

THE QUADRY

VIGNNESH A/L RAVINDRAN

**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**

May 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 : Vignesh A/L Ravindran _____

ID No. : 1902683 _____

Date : 17-05-2023 _____

APPROVAL FOR SUBMISSION

I certify that this project report entitled “**THE QUADRY**” was prepared by **Vignesh A/L Ravindran** 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 :  _____

Supervisor : Ts. Dr. Sugumaran _____

Date : 17-05-2023 _____

Signature : _____

Co-Supervisor : _____

Date : _____

The copyright of this report belongs to the author under the terms of the copyright Act 1987 as qualified by Intellectual Property Policy of Universiti Tunku Abdul Rahman. Due acknowledgement shall always be made of the use of any material contained in, or derived from, this report.

© 2023, Vignesh Ravindran. All right reserved.

ACKNOWLEDGEMENTS

The success and final outcome of this project required a lot of guidance and advice from many different people throughout this research. Therefore, I would like to take this opportunity to specially express my gratitude to my research supervisors and academic advisors Ts. Loo Yim Ling and Ts. Dr. Sugumaran for their valuable advice, supervisions, and constant support throughout this research. I would also like to thank Maurchand Ent. for allowing me to use this project as my Final Year Project.

I am extremely thankful and privileged to have everyone's support for their assistance in various ways for this research including my parents, friends, and lecturers.

ABSTRACT

Electronic commerce (e-commerce) sites and applications have been the talk of the century as it is one of the most visible business uses of the World Wide Web. The primary goal of e-commerce sites and applications are to provide/sell goods and services on a virtual platform. As the world moves towards a more electronically driven environment, the implementations of e-commerce have also been on a constant rise. However, there has been resistance in local convenient and retail stores in adopting this growing technology. Hence, this project focuses on introducing the use of e-commerce technology through a cross-platform mobile application called The Quadry and its benefits such as payment gateway integration to allow customers to purchase and place orders online, automated order tracking and delivery status updates, merchant details such as location integrated with Google Maps API, operating hours, ratings, and many other features to the resisting party. Users on the other hand get to experience the simplicity and convenience of purchasing products from their local convenience stores for a relatively cheaper price than what the current e-commerce application market offers.

TABLE OF CONTENTS

DECLARATION		ii
APPROVAL FOR SUBMISSION		iii
ACKNOWLEDGEMENTS		v
ABSTRACT		vi
TABLE OF CONTENTS		1
LIST OF TABLES		5
LIST OF FIGURES		6
 CHAPTER		
1	INTRODUCTION	1
1.1	Introduction	1
1.2	Problem Background	2
1.3	Problem Statement	4
1.3.1	Concentration on specific types of merchants	4
1.3.2	Lack of advertisement & persuasion to attract local convenience stores	4
1.3.3	Markup prices in products	5
1.4	Project Objectives	6
a.	To analyse project requirements and compare existing ecommerce applications.	6
b.	To develop a model of the e-commerce application by integrating specific payment gateway.	6
c.	To develop an e-commerce application in 14 weeks.	6
d.	To evaluate application functionalities/features using the results outcome by performing user acceptance testing.	6

1.5	Proposed Solution	6
1.5.1	Technologies	6
1.5.2	Proposed Solution Workflow	8
1.6	Proposed Approach	9
1.7	Project Scope	10
1.7.1	Users of the System	10
1.7.2	Scope Covered	10
1.7.3	Scope Not Covered	11
2	LITERATURE REVIEW	12
2.1	Introduction	12
2.2	Literature Review	12
2.2.1	E-Commerce	12
2.2.2	Transition to M-Commerce	15
2.2.3	Summary	19
2.3	Rapid Application Development Methodology	31
2.3.1	Requirements Planning	31
2.3.2	Prototyping	31
2.3.3	Rapid Construction	32
2.3.4	Implementation / Cutover	32
2.4	Similar Existing Application	33
2.4.1	Grab Superapp	33
2.4.2	FoodPanda	37
2.5	Comparison between e-commerce applications	39
2.6	Conclusion	40
3	METHODOLOGY AND WORK PLAN	41
3.1	Introduction	41
3.2	System Development Methodology	41
3.2.1	Project Phases	42
3.2.2	Development Tools	45
3.3	Communication Tools	48
3.4	Project Plan	48

		3
	3.4.1	Work Breakdown Structure (WBS) 48
	3.4.2	Gantt Chart 49
4	PRELIMINARY PROJECT SPECIFICATIONS	53
	4.1	Introduction 53
	4.2	Functional Requirements 53
	4.3	Non-Functional Requirements 54
	4.4	Use Case 54
	4.5	Use Case Description 56
	4.6	Screenshots with Descriptions 65
5	DATA MODELLING	77
	5.1	Introduction 77
	5.2	Entity-Relationship Diagram (ERD) 77
	5.2.1	Order Entity Relationships 78
	5.2.2	User – Payment Entity Relationships 79
	5.2.3	Shop – Product Entity Relationships 79
	5.2.4	Shop – Review Entity Relationships 79
	5.3	Firebase NoSQL Logical Design Diagram 80
	5.3.1	Products Collection 80
	5.3.2	SendGrid Email Templates Collection 81
	5.3.3	Shops Collection 82
	5.3.4	Users Collection 82
	5.3.5	Welcome New Users Mail Collection 86
	5.4	Data Dictionary 87
	5.4.1	Products Data Dictionary 87
	5.4.2	SendGrid Email Templates Data Dictionary 88
	5.4.3	Welcome New User Mail Data Dictionary 88
	5.4.4	Shops Data Dictionary 89
	5.4.5	Reviews Subcollection Data Dictionary 91
	5.4.6	Users Data Dictionary 91
	5.4.7	Orders Subcollection Data Dictionary 92

	5.4.8	Payments Subcollection Data Dictionary	94
	5.4.9	Cart Subcollection Data Dictionary	94
	5.5	Conclusion	95
6		IMPLEMENTATION	96
	6.1	Introduction	96
	6.2	User Authentication Design	96
	6.2.1	Registration	96
	6.2.2	Login	97
	6.3	Add Products to Cart Design	98
	6.4	Purchasing Products via Stripe Payment Gateway	99
	6.5	User Address Selection	101
	6.6	Order Status Automation & User Notification	104
	6.7	Summary	105
7		SYSTEM TESTING	106
	7.1	Introduction	106
	7.2	Testing Objectives	106
	7.3	Testing Scope	106
	7.4	Testing Type	107
	7.5	Testing Results	107
	7.5.1	Unit Testing	107
	7.5.2	Integration Testing	113
	7.5.3	User Acceptance Testing (UAT)	116
	7.6	Summary	116
8		CONCLUSION & FUTURE DEVELOPMENTS	117
	8.1	Introduction	117
	8.2	Achieving Objectives	117
	8.3	Future Developments	118
	8.4	Summary	118
		REFERENCES	119

LIST OF TABLES

Table 2.1: Summary of Literature Review (Matrix)	18
Table 2.2: Comparison between e-commerce applications	38
Table 3.1: Project Timeline for The Quadry	51
Table 5.1: Products Data Dictionary	86
Table 5.2: SendGrid Email Template Data Dictionary	87
Table 5.3: Welcome New User Mail Data Dictionary	87
Table 5.4: Shops Data Dictionary	88
Table 5.5: Reviews Subcollection Data Dictionary	90
Table 5.6: Users Data Dictionary	90
Table 5.7: Orders Subcollection Data Dictionary	91
Table 5.8: Payments Subcollection Data Dictionary	93
Table 5.9: Cart Subcollection Data Dictionary	94
Table 7.1: Unit Test Case for Create Account	106
Table 7.2: Unit Test Case for Verify Phone Number	107
Table 7.3: Unit Test Case for Verify Email Address	108
Table 7.4: Unit Test Case for Login to Account	109
Table 7.5: Unit Test Case for Add Products to Cart	109
Table 7.6: Unit Test Case for Place Order	110
Table 7.7: Unit Test Case for Order Completed Notification	111
Table 7.8: Unit Test Case for User Address Selection	112
Table 7.9: Unit Test Case for Stripe Payment Gateway	113
Table 7.10: Unit Test Case for Email Automation with SendGrid	114
Table 7.11: User Acceptance Testing Results	115

LIST OF FIGURES

Figure 1.1:	Current accustomed workflow of an e-commerce application.	3
Figure 1.2:	Retail subsector sales comparison (Susendran, 2018)	4
Figure 1.3:	SDLC – RAD Model (Tutorialspoint, n.d.)	9
Figure 2.1:	Consumer behaviour on E-commerce (TheFutureofCommerce, 2022)	14
Figure 2.2:	Mobile e-commerce is up and poised for further growth (Loesche & Richter, 2018).	14
Figure 2.3:	Ecommerce Mobile App Development Guide - Cost and features Samsukha (2022).	15
Figure 2.4:	Mobile Apps Conversion vs Mobile Web (Blair, I., 2022).	16
Figure 2.5:	Estimated mobile payment transaction volume in the US (Richter, F., 2017).	17
Figure 2.6:	US M-Commerce Volume (Blair, I., 2022).	17
Figure 2.7:	Grab Features Inventory (Bolilia et al., 2019).	33
Figure 2.8:	Overview of Grab Superapp home interface.	34
Figure 2.9:	Recommended engine model (Bolilia et al., (2019)	35
Figure 2.10:	Ad-Serve architecture by Grab (McCallum et al., 2022).	36
Figure 2.11:	Overview of FoodPanda home interface.	37
Figure 2.12:	FoodPanda shops interface with a variety of shopping categories	38
Figure 3.1:	Work Breakdown Structure for The Quadry.	49
Figure 3.2:	Gantt Chart for Project 1 (until Requirement Elicitation phase).	52
Figure 3.3:	Gantt Chart for Project 2	52
Figure 4.1:	Use Case Diagram.	55
Figure 4.2:	Register Page (1)	65

Figure 4.3:	Register Page (2)	65
Figure 4.4:	Mobile number verification page	65
Figure 4.5:	Email verification page	66
Figure 4.6:	Email verified page after clicking verification link	66
Figure 4.7:	Login Page	66
Figure 4.8:	Homepage	67
Figure 4.9:	Homepage Drawer	67
Figure 4.10:	Selected Shop Category Page (Zookeepers)	68
Figure 4.11:	Selected Shop Category Page (Vape)	68
Figure 4.12:	Selected Shop Category Page (Grocery)	68
Figure 4.13:	Main shop page	69
Figure 4.14:	Reviews page	69
Figure 4.15:	About page	69
Figure 4.16:	Generated dynamic link for merchant	69
Figure 4.17:	Product page sample 1	70
Figure 4.18:	Product page sample 2	70
Figure 4.19:	Product page sample 3	70
Figure 4.20:	Generated dynamic link for product	70
Figure 4.21:	iOS native sharing options	70
Figure 4.22:	Favourites page (Products - Default)	71
Figure 4.23:	Favourites page (Shops)	71
Figure 4.24:	Cart page (Empty)	72
Figure 4.25:	Cart page (Filled)	72
Figure 4.26:	User profile page	73
Figure 4.27:	Email page	73

Figure 4.28: Password page	73
Figure 4.29: Mobile number page	73
Figure 4.30: Address page	73
Figure 4.31: New address page	74
Figure 4.32: New address page – address selection	74
Figure 4.33: New address page – Delivery notice alert	74
Figure 4.34: Checkout page	75
Figure 4.35: Checkout page (2)	75
Figure 4.36: Stripe FPX payment selection	75
Figure 4.37: Stripe test payment page	75
Figure 4.38: Successful test payment redirect	75
Figure 4.39: Unsuccessful test payment redirect	75
Figure 4.40: Order page (Past orders)	76
Figure 4.41: Order page (Ongoing orders)	76
Figure 4.42: Ongoing order details	76
Figure 4.43: Ongoing order details (2)	76
Figure 4.44: Order Delivered Pop-up	76
Figure 4.45: Order Delivered Pop-up (2)	76
Figure 5.1: Entity-Relationship Diagram of collections	77
Figure 5.2: Data Structure of Products Collection	80
Figure 5.3: Data Structure of SendGrid Email Template Collection	81
Figure 5.4: Data Structure of Shops Collection	83
Figure 5.5: Data Structure of Users Collection	84
Figure 5.6: Data Structure of Welcome New User Mail Collection	85
Figure 6.1: User Registration Architecture	96

Figure 6.2:	User Login Architecture	97
Figure 6.3:	Adding Products to Cart Architecture	98
Figure 6.4:	Purchasing Products Architecture	100
Figure 6.5:	User Address Selection Architecture	102
Figure 6.6:	Order Status Automation Architecture	104

CHAPTER 1

INTRODUCTION

1.1 Introduction

This final year project is an external project in collaboration with Maurchand Enterprise. Maurchand Ent. is a local based start-up company focused on providing reliable, modern solutions to clients and in that context; The Quadry was introduced. The Quadry is an e-commerce cross-platform mobile application which provides users with the opportunity to purchase groceries, utilities, services, and many more items from various local stores across the country.

With the steep upward trend of digitalisation, many companies and organisations are engaging with the implementation of automation, Internet of Things (IoT), data analytics and many more for not only the growth of business operations, but to also stay current with the latest trends of technology as it impacts the society and their working environments.

The ideology and perception of The Quadry can be generalized into an ordinary cross platform e-commerce application such as the ones that currently exist. In section 1.2, the uses and distinctive qualities of The Quadry from existing e-commerce applications are explained. This includes an illustration of the current workflow of the application.

Brief problem statements with respect to the implementation of The Quadry are discussed in section 1.3. The objectives of this project have been analysed and specified in section 1.4 whereas a proposed solution has been described in section 1.5. The proposed solution and proposed approaches (in section 1.6) are to solve the problem statements described in section 1.3 and achieve the project objectives.

The scope of the proposed solution has been specified in section 1.7, covering the scope of users of the developed solution, functionalities and also what will not be covered in this study.

1.2 Problem Background

It is evident that the evolution of digitalisation has made a significant impact on the growth of e-commerce. Several ways of how these influences had taken place include the advancement of payment services, personalisation with artificial intelligence (A.I.), effective marketing with social media and more (Dominic, 2022).

The transition to e-commerce is paid much attention to by market leaders in various sectors including grocery stores such as AEON, Village Grocer, and Lotus. This is due to the heavy influence caused by the pandemic that started in 2020. Since then, almost all large merchants have transitioned to e-commerce while simultaneously maintaining the traditional method of shopping merely to increase their exposure and provide more convenience to the public. Many of these organisations such as Jaya Grocer have developed their own e-commerce applications due to sufficient funding and aimed to be independent whereas local retailers have opted to settle for third-party merchant applications such as Grab and FoodPanda. However, brick and mortar shopping still exist in many countries including Malaysia due to the lack of advertisement to local merchants, raised prices in products, and abundance of concentration on a specific type of merchant such as restaurants.

Maurchand Ent. aims to attract local convenient and retail stores into the e-commerce world by introducing a cross-platform mobile application called The Quadry. This application provides stores from varying sectors to digitalise their businesses and services. For example, currently there are no platform that offers e-commerce services for plumbing shops however with The Quadry, this issue can be resolved easily. Similarly, The Quadry is able to provide opportunities for these merchants to promote their designated services such as cleaning of fish tanks or piping services for plumbing shops.

In addition to merchant benefits, users of the application can also leverage the opportunity to purchase products from local convenience stores rather than same types of shops offered in almost every e-commerce application in the current market such as KK Super Mart, 99 Speedmart, etc.

The accustomed workflow for an e-commerce application is described in the figure below.

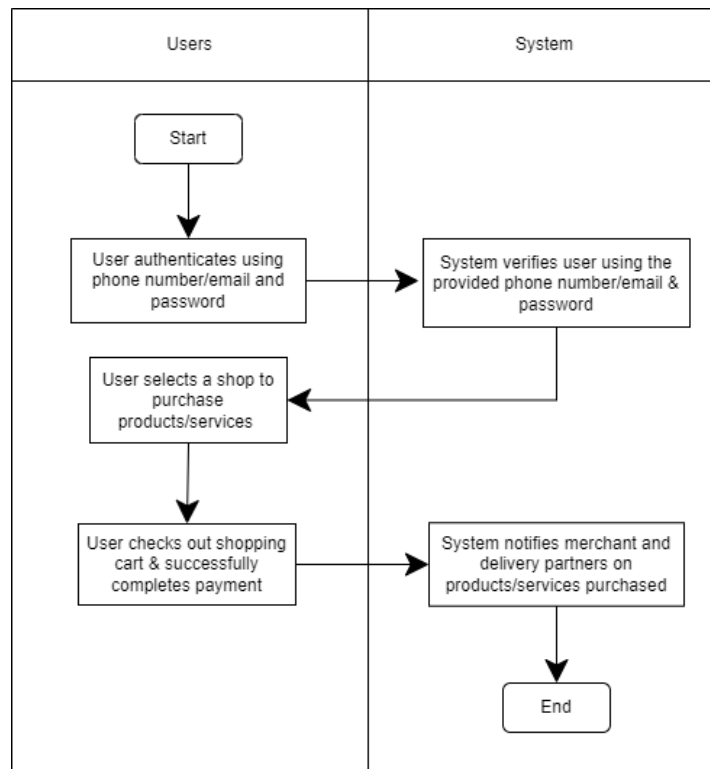


Figure 1.1: Current accustomed workflow of an e-commerce application.

