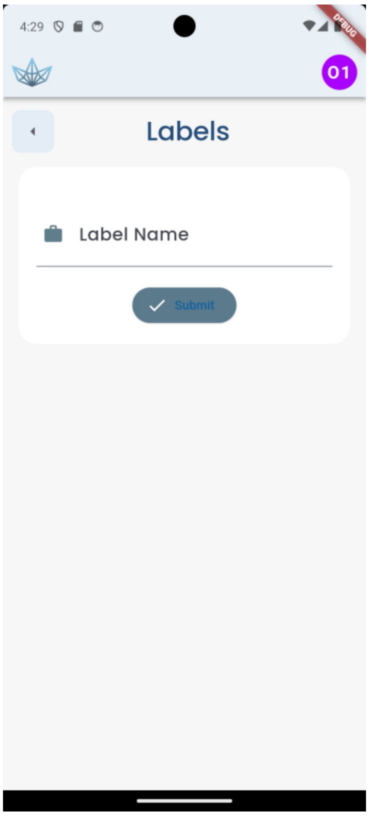


Figure 5.53: Create Label Page

The navigation of pages can be summarised from the interface diagram below:

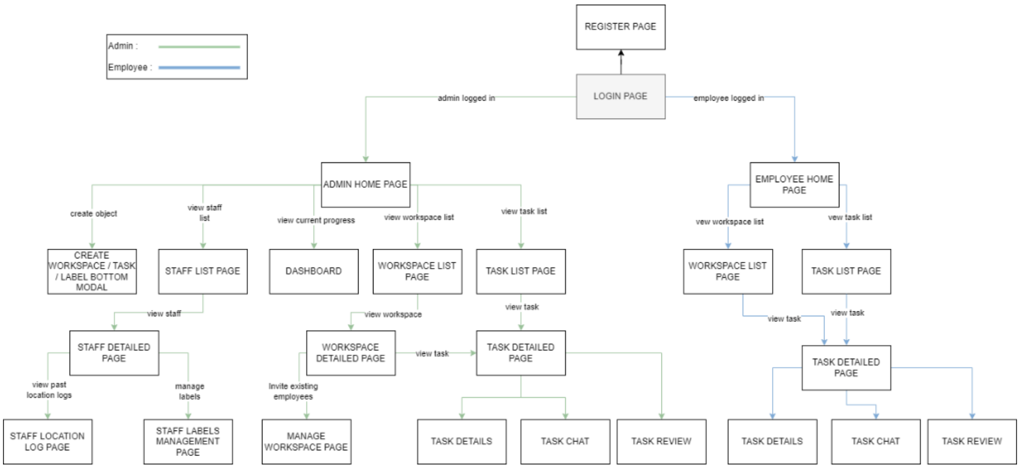


Figure 5.54: Interface Flow Diagram

CHAPTER 6

IMPLEMENTATION

6.1 Introduction

The implementation phase transforms design theories into tangible applications, serving as a pivotal stage in software development. This chapter outlines the methodologies and tools used to sculpt the system's architecture.

For front-end development, Flutter was selected. This open-source toolkit stands out for its ability to craft applications across various platforms from a singular codebase. In tandem, Laravel, an open-source PHP web framework, was employed for backend tasks. Laravel's capabilities extend to generating APIs, vital for seamless data communication.

Notably, JWT (JSON Web Tokens) has been integrated for secure data transfers and authentication, emphasising the system's commitment to security. Subsequent sections will further unpack the application's structural and functional aspects, highlighting the decision-making process behind each technological integration.

6.2 Flutter

Flutter is a framework with very rich UI libraries which provides ready implemented solutions which can fit to almost every need.

6.2.1 Flutter Libraries

It is indisputable that the library known as "flutter" serves as the cornerstone upon which all projects revolving around Flutter are constructed. This library is absolutely necessary for sculpting user interfaces that are interactive and responsive because it contains an extensive collection of widgets and tools that are required. Within the confines of this project, it gave birth to fundamental components such as MaterialApp and Scaffold, in addition to other basic widgets, which were all essential in composing an aesthetically pleasing arrangement and an intricate design. "flutter," which is recognized as the important dependency for crafting within the Flutter framework, provides developers with the necessary machinery to shape the very backbone of their applications. This is because "flutter" is recognized as the essential dependency for crafting within the Flutter framework. Due to the indispensability of its part, including it in this project was a no-brainer, and doing so ensured that the project would be successful and fulfill its intended purpose.

The library "google_fonts" offers a streamlined mechanism to integrate the vast selection from Google's font repository. This sophisticated integration heightens the typographical aesthetics of applications, making them more legible and thereby enhancing user engagement. Within the framework of this project, the "google_fonts" library was harnessed to adopt the 'Poppins' typeface, a deliberate choice to maintain design cohesion. By doing so, not only was the brand's visual identity solidified, but it also offered users a seamless, aesthetically pleasing interface. The emphasis on consistent typography, made possible by this library, undeniably amplified the project's visual appeal and overall user-centric design approach.

The "pie_chart" library offers a seamless approach to data visualisation. Serving as a pivotal tool within this project, it facilitates the effective portrayal of statistical data and intricate breakdowns, allowing for rapid comprehension by its users. Particularly within the dashboard section, this library plays a crucial role in presenting data, affording the Admin an immediate and lucid understanding of task progression. While the widget provides an array of chart types, the decision to harness solely the pie chart was made. Its inherent adaptability means that not only does it present data efficiently, but its presentation is also aesthetically refined, ensuring that even traditional data visualisations resonate with contemporary design principles.

The "expansion_tile_card" library presents a suite of collapsible card widgets. Their integration has been instrumental in refining the project's user interface, offering a decluttered view with the option to delve into further details as and when users deem it necessary. Leveraging this library offers a more efficient pathway than attempting to craft an analogous widget from scratch, particularly given its polished expand-collapse functionality. Within the confines of the 'MyTask' page, this widget has been deployed to categorise tasks based on their respective deadlines. This systematic grouping, distinguishing between tasks due imminently or those that are overdue, ensures that both users and administrators can effortlessly discern and prioritise tasks, promoting clarity and effective time management.

The library "flutter_typeahead" has proven invaluable in streamlining the introduction of search functionalities with intuitive suggestions. Traditionally, orchestrating a synchronised list, especially with features such as auto-clearing the search box post-selection, would have been an intricate and time-intensive endeavour. Within the context of this project, "flutter_typeahead" has found its place pervasively, especially when searching for entities like employees or tasks. The beauty of this library lies not only in its ease of implementation but also in the enhanced user experience it provides. Instead of relying on a manual search activation like pressing a 'search' button, this widget ensures immediate, relevant results as users type. Consequently, this tool has not only streamlined the user's search process but has also elevated the overall interaction experience, offering swift, relevant feedback.

The "omni_datetime_picker" library is employed to optimise date selection processes, ensuring they're both user-centric and designed to minimise input discrepancies. Within the ambit of this project, its integration is particularly evident during task updates, where the initiation and culmination dates of tasks need adjustments. What sets this library apart is its unique capability to simultaneously select both starting and ending dates – a feature not ubiquitously available in similar widgets. Such functionality provides a streamlined approach for users, allowing concurrent modifications to both dates if necessitated. Furthermore, it incorporates logical constraints, ensuring that the starting date is never erroneously set beyond the ending one. This strategic implementation eliminates the need for users to manually input dates, thereby curtailing the potential for errors and enhancing overall efficiency.

The libraries "google_maps_flutter" and its web equivalent have seamlessly incorporated the trusted Google Maps service into the project's application framework. This incorporation has been pivotal in ensuring that location-centric features are not only reliable but also tailored to provide an intuitive user experience. Given the renowned reputation of Google Maps, adopted by myriad businesses for its precision and expansive database, its inclusion became indispensable. During the development phases of this project, alternative map services were found lacking in customisation capabilities and pinpoint accuracy of locales. The choice to utilise "google_maps_flutter" was fortified by its consistent updates, meticulous place-naming accuracy, and vast database. As a result, users are granted a broad spectrum of location options, fostering precise selections. This precision aids staff in aligning seamlessly with project details. Furthermore, the interface allows for direct map activation with a mere click, negating the need for extraneous function implementations.

The library "syncfusion_flutter_signaturepad" played a pivotal role in procuring user signatures, which proved to be indispensable for authentication or validation protocols within the project's application. Introducing this feature became instrumental for staff when seeking endorsements from clients or authorised individuals upon the culmination of a project. The uniqueness of an individual's signature – inherently difficult to replicate – serves as a robust layer of verification. This ensures that the endorsing party is indeed the intended and correct individual. Such a signature-capturing mechanism is paramount for enterprises, as it fosters a culture of accountability and transparency. By incorporating this functionality, there's a tangible record of consent or approval from clients or supervisors. This not only solidifies the conclusion of projects but also underpins a consistent ethos of client or stakeholder satisfaction, given the significance of sign-offs in business operations.

The library "flutter_profile_picture" streamlined the integration of profile image functionalities, adding a nuanced, personal element to user accounts within the project's application. Given the data-intensive nature of such an application, the need for distinct and easily identifiable user profiles becomes paramount. This tool, therefore, stands as a cornerstone in distinguishing various user categories. It aids administrators by furnishing them with a diversified visual representation, ensuring each user type is easily discernible at a glance. Such a facet not only enhances operational efficiency but also enriches the user interface, making it both intuitive and aesthetically appealing. The implementation of this library underscores the project's commitment to merging functionality with user-centric design elements.

The "latlong2" library provided indispensable tools tailored for the adept management of geographical coordinates. Within the framework of this project, this library was central to harnessing and precisely handling location-centric data. By offering refined methodologies for geographical computations, it ensured the utmost accuracy in data representations. The use of the LatLng datatype, a key component of this library, was particularly instrumental in capturing and representing the exact latitude and longitude of employee locations. Additionally, during the task location selection process, "latlong2" fortified the precision, ensuring that every geographic detail registered within the system was both reliable and pinpoint. Such meticulous location management underscores the project's commitment to data accuracy and operational efficiency.

The library "http" was quintessential in orchestrating network interactions within the project's landscape. Tasked with both retrieving and transmitting data, this library fortified the application's seamless dialogue with backend infrastructures. Specifically, within the context of this project, the "http" library facilitated robust connections, drawing data from external APIs and liaising with the backend Laravel framework. This ensured not just data acquisition, but also the transmittal of requests and subsequent responses from the server. By leveraging the capabilities of "http", the project could maintain consistent and efficient data flows, reinforcing the system's reliability and responsiveness in real-time interactions.

The library "url_launcher" served as a pivotal bridge, facilitating the application's ability to initiate and access external URLs. Within the scope of this project, it became indispensable in seamlessly connecting to external web pages or digital resources, thus expanding the application's operational horizon. Its implementation was particularly strategic in streamlining communication processes; rather than navigating to the dial pad to initiate calls, the utility enabled direct call functionality. This meant that if there was a need to contact staff, the administrative interface could bypass redundant steps, launching calls straight from the application. This enhancement not only saved time but also elevated the user experience by offering a more integrated communication solution.

The library "shared_preferences" provided an efficient method to locally cache user settings and application preferences. Within the context of this project, this capability was instrumental in offering a tailored and consistent user experience on subsequent interactions with the application. The primary utilisation of this library was to securely retain tokens and basic user details. As a result, users could enjoy the convenience of sustained application activity in the background, eliminating the repetitive login requirement upon each engagement. Moreover, the ability to swiftly access stored user details fostered a more seamless experience. It negated the need for frequent backend queries whenever user-specific data was sought, enhancing the application's performance and responsiveness.

The incorporation of "background_location" into the project's infrastructure was paramount in enabling continuous location tracking, even when the application was not in active use. This feature held immense significance, especially in the context of real-time location-based functionalities. The demands of the current project necessitated the acquisition of real-time location data as a critical requirement. The library played a pivotal role in fulfilling this need by consistently fetching location data, even when the application operated in the background. This steadfast operation was pivotal in ensuring the accuracy of the collected data, which, in turn, provided administrators with precise and up-to-date information pertaining to staff locations. Such robust, real-time location tracking enhanced the overall efficiency and effectiveness of the project's location-based features.

The "permission_handler" library assumed a critical role in the management of permissions within the project's architecture. Its significance lay in the facilitation of secure access to user-sensitive data, establishing a foundation of transparency and trustworthiness. In the context of accessing certain data, particularly sensitive ones like location information, it is imperative to seek explicit permission from the data owner or user. This library streamlined the process of requesting location permissions, particularly vital when dealing with background location data. The necessity of such a library becomes apparent due to the stringent requirements imposed by Android's permission system. Without the user's explicit consent, Android restricts applications from accessing sensitive information. Moreover, compliance with legal frameworks like the Personal Data Protection Act underscored the library's indispensable role in ensuring data privacy and adhering to regulatory mandates.

The libraries "flutter_bloc" and "bloc" played a pivotal role in enhancing state management through the application of the BLoC (Business Logic Component) pattern. This tandem effectively segregated the application's logic from its user interface, thereby fortifying scalability and maintainability. In comparison to other state management libraries such as GetX and riverpod, which have their merits, "flutter_bloc" and "bloc" offered distinct advantages, particularly within the context of this project. One standout feature was their capacity to accommodate the hot reload of background location data—a feature crucial to the real-time nature of this application. The dynamic nature of "flutter_bloc" made it an ideal choice for maintaining a fluid user experience. For instance, it facilitated the seamless presentation of loading indicators, enhancing user engagement. While the initial setup may appear intricate, "flutter_bloc" demonstrated its prowess in managing a multitude of states efficiently. Furthermore, it firmly adhered to the best practice of isolating business logic from the view layer. This segregation not only streamlined data processing but also ensured that required data was readily accessible, effectively averting lazy load issues. In summary, "flutter_bloc" and "bloc" were indispensable tools in this project, embodying the principles of effective state management and software architecture. Their combined functionality facilitated not only an immersive user experience but also paved the way for effortless scalability and maintenance.

The "intl" library emerged as a pivotal tool in facilitating the localization of the project for diverse languages and regions, thereby extending its global accessibility. This library played a crucial role in customizing the presentation of dates and times, ensuring they adhered to the conventions of various regions and languages. One noteworthy application of "intl" was the formatting of dates and times to align with specific requirements, particularly when interacting with external APIs like the Laravel API. In many instances, the data received from these APIs did not conform to the standard date and time formats used by Dart. As such, the inclusion of "intl" became imperative to ensure seamless data integration, maintaining consistency and accuracy throughout the application. This library thus emerged as an essential component in enhancing the project's global reach and user experience.