The implementation and development of e-commerce systems such as The Quadry is essential in today's era for several reasons. Among them are:

- a. To drive additional business to local stores by expanding their products and services into an online space through an e-commerce platform.
- b. To enable local stores to remain competitive in the market by providing alternatives to the consumers to shop through an online platform while simultaneously preserving the brick-and-mortar shopping method.
- c. To provide local stores affected by the restrictions of the pandemic with an opportunity to bounce back from their financial struggles and continue their business on a virtual platform.
- d. To break the norm where only specific types of merchants can digitalise their business.

Despite having vast benefits, The Quadry still carries several flaws that could potentially cause as a hindrance to its progress.

1.3 Problem Statement

This section outlines the problems and its respective descriptions surrounding current e-commerce applications available in the market, in the aspects of merchant varieties, minimal efforts in attracting local merchants, and markup in product pricing. These are the problem statements that have led to the development of The Quadry.

1.3.1 Concentration on specific types of merchants

Most e-commerce and delivery applications focus on promoting their platform within the scope of restaurants and chain retail stores such as 7-Eleven, 99 Speedmart, and KK Mart.

This perspective can be broken and widened by introducing a variety of retailers such as department stores, grocery stores, internet/mobile stores, pet stores, etc. in a more local friendly centric where most of these businesses will not be chain retailers.

1.3.2 Lack of advertisement & persuasion to attract local convenience stores

It is evident that big chain retailers dominate the market due to their wide availability, however local convenience stores should also be provided the opportunity to digitalise their businesses to keep up with the competition. In an ideal situation, users of existing e-commerce and delivery applications are capable of only selecting mostly chained retailers due to the lack of variety.

By introducing local retail merchants, users will be able to select their favoured store to purchase an item due to the ability to make their own comparisons in terms of price, quality, store distance (as it affects the delivery amount), etc. Figure 1.2 portrays the comparison in growth of convenience stores to hypermarkets and supermarkets. For merchants however, digitalising their business is more towards a necessity than an option, otherwise they fail to reap the rewards and could potentially face losing relevance, competitive advantage, and social selling opportunities (Digital Marketing Institute, 2018).

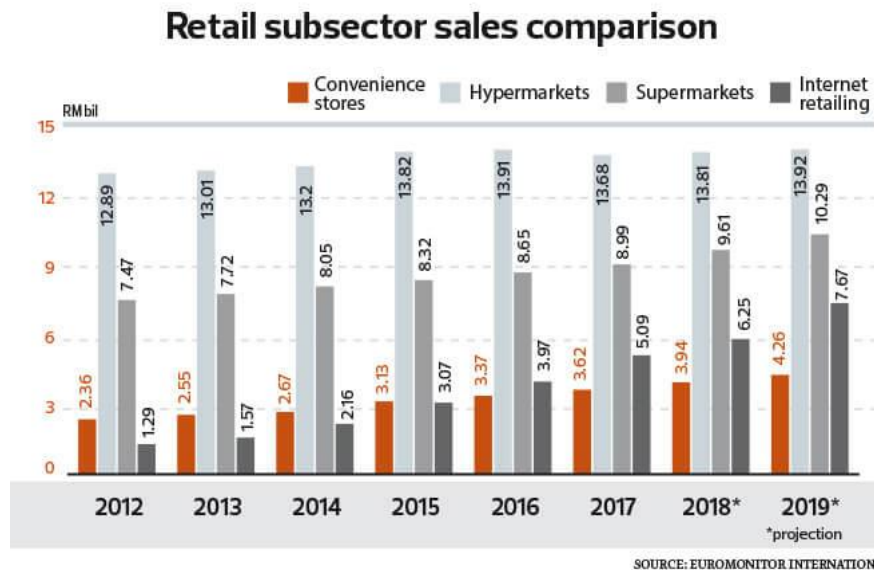


Figure 1.2: Retail subsector sales comparison (Susendran, 2018)

1.3.3 Markup prices in products

Notable e-commerce applications that are currently available in the Malaysian market such as FoodPanda, DeliverEat and RunningMan have price discrepancies in which price of items sold on the application differs from the buy-in price (Chan, 2017). Although these allegations have been denied by FoodPanda, there still lacks strict regulations on the policy of marking up prices and the transparency on how these prices are charged is yet to be clear.

As of today, Grab Superapp is one of the e-commerce applications that recommends and verifies merchants who have listed their prices on the application similar to their in-store prices after adopting to their 'no markup policy' (Grab, n.d.). By doing so, more users will be willing to order from that specific vendor and ultimately increasing the number of users for the application.

1.4 Project Objectives

This section defines the various project objectives that have been attained at the end of this project. These objectives will contribute towards solving the problem statements that have been defined in the previous section.

- a. To analyse project requirements and compare existing ecommerce applications.
- b. To develop a model of the e-commerce application by integrating specific payment gateway.
- c. To develop an e-commerce application in 14 weeks.
- d. To evaluate application functionalities/features using the results outcome by performing user acceptance testing.

1.5 Proposed Solution

This project was developed with technologies that are cross-platform supportable, scalable, and cost friendly. All technologies and proposed workflows have been discussed with Maurchand Ent. in advance.

1.5.1 Technologies

a. Android and iOS with Flutter (Dart)

- i. Both targeted operating systems are the most common among the Malaysian population. By using Flutter as the development framework, a cross-platform application can be developed.

b. Firebase

- i. Firebase is a Backend-as-a-Service platform (acquired by Google) that helps to build, scale, and improve an application. Implementing this technology will allow services such as cloud storage for various types of data needed for the application, authentication for users using a variety of methods such as Google and Facebook Sign In, and many more.

c. Google Cloud

- i. Google Cloud is a suite that offers a lot of cloud computing services. Among the services that could be used for this

development are cloud scheduler, Cloud Identity Access Management (IAM), etc.

d. Google Play Console & App Store

- i. Google Play Console allows for developers to better understand their applications in terms of performance, crashes, issues and more. This application will be deployed using this technology.
- ii. App Store is the Apple equivalent of the Google Play Store where developers or organisations will be able to upload their applications for user download. This platform will be used to host The Quadry application for iPhone users.

e. Google APIs

- i. The Google Maps, Google Places Autocomplete, and Google Places APIs provides valid and sufficient information regarding location details such as geo coordinates, place Ids, and more.

f. Stripe Payment Gateway

- i. Stripe is a financial and Software-as-a-Service (SaaS) company that provides payment gateway services. Among these services include accepting user payments using various types of transactional methods such as FPX, GrabPay, Debit/Credit cards, and more. Stripe will be used as the official payment gateway for Quadry due to its simplicity, feasibility, and vast types of services offered.

g. SendGrid

- i. SendGrid is a highly customisable customer communication platform that is used to deliver transactional and marketing emails to users via their API. Among the configurations that comes along with this technology is the ability to automate emails, use custom domain names, select/modify ready-made templates, and more. This technology is used in The Quadry application to send automated welcome emails to users who register for the first time.

1.5.2 Proposed Solution Workflow

All proposed project objectives specified in 1.4 have been achieved with the development of The Quadry. Methods and structure to achieve these goals respectively is briefly explained below.

To achieve objective 1.4.1., comparison of The Quadry was performed with a minimum of 2 existing and widely used e-commerce applications to identify its significance. Gathered general project requirements based on the analysis and comparison conducted and deduced justifications on how The Quadry will stand out and gain traction among merchants and targeted users. The duration of this activity was 2 weeks from week 3 to week 5 of the Project 1 period. This task is measured by the completion of at least 2 comparisons between The Quadry and other existing e-commerce applications and why it stands out. The objective of this task is to portray how this project will produce an e-commerce application that attracts local merchants. To ensure the goal of this task was achieved, a thorough research was conducted which includes attractive methods on how to appeal to these merchants. This task was completed on week 6 of the Project 1 period. However, it was frequently revised and updated as the project progressed.

Objective 1.4.2 was achieved by constructing a model for the entire e-commerce application by describing the overall framework and workflow with the integration of payment gateway platform. The objective of this task is to show the overall structure and data passage of The Quadry application and it is measured by the completion of said diagram which covers the entire structure of the application. This task was completed on week 10 of the Project 1 period. However, it was frequently revised and updated as the project progressed.

To achieve objective 1.4.3., a cross-platform mobile application was developed for this project to ensure users of either Android or iOS systems are accessible to the application. The fulfilment of this task is measured from the full coverage of project scope mentioned in Project 1. The accomplishment of this task was measured by the acceptance of the developed mobile application by Maurchand Ent. In this task, several technical stacks were used to meet the requirements

specified by Maurchand Ent. such as building a cross-platform mobile application that supports many users. Among the stacks that were used to facilitate this task include Flutter (Dart) framework, Firebase as the backend and database and Google Cloud to deploy and integrate required APIs. This task was carried out in a Rapid Application Development (RAD) business model to produce a rapid product delivery as per Maurchand Enterprise's requirements. This objective was compulsory and critical to achieve as it affects resources, schedules, and the user acceptance testing phase. The completion period of this task was on week 10 of Project 2. However, it was frequently revised and updated as the project progressed.

The last objective, 1.4.4, was achieved by focusing on developing test cases and test scripts for the developed mobile application. Among tests that were conducted include unit tests, integration tests and user acceptance testing and the breakdown for each test is as follows. Unit test comprises of functionalities of the application and to ensure they are operating as intended. Integration testing is comprised any third-party feature integration such as the chosen payment gateway (Stripe) and the mobile application to ensure payment is accepted and properly captured in case of any declinations. User acceptance testing was performed by Maurchand Ent. during the project completion phase. All tests were conducted during the project development phase in Project 2 period and was completed on week 10 of Project 2. However, it was frequently revised and updated as the project progressed. The completion of this task was derived through the test outcomes performed during development phases in Project 2.

1.6 Proposed Approach

The chosen development methodology for this project is Rapid Application Development (RAD) as its key benefit is fast project turnaround thus reducing planning time and iterating through prototype cycles. This methodology was selected to facilitate Maurchand Enterprise's requirements to develop the application as quickly as possible.

Figure 1.3 illustrates several processes in the RAD methodology and its workflow.

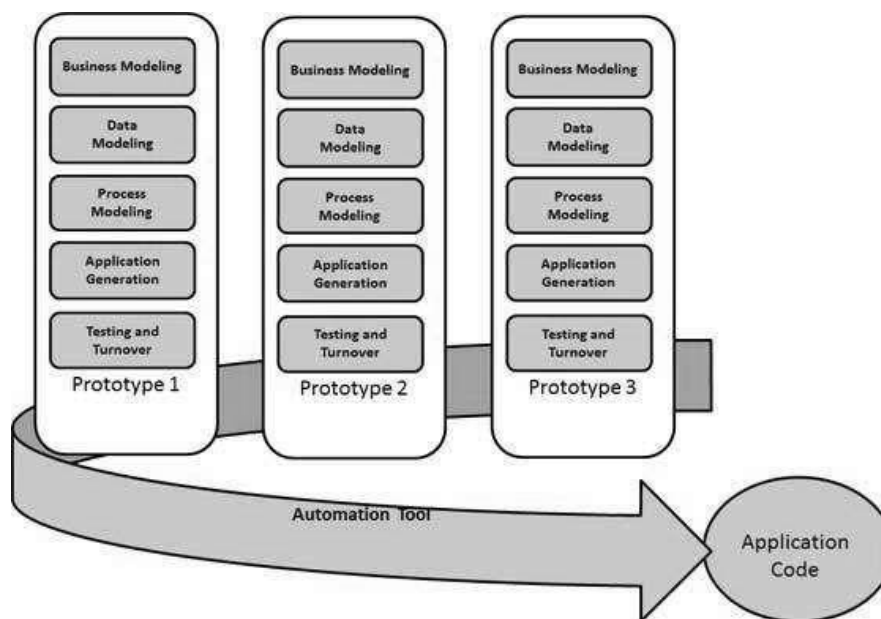


Figure 1.3: SDLC – RAD Model (Tutorialspoint, n.d.)

1.7 Project Scope

With the focus of digitalising local businesses and providing an online platform to local merchants, a cross platform application was developed. This project was scoped around any types of local stores such as pet stores, grocery stores, etc.

1.7.1 Users of the System

a. Online shoppers.

- i. Residents or more specifically, online shoppers are the target audience of this application. Users can use this application to purchase any products sold by their local stores.

1.7.2 Scope Covered

As a product of this project, an e-commerce application covers many entities and therefore, only a portion of the scope of this application is detailed in this project while the rest is omitted.

- a. A fully functional e-commerce application that contains various functionalities such as:

- i. Users can add any products from various merchants into their carts.
 - ii. Users can purchase variety of products via debit/credit card, bank transaction, or e-wallet (GrabPay).
 - iii. Users can update their personal information such as email, password, mobile number, and address.
 - iv. Users can register an account for themselves using the application with two-factor authentication.
 - v. Users can leave a comment and rating on a shop upon successfully receiving their purchased product(s).
 - vi. Users can share a particular store to their friends and family – enabled with dynamic linking.
 - vii. Users can favourite any products or stores available in the application.
 - viii. Users can contact the support team in case of any mishap.
 - ix. Users can track their past and current orders made from the application.
 - x. Users will receive an invoice upon every successful purchase.
- b. Only certain types of local stores such as pet stores, grocery stores and vape stores are displayed with detailed information such as:
- i. Location – integrated with Google Maps API.
 - ii. Products available for purchase through the application.
 - iii. Reviews and comments about their store & products from previous buyers.
 - iv. Operation hours & overall rating.

1.7.3 Scope Not Covered

The following aspects are not covered in this project.

- a. Local stores other than specified types of stores such as plumbing stores.
- b. Merchant application.
- c. Rider application.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In this chapter, a thorough explanation on e-commerce, its workflow, architecture, and comparison between three existing applications will be discussed. Several studies and research were conducted on the Rapid Application Development methodology to understand the best ways to implement it in this project.

This literature review is conducted by reading the official websites of the applications in comparison and exploring various related articles and research papers to understand the workflow in these applications.

2.2 Literature Review

2.2.1 E-Commerce

E-commerce, or popularly referred to as online shopping, is the activity of purchasing or selling of goods or services using the internet, in exchange for money and data to execute these transactions (Shopify, n.d.). A store or business is identified as an e-commerce store if they sell their products on an online platform.

Since its discovery, e-commerce platforms and technologies have come a long way as many freelancers, small businesses, and large corporations have adopted e-commerce to their fullest advantage. Four main categories of e-commerce models that can describe business transactions between the business and customer are as follows.

- a. Business to Customer (B2C)
 - i. This model describes the transaction process between a customer that buys a product from an online retailer. For example, buying a skateboard from an online store. It is mainly used by users of the e-commerce platform.
- b. Business to Business (B2B)
 - ii. This model describes when a business sells a product or service to another business. For example, a software company that

adopts Amazon Web Services (AWS), for their development and in this case, Amazon implements a B2B service since they sell their services for other businesses to use.

- c. Consumer to Consumer (C2C)
 - iii. This model describes a consumer selling their used/unused items to other consumers on an online platform such as Mudah (www.mudah.my).
- d. Consumer to Business (C2B)
 - iv. This model is applicable when a consumer sells their product/services to businesses. For example, a company that would like to promote a certain product can be done with the help of famous celebrities or influencers. In this case, the celebrity/influencer sells their service to the business.

E-commerce models are important for any business who wants to transition their business operations to a more hybrid setting such as maintaining the brick-and-mortar shopping whilst selling their products on an online platform. This is because, these models provide a clear picture as to what needs to be sold, to whom, and how can it be done properly. Once they have identified the right model, it would then be easier to select the right e-commerce platform to sell their products/services. As different e-commerce platforms provide services based on different models, the product of this project, The Quadry, will be developed mainly around the B2C model.

Most people would agree to the growth of e-commerce being significant since it was introduced, but the catalyst that resulted in the booming era of its usage today is the recent pandemic. According to (Koetsier, 2021), the Covid-19 pandemic has hugely sped up the growth of e-commerce by a staggering duration of 4 to 6 years in advance. Figure 2.1 illustrates the surge in e-commerce usage and purchase after the pandemic declaration. Both consumers and businesses have started to transition into modern times by adopting to e-commerce platforms to sell their products/services. Due to this transition, there has been a rise in development of various e-commerce websites/applications to keep up with the demands. The total m-commerce, also

known as mobile e-commerce, sales for the last 6 years is portrayed as Figure 2.2.

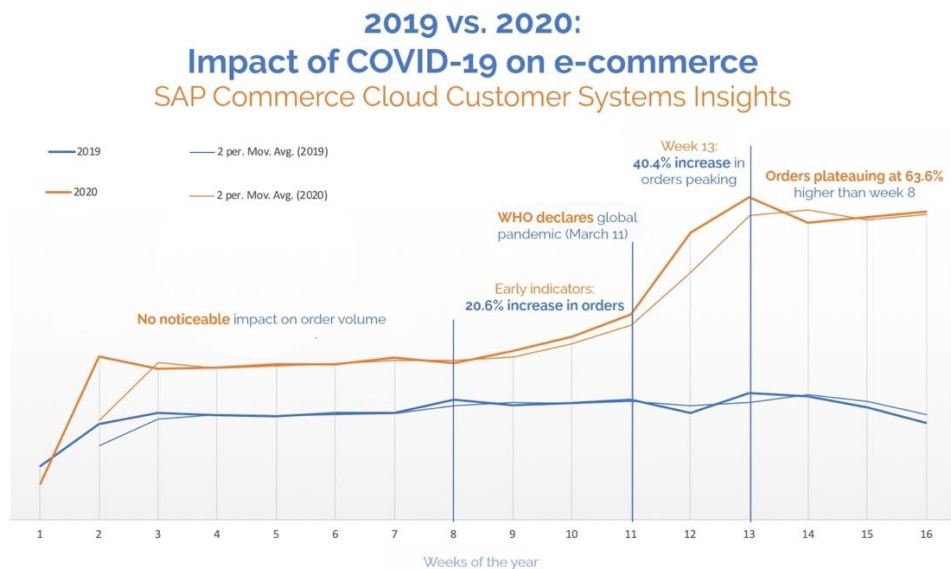


Figure 2.1: Consumer behaviour on E-commerce (TheFutureofCommerce, 2022).

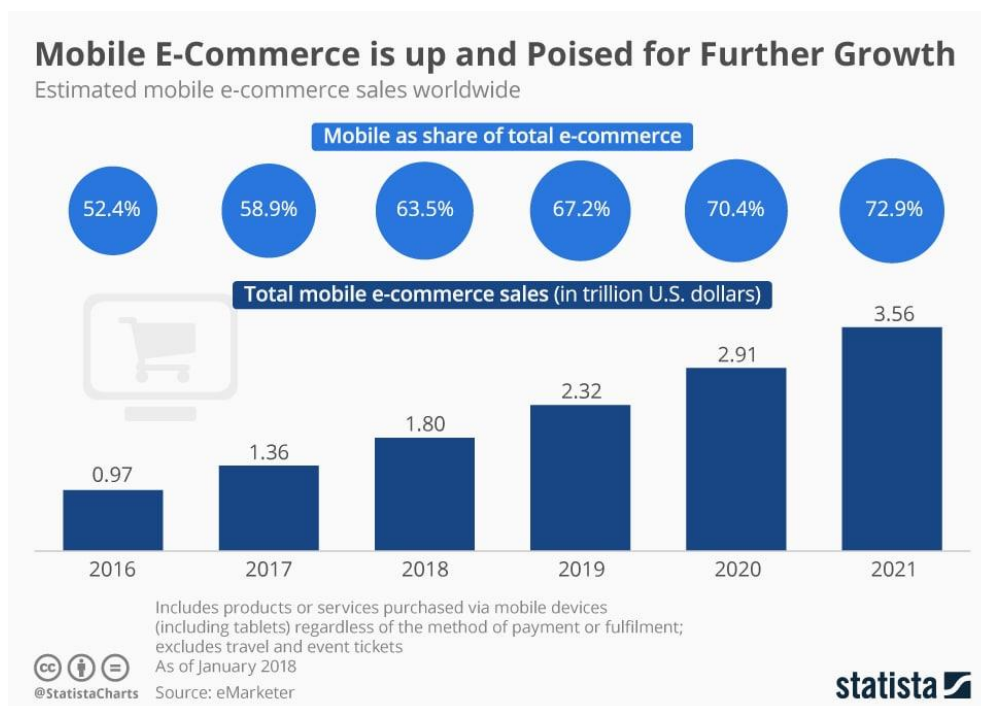


Figure 2.2: Mobile e-commerce is up and poised for further growth (Loesche & Richter, 2018).

To develop an e-commerce application that is secure, scalable, efficient, accurate, proper technological stacks must be implemented to sustain the load and traffics navigated to said application as shown in Figure 2.3. One

of the most important and crucial factors to consider before finalising the development stacks is the budget. Projects are constrained by budgets and every decision made must be in line with achieving the end goal within the project budget. Choosing the right technologies for a development is subjective but it is followed by a detailed analysis and affected by the features provided by the technology itself.

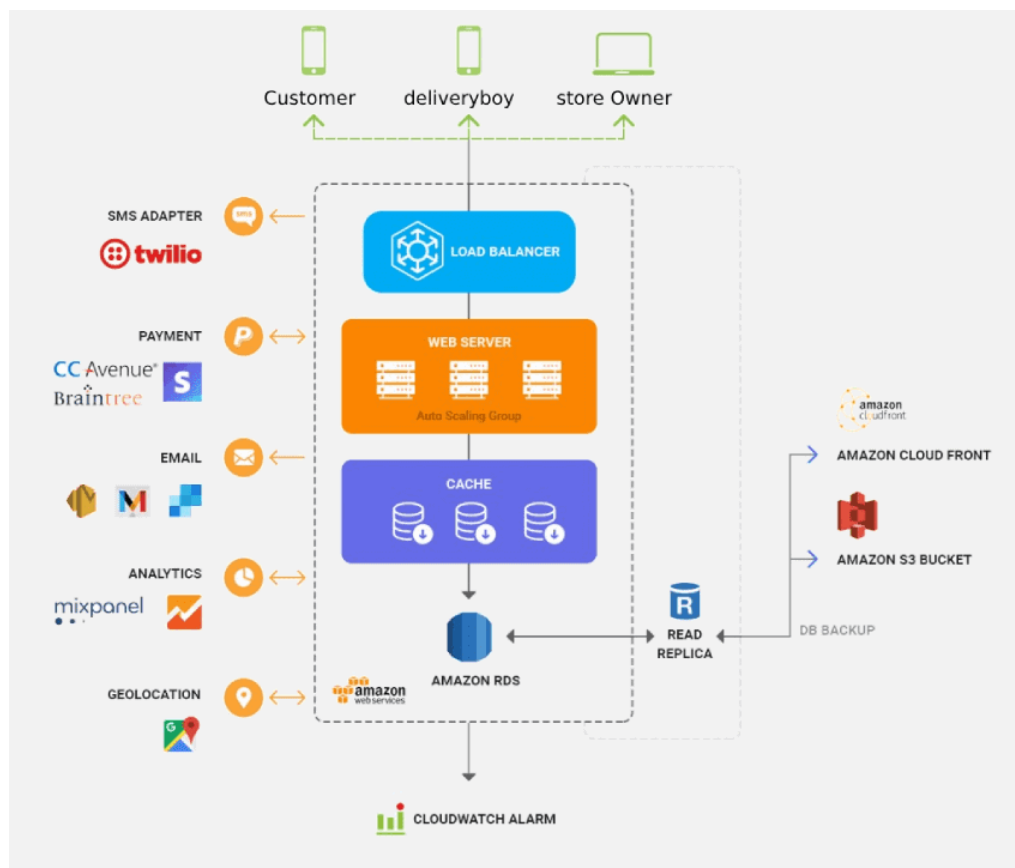


Figure 2.3: Ecommerce Mobile App Development Guide - Cost and features Samsukha (2022).

2.2.2 Transition to M-Commerce

With the rapid growth of e-commerce and the wide availability of various types of e-commerce platforms, the trending industry had seen a shift towards the adoption of m-commerce. M-Commerce, as described by Alvi, S., Laila, U., Khan, K. and Hussainy, S.K (2016), refers to the “transactions being done by wireless handles devices such as mobile phones. The context of m-commerce can be described in various ways with varying definitions as shown in the study m-commerce versus e-commerce (conducted in 2.2.3), but it often refers to the similar aspect which is the transaction of assets over mobile devices. It is also

a shared subset of e-commerce due to certain similarities in features. However, the unique features of m-commerce such as Near Field Communication (NFC) where users can use their mobile devices as a wallet is what makes m-commerce stand out and gain popularity (Omonedo, P. and Bocij, P, 2014). NFC is only one of the many reasons and features offered by m-commerce that had resulted in the steep rise of its domination.

A study conducted by BuildFire, in Figure 2.4, portrays that m-commerce applications continue to outperform mobile web sessions with a conversion rate of 157% higher. According to Statista, in Figure 2.5, the mobile payment volume is at a constant rate of growth of 62% yearly, and to break it down even further, Figure 2.6 shows the breakdown in m-commerce sales between smartphones and tablets respectively. These studies and statistics prove the growth of m-commerce and the focus of companies in developing mobile-commerce applications.

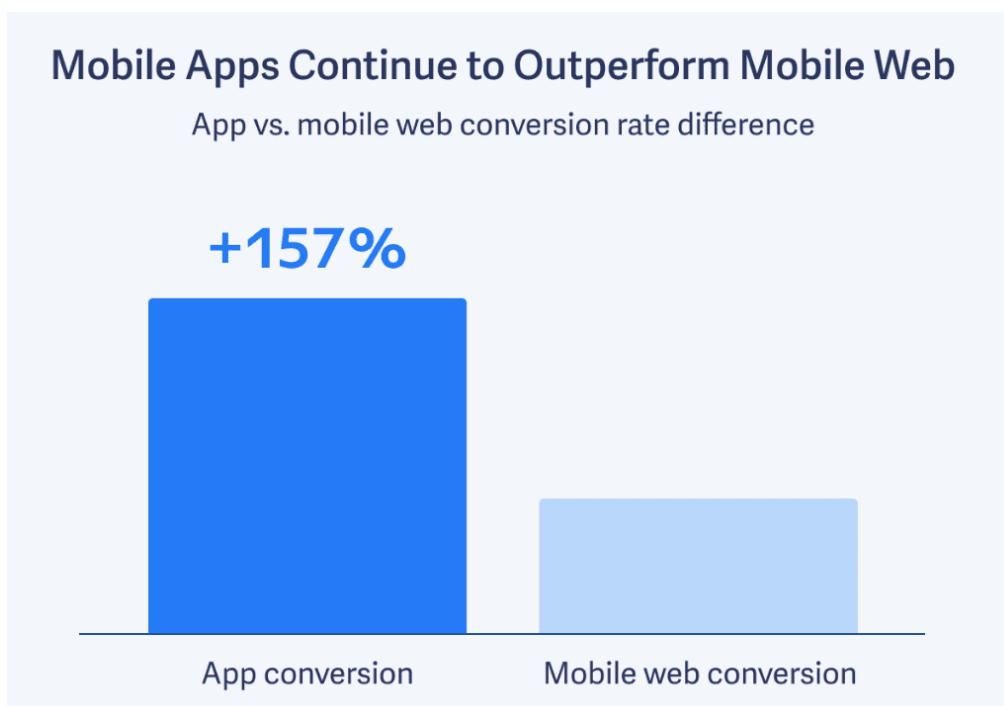


Figure 2.4: Mobile Apps Conversion vs Mobile Web (Blair, I., 2022).

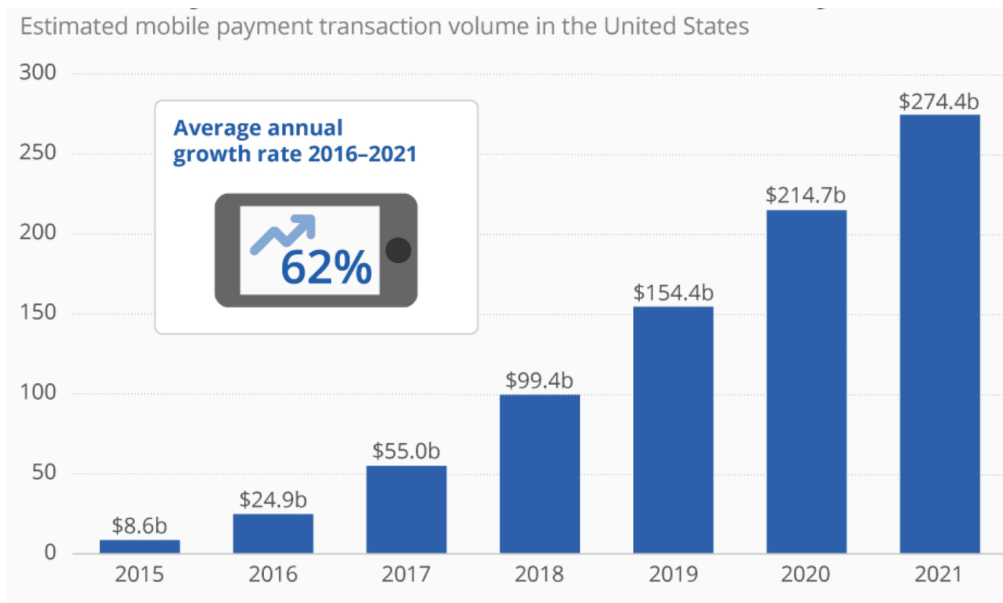


Figure 2.5: Estimated mobile payment transaction volume in the US (Richter, F., 2017).

FORECAST: US M-Commerce Volume

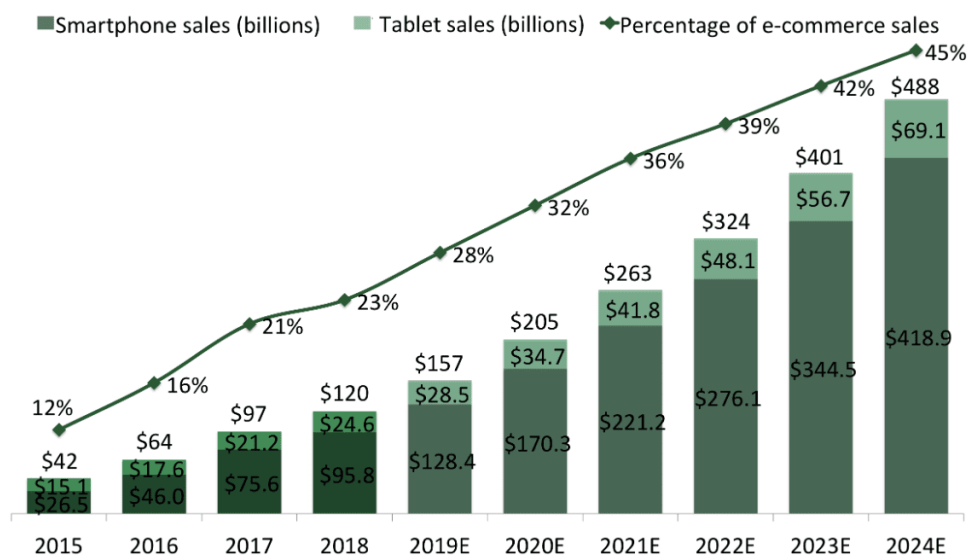


Figure 2.6: US M-Commerce Volume (Blair, I., 2022).

With the analysis of the current m-commerce trends in place, we will now dissect on the reasons for its popularity. Among the reasons why m-commerce has since an increase in growth and usage is due to its convenience and accessibility.

The existence of mobile-commerce application allows users to access various e-commerce services regardless of where they are via their mobile devices. Services ranging from internet banking, online shopping, downloading

media files are often performed through mobile devices. Smartphones these days are constantly being upgraded and optimised to be suited for everyday travels which make them very convenient to be carried around compared to a laptop which must be shut down and can be a burden at times to carry around.

In the context of accessibility, users can perform transactions by just connecting to a WIFI network, which is available almost everywhere today, or using their mobile data. The ease of internet connectivity on a mobile device reduces the redundancy of carrying a laptop with a broadband around. This also means that a significant amount of time can be saved by users to access, browse, and purchase items from m-commerce applications because the need to set up, wait for the laptop to load or plug in a charger to power up the laptop is no longer necessary.

Smartphones these days (which are not flagships) are also a cheaper alternative than purchasing a laptop. On average, a budget smartphone equipped with a 2GB RAM, sufficient capacity for internal and external storage can be acquire from the range of RM 400 – RM 500. Therefore, anyone with a smartphone can have access to a variety of categories of applications such as shopping, banking, social medias, etc. from either the Google Play Store or App Store as most of these applications are offered for free.

It can be concluded that the use of m-commerce applications is easy, convenient, accessible, and time saving. To support this claim even further, Ian Blair from BuildFire has also claimed that about 97% of online shopping carts are abandoned on mobile sites, 68% abandonment on desktop sites and only 20% abandonment on mobile applications (Blair, 2022). This is mainly due to the ability of ‘one-click ordering’ on mobile applications which allows users to purchase products in just a single click. To achieve this, mobile-commerce applications focus on saving customer information (after securely encrypting) such as user’s name, credit card number, credit card expiration, credit card verification number, and more to their respective profiles. By doing so, users will not have to input these details every time there is a transaction to purchase products.