The integration of the "equatable" library proved instrumental in simplifying the task of comparing objects for equality within the project. This enhancement significantly bolstered the efficiency of data operations and data processing. The primary application of this library was to align with the "bloc" library, which relies on object comparison to manage and process data effectively. With "equatable," the process of determining whether two objects are equal became much more straightforward and streamlined. This capability is particularly valuable in scenarios where data objects need to be compared and assessed for equivalence, a common requirement in many data-centric operations. In essence, the inclusion of the "equatable" library optimized the project's data handling capabilities, ensuring that data comparison and equality checks were both efficient and accurate. This not only enhanced the overall performance of the application but also adhered to best practices in data management within the "bloc" framework.

The "geolocator" library played a pivotal role in enabling geolocation functionalities within the project. Its primary function was to retrieve the precise location of the device, a critical requirement for any feature dependent on user location within the application. This library assumed a compulsory role within the project, as it was imperative to ascertain the user's current location accurately. This data was subsequently transmitted to external APIs, such as Laravel, where it was utilized for data processing and updating the user's current location. The "geolocator" library, therefore, served as the linchpin for location-dependent functionalities, ensuring that the application seamlessly accessed and leveraged accurate geospatial data, enhancing its overall utility and user experience.

6.3 Laravel

6.3.1 API List

Table 6.1: Api List Table

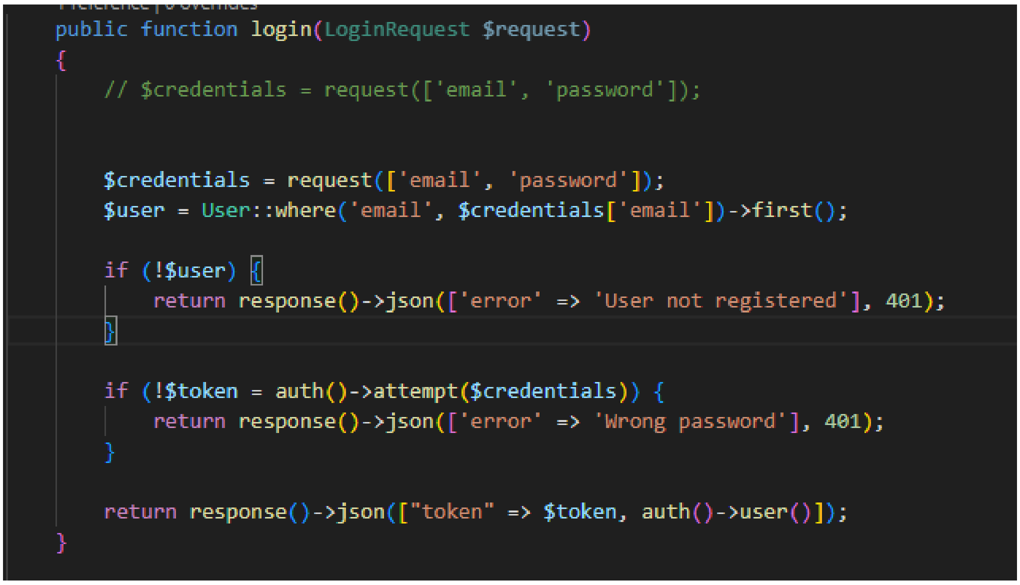
Endpoint	**Method**	**Description**
/login	POST	Authenticate a user and get an access token
/register-employee	POST	Register an employee user
/register-admin	POST	Register an admin/employer user
/user/checkin-status/{id}	GET	Check the check-in status of a user
/user/update-check-in/{id}	PUT	Update a user's check-in status

/user/update-user-location/{id}	PUT	Update a user's location
/logout	POST	Log the user out and invalidate their token
/me	GET	Get the authenticated user's details
/show-profile/{id}	GET	Show a user's profile details
/show-profile-labels/{id}	GET	Show a user's profile with associated labels
/show-all-employees	POST	List all employee users
/user-location/{id}	POST	Get a user's current location
/user-location-log/{id}	GET	Retrieve a user's location log/history
/user/{id}	GET	Get detailed information about a specific user
/check-employee-by-email	POST	Check if an employee exists by their email
/user/location/{id}	GET	Retrieve a user's location
/labels	GET	List all labels
/labels	POST	Create a new label
/labels/{id}	PUT	Update a specific label
/labels/{id}	GET	Retrieve details of a specific label
/labels/{id}	DELETE	Delete a specific label
/labels/add-label-to-user	POST	Add labels to a user's profile
/labels/remove-label-from-user/{id}	POST	Remove a label from a user's profile
/workspace/task-summary	GET	Get a summary of tasks in the workspace
/workspace/assign-label-to-task	POST	Assign a label to a task in the workspace
/workspace/remove-label-from-task	POST	Remove a label from a task in the workspace
/workspace	GET	List all workspaces

/workspace	POST	Create a new workspace
/workspace/{id}	PUT	Update a specific workspace
/workspace/{id}	GET	Retrieve details of a specific workspace
/workspace/{id}	DELETE	Delete a specific workspace
/workspace/invite-employee/{id}	POST	Invite an employee to a workspace
/workspace/remove-employee/{id}	POST	Remove an employee from a workspace
/task/task-suggestion-list/{id}	GET	Get a list of task suggestions for a user
/task/not-due-date-task	GET	List tasks without a due date
/task/generate-report	GET	Generate a task report
/task/overdue-task	GET	List overdue tasks
/task/overdue-delay-task	GET	List tasks that are overdue and delayed
/task/upcoming-task	GET	List upcoming tasks
/task/task-list-by-month	GET	List tasks by month
/task/completed-task-by-employee	GET	List tasks completed by an employee
/task/delay-task-by-employee	GET	List tasks delayed by an employee
/task/in-progress-task-by-employee	GET	List tasks in progress by an employee
/task/in-progress-open-task-list	GET	List tasks that are in progress or open
/task	GET	List all tasks
/task	POST	Create a new task
/task/{id}	PUT	Update a specific task
/task/{id}	GET	Retrieve details of a specific task
/task/{id}	DELETE	Delete a specific task

/task/assign-employee-task/{id}	POST	Assign an employee to a task
/task/remove-employee-task/{id}	POST	Remove an employee from a task
/task/upload-signature/{id}	POST	Upload a signature for a task
/subtask	GET	List all subtasks
/subtask	POST	Create a new subtask
/subtask/{id}	PUT	Update a specific subtask
/subtask/{id}	GET	Retrieve details of a specific subtask
/subtask/{id}	DELETE	Delete a specific subtask
/subtask/update-status/{id}	POST	Update the status of a subtask
/subtask-completed/{id}	POST	Mark a subtask as completed
/locationLog	GET	List all location logs
/locationLog	POST	Create a new location log
/locationLog/{id}	PUT	Update a specific location log
/locationLog/{id}	GET	Retrieve details of a specific location log
/locationLog/{id}	DELETE	Delete a specific location log
/chat	GET	List all chat messages
/chat	POST	Create a new chat message
/chat/{id}	PUT	Update a specific chat message
/chat/{id}	GET	Retrieve details of a specific chat message
/chat/{id}	DELETE	Delete a specific chat message

6.3.2 Authentication and Authorization



```
1 reference | 0 overrides
public function login(LoginRequest $request)
{
    // $credentials = request(['email', 'password']);


    $credentials = request(['email', 'password']);
    $user = User::where('email', $credentials['email'])->first();

    if (!$user) {
        return response()->json(['error' => 'User not registered'], 401);
    }

    if (!$token = auth()->attempt($credentials)) {
        return response()->json(['error' => 'Wrong password'], 401);
    }

    return response()->json(["token" => $token, auth()->user()]);
}
```

Figure 6.1: Laravel project login code

In this project, the mechanism for both authentication and authorisation is orchestrated through the AuthController. It adeptly manages user authentication, subsequently dispensing JWT tokens for the crucial task of authorisation. At the heart of this process lies a construct method. This particular method is structured to mandate that all functionalities, with the exception of 'login' and commonly accessed pages like 'register', forgo the authorisation safeguard. To access the exclusive pages, users are compelled to undergo the login process.

The onus of managing user login and JWT token issuance rests with the 'login' method. In scenarios where a user embarks on the login journey by presenting their unique email and password, the method rises to the challenge of validating these credentials. Encountering an email not indexed in the database—indicative of an unregistered user—triggers an error response.

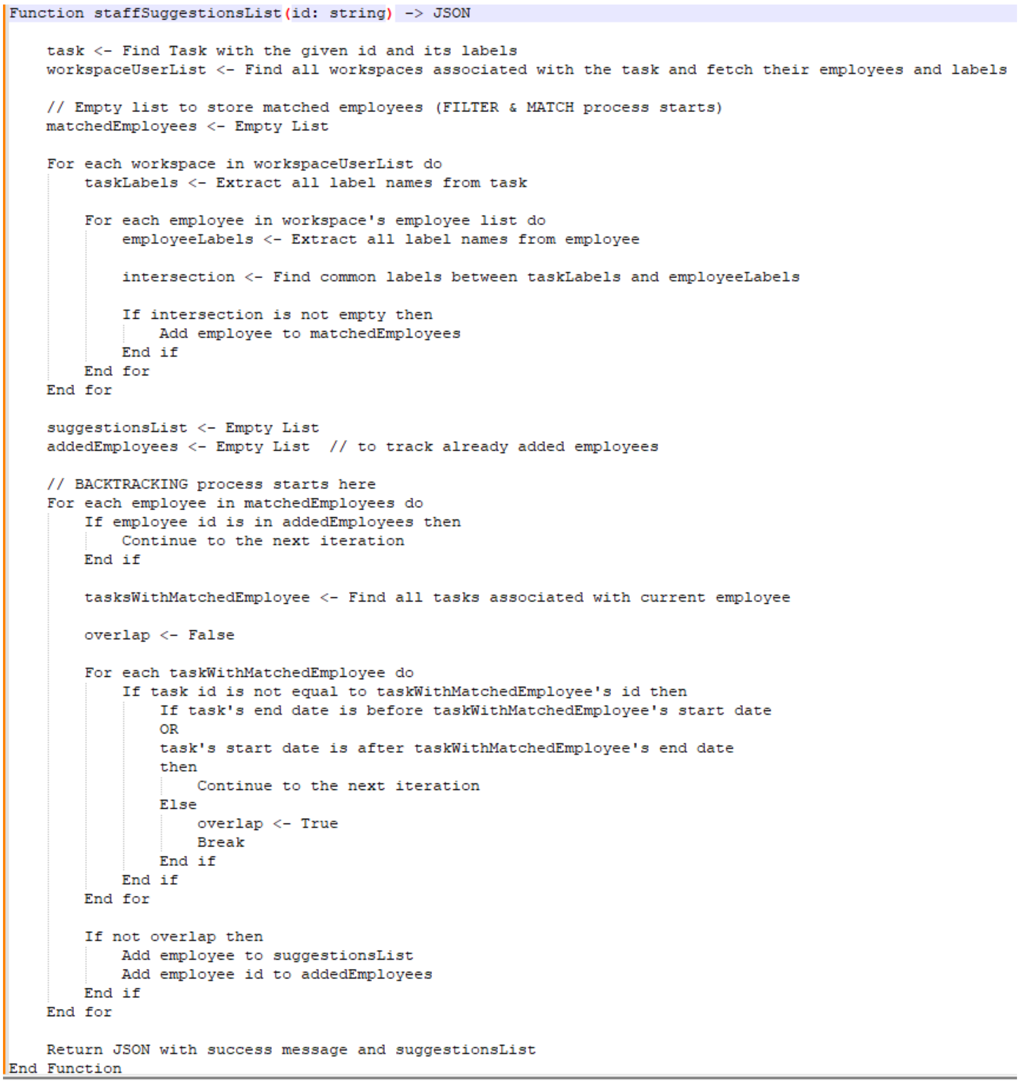
Conversely, recognising a registered user, the system then endeavours to validate the user using Laravel's intrinsic auth()->attempt($credentials) method. A mismatch in the password swiftly results in an error response. Nevertheless, a triumphant authentication facilitates the generation of a JWT token via auth()->attempt, which is then relayed alongside the user's salient information within the response. It's noteworthy to mention that the act of logging out is paired with a token refresh function, which rejuvenates the token to preclude any unsanctioned usage.

A pragmatic expiry timeframe of 7 days has been affixed to the token. It operates under the assumption that, on a weekly basis, employees engage with the login procedure at least once. This ensures the application consistently interacts with an authorised individual. JWT tokens play a pivotal role in safeguarding user access. The established protocol demands users to present their email and password. Successful authentication bestows upon them a JWT token. This newly acquired token becomes an integral part of subsequent requests to sequestered routes, granting the token bearer access to limited resources. The architecture also incorporates the capacity for token renewal, fortifying both security parameters and user sessions.

To bolster security further, middleware has been interwoven to protect routes demanding authentication. Middleware conditions are judiciously assigned to those routes where user access necessitates restrictions. Should an unverified user audaciously attempt to penetrate these routes devoid of a legitimate JWT token, the middleware promptly intercedes, revoking access. This guarantees that any rogue endeavours to circumvent this security cordon are thwarted. Once authentication triumphs, users find themselves privy to a cache of data tailored to them—encompassing aspects like workspace details, labels, location logs, and specific tasks. Such granularity ensures a tailored experience for every user within the application's realm.

6.3.3 Application of algorithm inside the staffSuggestionsList

The staffSuggestionsList function represents an advanced algorithmic solution crafted to identify an optimal list of suggested employees for a specific task. This algorithm seeks to balance both qualitative (matching based on labels) and quantitative (ensuring time availability) aspects to provide a curated list of suitable employees. The detailed description provided below elucidates its methodological intricacies.



```
Function staffSuggestionsList(id: string) -> JSON

    task <- Find Task with the given id and its labels
    workspaceUserList <- Find all workspaces associated with the task and fetch their employees and labels

    // Empty list to store matched employees (FILTER & MATCH process starts)
    matchedEmployees <- Empty List

    For each workspace in workspaceUserList do
        taskLabels <- Extract all label names from task

        For each employee in workspace's employee list do
            employeeLabels <- Extract all label names from employee

            intersection <- Find common labels between taskLabels and employeeLabels

            If intersection is not empty then
                Add employee to matchedEmployees
            End if
        End for
    End for

    suggestionsList <- Empty List
    addedEmployees <- Empty List  // to track already added employees

    // BACKTRACKING process starts here
    For each employee in matchedEmployees do
        If employee id is in addedEmployees then
            Continue to the next iteration
        End if

        tasksWithMatchedEmployee <- Find all tasks associated with current employee

        overlap <- False

        For each taskWithMatchedEmployee do
            If task id is not equal to taskWithMatchedEmployee's id then
                If task's end date is before taskWithMatchedEmployee's start date
                OR
                task's start date is after taskWithMatchedEmployee's end date
                then
                    Continue to the next iteration
                Else
                    overlap <- True
                    Break
                End if
            End if
        End for

        If not overlap then
            Add employee to suggestionsList
            Add employee id to addedEmployees
        End if
    End for

    Return JSON with success message and suggestionsList
End Function
```

Figure 6.2: Pseudocode of staffSuggestionsList

The function accepts a singular input, which is the identifier (id) of a task. This identifier serves as the foundational data point, leading the algorithm to gather associated task details, such as its labels and associated workspaces.