2.2.3 Summary

No	Citation Author (year)	Purpose, rationale, research question	Major findings, contributions	Research methods, sample, variables	Strengths/ limitations	Significance/ Implication for practice/ research	Make note on how this research is linked to other studies reviewed
Theme/ Area of Studied: E-Commerce and Traditional “brick-and-mortar” shopping							
Study 1	(Bakos 2001)	To focus on how the global e-commerce affects retail markets and how they are compared to traditional markets in terms of search costs, patterns of competition, mechanism for price discovery and types of intermediation.	<ul style="list-style-type: none"> - The reduce in search costs for buyers and sellers. This is achieved via the Internet where sellers communicate product and cost information effectively to potential buyers. - Customers are able to benefit from the low pricing competition in digital markets. - Due to retailing business being digitalised, consumers are able to compare price levels and disparities of online markets to traditional retailers. 	This study is an exploratory study to better understand the impact of e-commerce towards traditional retailers.	The limitations of this study is the lack of samples or interviews with consumers/retailers as it is an empirical analysis based on various sources.	To make a thorough analysis and comparison on e-commerce and traditional “brick-and-mortar” shopping.	The decrease in search cost in the retail market may improve or deteriorate social welfare (Wu et al., 2004).

			<p>- Roles of intermediaries will make a shift from being the distributor to providing trust relationships, ensuring the integrity of the market and laying out marketing information to suppliers.</p>				
Study 2	Payaro et al. (2021)	<p>To analyse the online commerce for food groceries and compare the evolution from 2017 to 2019 and calculate the cost of to prepare the expenses and satisfy the order of customers.</p>	<p>- From the sample of research in 2017 and 2019, it can be seen that the number of retailers that have a website without online commerce had dropped by 16 and retailers that have a website with online commerce had increased by 17.</p> <p>- The number of brands that have failed their online commerce transition is 2.</p> <p>- In 2019, 6 brands have adopted the home delivery</p>	<p>This research contains a significant percentage of samples and analysed after two years using the same samples to report the number of grocery retailers have adopted e-commerce.</p> <p>Every Internet site was visited and some managers were</p>	<p>The sample size (99%) represents almost all of Italy's brands of brick and mortar supermarkets and hypermarkets.</p>	<p>This research proposes the reasons why only a limited number of grocery brands sell food over the Internet.</p>	<p>Opinions voiced by consumers indicate that the issue of consumer's trust and acceptance relates to the integrity of an e-commerce business (Nguyen et al., 2000).</p>

			<p>model, 10 have adopted the drive through model, and 15 with both options.</p> <p>- This research indicates that the lack of quality assurance in the e-commerce sector deters some Italians from moving into the Internet for grocery shopping.</p>	interviewed in this research.			
Study 3	Steinfeld et al. (2002)	<p>To explore the relationship between traditional and electronic channels and illustrate particular management strategies that firms developed to achieve synergy and avoid conflicts between their Internet and physical channels.</p>	<p>- One key element of the synergy approach in a national electronics retail chain is the integration between the Internet and existing databases, and other legacy systems.</p> <p>- Physical infrastructures such as warehouses, and distribution systems enhances the value of e-commerce.</p> <p>- All ten cases analysed collectively illustrate efforts by firms to convince existing physical outlets that e-commerce channels worked for their benefit, rather than in direct competition.</p> <p>- To ensure adequate e-commerce capabilities,</p>	<p>This research is an empirical study based on a series ten case studies of the United State firms completed in the late 2000 and early 2001.</p>	<p>Ability to further elaborate on effective click and mortar development was limited due to sampling strategy.</p>	<p>This research was carried out to provide a framework for understanding where the potential payoffs are when integrating physical and e-commerce channels.</p>	<p>The creation of a synergistic platform where consumers, merchants and service providers connect results in the increasing efficiency in their value activities and expanding their market (Wu et al., 2016).</p>

			<p>“several firms formed alliances to fill in important gaps in either their physical or virtual channels. This framework emphasizes the use of synergy strategies to lower costs, differentiate products and services, build trust, and extend into new markets.”</p>				

Theme/ Area of Studied: M-commerce versus E-commerce, and their significance							
Study 4	Omonedo and Bocij (2014)	To identify the attributes that assist in defining e- and m- commerce.	<p>- Setting a perspective on e-commerce, the definition can be anything that involves the transaction of goods or services and not necessarily something that directly involves monetary value.</p> <p>- Setting a perspective on m-commerce, the research suggests that it can be defined as “the conduct of activities that involves content delivery (notification and reporting) and transactions (purchasing, transfers, data entry) on mobile devices capable of gaining access to a network, particularly the internet, which provides direct or indirect commercial benefit”.</p> <p>- On the aspect of power supply, an e-commerce user would have to be close to a power source whereas a m-commerce user can conduct transactions simply by using their phones which relatively</p>	This research is an empirical study with several credible references. No samples were collected or interviews were conducted for this paper.	The strengths of this paper includes the vivid line drawn to separate both e- and m- commerce, brief explanations on their relationship and descriptions on their individual uniqueness.	Finalises that the attributes of mobility, ubiquity and immediacy provide a more vivid and comprehensible template to differentiate e-Commerce from m-Commerce.”	“It can be said that M-Commerce is a subset of E-commerce” Alvi et al. (2016).

			<p>have a longer lasting batter life.</p> <ul style="list-style-type: none"> - M-commerce have several advantages such as conducting business on-the-go, localisation and instant connectivity, Near Field Communication (NFC), etc. - Limitations with m-commerce include the screen size, which can be resolved through e-commerce. 				
Study 5	Alvi et al. (2016)	To help explore how and the reasons m-commerce is gaining continuous fame whilst taking over e-commerce and exploring m-commerce in banking sectors.	<ul style="list-style-type: none"> - On the aspect of usability and convenience, m-commerce takes the cake as users are able to perform a certain task regardless of where they are compared to e-commerce. - Mobile phones require strict controls over security as it is an important aspect in any commerce applications. - Based on a study, many people are now progressing towards m-banking systems due to the accessibility of financial services in a convenient way. 	<p>“Simple random sampling technique has been used to identify the segment whereas customers of four banks are taken as respondents in this study”.</p> <p>The obtained demographic include young adults from 18 years onwards, have access to the Internet, mobile phone and a bank account.</p>	<p>Questionnaires are well validated before conducting analysis. Sample size was 200 respondents which makes this research credible in terms of feedback received.</p> <p>Data collection was only conducted through an online medium such as Google Docs.</p> <p>This research was only limited to the</p>	Summarises the progress and future of m-commerce which depicts the introduction of it has increased the margins of effective commerce revolution.	Categorising m-commerce as an integral subset of e-commerce is significantly becoming less practical (Omonedo and Bocij, 2014).

			- This research indicates that the altitude of understanding consumer's adoption of m-commerce is difficult due to its early stages .	Questionnaires used are based on a 5 point Likert scale.	city Karachi in Pakistan.		
Study 6	Niranjanamurthy et al. (2013)	To study and understand the relationship between e-commerce and m-commerce , discuss their pros and cons, analyse e-commerce sales, identify m-commerce applications and study the security issues in online marketing.	<p>- Several advantages of e-commerce as portrayed in this paper includes no need for physical stores, common availability of coupons and deals, easier to compare prices, variety of choice, etc.</p> <p>- Several disadvantages of e-commerce are the inability to experience products before purchase, credit card fraud, delay in receiving goods, expense and expertise for e-commerce infrastructure.</p> <p>- Advantages of m-commerce are accessibility, convenient, provides a wider reach, competitive pricing, time efficient, etc.</p> <p>- Disadvantages of m-commerce include technological constraints of mobile devices (displays), limited bandwidth, smaller</p>	This research lacks the sample size, description on instruments used and clarification on how the results were obtained.	<p>Among the strengths of this paper is the vast number of advantages and disadvantages of e- and m- commerce respectively.</p> <p>The limitations however is the lack of sample and explanation on how this research was carried out.</p> <p>From my observation and analysis, this research maybe an empirical study.</p>	Provides a brief understanding on the pros and cons of e-/m- commerce and the integrity of their security challenges.	Opinions voiced by consumers indicate that the issue of consumer's trust and acceptance relates to the integrity of an e-commerce business (Nguyen et al., 2000).

			<p>interface makes it harder to learn application UI, etc.</p>				
--	--	--	---	--	--	--	--

- Among the security features that must be present in e- and m- commerce applications are **authentication** to verify user, **encryption** to ensure data is hidden safely, **auditing** to keep records of operations, **integrity** to avoid unauthorised data manipulation and **availability** to prevent data delays.

Theme/ Area of Studied: An overview on native and cross-platform development							
Study 7	Pinto et al. (2018)	To discuss the ideology of cross-platform development , existing theories and technologies, mobile development and concepts.	<p>- Native approach is the implementation of system programming languages such as Java and Swift to build Android and iOS applications respectively. This requires the project to be developed twice, each operating system on its own programming language and framework.</p> <p>- The web approach specifies the traditional approach where HTML, CSS and JavaScript are used. This approach is popular, standardised and relatively simple yet powerful. Several frameworks that facilitate this approach are jQuery, Bootstrap and Angular.</p> <p>- The Hybrid Approach is the modern way of building cross-platform applications where specific programming languages and frameworks that support both Android and iOS operating systems</p>	This is an empirical study where no samples are needed however, thorough research and credible references were cited.	The limitations to this paper is the lack of mobile development approaches and predictions on the technology that would dominate the software industry in coming years.	Comparison made on various types of technologies and further breakdown on the speed, functionalities, difficulties, costs, etc. were provided. Native applications have the overall better performance .	The production of native applications can be expensive thus building cross-platform applications can save time and resources (Grzmil et al., 2017).

			<p>are used. Several of these frameworks include Xamarin, Firebase, React Native, Meteor and Ionic.</p> <p>- In this project, native approach has proven to be faster and covers more functionalities whereas the hybrid approach with ionic framework is less difficult to learn and costs lesser to develop.</p>				
Study 8	Nawrocki et al. (2021)	To analyse the two most used approaches in mobile application development and compare identical applications developed using various tools.	<p>- The first comparison between React Native and Android focused primarily on framework performance. Almost all measurements favoured the native Android framework although the disparity was not significant.</p> <p>- The second comparison was between two cross-platform applications; Flutter and React Native. Disk I/O and FPS were the primary focus. React Native was deemed faster at I/O operations whereas Flutter offered a better developer experience.</p>	<p>This research compares CPU and random-access memory (RAM) usage on native applications and 3 cross-platform applications each built in Xamarin, React Native and Flutter respectively.</p> <p>Several metrics were used as measurements such as GPU frequency, CPU load, memory usage and power consumption.“ The measurements were</p>	<p>This was a thorough research and effective test with proper measurement metrics and tools used to obtain the most accurate result possible.</p> <p>Useful study for developers who are in the software development industry.</p>	<p>Supports that native applications have overall better performance compared to cross platform applications.</p>	<p>“Native approach has proven to be faster” (Pinto et al., 2018).</p>

			<p>- The final results revealed that the native applications had the fastest start time, lowest memory usage and smallest application size. CPU was at a medium usage for Flutter and the Native applications but medium to high for Xamarin and React Native. Flutter provided the best developer experience overall.</p>	<p>conducted with remarkable accuracy and illustrated using diagrams”.</p> <p>An average of 10 measurements yielded the final result for the native application comparison.</p> <p>Android Studio profiler was used for the Android application whereas XCode Instruments were used for the iOS application.</p> <p>All applications were standardised to display the similar UI to ensure a fair test.</p>			
Study 9	Grzmil et al. (2017)	To analyse the performance of mobile applications produced natively and the cross-platform way.	<p>- Computing performance: Native application completed in a faster time compared to Xamarin (hybrid).</p> <p>- Save and read file: Xamarin had a better read time on Android but slower</p>	<p>Analysis was performed to compare execution times of selected tasks.</p> <p>Methods used were numerical calculations, saving</p>	Comparison and analysis were done in a proper manner with best practices to obtain accurate measurements.	Cross-platform application performs certain tasks better than native application and vice versa.	When it comes to the comparison of frameworks for developing mobile applications, results are rather subjective (Nawrocki et al.,

			<p>on iOS. Saving time is vice versa.</p> <p>- Downloading image: Native application took a lesser time but disparity was insignificant.</p> <p>- Location: Native application completed in a faster time than Xamarin.</p>	<p>and reading from a file, network support, determination of location.</p> <p>Constant variables were the file size and same WiFi speed (network support).</p>			<p>2021). This is because different framework and languages perform better at various tasks.</p>
--	--	--	---	---	--	--	---

Table 2.1: Summary of Literature Review (Matrix)

2.3 Rapid Application Development Methodology

For this project, the chosen approach is the Rapid Application Development (RAD) methodology as it facilitates the requirements given by Murchand. This methodology is an agile project management strategy popular in software development as it focuses on achieving the end goal such as a complete working application, as quickly as possible. The rapid development pace is made possible by minimising the planning stage and shifting a large focus to maximise the prototype development (LucidContentTeam, 2018). By doing so, project managers and stakeholders and result in better communication and progress measurement on evolving issues (LucidContentTeam, 2018). The following are several phases that can be derived from this methodology.

2.3.1 Requirements Planning

Similar to every other methodology, the requirement elicitation phase is inevitable and must be performed to gather sufficient requirements and a deeper understanding of the end product. However, although it is carried out in RAD, this phase is much condensed in comparison to other methodologies.

During this stage, stakeholders discuss to determine the project requirements such as project goals, expectations, timelines, and estimated budget to complete said project (Chien, 2020). On top of that, ways to overcome potential issues that may arise during the development phase will also be discussed by all parties. This stage is critical as it ensures all parties are on the same page before the development begins, to avoid misunderstanding, and prevention of costly mistakes. One unique feature to note about this methodology is its flexibility to adapt to changes made in the requirements at any time during the development stage. As soon as the requirements and scope of the project has been outlined, the development team can move on to the next stage.

2.3.2 Prototyping

This phase is the core of this methodology as it consists of various iterations until the client's needs are met and the final product is ready. The goal of this stage is to rapidly produce working prototypes with different user designs and the clients will be able to test the prototypes at each stage.

The process of prototyping is as follows; development team designs and builds a prototype for their client, client tests and provides feedback based on user-interface, functionalities, etc., development team resolves all issues and bugs based on the feedback received before providing their client with a new prototype. This cycle continues until the client is satisfied with the design and functionalities of the application. It is crucial that both the developer and client work hand in hand during this stage to identify the pros and cons of the prototype at each iteration as it avoids any miscommunication and time-consuming changes/repairs.

2.3.3 Rapid Construction

With an approved and satisfied prototype in the last stage, developers will now focus on converting said prototype into a fully functional model. Since majority of the bugs and problems were already addressed in the prototype iterations, developers will be able to construct the final model more quickly than they could if they followed a traditional project management approach (LucidContentTeam, 2018).

However, compared to previous prototypes, various tests such as unit testing, integration testing, and system testing will be performed to ensure its stability, peak performance, and client satisfaction. This stage will require client and end users' involvement as developers collect thorough feedback to continually improve the product in all aspects (Chien, 2020). This also provides clients with the opportunity to suggest amendments to resolve any issues that has been identified throughout this phase. At the end of this stage, depending on the feedback received, the development team can either return to prototyping or move onto the final stage.

2.3.4 Implementation / Cutover

This is the final stage of the RAD methodology where the developed application is deployed into a live environment. Prior to that, developers ensure that the application is well maintainable and stable in all targeted systems.

Development teams will document the completed product and continue to look for bugs in the application before confidently handing the product over to their client.

2.4 Similar Existing Application

As previously mentioned, the effect of pandemic had caused a rise in the number of e-commerce applications available in the market. Two widely used m-commerce/mobile commerce applications have been analysed and studied to identify key factors and unique features that would make them stand out among competitors.

Screenshots of the applications and their architectures have been included to support the analysis. The two applications chosen for this study are Grab Superapp and FoodPanda.

2.4.1 Grab Superapp

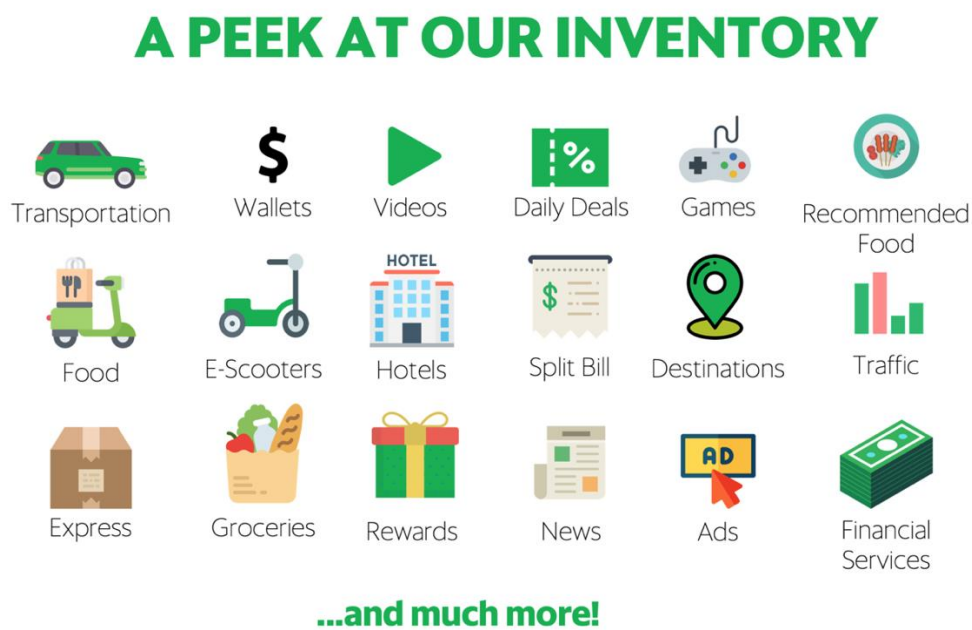


Figure 2.7: Grab Features Inventory (Bolilia et al., 2019).

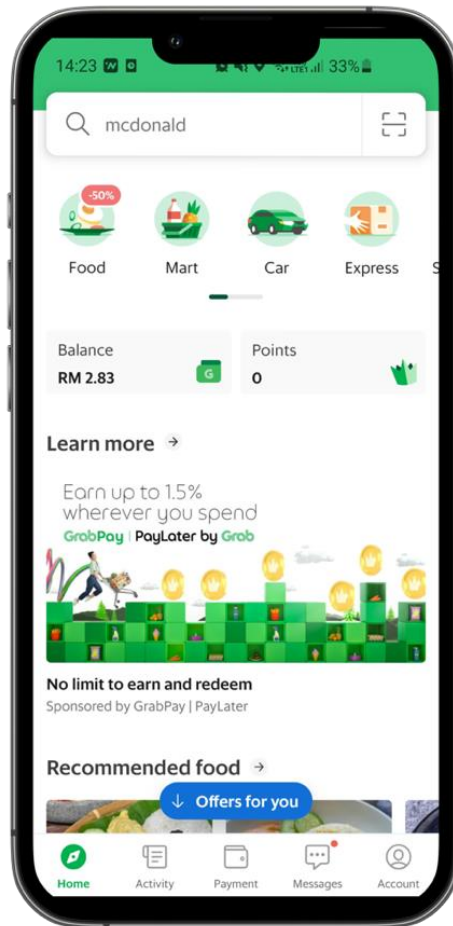


Figure 2.8: Overview of Grab Superapp home interface.

2.4.1.1 Grab Superapp – Key Features

a. Multiple services

Grab focuses on providing various types of services on its platform making it a unique superapp compared to its competitors. Several of these offered services include GrabFood which makes allows users to order food and drinks from a variety of chain restaurants, GrabMart which allow users to shop for produce and have them delivered to their doorstep, GarbCar the founding feature of this application that provides users transportation services for a certain price, GrabExpress which allow users to deliver items from one place to another and many more.

b. User Recommendations

Due to the vast number of services offered by Grab, the team had faced with a problem about inferring preferred contents to specific types of

consumers (Bolilia et al, 2019). To resolve this issue, a recommendation engine was built to suggest type of content to specific types of users depending on how they have used the platform. Figure 2.6 illustrates the recommendation engine model that was built by Grab.

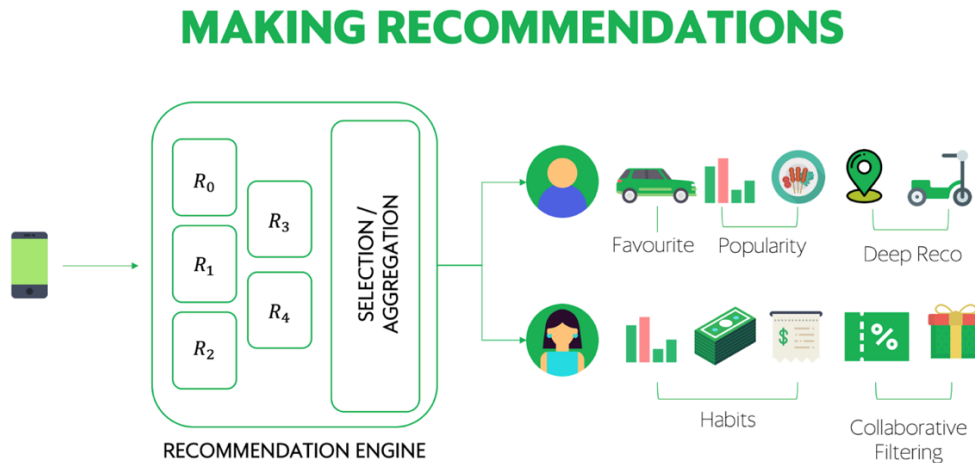


Figure 2.9: Recommended engine model (Bolilia et al., (2019).

c. GrabAds

GrabAds is a service that provides businesses with the opportunity to advertise their products to the Grab's consumer base (McCallum et al., 2022). Grab had foreseen this opportunity as the demand in its service gained a steep incline during the pandemic. As a result, a dedicated ad server was built after initially experimenting with an off-the-shelf solution to gain the public's reception towards this feature. The decision to build an inhouse solution was based on seamlessly integration, overall business growth and personalisation handling using machine learning (McCallum et al., 2022). Figure 2.7 portrays the ad server architecture built by Grab.

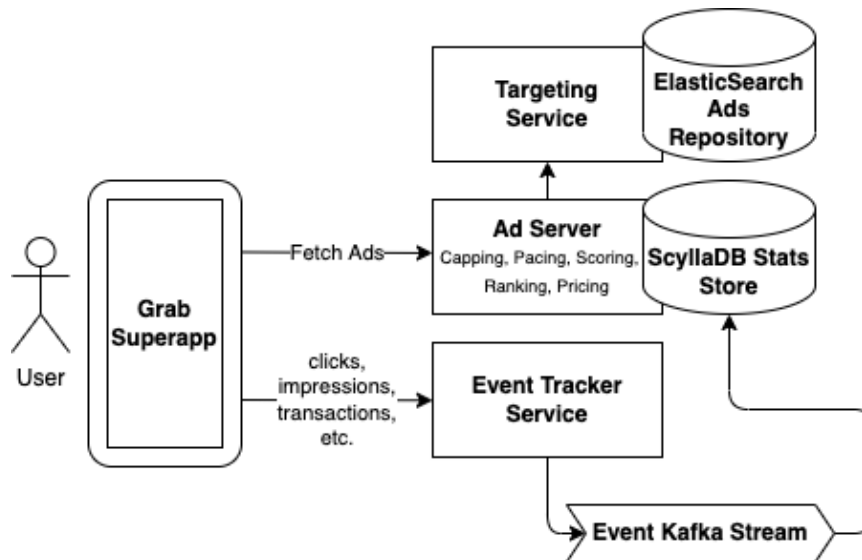


Figure 2.10: Ad-Serve architecture by Grab (McCallum et al., 2022).

2.4.1.2 Grab Superapp – Evaluation

Grab Superapp can be deemed as the leading e-commerce application in Southeast Asia due to its attractive features and variety of functionalities. With efficient in-house servers and complex machine learning algorithms to provide the best recommendations to users, they stand out in many ways. They are constantly experimenting and implementing new features as well as optimising the application and their servers to maximise performance.

There are not many applications with the levels of offerings such as Grab in the current market, and this has proven to be the selling point to consumers. It allows consumers to order e-hailing services which was the starting point when this application was developed. Now, consumers can transport items from one place to another, order meals from available restaurants, buy groceries and have them delivered, pay using Grab's dedicated e-wallet and more.

However, several issues remain a concern such as the absence of local convenience stores on the platform, and lack of variety in the retail business sector such as pet stores, and shops that provide their services such as plumbing and installations.

In conclusion, if a customer would like to purchase a cage for their pet or air conditioning services from an electrical store, Grab Superapp would not be their ideal option.

2.4.2 FoodPanda

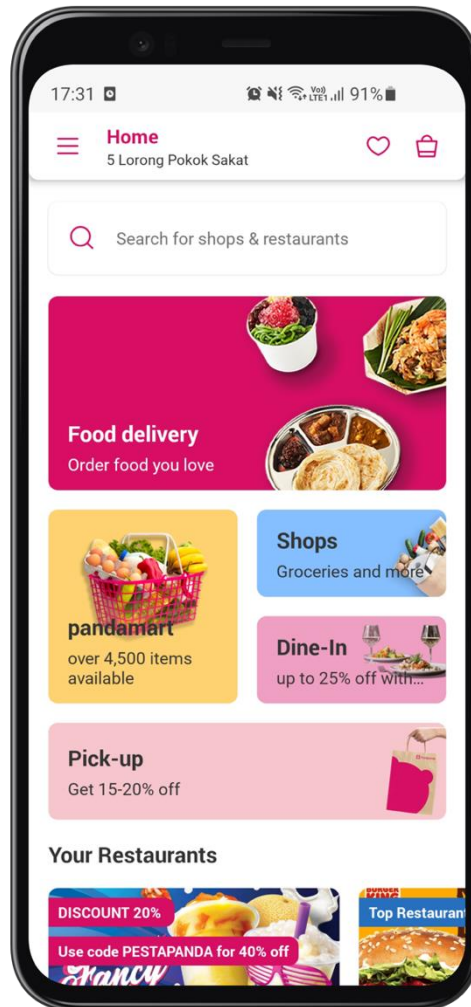


Figure 2.11: Overview of FoodPanda home interface.

2.4.2.1 FoodPanda – Key Features

a. Shops

Shops is the one-stop shopping feature developed by FoodPanda to provide consumers with the ability to shop for not only groceries but various items. Items categories of shops on this section ranges from groceries, medications, electronics, and clothing. It is a unique feature as not many applications have such a broad clientele. Figure 2.9 shows the different categories of stores that are offered in the application.

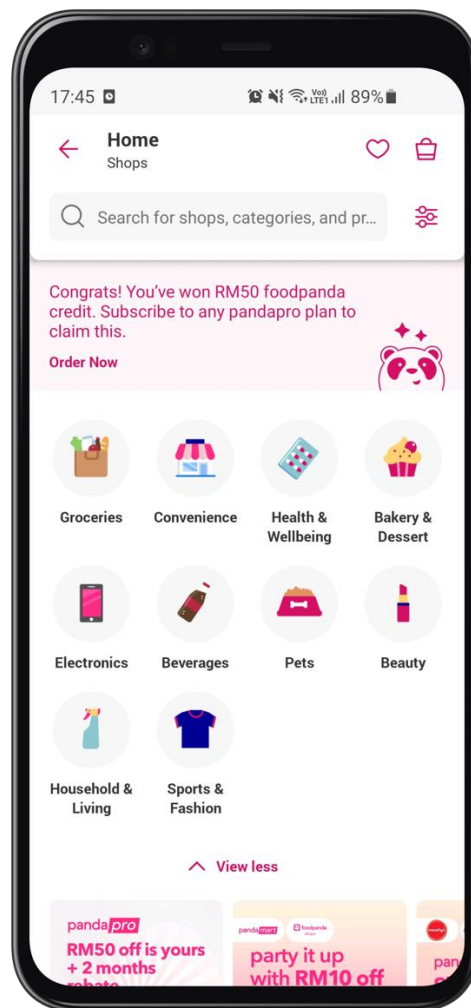


Figure 2.12: FoodPanda shops interface with a variety of shopping categories.

b. Cutting-edge technologies

FoodPanda has been experimenting with the usage of drones to deliver food items in a radius up to 3km in the city state (Subramaniam, 2021). With predictions to implement this technology in the coming years, they would lead the e-commerce delivery by solving on-demand needs and providing logistic companies with the opportunity to send necessities to communities that are not within its serviceable range (Subramaniam, 2021).

2.4.2.2 FoodPanda – Evaluation

FoodPanda has proven to be one of the most competitive applications in the current e-commerce market. Although the application does not offer as many services to the extent of a superapp, they have constantly managed to keep up

with the current technological trend by providing relevant features and experimenting with new technologies such as drone deliveries. They have managed to nail the fundamental features and aspects of an e-commerce application such as effective payment gateway, user-merchant integration, geolocation tracking, push notification, and more.

Besides that, the application offers a variety of local convenience stores depending on the user's location which makes it the go-to application for users who would like to purchase products other than groceries. However, even with the existence of local merchants, they still lack in the aspect of selling services instead of goods.

In conclusion, despite its variety in merchants and products, FoodPanda still lacks the features and ability to purchase professional services.

2.5 Comparison between e-commerce applications

The table below compares several key features and functionalities between the e-commerce applications: Grab, FoodPanda and The Quadry.

Applications Criteria	Grab	FoodPanda	The Quadry
Merchant delivery radius	7 km – 10 km	Within 5 km	5 km – 7 km
Payment methods	5 (Cash, Grab Pay, Credit/Debit card, PayPal, FPX)	4 (Cash, Credit/Debit card, FPX, Boost)	4 (Cash, Credit/Debit card, GrabPay, FPX)
Invoice/Receipt automation	Available	Available	Available
Two-factor authentication	Available	Not Available	Available
Push notifications via smartphones	Available	Available	Available
Biometrics support	Available	Not Available	Not Available
Automatic geolocation tracking	Available	Available	Available

Automated help centre	Available	Not Available	Available
-----------------------	-----------	---------------	-----------

Table 2.2: Comparison between e-commerce applications

2.6 Conclusion

To conclude, the existing e-commerce applications that have been analysed and compared has attractive features that make each unique among the rest. Most security and effective features can be reproduced whereas the flaws that exist within these applications can be addressed using the proposed solutions.

Some of the features and functionalities from compared e-commerce applications that have been found suitable for this project are:

a. Invoice/Receipt automation

As an e-commerce application, it is important to address payments in the most secure and efficient manner. Implementing an automation process for invoice or receipt generation upon payment will save a lot of time and makes the payment process efficient as users will be able to receive their invoice as soon as the payment is complete.

b. Two-factor authentication

Any application that deals with the transfer of user's assets must be well protected and all data must be encrypted before storage. By implementing this feature, users will have a higher level of trust in using The Quadry.

c. Automated help centre

Chatbots are very useful in the mobile development industry as they help users solve some of their concerns and issues. Certainly not all issues can be solved but integrating a chatbot into an application reduces weight off the customer service representatives and allows them to focus on bigger and more serious issues faced by the users.

CHAPTER 3

METHODOLOGY AND WORK PLAN

3.1 Introduction

In this chapter, implementation of the chosen system development methodology for this project will be briefly outlined by emphasising on the phases and stages of the development.

Various development instruments and software were identified. This includes development tools ranging from environments, programming languages, communication medium, and development frameworks and libraries.

Project plan had also been developed by constructing work-breakdown structure and translating it into Gantt Chart.

3.2 System Development Methodology

The system development methodology is often referred to as structured processes embedded into a project before, during, and after its development process. “The goal is to provide a systematic approach to software development” (Nikolaieva, n.d.). As this project is a collaborative development with Maurchand, the Rapid Application Development (RAD) methodology will be adopted as it is one of the project requirements as requested by Maurchand.

RAD was chosen as the development methodology for this project due to its ability to produce quick successful product releases within a stipulated timeline through multiple iterations of prototyping, testing, and implementation until the customer (in this case Maurchand) is happy. In each iteration, several features will be designed and thoroughly tested for the client and end users’ satisfaction. It also addresses bug fixes and adjustments according to client and end users’ experience in upcoming iterations. The iterations will come to an end when all stakeholders are satisfied and implementation such as documentation, adaptability in target systems and maintainability is ensured by the developers. Any refinements of the project will trigger a new requirements elicitation phase in which all stakeholders will specify exactly the type of feature(s) that will need to be added before commencing a new prototyping cycle.

3.2.1 Project Phases

This project consists of 4 phases which are preliminary phase, requirement elicitation phase, development phase, and deployment phase. The actions performed in each phase are broken down and specified in their respective sections below.

3.2.1.1 Actions in Preliminary Phase

Preliminary phase in the context of a project is the discussion and planning stage that takes place at the beginning of the project. Similarly, the preliminary phase of this project was carried out to gain an overall understanding on the concepts of e-commerce, m-commerce, and its trends leading towards the adoption of m-commerce over e-commerce. A general overview on these concepts will help in identifying the current workflow of developing an e-commerce application.

Based on the current workflow in the literature review above, problem statements were derived in this phase to identify the shortcomings and problems faced by current e-commerce applications. By identifying the problems, research was conducted on how to resolve these issues, and solutions were proposed which includes its workflow and key technologies that was used. The proposed project approach was specified by identifying a suitable development methodology after communicating with Murchand. Finally, the project scope was outlined in this phase which includes the modules covered and scope not covered in this project.

3.2.1.2 Actions in Requirements Elicitation Phase

Literature review was conducted to obtain a deep comprehension on e-commerce, its workflow, trends, and architecture. This was achieved by studying the features and abilities of current existing e-commerce applications and its trend in growth for the past several years. The transition to m-commerce and its comparison to e-commerce was analysed and described as well.