Utilising the provided task id, the algorithm fetches relevant task details, predominantly focusing on its labels. Furthermore, it draws information about all the workspaces that are linked to this task, concurrently retrieving the list of employees and their associated labels within these workspaces.

One of the central computational segments of this function revolves around filtering potential employees based on matching labels. An empty list named matchedEmployees is initialised to collate this filtered data. For each workspace and its associated employees, the algorithm examines if the labels of the task have any commonality with the labels of the employee. If an intersection of labels is identified, the employee is deemed a potential fit and thus added to the matchedEmployees list.

Post the filter and match process, the algorithm delves into a more intricate phase where it evaluates the suitability of these potential employees based on their availability. Two new lists, suggestionsList and addedEmployees, are initialised to aid this phase. The suggestionsList is designed to store the final list of suggested employees, whereas the addedEmployees list ensures that no employee is considered more than once.

For every employee in matchedEmployees, the algorithm fetches all tasks associated with that specific employee to evaluate any time overlaps or conflicts with the current task. If an overlap in timelines is detected, the employee is not considered a viable suggestion. However, if no overlap is found, the employee is deemed suitable and added to the suggestionsList, with their identifier also being appended to the addedEmployees list to ensure they are not re-evaluated.

The culmination of this computational journey results in a JSON response. This structured output encapsulates a success message, indicative of the process's completion, and more crucially, the suggestionsList containing the optimal list of suggested employees for the given task.

The staffSuggestionsList function presents a methodological amalgamation of filtering, matching, and backtracking mechanisms. Its capability to discern based on labels and efficiently resolve time overlaps makes it a robust and functional solution for the problem at hand. The systematic breakdown of its operations, from input retrieval to the structured output, underscores the meticulous thought process and architectural integrity embedded within.

CHAPTER 7

TESTING AND EVALUATION

7.1 Introduction

In the realm of software development, testing and evaluation stand as pivotal phases ensuring that the delivered system aligns with the specified requirements and functions optimally in the intended environment. Within this chapter, a comprehensive exploration of two fundamental testing techniques will be undertaken: User Acceptance Testing (UAT) and Unit Testing.

User Acceptance Testing (UAT) underscores the importance of end-user satisfaction. It is the final testing phase where the target audience examines the software to certify that it meets their expectations and use-case scenarios. This ensures that the software is not only technically sound but also resonates with the actual needs and preferences of its end users.

Conversely, Unit Testing hones in on the smallest parts of the software, the individual units or components, to verify that each functions correctly in isolation. By ensuring that these individual blocks operate flawlessly, a solid foundation is established upon which the larger system architecture is built.

Together, these tests offer a holistic approach to software verification, focusing both on its foundational elements and its end-user experience. In the subsequent sections, an in-depth dive into the implementation, methodologies, and results of these tests will be presented, casting light on the architectural integrity and functional viability of the system in question.

7.2 Unit Testing

For this Employee Onsite Task Assignment Management And Tracking App, unit testing has played a pivotal role in ensuring its robustness and reliability. As we delved into the intricacies of the app's development, each unit test served as a rigorous validation check for individual components, ensuring they functioned precisely as intended. The benefits have been manifold. Firstly, these tests have significantly reduced the incidence of bugs, allowing us to maintain a consistently high quality of code. Moreover, they have facilitated rapid and confident iterations; every time a change was introduced, the unit tests acted as a safety net, ensuring that existing functionalities remained unaffected. This not only accelerated the development pace but also bolstered the the confidence in the app's stability.

Additionally, the presence of well-structured unit tests has simplified the onboarding process for new members, offering clear insights into the functionality and expected behavior of code components. In essence, unit testing has been instrumental in elevating the Employee Onsite Task Assignment Management And Tracking App's quality, efficiency, and maintainability to new heights.

7.2.1 Summary of Unit testing result

Test Case Id	Test Case Title	Status
TC01	Register as Admin (Valid)	Pass
TC02	Register as Employee (Invalid)	Pass
TC03	Register as Employee (Valid)	Pass
TC04	Login as Admin (Valid)	Pass
TC05	Login as Staff (Valid)	Pass
TC06	Login as Staff (Invalid)	Pass
TC07	Login as Staff (Invalid)	Pass
TC08	Create Task (Valid)	Pass
TC09	Create Task (Valid)	Pass

TC10	Create Task (Invalid)	Pass
TC11	Create Label (Valid)	Pass
TC12	Create Label (Invalid)	Pass
TC13	View workspace List	Pass
TC14	View task list	Pass
TC15	View Staff List	Pass
TC16	View detail staff	Pass
TC17	Call staff button	Pass
TC18	View location logs	Pass
TC19	Assign label to staff	Pass
TC20	View Detailed Workspace	Pass
TC21	Delete task	Pass
TC22	Invite / Remove existing employees from workspace	Pass
TC23	Invite new employees (Valid)	Pass
TC24	Invite new employees (Invalid)	Pass

7.3 User Acceptance Test

User Acceptance Testing (UAT), a pivotal phase in the software development lifecycle, ensures that the final product aligns with users requirements and functions as intended. Conducted in a controlled environment, UAT involves actual users testing the application to validate its fitness for purpose. Crucially, in the UAT conducted, all participating users provided their sign-off, indicating their satisfaction level exceeded 50%. This feedback not only validates the solution's effectiveness but also emphasises the importance of involving users in the evaluation of software functionality and design. The following report delves into the structural and functional intricacies of UAT, highlighting its paramount importance in delivering a product that aligns with user expectations and requirements. The user acceptance form and results could be viewed under

7.3.1 Supervisor UAT summary

During the User Acceptance Testing, several key functionalities were evaluated to ensure the software meets user requirements. The functionalities assessed, along with their corresponding step-by-step procedures, were: (1) Login to account, (2) Creation of a workspace, task, and label, (3) Viewing lists for workspaces, tasks, and staff, (4) Inviting new staff to a workspace, (5) Assigning labels to staff members, (6) Viewing staff location logs, (7) Updating task details, and (8) Sending a message within a task. Remarkably, each of these functionalities achieved a 100% pass rate, indicating a high level of software reliability.

The application was assessed on various functionalities, each relating to specific productivity and usability aspects. The app excelled in providing easy access to task details and documentation with an 84% approval rate, and in showcasing visual representations of team progress at 86.7%. Communication between team members was found to be smooth, rated at 82.7%, and the ability to efficiently reallocate or adjust tasks received the same score. There was a positive response to the user interface's user-friendliness, as well as the system's capability to assist with efficient task assignment and the simplification of task monitoring – all at around 77.3%. The platform also aids in reducing manual follow-ups, with a 78.7% rating, and in logging task-related issues, receiving a 74.7% score. Feedback for completed tasks' process was streamlined at 70.7%. Overall, the application demonstrated a commendable performance in enhancing team productivity and collaboration.

Table 7.1: Supervisor Summary of Test Functionality

Question	Description	Related Functions	% Of passes
Are you able to login	Steps as below: 1. Navigate to the login page. 2. Enter the valid email and password. (Email: owner@gmail.com Password: password) 3. Submit the login form.	Login account	100%
Are you able to create a workspace	Steps as below: 1. Click the + button at the bottom center 2. Select create workspace 3. Fill in workspace name - You're	Create Workspace	100%

	welcome to write anything. 4. Click submit button		
Are you able to create a task	Steps as below:1. Click the + button at the bottom center2. Select create task3. Select FYP workspace4. Fill in task name - You're welcome to write anything.5. (Optional) Fill in task description - You're welcome to write anything.6. Click submit button	Create Task	100%
Are you able to create a label	Steps as below:1. Click the + button at the bottom center2. Select create label3. Fill in label name - You're welcome to write anything.4. Click submit button	Create Label	100%
Are you able to view all workspace list	Steps as below:1. Navigate to workspace page (click workspace in the bottom bar)	View Workspace List	100%
Are you able to view all task list	Steps as below:1. Navigate to task page (click my task in the bottom bar)	View All Task List	100%
Are you able to view all staff list	Steps as below:1. Navigate to staff page (click my staff in the bottom bar)	View All Staff List	100%
Are you able to invite new staff into your workspace	Steps as below:1. Click the Staff list page2. Beside the search box click the add members icon3. Search for testStaff@gmail.com (new staff will be deleted each timetesting session finishes)4. Click the search Icon5. Select your previously created workspace 6. Click add	Invite employee into workspace	100%
Are you able to assign label to a staff	Steps as below:1. Click the Staff list page2. In the search box , type Shinyun3. Select staff Shin yun 4. Click the settings Icon , upper left corner5. Search for your previously created	Assign label to staff	100%

	label6. Click on your created label or click the add icon beside of your label		
Are you able to view staff location log	Steps as below:1. Click the Staff list page2. In the search box , type Shinyun3. Select staff Shin yun 4. Click the location Icon , upper right corner	View Staff Location	100%
Are you able to update task details	Steps as below:1. Navigate to the previously created task2. Amend the start date & end date3. (Optional) Edit labels - assign your previously assigned labels4. (Optional) Edit location5. (Optional) Update status6. (Optional) Update task description	Update task details	100%
Are you able to send a message	Steps as below:1. Navigate to the previously created task2. Click on the chat icon3. Type hello and click send	Send message	100%

Table 7.2: Supervisor Summary of Satisfaction towards implemented solution

Question	In aspects of	Rate Count						
		1	2	3	4	5	Average	%
This application helps with efficiently assigning tasks to on-site team members.	Ease the process of assigning task to on-site staff	0	0	5	7	3	0.7733333333	77.3
This application helps with simplifying the monitoring of multipletasks at once.	Monitor progress of tasks to ensure timely delivery	0	0	5	7	3	0.7733333333	77.3
This application helps with reducing the need for manual follow-upsand reminders.	Reduce the traditional way of texting and alling to ask and follow up progress	0	0	5	6	4	0.7866666667	78.7
This application helps with facilitating smooth	Communication between team members	0	0	3	7	5	0.8266666667	82.7

communicationbetween team members.								
This application helps with easy access to task details and relevantdocumentation.	Access the task requirements easily	0	0	3	6	6	0.84	84.0
This application helps with the visual representation (charts) of teamprogress on task.	Providing a summary of the overall tasks assigned on its status	0	0	3	4	8	0.8666666667	86.7
This application helps with streamlining the feedback process forcompleted tasks.	Ensuring external satisfactory % improves	0	0	7	8	0	0.7066666667	70.7
This application helps with the efficient reallocation or adjustment oftasks.	Reallocate resources when changes happen	0	0	2	9	4	0.8266666667	82.7
This application helps with logging any issues or concerns related totasks.	Enable team collaboration on updating any roadblocks to task	0	0	5	9	1	0.7466666667	74.7
This application helps with providing a user-friendly interface that'seasy to navigate.	An understandable interface without too much training resources required	0	0	5	7	3	0.7733333333	77.3

7.3.2 Staff UAT Summary

During the User Acceptance Testing, the software was evaluated on its essential functionalities. Users successfully logged into their accounts and utilised the 'clock in' feature, both with a 100% pass rate. The application facilitated easy navigation, allowing users to view workspace and task lists seamlessly. Users could also access detailed task information, particularly for the 'Create Gantt Chart' task. The software integrated external tools such as Google Maps for direct location access. Additionally, communication was streamlined, enabling users to send messages and update their teams efficiently. Finally, the platform supported a sign-off feature for clients, a crucial step in ensuring task completion satisfaction. Every functionality tested yielded a 100% success rate, indicating the software's robustness and user-friendliness.

The application was assessed on multiple aspects to understand its efficiency and user-friendliness. A notable 93.3% of users found that the application helped them in receiving and understanding their daily task assignments. The tool was also beneficial in submitting updates, with 77.3% user satisfaction, and made adjusting tasks due to unforeseen changes a smoother process at 78.7%. Users appreciated the application's ability to track task progress and imminent deadlines, resulting in an 82.7% approval rate. Features like viewing a summary of tasks and the mobile-friendly experience garnered 84% and 86.7% satisfaction rates, respectively. The direct communication feature and access to a comprehensive history of tasks both received an 82.7% rating. Additionally, accessing relevant resources and storing information for each task were well-received, obtaining scores of 74.7% and 77.3% respectively. Overall, the application was deemed highly effective for managing and tracking tasks.

Table 7.3: Staff Summary of Test Functionality

Question	Description	Related Functions	% Of passes
Are you able to login	Steps as below: 1. Navigate to the login page. 2. Enter the valid email and password. (Email: user1@gmail.com Password:password) 3. Submit the login form.	Login account	100%
Are you able to clock in	Steps as below: 1. Click the clock button at the bottom center 2. Toggle the check in button	Check in	100%
Are you able to view all workspace list	Steps as below:1. Navigate to workspace page (click workspace in the bottom bar)	View workspace list	100%
Are you able to view all task list	Steps as below:1. Navigate to task page (click my task in the bottom bar)	view task list	100%

Are you able to view the task details	Steps as below:1. Click the Task list page2. Select the 'Create Gantt Chart' task	view task details	100%
Are you able to be redirected to google maps	Steps as below:1. Click the Task list page2. Select the 'Create Gantt Chart' task3. In the task detail page under details tab, click the location name	Redirect to google maps to direct access location and head over to the specific location	100%
Are you able to send a message	Steps as below:1. Click the Task list page2. Select the 'Create Gantt Chart' task3. In the task detail page under chat tab4. Type in the chat fi eld5. Click send icon	Send message to teams and inform updates easily	100%
Are you able to let client sign-off (you can imitate as the client as ofnow)	Steps as below:1. Click the Task list page2. Select the 'Create Gantt Chart' task3. In the task detail page under client tab4. Click the rectangular box 5. Draw a signature 6. Click the Upload	Lets client review and sign off to indicate that client is satisfacted upon task completion and ensure duly time of delivery	100%

Table 7.4: Staff Summary of Satisfaction towards implemented solution

Question	In aspects of	Rate Count						
		1	2	3	4	5	Average	%
This application helps with receiving and understanding my daily task assignments clearly.	Receive assigned task	0	0	0	5	10	0.9333333333	93.3
This application helps with submitting updates or progress on tasks with ease.	Updating task and inform members	0	0	5	7	3	0.7733333333	77.3
This application helps with adjusting or rescheduling my tasks when unexpected changes occur.	Informed of the task that is assigned to itself	0	0	5	6	4	0.7866666667	78.7

This application helps with keeping track of my overall task progress and deadlines.	Be informed of the task that is upcoming and due	0	0	3	7	5	0.8266666667	82.7
This application helps with viewing a summary of my completed and pending tasks.	Be informed of the task that has not been completed and needs attention	0	0	3	6	6	0.84	84.0
This application helps with offering a mobile-friendly experience for on-site access.	Ease of use especially for no tech exxpertise workers	0	0	3	4	8	0.8666666667	86.7
This application helps with facilitating direct communication with my supervisors or team members.	Enable seamless communication	0	0	7	8	0	0.7066666667	70.7
This application helps with accessing a history of all tasks, feedback, and comments.	View past history tasks and review back details	0	0	2	9	4	0.8266666667	82.7
This application helps with accessing necessary resources or tools related to my tasks.	Easy understanding of task assigned and can re confirm what tools will be needed	0	0	5	9	1	0.7466666667	74.7
This application helps with storing relevant information and notes for each task.	May always access the task assigned and confirm what is needed to be done	0	0	5	7	3	0.7733333333	77.3

CHAPTER 8

CONCLUSION & FUTURE WORK

8.1 Introduction

In this chapter, the culmination of a 4-month effort aimed at the development of an Employee Onsite Task Assignment Management and Tracking App will be explored. Throughout the preceding sections, the report has scrutinized various aspects of this project, from its inception to its execution. As the conclusion of this journey approaches, this sub-chapter will shed light on several critical facets: the achievements of the project's objectives, the limitations encountered during its lifespan, and, finally, recommendations for future endeavors in the realm of employee onsite task management and tracking.

8.2 Objective achivements

Designing and Developing the Mobile Application: The primary objective was to create a mobile application tailored for use by supervisors and employees to streamline onsite task management and assignment. This objective was achieved by conducting extensive research into user requirements, gathering feedback through surveys and interviews, and collaborating closely with a team of developers and designers. The result is an intuitive, user-friendly application that fulfills the needs of both supervisors and employees.

Analyzing User Challenges: To address the issues and obstacles encountered by users in managing and tracking onsite employees, a thorough analysis phase was undertaken. This involved collecting data on pain points, bottlenecks, and user preferences. The insights gained from this analysis informed the design and development process, allowing us to incorporate solutions directly addressing these challenges.

Creating Key Features: The creation of the mobile application with specific features was a pivotal aspect of the project's success:

Onsite Task Assignment System: We developed a robust system for onsite task assignment. This system captures essential details such as project location, project specifics, duration, and personnel requirements. Users can now effortlessly document and manage task assignments through the application.

Employee Assignment Flexibility: Our application allows for the seamless assignment of employees to various projects. Moreover, it permits the reassignment of tasks at any given moment. This flexibility empowers supervisors to adapt to changing project needs swiftly.

GPS-Based Location Tracking: The inclusion of GPS-based location tracking is a key milestone. It enables real-time monitoring of employee whereabouts and the automatic recording of check-in and check-out information. This feature enhances accountability and security.

Task Completion Monitoring and Visualization: We implemented a task completion monitoring feature that displays relevant data through an easy-to-understand dashboard. Utilizing visual representations such as graphs and pie charts, this feature provides a clear overview of project progress and employee performance.

These achievements signify the project's successful execution, underpinned by a thorough understanding of user needs, meticulous development, and a commitment to addressing the identified challenges comprehensively. The resulting mobile application stands as a testament to our dedication to delivering an efficient solution for employee onsite task management and tracking.

8.3 Project Limitations

In the course of developing the Employee Onsite Task Assignment Management and Tracking App within the stipulated 4-month timeframe, several limitations were encountered. These limitations are outlined below:

Lack of Clear Success and Error Messages: The application currently lacks clear and comprehensive success and error messaging. This absence of proper indication of states makes it challenging for users to understand whether their actions have been successful or if errors have occurred during their interactions with the app.

Performance Speed: One of the notable limitations is the speed of the application. Users have reported that the app is not fast enough, resulting in delays when navigating, assigning tasks, or retrieving information. This impacts the overall user experience and efficiency of the app.

Absence of Task Notifications: The app currently does not provide notifications to inform staff about incoming tasks. This limitation can lead to delays in task assignment and execution, as staff may not be promptly aware of their responsibilities.

Lack of Time Tracking: Another significant limitation is the absence of a time tracking feature. Staff members are unable to record the time it takes to complete tasks, including any overtime (OT) hours worked. This limitation hinders accurate tracking of task durations and personnel resource allocation.

Addressing these limitations will be crucial to enhance the functionality and usability of the Employee Onsite Task Assignment Management and Tracking App. In the subsequent section, recommendations for future work will be explored to provide potential solutions and improvements.

8.4 Recommendations

In considering the future work for the Employee Onsite Task Assignment Management and Tracking App, it's essential to address the identified limitations. Here are recommendations for future improvements:

Enhanced User Feedback and Messaging:

Implement a comprehensive system of success and error messaging within the app. Ensure that users using the application receive clear, descriptive messages that indicate the outcome of their actions. This will significantly improve user understanding and reduce frustration.

Optimize App Performance:

Conduct thorough performance testing and optimization to make the app faster and more responsive. Address bottlenecks in the app's code and database structure to improve navigation speed and task execution efficiency.

Task Notification System:

Integrate a real-time notification system to alert staff about incoming tasks. Notifications can be sent via push notifications or in-app alerts, ensuring that staff are promptly informed of new assignments. This feature will enhance task assignment and execution efficiency.

Time Tracking and Reporting:

Develop a time tracking feature within the app, allowing staff to record the time it takes to complete tasks, including any overtime (OT) hours worked. This data can be used for better task planning, resource allocation, and performance evaluation. Additionally, create reporting capabilities to generate insights from time tracking data.

User Feedback and Testing:

Continuously gather feedback from users using the app to identify any emerging issues or areas for improvement. Regular usability testing with users can help in refining the user experience and ensuring that the app meets evolving needs.

Security and Data Privacy:

Strengthen security measures to protect sensitive employee data and project information. Ensure compliance with data privacy regulations and standards, such as GDPR, to safeguard users information.

Documentation and Training:

Provide comprehensive user documentation and training resources to assist users in making the most of the app's features. Clear user guides and tutorials can improve user adoption and proficiency.

Integration with Other Systems:

Explore opportunities for integrating the app with other relevant systems or platforms used within the organization to streamline data sharing and workflow efficiency.

By addressing these recommendations in future development efforts, the Employee Onsite Task Assignment Management and Tracking App can evolve into a more robust and user-friendly tool, better suited to meet the needs of users and their organizations.

REFERENCES

4 phases of Rapid Application Development Methodology (2018) 4 Phases of Rapid Application Development Methodology | Lucidchart Blog. Available at: https://www.lucidchart.com/blog/rapid-application-development-methodology (Accessed: February 21, 2023).

Abubakar Shema, A. (no date) Figure2. 4: SDLC phases by Dennis et al.,(2005) - researchgate, Research Gate. Research Gate. Available at: https://www.researchgate.net/figure/Figure2-4-SDLC-phases-By- Dennis-et-al-2005_fig1_336022622 (Accessed: April 20, 2023).

Alaghbari, W. et al. (2007) The significant factors causing delay of building construction projects in Malaysia, Engineering, Construction and Architectural Management. Emerald Group Publishing Limited. Available at: https://www.emerald.com/insight/content/doi/10.1108/09699980710731308/full/html

Altvater, A. (2023) What is SDLC? understand the software development life cycle, What is the software development life cycle? Stackify. Available at: https://stackify.com/what-is-sdlc/ (Accessed: April 21, 2023).

Arumugam, T. (2023) Govt healthcare service at 'Breaking point', think tank calls for ..., Govt healthcare service at 'breaking point', think tank calls for urgent intervention. New Straits Time. Available at: https://www.nst.com.my/news/nation/2023/02/876409/govt-healthcare-service-breaking-point-think-tank-calls-urgent (Accessed: April 19, 2023).

Atlassian (no date) Scrum - what it is, how it works, and why it's awesome, Atlassian. Available at: https://www.atlassian.com/agile/scrum (Accessed: April 21, 2023).

Brikoshina, I.S., Birukov, A.P. and Geokchakyan, A.G. (1970) Smart transformation of the project management system and processes as a factor in increasing the efficiency and competitiveness of the project, SpringerLink. Springer International Publishing. Available at: https://link.springer.com/chapter/10.1007/978-3-030-59126-7_176 (Accessed: February 19, 2023).

Chien, C. (2020) What is rapid application development (RAD)?, Codebots. Available at: https://codebots.com/app-development/what-is- rapid-application-development-rad (Accessed: February 21, 2023).

Creasey, T. (2022) The costs and risks of poorly managed change, Prosci. Prosci Inc. Available at: https://www.prosci.com/blog/the-costs-risks-of-poorly-managed-change#:~:text=If%20the%20change%20does%20not,have%20invested%20in%20other%20projects (Accessed: April 19, 2023).

Definition of 'onsite' (no date) Collins Dictionary. HarperCollins Publishers Ltd. Available at: https://www.collinsdictionary.com/dictionary/english/on-site. (Accessed: February 19, 2023).

Deniz (2022) Bagaimana Untuk Menukar Teks Kepada Pertuturan Pada Android Pada 2022?, Speaktor. Speaktor. Available at: https://speaktor.com/ms/text-to-speech-android/ (Accessed: April 21, 2023).

Dey, S. (2021) Multitasking affects productivity and Brain Health, StartupTalky. StartupTalky. Available at: https://startuptalky.com/brain-health-multitasking/ (Accessed: April 19, 2023).

E. Ortiz, L. (2006) Practice problems for Constraint Satisfaction Problems, Notes on Algorithms for Constraint Satisfaction Problems 1. MIT EECS . Available at: http://www-personal.umd.umich.edu/~leortiz/teaching/6.034f/Fall06/csp/csp_probs.pdf (Accessed: April 20, 2023).

Egeonu, E. (2022) The advantages and disadvantages of the Rad Model, DistantJob. Available at: https://distantjob.com/blog/rad-model-advantages-and-disadvantages/ (Accessed: February 21, 2023).

Francis, A. (2018) A phased approach to software development, MBA Knowledge Base. MBA Knowledge Base. Available at: https://www.mbaknol.com/information-systems-management/a-phased- approach-to-software-development/ (Accessed: April 21, 2023).

Francis, A. (2018) A phased approach to software development, MBA Knowledge Base. MBA Knowledge Base. Available at: https://www.mbaknol.com/information-systems-management/a-phased- approach-to-software-development/ (Accessed: April 21, 2023).

Greedy algorithms introduction - javatpoint (no date) www.javatpoint.com. Available at:

https://www.javatpoint.com/greedy-algorithms#:~:text=Pseudo%20code%20of%20Greedy%20Algorithm&text=Initially%2C%20the%20solution%20is%20assigned,then%20we%20perform%20the%20union. (Accessed: April 21, 2023).

Hwang, B.-G. and Low, L.K. (2012) "Construction project change management in Singapore: Status, importance and impact," International Journal of Project Management, 30(7), pp. 817–826. Available at: https://doi.org/10.1016/j.ijproman.2011.11.001.

Jayathilaka, C. (2020) Waterfall vs. Agile Methodology, Medium. Medium. Available at: https://medium.com/@chathmini96/waterfall-vs- agile-methodology-28001a9ca487 (Accessed: April 21, 2023).

Joubert, S. (2020) The critical role of Communication in Project Management, Northeastern University Graduate Programs. Northeastern University. Available at: https://www.northeastern.edu/graduate/blog/communication-in-project-management/ (Accessed: February 21, 2023).

KUMAR PAL, S.A.Y.A.N. (2023) Software engineering: Classical waterfall model, GeeksforGeeks. GeeksforGeeks. Available at: https://www.geeksforgeeks.org/software-engineering-classical-waterfall-model/ (Accessed: April 21, 2023).

Kuroda, S. and Yamamoto, I. (2018) Why do people overwork at the risk of impairing mental health?, SpringerLink. Springer Netherlands. Available at: https://link.springer.com/article/10.1007/s10902-018-0008- x (Accessed: April 20, 2023).

Lessing, B., Thurnell, D. and Durdyev, S. (2017) Main factors causing delays in large construction projects: Evidence from New Zealand, SSRN. Available at: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3044315 (Accessed: February 21, 2023).

Lutkevich, B. and Lewis, S. (2022) What is the waterfall model? - definition and guide, Software Quality. TechTarget. Available at: https://www.techtarget.com/searchsoftwarequality/definition/waterfall-model (Accessed: April 21, 2023).

Mallawaarachchi, V. (2020) Introduction to genetic algorithms - including example code, Medium. Towards Data Science. Available at: https://towardsdatascience.com/introduction-to-genetic-algorithms-including-example-code-e396e98d8bf3 (Accessed: April 21, 2023).

Martinez, P. (2023) The definition of throwaway prototyping, The Definition of Throwaway Prototyping. Wondershare. Available at: https://mockitt.wondershare.com/prototyping/throwaway- prototyping.html (Accessed: April 21, 2023).

McQuillan, R. (2022) What is throwaway prototyping? in-depth guide, Budibase. Budibase. Available at: https://budibase.com/blog/inside-it/throwaway-prototyping/ (Accessed: April 21, 2023).

Muthohari, F. (2016) Pengembangan Sistem Informasi, Futuhm. Available at: https://futuhm.wordpress.com/2016/03/03/pengembangan- sistem-informasi/ (Accessed: April 21, 2023).

O'Neill, M. (2018) Communication failure: Why it happens and how Team Chat Apps can help, Samewave. Available at: https://www.samewave.com/posts/how-to-avoid-communication-fails-with-teamwork-chat-apps (Accessed: February 22, 2023).

Online, T.S. (2023) Delay of Monorail Project Cost Govt almost rm800mil, The Star. Available at: https://www.thestar.com.my/news/nation/2023/02/17/delay-of-monorail-project-cost-govt-almost-rm800mil (Accessed: February 21, 2023).

Online, T.S. (2023) Public Healthcare in Malaysia is understaffed – yet ministry is still not hiring permanent staff, The Star. The Star. Available at: https://www.thestar.com.my/opinion/letters/2023/01/17/public-healthcare-in-malaysia-is-understaffed---yet-ministry-is-still-not-hiring-permanent-staff (Accessed: April 20, 2023).

Online, T.S. (2023) Thai man dies from overworking; family to get compensation from company, The Star. The Star. Available at: https://www.thestar.com.my/aseanplus/aseanplus-news/2023/02/08/thai-man-dies-from-overworking-family-to-get-compensation-from-company (Accessed: April 20, 2023).