Several studies on the aspects of e-commerce and traditional “brick-and-mortar” shopping, m-commerce versus e-commerce and their significance, and an overview of native and cross-platform development was carried out and summarised in the literature review. This was conducted to understand the need

of m-commerce applications and backed by its growth that was catalysed by the pandemic that started back in 2020.

Literature review on the overall process of the Rapid Application Development (RAD) methodology was conducted to justify and understand the methodology of choice for this project. The chosen methodology for this project was briefly described with its processes and cycles.

Next, similar existing e-commerce applications were analysed by selecting two widely used mobile commerce applications. Key features of these chosen applications were identified and outlined with attached screenshots of said features. Thorough analysis and assessment were conducted for both applications to identify their strengths and weaknesses respectively. Doing so, provided an overview on the features that can be garnered and implemented in this project replicating either the similar version or altering to an optimised version. Finally, with the analysis done, a comparison between the two e-commerce applications and The Quadry was performed to compare their key features and available functionalities. A conclusion was drawn at the end to identify the strengths and weaknesses of each application and key features that were selected to be implemented in The Quadry was included with their respective reasonings.

The development tools used in this project were determined and specified along with the main communication medium used to communicate with Maurchand. Moving on, the requirements of this project were specified by identifying functional and non-functional requirements. These requirements were also added, amended, and adjusted after a detailed discussion with Maurchand during this phase. All requirements gathered was then be modelled with a use case diagram with its descriptions to explain the derived use cases.

3.2.1.3 Actions in Development Phase

The development phase of this project follows the RAD methodology where processes are broken down into cycles, or also referred to as iterations. Each iteration is based on the feedback received from clients about the previous iteration, therefore iterations may consist of performance optimisations, new features, designs, and bug fixes.

a. **First Iteration**

The first iteration for the development of this project was primarily focused on developing the first prototype of this application which covers all critical contents in the application such as viewing shops, products sold in said shops with their detailed description, drawer, and bottom-bar navigations with their respective contents, etc.

Upon the completion of all fundamental and critical pages/features covering both the frontend and backend, the prototype was thoroughly tested. This testing involved Maurchand and end-users' to receive feedbacks on the features to be kept and removed. The feedbacks received was noted for the next iteration.

b. **Second Iteration**

With feedbacks obtained in the first iterations, the received feedbacks were analysed to identify and select features to add, update or remove and amendments were implemented accordingly. Unfinished features and pages were completed and bugs that have been found were fixed during this iteration, producing a fully functioning prototype, and fulfilling all requirements.

As soon as all requirements have been addressed in this iteration, the prototype was once again thoroughly tested to reveal more bugs that need to be fixed before proceeding to the feedback gathering.

In feedback gathering, suggestions and feedbacks were once again be gathered from clients and end users' to identify functionalities to be improved.

c. **Third Iteration**

This iteration repeats implementations from the second iteration but more narrowed towards addressing the negative feedbacks, because all functional and non-functional requirements have already been met at the end of the previous iteration. However, if there were to be changes in the requirements, the prototype will be updated accordingly to fulfil those requirements.

These iterations were continually repeated, and concerns were addressed until all stakeholders are satisfied with the outcome.

d. **Rapid Construction**

When all stakeholders are satisfied with the outcome, rapid construction phase was commenced, this involved building the actual application by converting the prototype into a fully functional application ready to be deployed. This is a fast development process due to bugs and changes being addressed in the previous iteration.

Once the construction was completed, the application underwent thorough testing such as integration testing, system testing, and user-acceptance testing before moving forward with the deployment phase.

3.2.1.4 Actions in Deployment Phase

The developed application with approvals from all stakeholders was now moved to production under the deployment phase. In this phase, The Quadry will be deployed onto Google Play Console and App Store to be publicly accessed nationwide. The documentations and test reports on all conducted tests of this project was then compiled.

3.2.2 Development Tools

Emulators, cloud services and programming tools are required for the development of mobile applications. More so, considering, the number of estimated users of this application introduces the need for more advanced technological stacks to help support the scalability of the application. With a large repository of open-source tools and frameworks made available for mobile application developers such as Dart programming language and frameworks such as Flutter and its supported libraries, the development of any type of mobile application can be made possible. The technologies listed below are part of the requirements from Maurchand as these tools not only facilitate with the development of The Quadry, but it is also the best suited technologies for their business operations.

a. Dart

Dart is a programming language that is rapidly growing in terms of popularity, contributors, and repository. Its simplicity, modern day functionalities and reliability due to being backed by a tech giant (Google) are among the catalyst for its growth. Dart had been selected as the programming language to develop The Quadry due to its ability to support the development of cross-platform applications. It was also selected as the primary programming language for the development of this project because it is the main language that is supported by the Flutter framework. Since Flutter was written in the Dart language, performance will be more superior when using Dart compared to other programming languages as it goes hand in hand.

b. Flutter

Flutter is an open-source front end software development kit developed by Google using the Dart language. Flutter is used for the development of cross-platform applications across many different operating systems such as Android, iOS, Windows, MacOS, Linux, Web platform, and Fuchsia. It had been selected as the framework for the development of this project due to its ability to support multiple platforms from a single codebase and the simplicity it offers which is suitable for a project that is being developed using Rapid Application Development (RAD). Other frameworks such as React Native and Xamarin are relatively complex which repels against the RAD methodology.

c. Google Maps

Google Maps is a web mapping platform that offers satellite imagery, aerial photography, street maps, route planning and more. This service was integrated into The Quadry to provide users with the ease of locating and navigating to a certain merchant through its Geolocation API. Doing so, presents users with the ability to directly obtain the location of a merchant, thus avoiding the trouble of manually searching the merchant's location.

d. Stripe

Stripe is a financial and Software-as-a-Service (SaaS) company that primarily offers payment processing services. Stripe had been chosen as the payment service handler for this project due to its scalability and reliability. Certain e-commerce giants such as Grab, Shopify, Jaya Grocer, DahMakan, etc. uses Stripe payment services due to its abundance of features such as invoicing, frontend UI components, and more. The widely used SaaS company also supports various types of programming languages such as iOS, Android, React Native, etc.

e. Firebase

Firebase is an application development platform built by Google to assist developers in building applications or games of any genre. It had been chosen as the Backend-as-a-Service (BaaS) provider for this project due to the various products offered. Among the offered products that were used in this project are Firebase Authentication to handle end-to-end identity logics such as user login and registrations, Cloud Storage to store images of products offered by merchants, Cloud Messaging to send push notifications to the end users, Cloud Functions to automate certain operational and automated processes of the application such as invoice generation and email upon product purchase and welcome email upon user registration, Cloud Firestore a NoSQL database to store a variety of information, Remote Config to monitor the application version code and notify users when an update is available and Dynamic Links to provide end users with the feature to share any of the merchant profiles or products in the application as it also survives installation process in the event where a user does not have the application installed on their mobile device.

f. GitHub

GitHub is a popular internet hosting service for software development and an online version control application using Git. This system was used in this project to manage source code versions and as the project's repository. It had been chosen as the version control system to be used

due to its simplicity to manage and publish code changes, collaborate with other developers, and track the development progress.

3.3 Communication Tools

a. Email

Since this is an outsourced project, the communication tool that was used to be in contact with Maurchand is email. It had been chosen as it is Maurchand's preferred communication channel in the event of handling confidential documents.

b. Zoom

Zoom is an online video conferencing platform that was used to communicate directly with the Directors of Maurchand for requirements elicitations, feedback gathering, discussions, and feature planning. This platform was also used in case of changes to the requirements or major updates such as partnership collaborations, etc.

3.4 Project Plan

The project plan was constructed by producing activities in the form of Work Breakdown Structure (WBS) and have been scheduled accordingly in a Gantt Chart.

3.4.1 Work Breakdown Structure (WBS)

The Work Breakdown Structure in the context of project management refers to a deliverable-oriented breakdown of a project into smaller components. This allows for project activities and progresses to be measured easily and accurately to estimate the approximate completion period. The goal of implementing WBS in projects is to decompose large projects into smaller activities as it results in the project being more manageable.

Top-down approach was used to produce the Work Breakdown Structure for this project as it focuses on identifying and completing major tasks within the project. These major approaches were then broken down into smaller tasks and for each completed small tasks results in completion of the parent

activity. Figure 3.1 shows the WBS that was used in this project based on the project phases specified in 3.2.1.

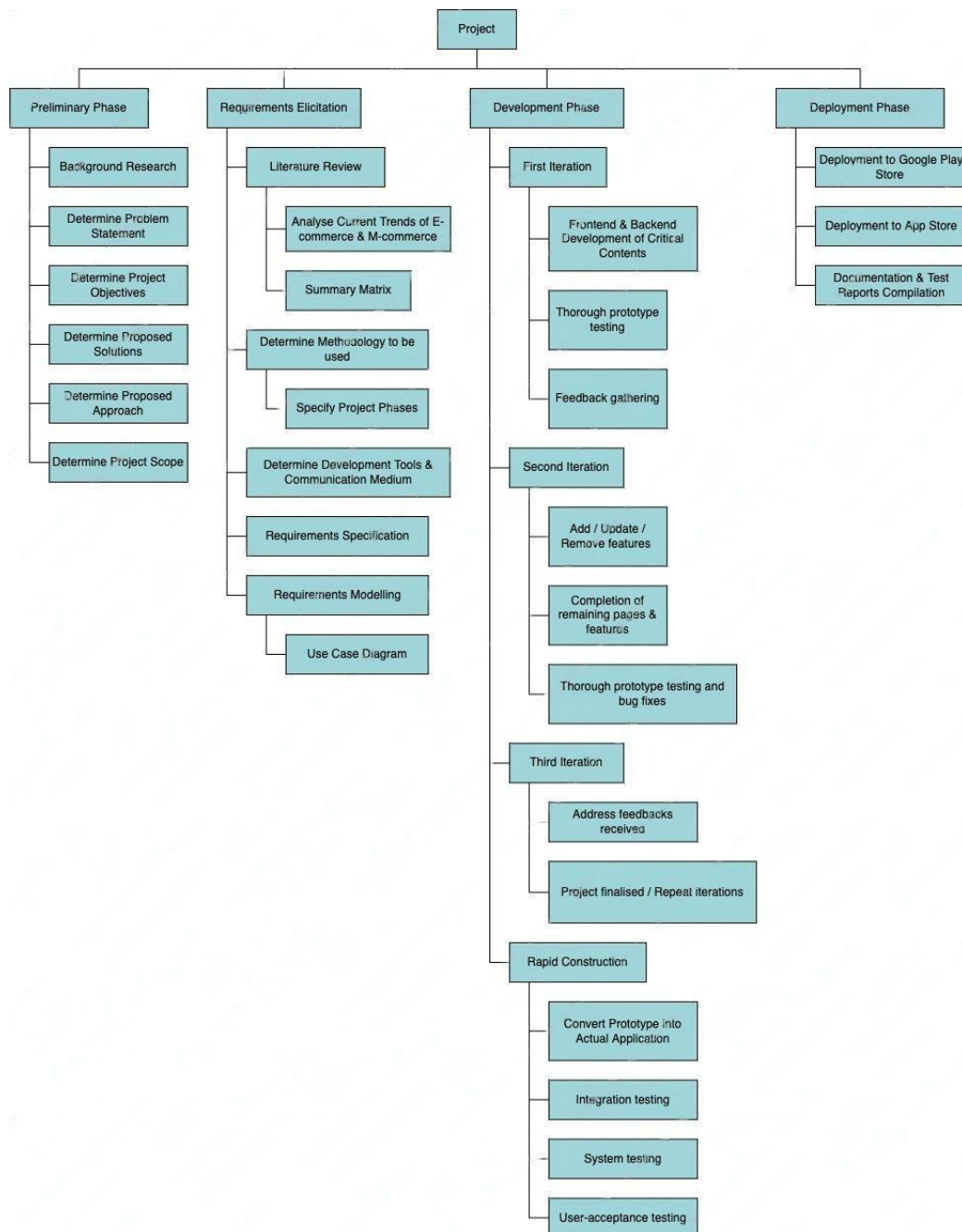


Figure 3.1: Work Breakdown Structure for The Quadry.

3.4.2 Gantt Chart

The implementation of Gantt Chart provides an overview on the order of which activities should be completed, their schedule such as the estimated start date, end date, and duration of project. Each main activity in the Work Breakdown

Structure (WBS) is broken down into smaller tasks and provided with estimated start date and end date to obtain the duration of each task. The Gantt Chart that had been tabulated for this project is shown in Table 3.1 below.

Tasks	Duration (Days)	Start Date	End Date
Preliminary phase	12	13 June 2022	24 June 2022
Background Research	3	13 June 2022	15 June 2022
Determine Problem Statement	2	16 June 2022	17 June 2022
Determine Project Objectives	2	18 June 2022	19 June 2022
Determine Proposed Solutions	2	20 June 2022	21 June 2022
Determined Proposed Approach	2	22 June 2022	23 June 2022
Determine Project Scope	1	24 June 2022	24 June 2022
Requirement Elicitation	30	25 June 2022	9 Sep 2022
[Literature Review]	6	25 June 2022	30 June 2022
<i>Current Trends Analysis</i>	4	25 June 2022	28 June 2022
<i>Summary Matrix</i>	2	29 June 2022	30 June 2022
[Determine Methodology]	3	1 July 2022	3 July 2022
<i>Specify Project Phases</i>	3	1 July 2022	3 July 2022
Determine Development & Communication Tools	3	4 July 2022	6 July 2022
Requirements Specification	5	7 July 2022	11 July 2022
[Requirements Modelling]	-	-	-
<i>Use Case Diagram & Descriptions</i>	4	12 July 2022	15 July 2022
Development Phase	61	16 Jan 2023	17 Mar 2023
[First Iteration]	17	16 Jan 2023	1 Feb 2023

<i>Frontend & Backend Development of Critical Contents</i>	12	16 Jan 2023	27 Jan 2023
<i>Testing (Unit Testing)</i>	3	28 Jan 2023	30 Jan 2023
<i>Feedback Gathering</i>	2	31 Jan 2023	1 Feb 2023
[Second Iteration]	14	2 Feb 2023	15 Feb 2023
<i>Add / Update / Remove Features</i>	3	2 Feb 2023	4 Feb 2023
<i>Complete Remaining Pages & Features</i>	7	5 Feb 2023	11 Feb 2023
<i>Testing (Unit Testing) & Bug Fixes</i>	4	12 Feb 2023	15 Feb 2023
[Third Iteration]	7	16 Feb 2023	22 Feb 2023
<i>Address feedbacks received</i>	3	16 Feb 2023	18 Feb 2023
<i>Testing (Unit Testing) & Bug Fixes</i>	4	19 Feb 2023	22 Feb 2023
[Rapid Construction]	23	23 Feb 2023	17 Mar 2023
<i>Prototype Conversion to Actual App</i>	7	23 Feb 2023	1 Mar 2023
<i>Integration Testing</i>	2	2 Mar 2023	3 Mar 2023
<i>System Testing</i>	7	4 Mar 2023	10 Mar 2023
<i>User-Acceptance Testing</i>	7	11 Mar 2023	17 Mar 2023
Deployment Phase	9	18 Mar 2023	26 Mar 2023
<i>Deployment – Google Play Store</i>	3	18 Mar 2023	20 Mar 2023
<i>Deployment – App Store</i>	3	21 Mar 2023	23 Mar 2023
<i>Documentation & Test Report Compilation</i>	3	24 Mar 2023	26 Mar 2023

Table 3.1: Project Timeline for The Quadry

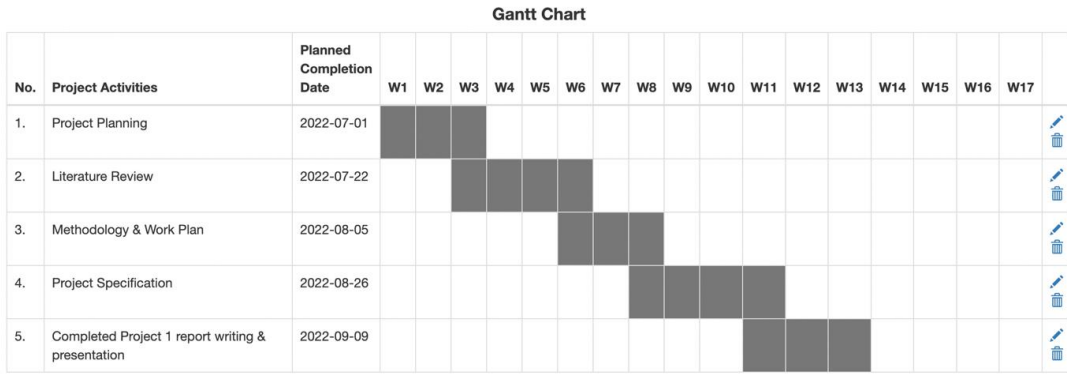


Figure 3.2: Gantt Chart for Project 1 (until Requirement Elicitation phase).