Orechwa, W. (2021) Closing the communication gap with Field Workers, LinkedIn. LinkedIn. Available at: https://www.linkedin.com/pulse/closing-communication-gap-field-workers-walter-orechwa (Accessed: February 21, 2023).

PadaKuu (2023) Prototyping model, www.padakuu.com. PadaKuu. Available at: https://padakuu.com/prototyping-model-598-article (Accessed: April 21, 2023).

Peek, S. (2023) What is agile scrum methodology?, What Is Agile Scrum Methodology? Business News Daily. Available at: https://www.businessnewsdaily.com/4987-what-is-agile-scrum-methodology.html (Accessed: April 21, 2023).

Schindler, E. (2010) 5 reasons to take a phased approach to software development, Computerworld. ITworld. Available at:https://www.computerworld.com/article/2749336/5-reasons-to-take-a- phased-approach-to-software-development.html (Accessed: April 21, 2023).

Sharma, A. (2022) Why should developers use Kotlin for Android App Development, Appinventiv. Appinventiv. Available at: https://appinventiv.com/blog/kotlin-for-android-app-development-the- whys-and-hows-bonus-tips/ (Accessed: April 21, 2023).

Stoop, P.P.M. and Wiers, V.C.S. (1996) "The complexity of scheduling in practice," International Journal of Operations & Production Management, pp. 37–40.

Supercool Creative Culture sdn bhd :: Powered by Universe2u.com.my (no date) Supercool Creative Culture Sdn Bhd :: Powered by universe2u.com.my. Available at: http://www.supercool.universe2u.com.my/ (Accessed: April 21, 2023).

Suri, A. (2021) Theming in flutter, WalkingTree Technologies. Available at: https://walkingtree.tech/theming-in-flutter-blog/ (Accessed: April 21, 2023).

What is Scrum? (no date) Scrum.org. Available at: https://www.scrum.org/learning-series/what-is-scrum (Accessed: April 21, 2023).

APPENDICES

Appendix A: Unit Test Cases

TEST CASE ID	TC01	**TEST NAME**	Register as Admin (Valid)	**PASS/ FAIL**	PASS
TEST CASE DESCRIPTION		Register as an admin using valid data.			
TEST CASE SCENARIO				**TEST DATA**	
1. Navigate to the registration page. 2. Toggle the button to select "Admin" registration. 3. Fill in the required fields with valid data. 4. Submit the registration form.				- Email: \`adminuser@mysite.my\` - Password: \`ValidPass123\` - Name: \`Ali Bin Abu\` - Address: \`Jalan Bukit Bintang, 55100 Kuala Lumpur\` - Phone Number: \`+60 12-345 6789\`	
EXPECTED RESULT				**ACTUAL RESULT**	
- User account with admin privileges should be created. - User should be redirected the login page.				Account created and redicrected to login page	

TEST CASE ID	TC02	**TEST NAME**	Register as Employee (Invalid)	**PASS/ FAIL**	PASS
TEST CASE DESCRIPTION		Attempt to register as an employee using invalid data.			
TEST CASE SCENARIO				**TEST DATA**	
1. Navigate to the registration page. 2. Ensure the button is untoggled to select "Employee" registration. 3. Fill in the required fields with invalid data (null). 4. Submit the registration form.				Null data	
EXPECTED RESULT				**ACTUAL RESULT**	
- User account should not be created. - User should receive an error message detailing the issues with the provided data.				User account is not created Required fields error message are shown	

TEST CASE ID	TC03	**TEST NAME**	Register as Employee (Valid)	**PASS/ FAIL**	PASS
TEST CASE DESCRIPTION		Register as an employee using valid data.			
TEST CASE SCENARIO				**TEST DATA**	
1. Navigate to the registration page. 2. Ensure the button is untoggled to select "Employee" registration. 3. Fill in the required fields with valid data. 4. Submit the registration form.				Email: \`employeeuser@mysite.my\` - Password: \`EmployeePass456\` - Name: \`Siti Binti Ahmad\` - Address: \`Jalan Sultan Ismail, 50250 Kuala Lumpur\`	

				- Phone Number: `+60 13-456 7890`	
EXPECTED RESULT				**ACTUAL RESULT**	
- User account with employee privileges should be created. - User should be redirected the login page.				Account created and redicrected to login page	
TEST CASE ID	TC04	**TEST NAME**	Login as Admin (Valid)	**PASS/ FAIL**	PASS
TEST CASE DESCRIPTION		.Test logging in with correct email and password (admin)			
TEST CASE SCENARIO				**TEST DATA**	
1. Navigate to the login page. 2. Enter the valid email and password. 3. Submit the login form.				- Email: `adminuser@mysite.my` - Password: `ValidPass123`	
EXPECTED RESULT				**ACTUAL RESULT**	
- User should be successfully logged in. - User should be redirected to their dashboard				Logged in and redirected to dashboard	
TEST CASE ID	TC05	**TEST NAME**	Login as Staff (Valid)	**PASS/ FAIL**	PASS
TEST CASE DESCRIPTION		.Test logging in with correct email and password (staff)			
TEST CASE SCENARIO				**TEST DATA**	
1. Navigate to the login page. 2. Enter the valid email and password. 3. Submit the login form.				Email: `employeeuser@mysite.my` - Password: `EmployeePass456`	
EXPECTED RESULT				**ACTUAL RESULT**	
- User should be successfully logged in. - User should be redirected to their dashboard				Logged in and redirected to Workspace page	
TEST CASE ID	TC06	**TEST NAME**	Login as Staff (Invalid)	**PASS/ FAIL**	PASS
TEST CASE DESCRIPTION		Test logging in with incorrect email and password.			
TEST CASE SCENARIO				**TEST DATA**	
1. Navigate to the login page. 2. Enter an invalid email and password. 3. Submit the login form.				- Email: `wronguser@example.com` - Password: `WrongPassword`	
EXPECTED RESULT				**ACTUAL RESULT**	
- User should remain on the login page. - User should receive an error message stating invalid credentials.				Not redirected and has error message	
TEST CASE ID	TC07	**TEST NAME**	Login as Staff (Invalid)	**PASS/ FAIL**	PASS

TEST CASE DESCRIPTION		Test logging in with either a missing email or password.			
TEST CASE SCENARIO				**TEST DATA**	
1. Navigate to the login page. 2. Enter only the email (leave the password field empty) or vice versa. 3. Submit the login form.				- Option 1: - Email: `employeeuser@mysite.my` - Password: (leave blank) - Option 2: - Email: (leave blank) - Password: `EmployeePass456`	
EXPECTED RESULT				**ACTUAL RESULT**	
- User should remain on the login page. - User should receive an error message stating the missing credentials.				Not redirected and has error message	

TEST CASE ID	TC08	**TEST NAME**	Create Task (Valid)	**PASS/ FAIL**	PASS
TEST CASE DESCRIPTION		Create a task with a valid name and description in a specific workspace.			
TEST CASE SCENARIO				**TEST DATA**	
1. Navigate to the create task page. 2. Select a workspace. 3. Enter a valid task name. 4. Enter a valid task description. 5. Click the "create" button.				- Workspace: `Workspace 1` - Task Name: `Design Logo` - Task Description: `Create a new logo for the rebranding project.`	
EXPECTED RESULT				**ACTUAL RESULT**	
- Task should be successfully created within "Workspace 1". - User should be navigated to the task page				Navigated to task page and created successfullty	

TEST CASE ID	TC09	**TEST NAME**	Create Task (Valid)	**PASS/ FAIL**	PASS
TEST CASE DESCRIPTION		reate a task with a valid name and no description in a specific workspace.			
TEST CASE SCENARIO				**TEST DATA**	
1. Navigate to the create task page. 2. Select a workspace. 3. Enter a valid task name. 4. Leave the task description field empty. 5. Click the "create" button.				- Workspace: `Workspace 1` - Task Name: `Develop Homepage` - Task Description: (leave blank)	
EXPECTED RESULT				**ACTUAL RESULT**	
" - Task should be successfully created within "Workspace 1" - User should be navigated to the task page"				Navigated to task page and created successfullty	

TEST CASE ID	TC10	**TEST NAME**	Create Task (Invalid)	**PASS/ FAIL**	PASS

TEST CASE DESCRIPTION	Attempt to create a task without entering a task name.

TEST CASE SCENARIO	**TEST DATA**
1. Navigate to the create task page. 2. Select a workspace. 3. Leave the task name field empty. 4. Enter a task description. 5. Click the "create" button.	" - Workspace: `Workspace 1` - Task Name:(leave blank) - Task Description: (leave blank) "

EXPECTED RESULT	**ACTUAL RESULT**
- Task should not be created. - User should remain on the create task page. - User should receive an error message indicating the missing task name.	Task not created and remained in the task page

TEST CASE ID	TC11	**TEST NAME**	Create Label (Valid)	**PASS/ FAIL**	PASS

TEST CASE DESCRIPTION	Test creating a label with a valid name.

TEST CASE SCENARIO	**TEST DATA**
1. Navigate to the create label page. 2. Enter a valid label name. 3. Click the "create" button.	- Label Name: `Urgent`

EXPECTED RESULT	**ACTUAL RESULT**
- Label should be successfully created. - User should receive a confirmation message, such as "Label created successfully". - The label list should display the new label "Urgent".	Label is created and present in the label list

TEST CASE ID	TC12	**TEST NAME**	Create Label (Invalid)	**PASS/ FAIL**	PASS

TEST CASE DESCRIPTION	Attempt to create a label without entering a name.

TEST CASE SCENARIO	**TEST DATA**
1. Navigate to the create label page. 2. Leave the label name field empty. 3. Click the "create" button.	- Label Name: (leave blank)

EXPECTED RESULT	**ACTUAL RESULT**
- Label should not be created. - User should remain on the create label page. - User should receive an error message indicating the missing label name.	Label is note created and error message is shown

TEST CASE ID	TC13	**TEST NAME**	View workspace List	**PASS/ FAIL**	PASS

TEST CASE DESCRIPTION	Validate that users can successfully view the workspace list by navigating to the workspace tab.

TEST CASE SCENARIO	TEST DATA
1. Start at the main/dashboard page. 2. Click/navigate to the workspace tab.	- N/A (since no input is required from the user)
EXPECTED RESULT	**ACTUAL RESULT**
- The user should be redirected to the workspace tab/page. - A list of workspaces (if any exist) should be displayed clearly and without any errors.	Workspace list is displayed

TEST CASE ID	TC14	**TEST NAME**	View task list	**PASS/ FAIL**	PASS
TEST CASE DESCRIPTION	Validate that users can successfully view the task list grouped by deadline when navigating to the "my task" tab.				

TEST CASE SCENARIO	TEST DATA
1. Start at the main/dashboard page. 2. Click/navigate to the "my task" tab.	- N/A (since no input is required from the user)
EXPECTED RESULT	**ACTUAL RESULT**
- The user should be redirected to the "my task" tab/page. - Tasks should be displayed and grouped according to their deadlines. - Each group of tasks should be distinguishably separated, making it clear which tasks fall under which deadline.	Task list is displayed according to the deadline group

TEST CASE ID	TC15	**TEST NAME**	View Staff List	**PASS/ FAIL**	PASS
TEST CASE DESCRIPTION	Validate that users can successfully view the entire staff list and can optionally use a search function to display the appropriate staff members.				

TEST CASE SCENARIO	TEST DATA
1. Start at the main/dashboard page. 2. Click/navigate to the "staff list" tab. 3. (Optional) Type into the search box to filter staff.	- Search Input (optional): `Shinyun`
EXPECTED RESULT	**ACTUAL RESULT**
- Upon navigating to the "staff list" tab, all staff members should be displayed. - If a search input (like 'Shinyun') is provided, only staff members with names matching or related to 'Shinyun' should be shown. - Non-matching staff entries should be filtered out from the displayed list. - If the search box is cleared, the full list of staff should reappear.	All staff members are displayed appropriately

TEST CASE ID	TC16	**TEST NAME**	View detail staff	**PASS/ FAIL**	PASS

TEST CASE DESCRIPTION	Validate that users can access a detailed staff profile by clicking on a user tile from the staff list, showcasing workspaces the staff is a part of, tasks assigned to them, and their current location (if they are clocked in).
TEST CASE SCENARIO	**TEST DATA**
1. Start by navigating to the "staff list" tab. 2. Browse through the list and click on a specific user tile to view their detailed profile.	- Selected Staff: \`Shinyun\`
EXPECTED RESULT	**ACTUAL RESULT**
- The detailed profile of \`Shinyun\` should open. - The profile should display a list of workspaces that \`Shinyun\` has been invited to. - The profile should also show a list of tasks that \`Shinyun\` is assigned to or involved with. - If \`Shinyun\` is currently clocked in, the system should display her current location. If she isn't clocked in, the location should either be hidden or display a status indicating she's not currently active. - All displayed data should be relevant only to \`Shinyun\` and should not mix with data from other staff members.	Staff detail is displayed appropriately

TEST CASE ID	TC17	**TEST NAME**	Call staff button	**PASS/ FAIL**	PASS
TEST CASE DESCRIPTION	Validate that the call button, when clicked, initiates a call to the respective staff member.				
TEST CASE SCENARIO				**TEST DATA**	
1. Navigate to the desired staff profile or staff list. 2. Locate the "call" button associated with the specific staff member. 3. Click the "call" button.				- Staff Member to be called: \`Shinyun\`	
EXPECTED RESULT				**ACTUAL RESULT**	
- After clicking the "call" button for \`John Doe\`, the system should initiate a call to him. - The user should be redirected to the dial pad with the phone number readily available and same with user				The page is redirected to dial pad page	

TEST CASE ID	TC18	**TEST NAME**	View location logs	**PASS/ FAIL**	PASS
TEST CASE DESCRIPTION	Validate the ability to view the location logs of a specific staff member, which updates every 30 minutes, by clicking on the location button.				
TEST CASE SCENARIO				**TEST DATA**	
1. Navigate to the respective staff member's profile or detail view. 2. Identify and click on the "location" button. 3. Review the displayed location logs.				- Selected Staff: \`Shinyun\`	
EXPECTED RESULT				**ACTUAL RESULT**	

- After clicking the "location" button for `Shinyun`, the system should display a list of locations she has been to. - The locations should be timestamped and the logs should be in chronological order, with the most recent log at the top. - The system should reflect updates to the log every 30 minutes if there are new locations or movements. - The user interface should clearly show the location data, possibly with additional details like duration at each location or distance traveled. - If `Shinyun` hasn't checked in during that duration system will show no logs available	The logs are appropriately show and only shown if user has checked in

TEST CASE ID	TC19	**TEST NAME**	Assign label to staff	**PASS/ FAIL**	PASS
TEST CASE DESCRIPTION	Validate the user's ability to navigate to the assign label widget page, view assigned labels, delete them, search for specific labels, and assign new labels to a staff member.				
TEST CASE SCENARIO				**TEST DATA**	
1. Navigate to the desired staff member's profile or detail view. 2. Click on the "settings" icon. 3. Once on the assign label widget page, view the labels currently assigned to the staff member. 4. Optionally, click the cross icon to delete a specific label. 5. Use the search box to filter available labels. 6. Click the "+" button next to a label to assign it to the staff member.				- Selected Staff: `Shinyun` -Label to be added : 'Construction' -Label to be deleted : 'Designer'	
EXPECTED RESULT				**ACTUAL RESULT**	
- After clicking the "settings" icon for `Shinyun`, the user should be navigated to the assign label widget page. - At the top, the assigned labels should be displayed. If `Designer` is among them, it should be removed upon clicking the cross icon. - Using the search box and entering the term "Construction" should filter the list of available labels to those containing the word "Construction". - Clicking the "+" button next to the `Construction` label should add it to `Shinyun`'s assigned labels.				The add and delete function of labels upon staff is successful and search is also working	

TEST CASE ID	TC20	**TEST NAME**	View Detailed Workspace	**PASS/ FAIL**	PASS
TEST CASE DESCRIPTION	Validate the user's ability to navigate to the workspace tab, select a specific workspace, and view a detailed breakdown including the list of tasks associated with the chosen workspace.				
TEST CASE SCENARIO				**TEST DATA**	

1. Navigate to the workspace tab within the application. 2. Browse through the list of available workspaces. 3. Select a specific workspace to view its details. 4. Upon selection, review the list of tasks associated with the chosen workspace.	Select workspace: 'Workspace 1'
EXPECTED RESULT	**ACTUAL RESULT**
- After navigating to the workspace tab, the user should be presented with a list of available workspaces. - Selecting `Workspace 1` should take the user to a detailed view of that workspace. - Inside the detailed view, a comprehensive list of tasks associated with `Workspace 1` should be displayed. Tasks should be clear, readable, and properly associated with the chosen workspace. - The interface should be intuitive, ensuring the user understands the relationship between the selected workspace and its associated tasks.	The workspace displays all appropriate details of tasks

TEST CASE ID	TC21	**TEST NAME**	Delete task	**PASS/ FAIL**	PASS

TEST CASE DESCRIPTION	Validate the user's ability to delete a task from within the detailed workspace view by clicking on the red bin icon associated with the specific task.

TEST CASE SCENARIO	**TEST DATA**
1. Navigate to the workspace tab within the application. 2. Browse through the list of available workspaces. 3. Select a specific workspace to view its details. 4. In the detailed view, expand the task list. 5. Locate a specific task and click on its associated red bin icon to initiate deletion.	- Workspace to Select: `Workspace 1` - Task to Delete: `Create Gantt Chart`
EXPECTED RESULT	**ACTUAL RESULT**
- After navigating to the workspace tab, the user should see a list of available workspaces. - Upon selecting `Workspace 1`, the user should be taken to a detailed view of that workspace. - In the detailed view, tasks associated with `Workspace 1` should be presented, and expanding the task list should reveal individual tasks with their associated red bin icons. - Clicking on the red bin icon next to `Create Gantt Chart` the task should be deleted.	The task is deleted

TEST CASE ID	TC22	**TEST NAME**	Invite / Remove existing employees from workspace	**PASS/ FAIL**	PASS

TEST CASE DESCRIPTION	Validate the user's ability to invite existing employees to a workspace from the detailed workspace view using the invite employee modal.

TEST CASE SCENARIO	TEST DATA
1. Navigate to the workspace tab and select a specific workspace to view its details. 2. Within the detailed view, click on the "settings" button. 3. Upon click, the "invite employee workspace" modal should appear displaying a full list of employees. 4. From this list, select certain employees to invite to the workspace. 5. Optionally, deselect any employees to remove their invitation. 6. Once satisfied with the selection, close the dialog.	- Workspace to Access: `Workspace 1` - Employees to Invite: `Shinyun, Employee2` - Employee to Deselect (assuming previously selected): `Employee 1`
EXPECTED RESULT	**ACTUAL RESULT**
- After accessing the detailed view of `Workspace 1` and clicking the "settings" button, the "invite employee workspace" modal should pop up. - The modal should provide a clear, readable list of all employees. - Upon selecting `Shinyun` and `Employee2`, they should be visibly marked as selected. - If `Employee 1` was previously selected, deselecting should be straightforward and should visibly mark him as unselected. - Post-closing the dialog, the workspace (in other features or in future accesses to the modal) should reflect the changes in invited employees.	The interface shows that the employee is successfully added and deselected

TEST CASE ID	TC23	TEST NAME	Invite new employees (Valid)	PASS/ FAIL	PASS
TEST CASE DESCRIPTION		Validate the user's ability to invite a registered employee to a workspace from the staff list page.			
TEST CASE SCENARIO				**TEST DATA**	
1. Navigate to the staff list page within the application. 2. Click on the "add members" icon button. 3. Type in the email address of a registered employee. 4. Upon search, the staff details related to the entered email should be displayed. 5. Select one of the available workspaces from the list. 6. Click "confirm" to finalize the invitation.				- Employee Email: `alice@email.com` - Workspace to Invite to: `Workspace 1`	
EXPECTED RESULT				**ACTUAL RESULT**	
- After entering the email `alice@email.com`, Alice's staff details should appear. - Upon selecting `Project Delta` and clicking "confirm", Alice should successfully be invited to `Workspace 1`.				Inside the workspace members settings it shows that Alice is already in the workspace	

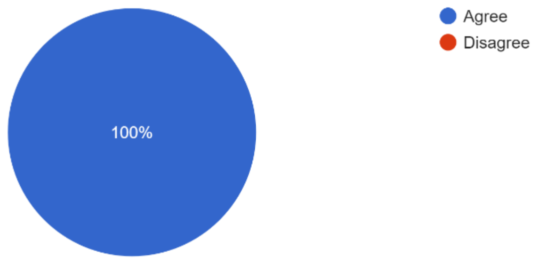
<table>
<tr><td><b>TEST CASE ID</b></td><td>TC24</td><td><b>TEST NAME</b></td><td>Invite new employees (Invalid)</td><td><b>PASS/ FAIL</b></td><td>PASS</td></tr>
<tr><td colspan="2"><b>TEST CASE DESCRIPTION</b></td><td colspan="4">Validate the system's response when attempting to invite an unregistered employee to a workspace from the staff list page.</td></tr>
<tr><td colspan="4"><b>TEST CASE SCENARIO</b></td><td colspan="2"><b>TEST DATA</b></td></tr>
<tr><td colspan="4">1. Navigate to the staff list page.
2. Click on the "add members" icon button.
3. Type in the email address of an unregistered employee.
4. Observe the system's response.</td><td colspan="2">- Employee Email: `unregistered@email.com`</td></tr>
<tr><td colspan="4"><b>EXPECTED RESULT</b></td><td colspan="2"><b>ACTUAL RESULT</b></td></tr>
<tr><td colspan="4">- After entering the email `unregistered@email.com`, the system should display an error or notification indicating that the user is not registered and therefore cannot be invited.</td><td colspan="2">Comply with above showing no employee is selected</td></tr>
</table>

Appendix B: User Acceptance Test Summary

Section 1 : Introduction and demographics

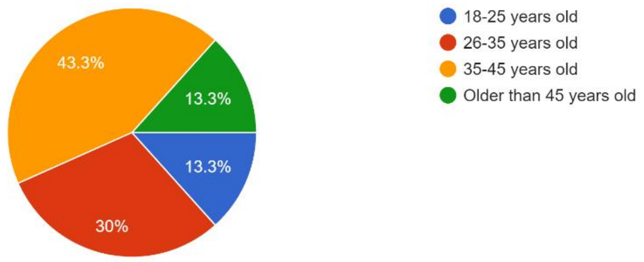
By clicking "Agree" below, you are indicating that you are

30 responses



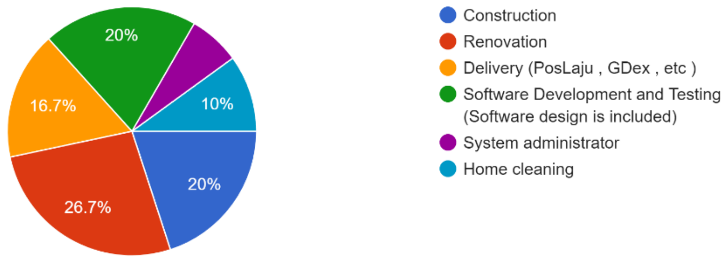
Age

30 responses



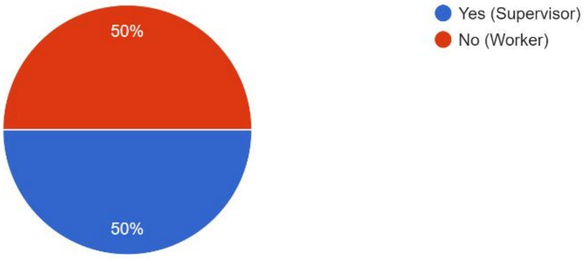
Nature field of work

30 responses



Are you involved in supervising other individuals at your workplace?

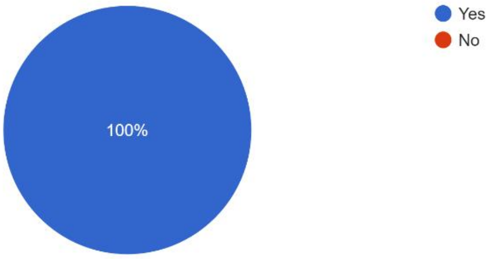
30 responses



Section 2 : Supervisor Section UAT

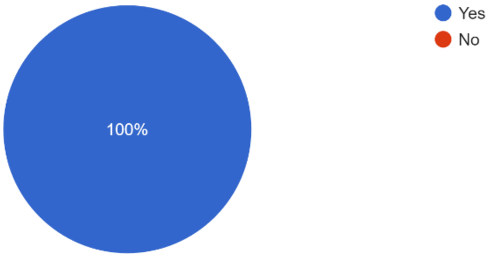
Are you able to login

15 responses



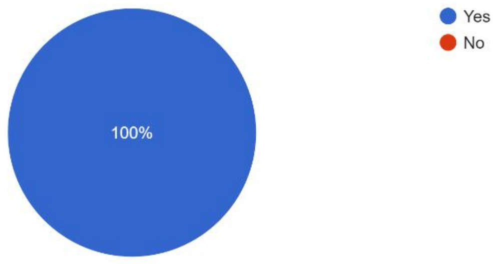
Are you able to create a workspace

15 responses



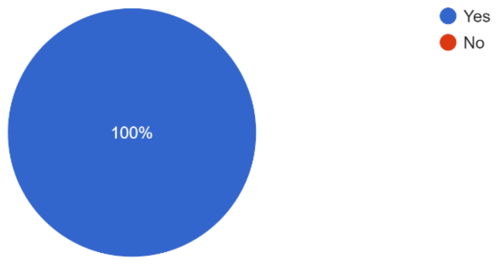
Are you able to create a label

15 responses



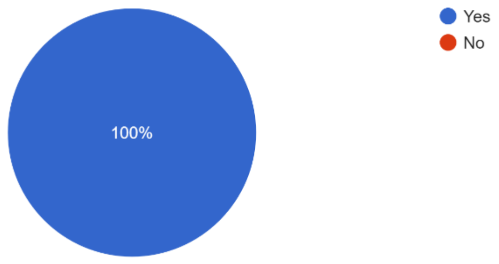
Are you able to view all workspace list

15 responses



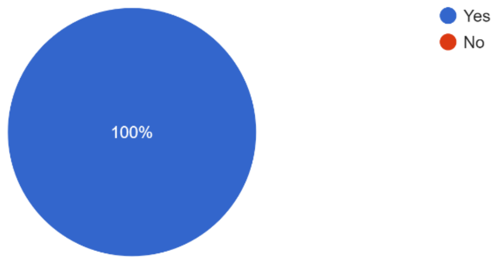
Are you able to view all task list

15 responses



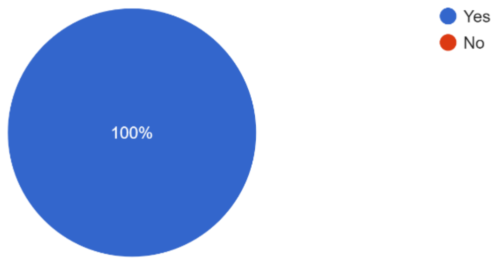
Are you able to view all staff list

15 responses



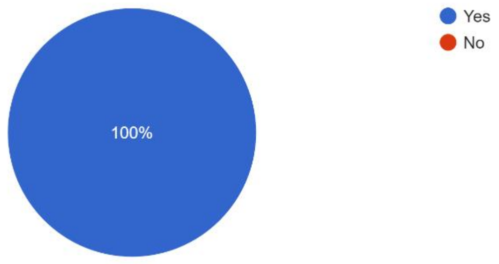
Are you able to invite new staff into your workspace

15 responses



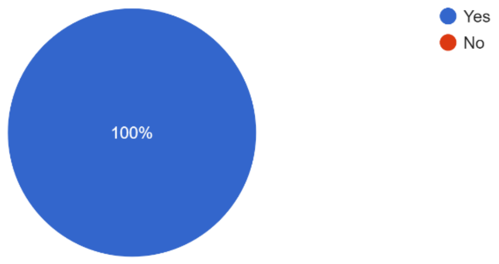
Are you able to assign label to a staff

15 responses



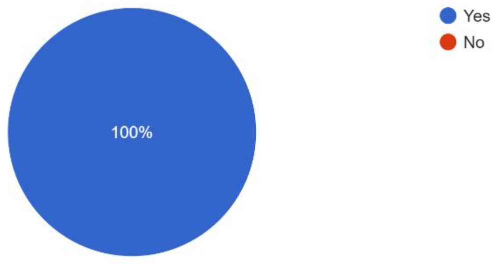
Are you able to view staff location log

15 responses



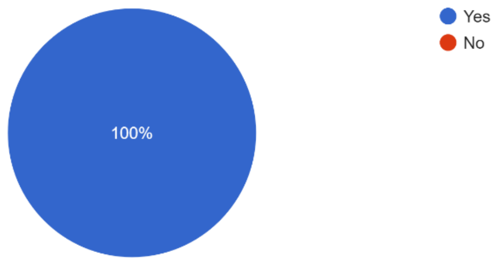
Are you able to update task details

15 responses



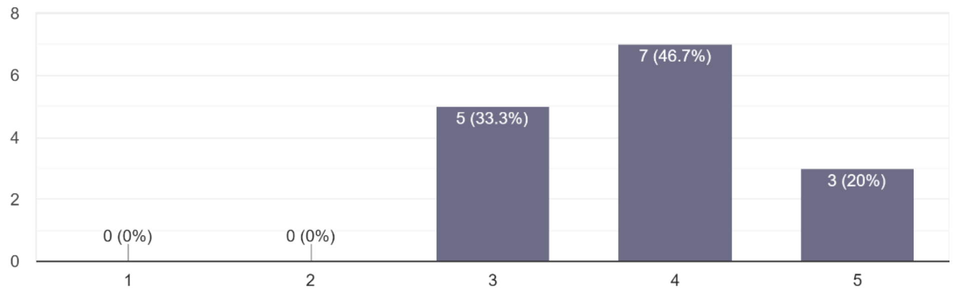
Are you able to send a message

15 responses



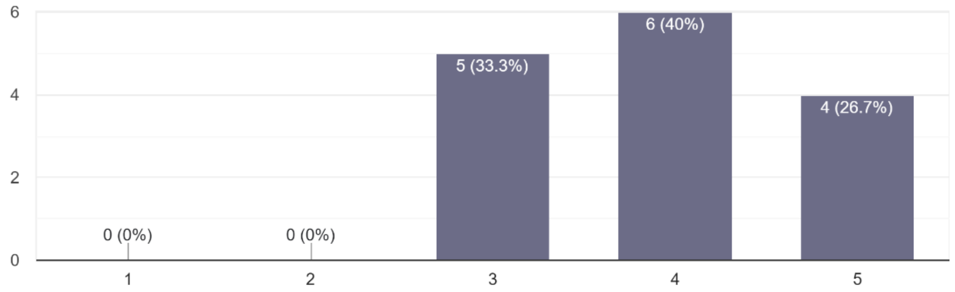
This application helps with efficiently assigning tasks to on-site team members.