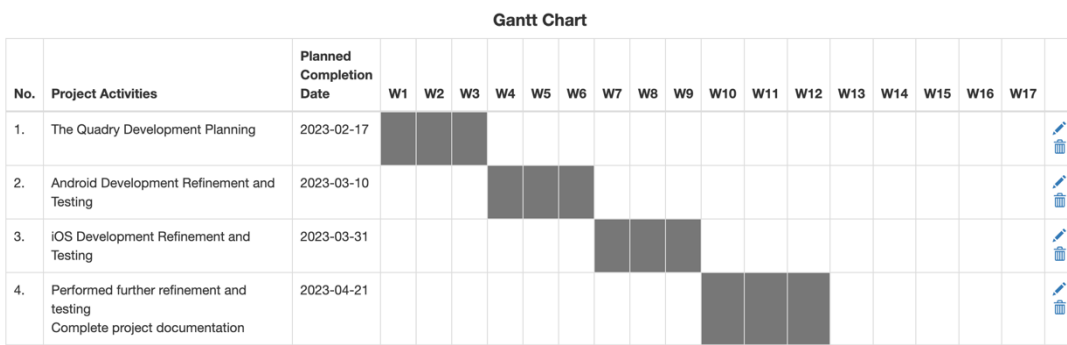


Figure 3.3: Gantt Chart for Project 2

CHAPTER 4

PRELIMINARY PROJECT SPECIFICATIONS

4.1 Introduction

This chapter specifies the project initial requirements covering both functional and non-functional requirements.

The requirements are then modelled with use case diagrams and use case descriptions have been developed in respect to the respective items in the use case diagram.

Screenshots of the prototype had also been enclosed in this chapter describing the overall flow of the application, how and why they meet the requirements.

4.2 Functional Requirements

- a. The application shall allow users to register into the system via two-factor authentication.
- b. The application shall allow users to sign into the system using the users' email and password.
- c. The application shall allow users to sign out of the application.
- d. The application shall allow users to view the users' account information such as email, mobile number, and registered addresses.
- e. The application shall allow users to view their current orders, completed orders, and past orders.
- f. The application shall allow users to favourite any products or merchants that the users favour.
- g. The application shall send and allow users to view notifications regarding status of purchased products, and any promotions.
- h. The application shall allow users to view the terms and conditions of using the application.
- i. The application shall allow users to post reviews on merchant(s) upon completing their purchase with said merchant(s).

- j. The application shall allow users to view information about the merchants such as overall ratings, operational hours, operation status, and merchant location.
- k. The application shall allow users to share any merchant page or items displayed in the application.
- l. The application shall allow users to add various products from different merchants into the users' shopping carts.
- m. The application shall allow users to purchase any products from any merchants via cash, debit/credit card, GrabPay, and FPX.

4.3 Non-Functional Requirements

- a. Usability – The user interface must be simple and straight-forward to be used.
- b. Development – The application will be built using Dart, Flutter, Firebase.
- c. Performance – The application must respond within 5 seconds.
- d. Security – The user shall sign into the application using their email and password.
- e. Space – A combined storage of 300MB is required for application installation.

4.4 Use Case

Main functionalities from the requirements specification described in [4.2](#) and [4.3](#) had been translated into the following use case diagram. Figure 4.1 shows the use case diagram for this project.



Figure 4.1: Use Case Diagram.

4.5 Use Case Description

a. Use Case Description – Register for account

Use Case Name: Register for account	ID: 1	Importance Level: High
Primary Actor: User	Use Case Type: Detailed, Essential	
Stakeholders and Interests: User – Wants to register for an account to use the application functionalities		
Brief Description: This use case describes how the user will register for an account using the application.		
Trigger: The user wants to purchase products or use the application.		
Relationships: Association: - Include: - Extend: - Generalization: -		
Normal Flow of Events: <ol style="list-style-type: none"> 1. The user selects the create an account button. 2. The user inputs personal details such as name, email, password & mobile number. <u>Proceed to S1.</u> 		
SubFlows: S1. 2.1 The application validates if user email and mobile number is correct. S1. 2.2 The application sends a verification email to the user’s email for verification. S1. 2.3 The application sends an OTP code to the user’s mobile device for verification. S1. 2.4 The application checks if the user exists in the current database. S1. 2.5 The application registers user and inserts user information into the database.		
Alternate/Exceptional Flows: <ol style="list-style-type: none"> 1. If the user email or mobile number is not valid, the application will prompt the user to re-enter the email or mobile number. 		

b. Use Case Description – Sign into account

Use Case Name: Sign into account	ID: 2	Importance Level: High
Primary Actor: User	Use Case Type: Detailed, Essential	
Stakeholders and Interests: User – Wants to sign into their account to use the application functionalities		
Brief Description: This use case describes how the user will sign into their account using the application.		
Trigger: The user wants to purchase products or use the application.		
Relationships: Association: - Include: - Extend: - Generalization: -		
Normal Flow of Events: <ol style="list-style-type: none"> 1. The user inputs credentials such as email & password. 2. The user selects the login button. <u>Proceed to S1.</u> 3. The application authenticates user into the application. 		
SubFlows: S1. 2.1 The application validates if user email is correct and registered in the database.		
Alternate/Exceptional Flows: <ol style="list-style-type: none"> 1. If the user credentials are incorrect, the application prompts user re-enter email & password after displaying error message. 		

c. Use Case Description – Sign out of account

Use Case Name: Sign out of account	ID: 3	Importance Level: High
Primary Actor: User	Use Case Type: Detailed, Essential	
Stakeholders and Interests: User – Wants to sign out of their account.		
Brief Description: This use case describes how the user will sign out of their account using the application.		
Trigger: The user wants to switch accounts or no longer use the application.		
Relationships: Association: - Include: - Extend: - Generalization: -		
Normal Flow of Events: <ol style="list-style-type: none"> 1. The user selects the logout button. 2. The application logs user out of the application. 		
SubFlows:		
Alternate/Exceptional Flows:		

d. Use Case Description – Share merchant page or product

Use Case Name: Share merchant page or product	ID: 4	Importance Level: High
Primary Actor: User	Use Case Type: Detailed, Essential	
Stakeholders and Interests: User – Wants to share details about a merchant or a product.		
Brief Description: This use case describes how the user will share information about any merchant or products sold using the application.		
Trigger: The user wants to share the location of a merchant, or an item sold by a particular merchant.		
Relationships: Association: User Include: Generate survivable dynamic link Extend: - Generalization: -		
Normal Flow of Events: 1. The user log into their account. <u>Proceed to S1.</u> 2. The user selects a merchant or product. 3. The user selects the share button on the top right of the item/merchant page.		
SubFlows: S1. 1.1 The application validates if user email is correct and registered in the database.		
Alternate/Exceptional Flows: 1. If the user credentials are incorrect, the application prompts user re-enter email & password after displaying error message.		

e. Use Case Description – Generate survivable dynamic link

Use Case Name: Generate survivable dynamic link	ID: 5	Importance Level: High
Primary Actor: User	Use Case Type: Detailed, Essential	
<p>Stakeholders and Interests:</p> <p>User – The dynamic link is generated specifically for the merchant/product that is wished to be shared.</p>		
<p>Brief Description:</p> <p>This use case describes how the system will generate sharable dynamic links about any merchant or products from the application.</p>		
<p>Trigger:</p> <p>The user performs share operation by selecting the share button to generate a link that can be shared.</p>		
<p>Relationships:</p> <p>Association: -</p> <p>Include: Generate survivable dynamic link</p> <p>Extend: -</p> <p>Generalization: -</p>		
<p>Normal Flow of Events:</p> <ol style="list-style-type: none"> 1. The user performs sharing feature by selecting the share icon button. 2. The application communicates with the backend API to generate a specific dynamic link for the product/merchant to be shared. 		
SubFlows:		
Alternate/Exceptional Flows:		

f. Use Case Description – View current order status

Use Case Name: View current order status	ID: 6	Importance Level: High
Primary Actor: User	Use Case Type: Detailed, Essential	
Stakeholders and Interests: User – Views current order status after successful payment.		
Brief Description: This use case describes how a user can view details about their order status upon successful payment.		
Trigger: The user successfully purchases any product from any merchant in the application.		
Relationships: Association: - Include: - Extend: - Generalization: -		
Normal Flow of Events: <ol style="list-style-type: none"> 1. The user log into their account. <u>Proceed to S1.</u> 2. The user selects a product to purchase. 3. The user checkout with item(s) in shopping cart and continues with payment. <u>Proceed to S2.</u> 4. The user selects ‘orders’ button in the drawer navigation to view more details. 		
SubFlows: S1. 1.1 The application validates if user email is correct and registered in the database. S2. 2.1 The application validates if payment information given by user is valid.		
Alternate/Exceptional Flows: <ol style="list-style-type: none"> 1. If the user credentials are incorrect during authentication, the application prompts user re-enter email & password after displaying error message. 2. If the payment information / transaction is rejected, the application prompts user to try again after displaying the error message. 		

g. Use Case Description – View available products for each merchant

Use Case Name: View available products for each merchant	ID: 7	Importance Level: High
Primary Actor: User	Use Case Type: Detailed, Essential	
Stakeholders and Interests: User – Views all available products offered by each merchant registered in the application.		
Brief Description: This use case describes how a user can view all available products offered by various merchant in the application.		
Trigger: The user wants to browse / purchase items from a merchant.		
Relationships: Association: User Include: - Extend: Purchase products Generalization: -		
Normal Flow of Events: 1. The user log into their account. <u>Proceed to S1.</u> 2. The user selects a merchant. 3. The user can view / browse the available products.		
SubFlows: S1. 1.1 The application validates if user email is correct and registered in the database.		
Alternate/Exceptional Flows: 1. If the user credentials are incorrect during authentication, the application prompts user re-enter email & password after displaying error message.		

h. Use Case Description – Send payment authorisation request

Use Case Name: Send payment authorisation request	ID: 8	Importance Level: High
Primary Actor: User	Use Case Type: Detailed, Essential	
<p>Stakeholders and Interests:</p> <p>User – The authorisation request is sent by the application to the user’s service provider depending on the chosen payment method.</p>		
<p>Brief Description:</p> <p>This use case describes how the application authorises payments made by the user.</p>		
<p>Trigger:</p> <p>The user purchased item(s) using the application.</p>		
<p>Relationships:</p> <p>Association: User</p> <p>Include: Purchase products</p> <p>Extend: -</p> <p>Generalization: -</p>		
<p>Normal Flow of Events:</p> <ol style="list-style-type: none"> 1. The user log into their account. <u>Proceed to S1.</u> 2. The user selects a merchant. 3. The user checkout with item(s) in shopping cart and makes payment. <u>Proceed to S2.</u> 4. The application redirects user to the dedicated payment service provider. <u>Proceed to S3.</u> 5. The application displays the response message accordingly. 		
<p>SubFlows:</p> <p>S1. 1.1 The application validates if user email is correct and registered in the database.</p> <p>S2. 3.1 The application validates if payment information by user is valid.</p> <p>S3. 4.1 The application waits for response by service provider to identify if payment was successful.</p>		
<p>Alternate/Exceptional Flows:</p> <ol style="list-style-type: none"> 1. If the user credentials are incorrect during authentication, the application prompts user re-enter email & password after displaying error message. 		

i. Use Case Description – Purchase product

Use Case Name: Purchase product	ID: 9	Importance Level: High
Primary Actor: User	Use Case Type: Detailed, Essential	
Stakeholders and Interests: User – Purchases products from the application.		
Brief Description: This use case describes how a user can purchase products from merchants using the application.		
Trigger: The user wants to purchase items using the application.		
Relationships: Association: User Include: Purchase products Extend: View available products for each merchant Generalization: -		
Normal Flow of Events: <ol style="list-style-type: none"> 1. The user log into their account. <u>Proceed to S1.</u> 2. The user selects a merchant. 3. The user checkout with item(s) in shopping cart and makes payment. <u>Proceed to S2.</u> 		
SubFlows: S1. 1.1 The application validates if user email is correct and registered in the database. S2. 3.1 The application validates if payment information by user is valid.		
Alternate/Exceptional Flows: <ol style="list-style-type: none"> 1. If the user credentials are incorrect during authentication, the application prompts user re-enter email & password after displaying error message. 2. If the payment information is invalid, the application prompts user to try again after displaying the error message. 		

4.6 Screenshots with Descriptions

Screenshots depicting the potential interface of this application are attached below along with their respective descriptions, detailing how they fulfil the requirements of this project.

Register page

Below are the screenshots of the application where users will be able to perform account registration in order to access the application functionalities. Figure 4.2 & 4.3 shows the account registration form with required information from the user. Next, Figure 4.4 shows a two-factor authentication where a verification code has been sent to the user's mobile device and is required for user to successfully sign up an account. Once completed, the application will require email verification as displayed in Figure 4.5. A verification email will be sent to the user's email and once clicked, the user will be redirected to authorisation site as shown in Figure 4.6.

Both email and mobile number authentication is required for users to access the application. This fulfils the no 1. functional requirement described in [4.2](#) which is to allow users to register into the system via two-factor authentication.

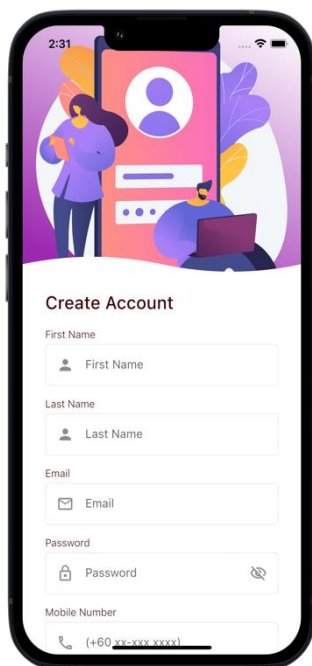


Figure 4.2: Register page (1)

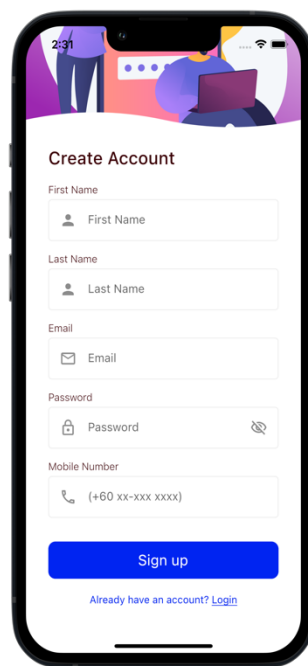


Figure 4.3: Register page (2)

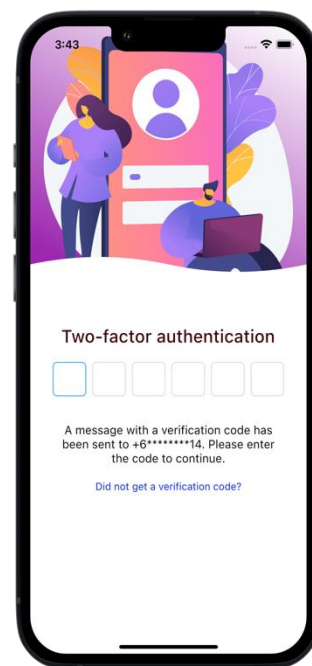


Figure 4.4: Mobile number verification page

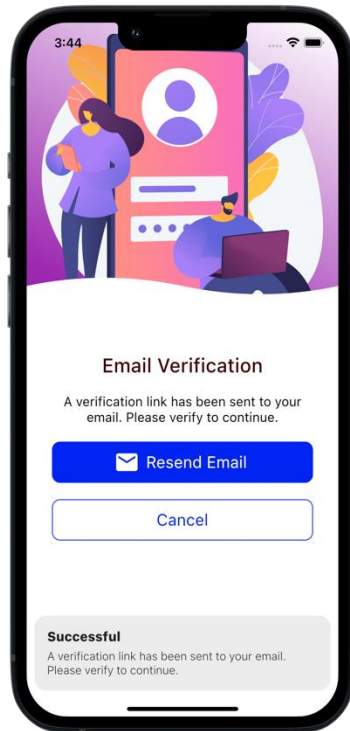


Figure 4.5: Email verification page

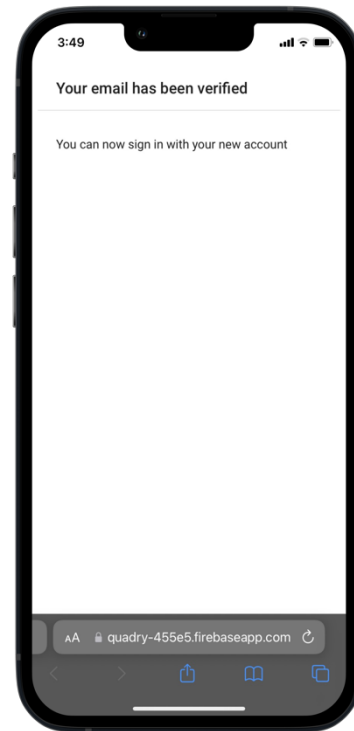


Figure 4.6: Email verified page after clicking verification link

Login page

Figure 4.7 shows the login page of the application which requires two inputs from the users: the registered email and respective password. This fulfils the no 2. functional requirement described in [4.2](#).