15 responses



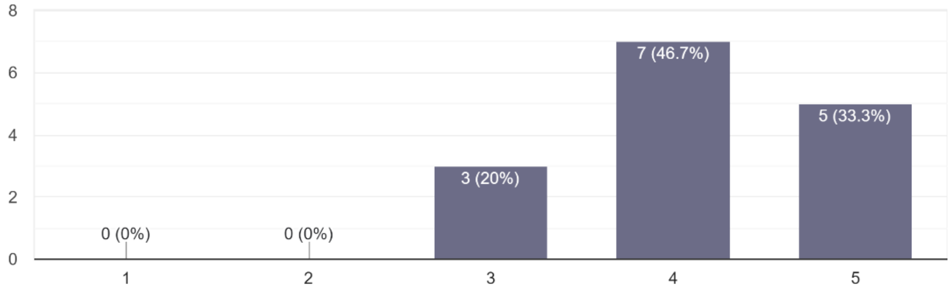
This application helps with simplifying the monitoring of multiple tasks at once.

15 responses



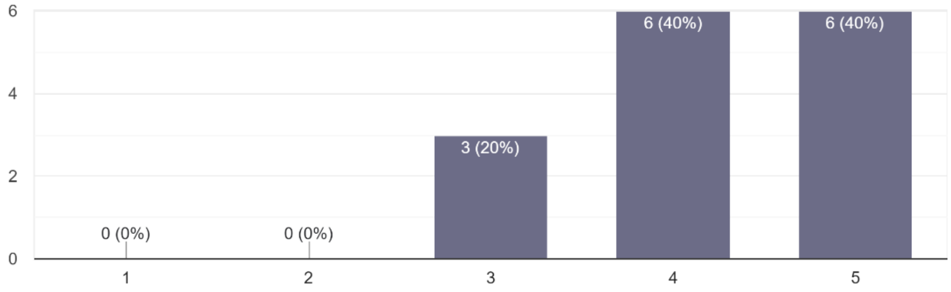
This application helps with reducing the need for manual follow-ups and reminders.

15 responses



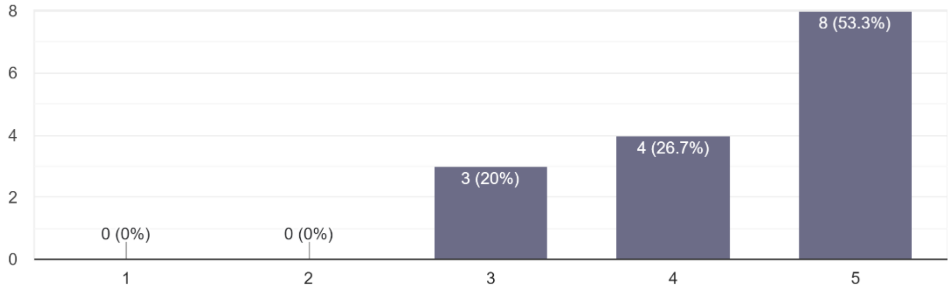
This application helps with facilitating smooth communication between team members.

15 responses



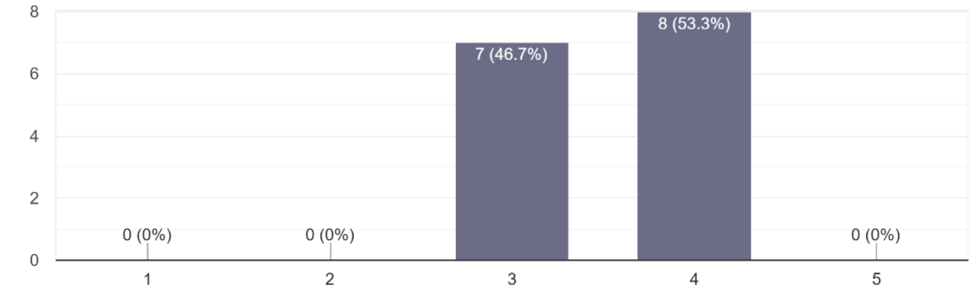
This application helps with easy access to task details and relevant documentation.

15 responses



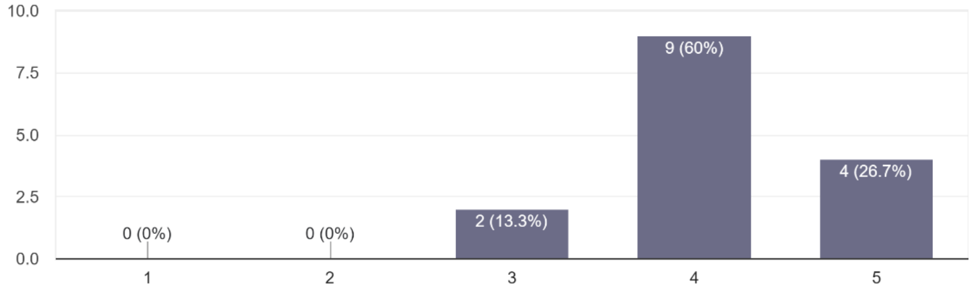
This application helps with the visual representation (charts) of team progress on task.

15 responses



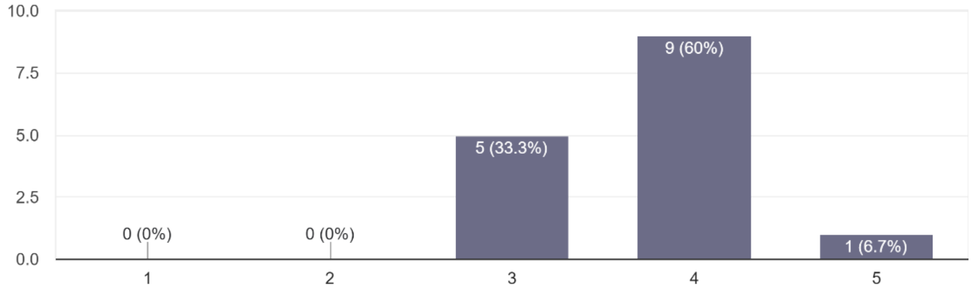
This application helps with streamlining the feedback process for completed tasks.

15 responses



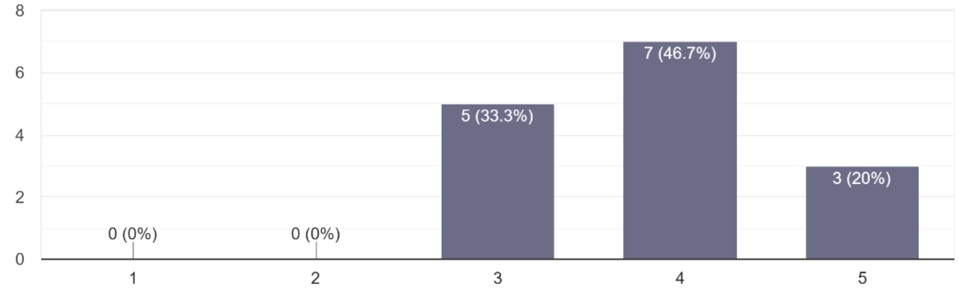
This application helps with the efficient reallocation or adjustment of tasks.

15 responses



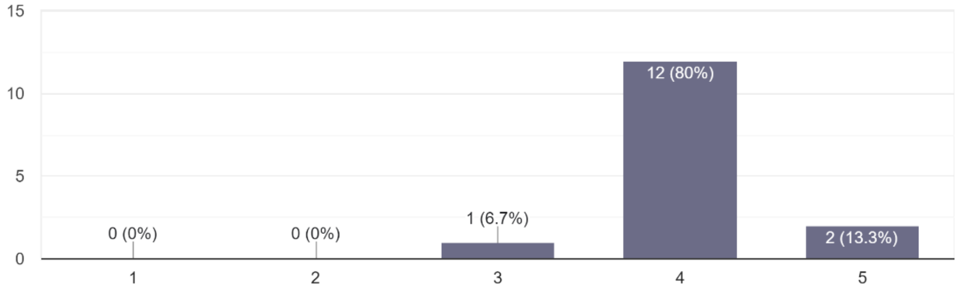
This application helps with logging any issues or concerns related to tasks.

15 responses



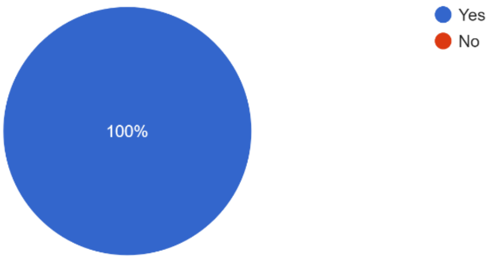
This application helps with providing a user-friendly interface that's easy to navigate.

15 responses



As a user , would you consider to sign-off the user acceptance test ?

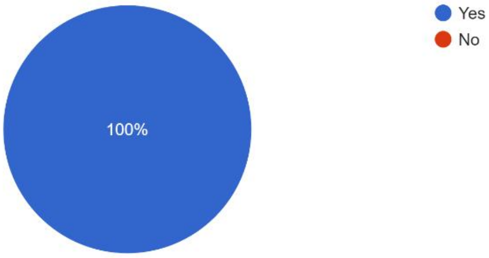
15 responses



Section 3 : Staff Section UAT

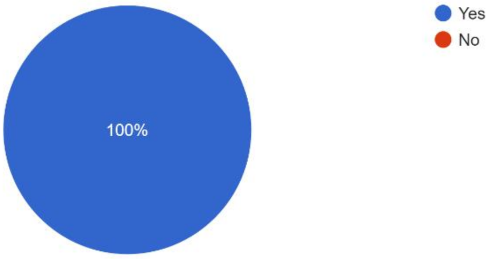
Are you able to login

15 responses



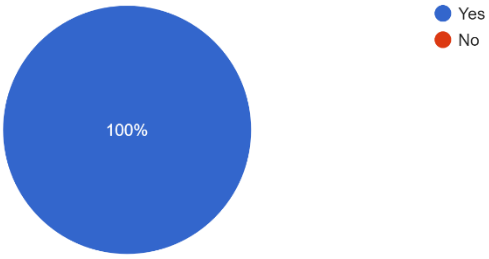
Are you able to clock in

15 responses



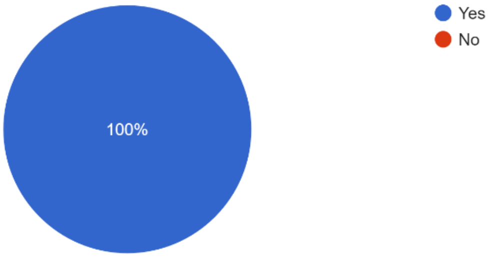
Are you able to view all workspace list

15 responses



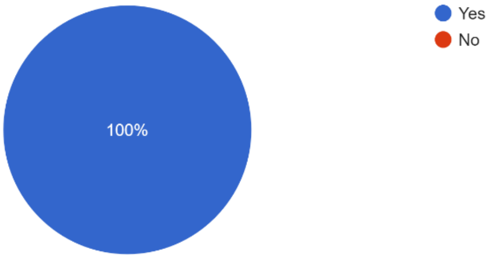
Are you able to view all task list

15 responses



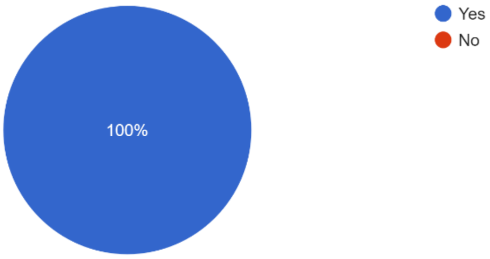
Are you able to view the task details

15 responses



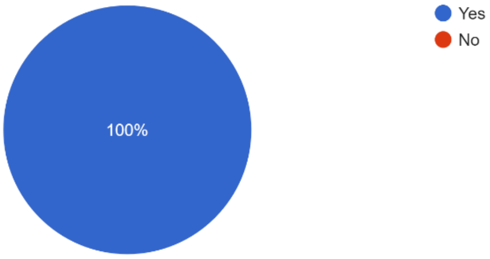
Are you able to be redirected to google maps

15 responses



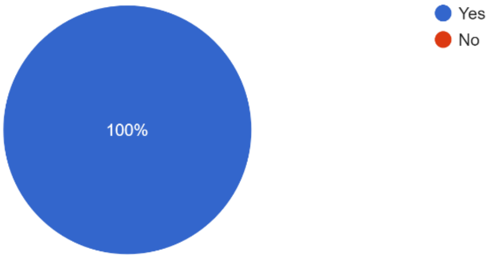
Are you able to send a message

15 responses



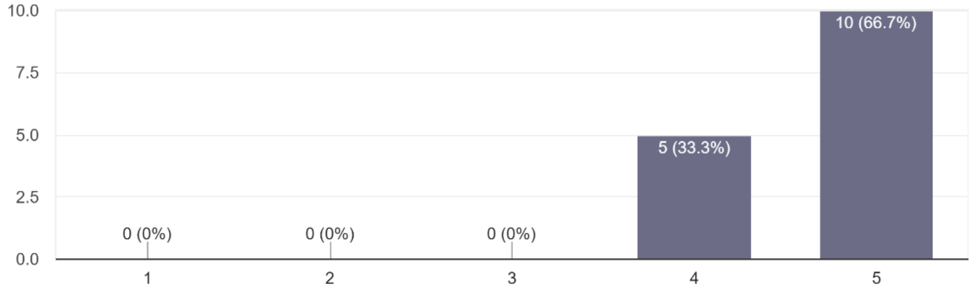
Are you able to let client sign-off (you can imitate as the client as of now)

15 responses



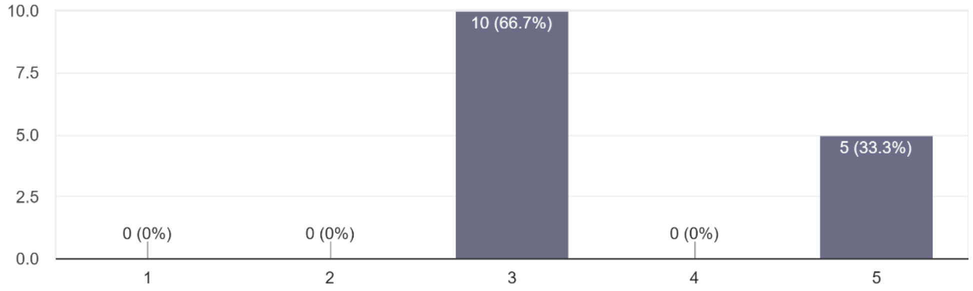
This application helps with receiving and understanding my daily task assignments clearly.

15 responses



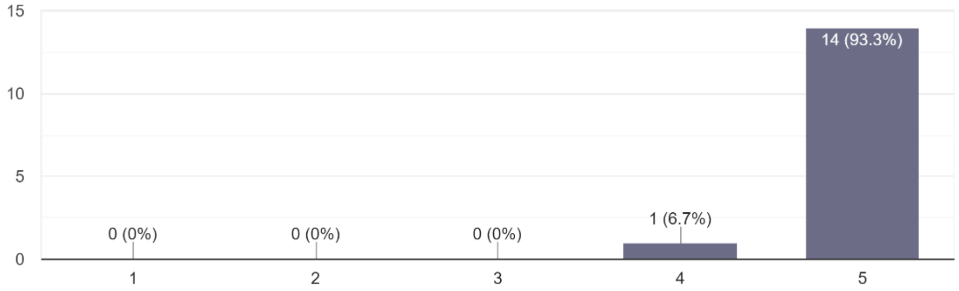
This application helps with submitting updates or progress on tasks with ease.

15 responses



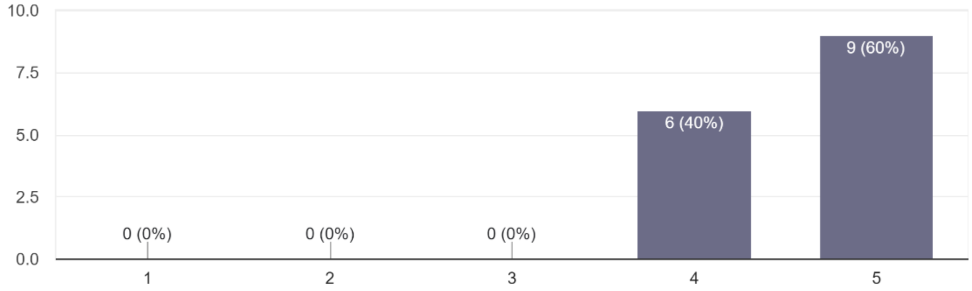
This application helps with adjusting or rescheduling my tasks when unexpected changes occur.

15 responses



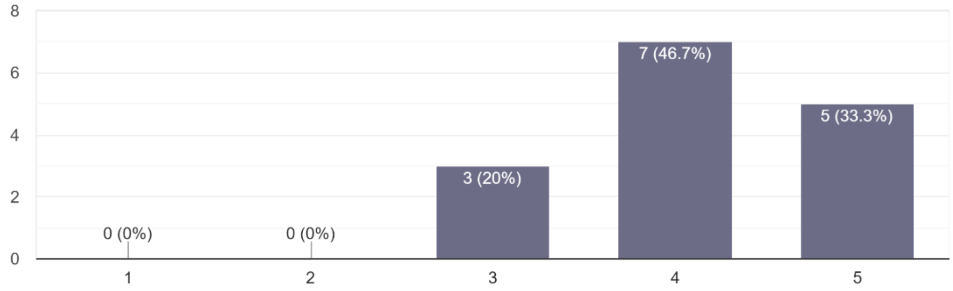
This application helps with keeping track of my overall task progress and deadlines.

15 responses



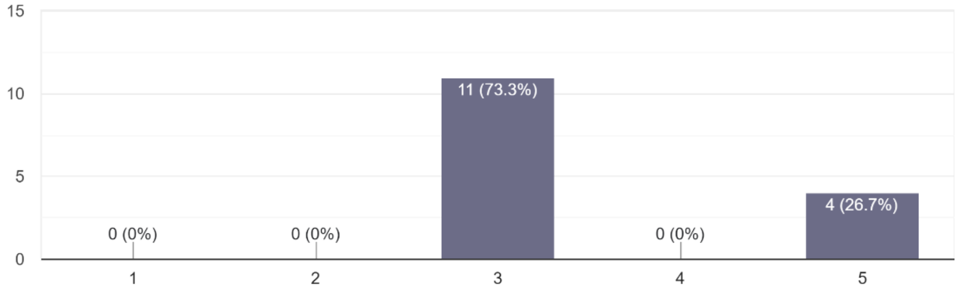
This application helps with viewing a summary of my completed and pending tasks.

15 responses



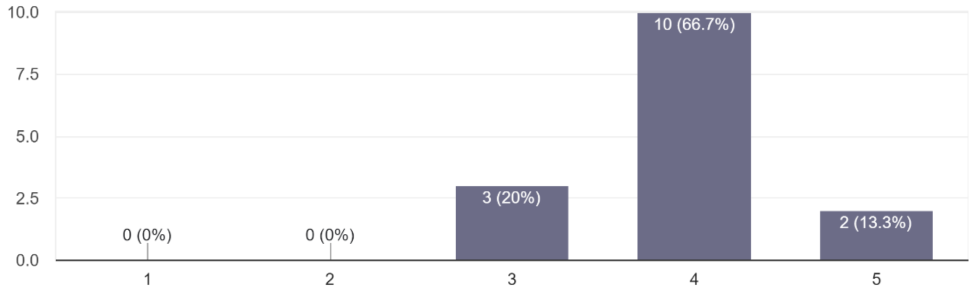
This application helps with offering a mobile-friendly experience for on-site access.

15 responses



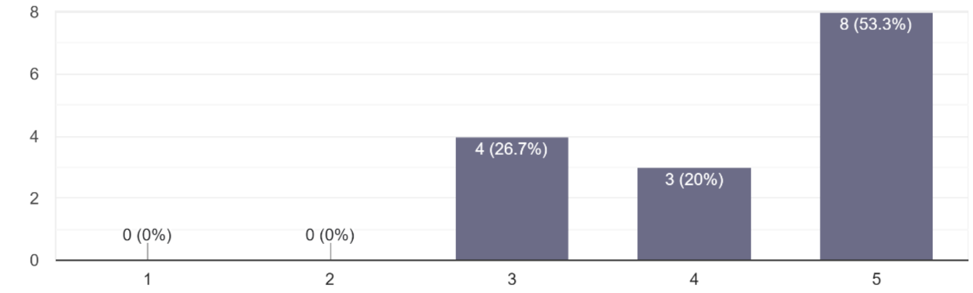
This application helps with facilitating direct communication with my supervisors or team members.

15 responses



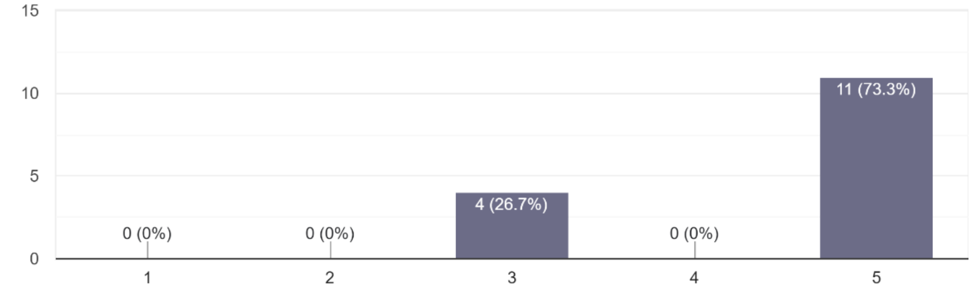
This application helps with accessing a history of all tasks, feedback, and comments.

15 responses



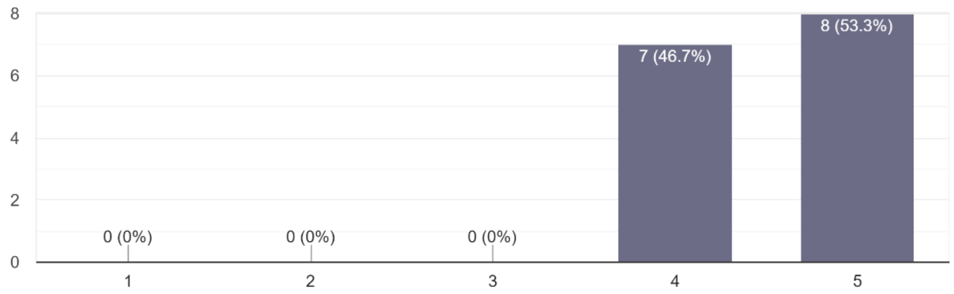
This application helps with accessing necessary resources or tools related to my tasks.

15 responses



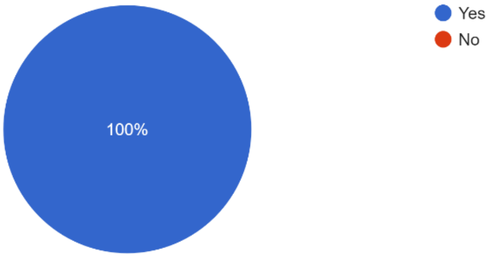
This application helps with storing relevant information and notes for each task.

15 responses

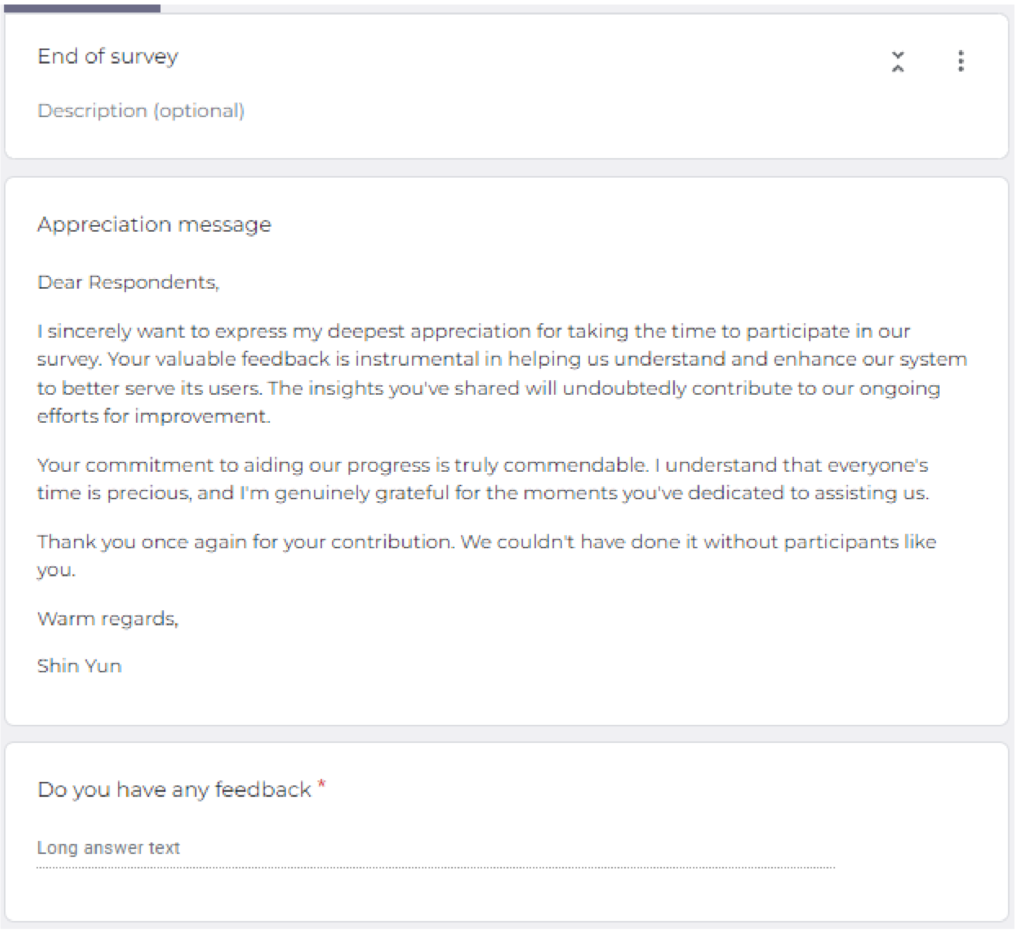


As a user , would you consider to sign-off the user acceptance test ?

15 responses



Section 4 : End of survey and appreciation message



End of survey

Appreciation message

Dear Respondents,

I sincerely want to express my deepest appreciation for taking the time to participate in our survey. Your valuable feedback is instrumental in helping us understand and enhance our system to better serve its users. The insights you've shared will undoubtedly contribute to our ongoing efforts for improvement.

Your commitment to aiding our progress is truly commendable. I understand that everyone's time is precious, and I'm genuinely grateful for the moments you've dedicated to assisting us.

Thank you once again for your contribution. We couldn't have done it without participants like you.

Warm regards,

Shin Yun

Do you have any feedback *

FYP Final V1

ORIGINALITY REPORT

6%	5%	0%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Universiti Tunku Abdul Rahman Student Paper	2%
2	eprints.utar.edu.my Internet Source	1%
3	Submitted to The University of Texas at Arlington Student Paper	<1%
4	www.slideshare.net Internet Source	<1%
5	Submitted to Middlesex University Student Paper	<1%
6	Submitted to Central Queensland University Student Paper	<1%
7	Submitted to University of Bradford Student Paper	<1%
8	www.seu1.org Internet Source	<1%
9	Submitted to University of Melbourne Student Paper	<1%