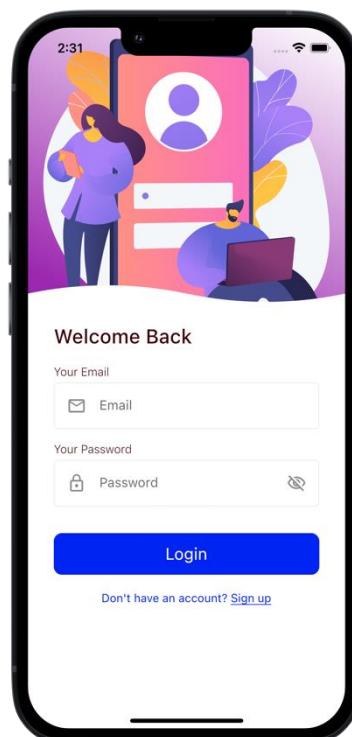


Figure 4.7: Login page

Homepage

Figures 4.8 and 4.9 shows the homepage of the application where different types of shop categories are displayed to the user. This design corresponds to the business model of The Quadry which is to offer different types of merchant categories/services.

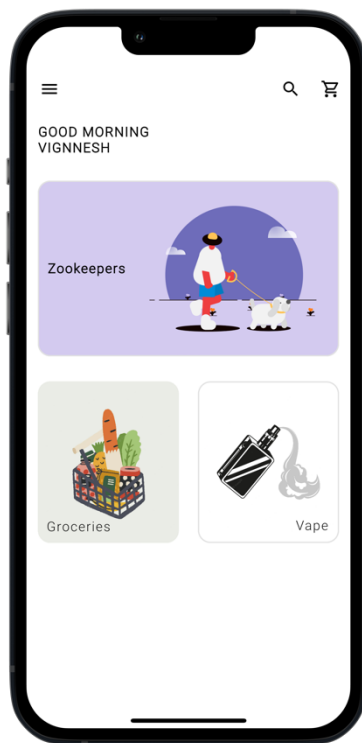


Figure 4.8: Homepage

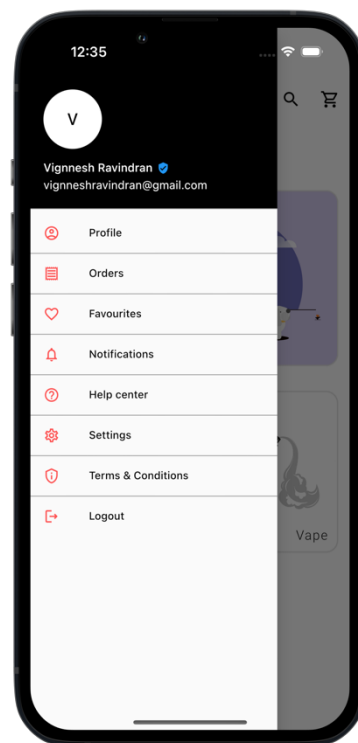


Figure 4.9: Homepage Drawer

Selected shop type page

Figures 4.10 to 4.12 shows the homepage of the application designed to contain the necessary functions stated in [4.2](#). With the implementation of this design, almost all functional requirements can be met.

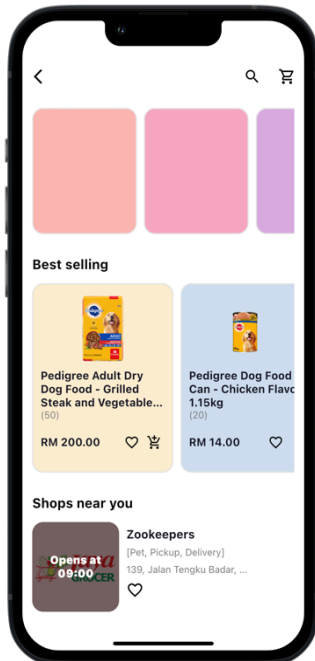


Figure 4.10: Selected Shop Category page (Zookeepers)

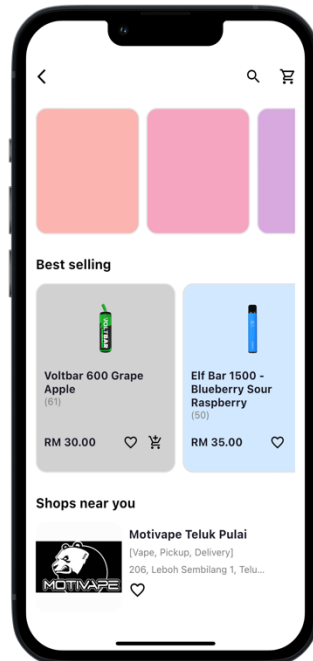


Figure 4.11: Selected Shop Category page (Vape)

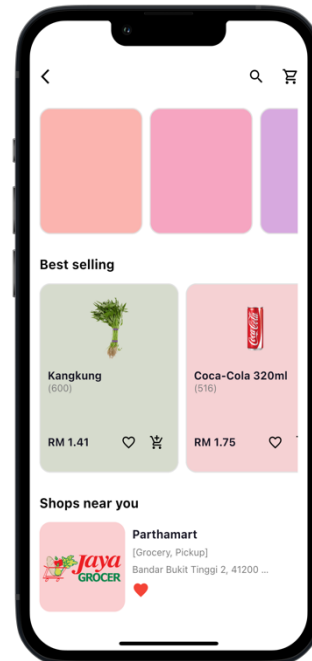


Figure 4.12: Selected Shop Category page (Grocery)

Merchant page

Figure 4.13 reveals the main page of merchants with a top navigation widget that includes a reviews section (Figure 4.14) with comments about the merchant from other users and an about section (Figure 4.15) with brief descriptions about the merchant such as operation hours, location, and more. The implementation of this design meets several functional requirements stated in 4.2 such as number 10, 11, and 12.

Figure 4.16 shows the dynamic link that has been generated and printed onto the console when the user clicks the share icon button at the top-right corner of the merchant page. Upon successful link generation, the application opens the native sharing options for the user to select the sharing destination, ex. WhatsApp, Email, Notes, etc.

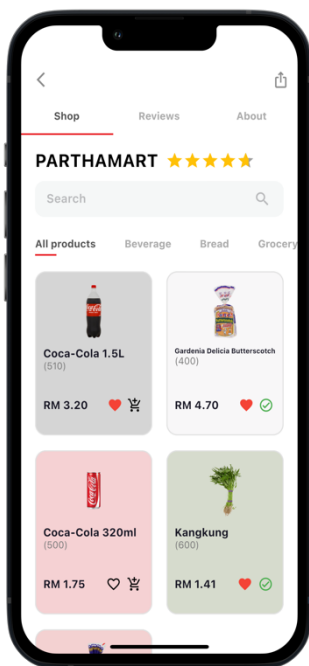


Figure 4.13: Main shop page

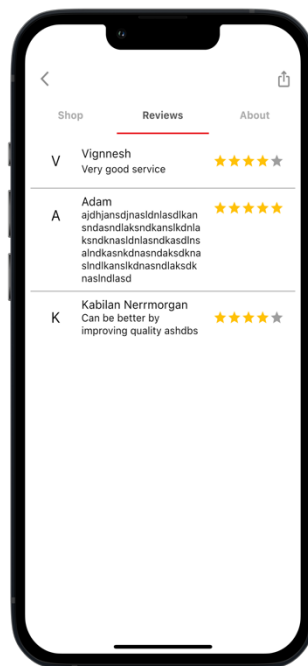


Figure 4.14: Reviews page

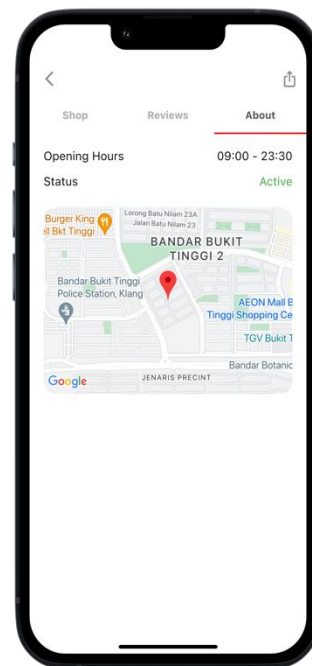


Figure 4.15: About page

flutter: <https://thequadry.com/sQXj>

Figure 4.16: Generated dynamic link for merchant

Product page

Figures 4.17, 4.18, and 4.19 displays the product page when user clicks a specific product. The background colour (chosen based on the most dominant colour of the product picture), product description, and price changes respectively. Dynamic link that has been generated is shown in Figure 4.20 and the native sharing options opened by the application is shown in Figure 4.21.

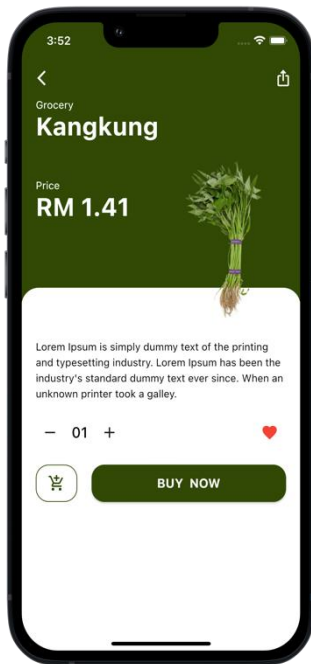


Figure 4.17: Product page sample 1

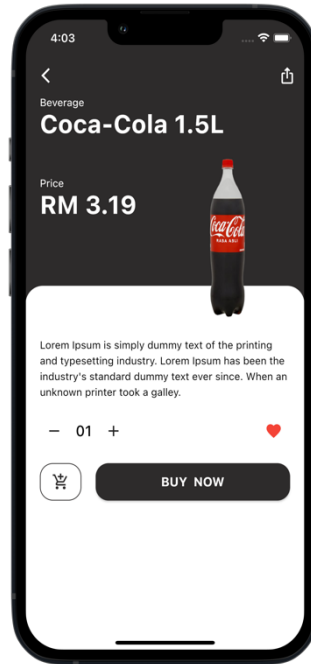


Figure 4.18: Product page sample 2

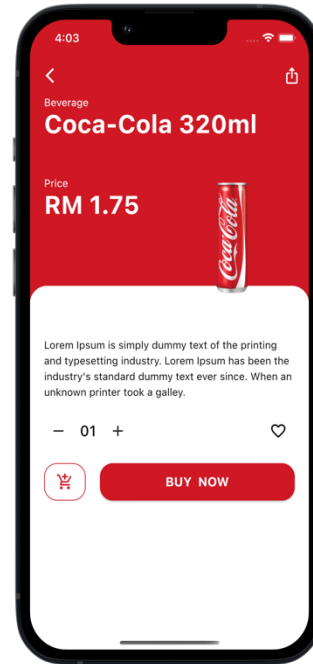


Figure 4.19: Product page sample 3

flutter: <https://thequadry.com/HNyL>

Figure 4.20: Generated dynamic link for product

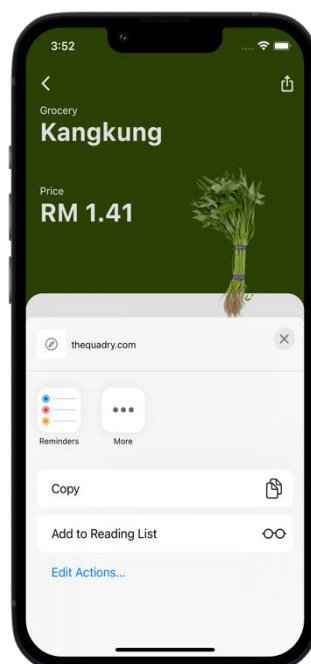


Figure 4.21: iOS native sharing options

Favourites page

Figures 4.22 and 4.23 portrays the favourites page where users can view their favourited merchants and products. The 'product' and 'shops' button allows users to dynamically switch the application to display the chosen option respectively.

Users are also provided with the favourite button (heart icon) to unlike any items which updates the state of the application in real-time. The implementation of this page checks functional requirement number 6 stated in [4.2](#).

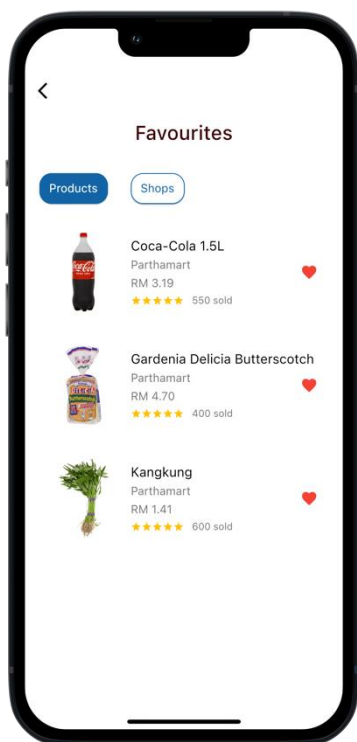


Figure 4.22: Favourites page (Products - Default)

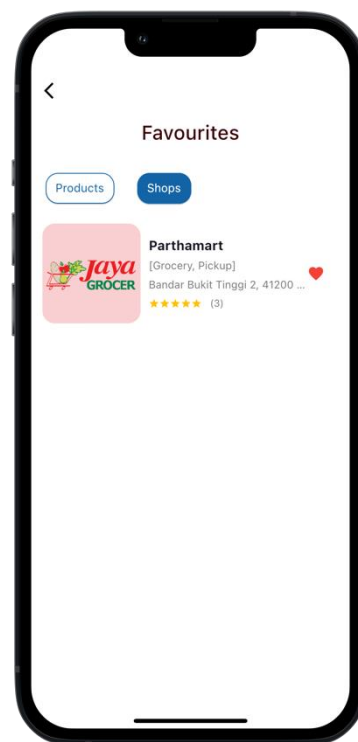


Figure 4.23: Favourites page (Shops)

Cart page

Figure 4.24 unveils the cart page when the user's shopping cart is empty whereas Figure 4.25 shows the cart page when the user shopping cart is not empty. The cumulative number of products, products from each merchant and total amount to be paid (based on quantity) is dynamically updated when the shopping cart is not empty. The implementation of this page fulfils functional requirement number 13 stated in 4.2.



Figure 4.24: Cart page (Empty)

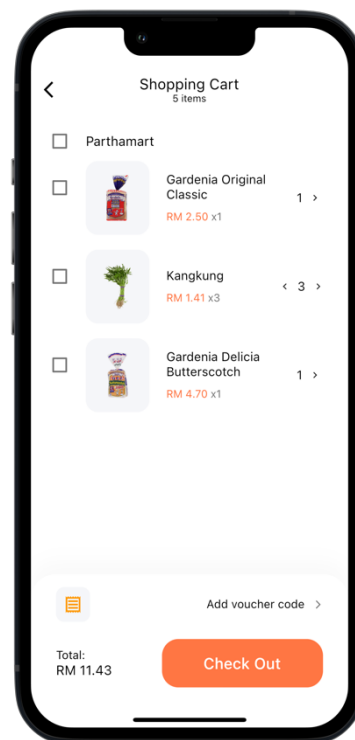


Figure 4.25: Cart page (Filled)

User profile page

Figure 4.26 shows the user profile page with all user information that was obtained during the registration process except user address. The user address data will be required to be provided by user before any purchase which will then be added to the database. Figures 4.27 – 4.30 shows the respective verification pages when the user clicks the email, password, mobile number, or address containers to either change their data or add new address. The implementations of these pages meet functional requirement number 4 stated in [4.2](#).

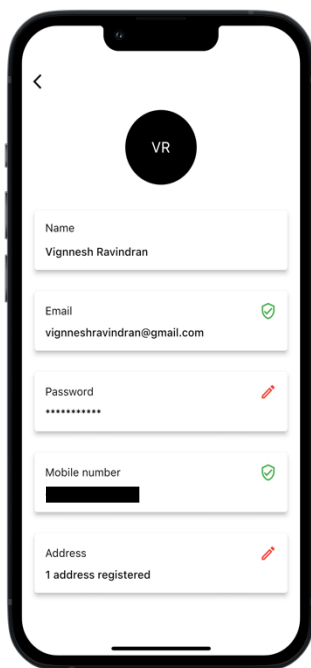


Figure 4.26: User profile page

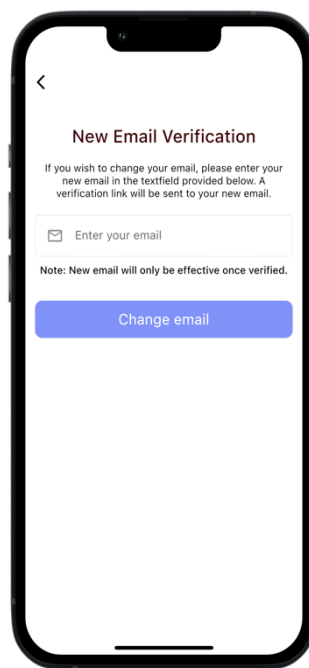


Figure 4.27: Email page

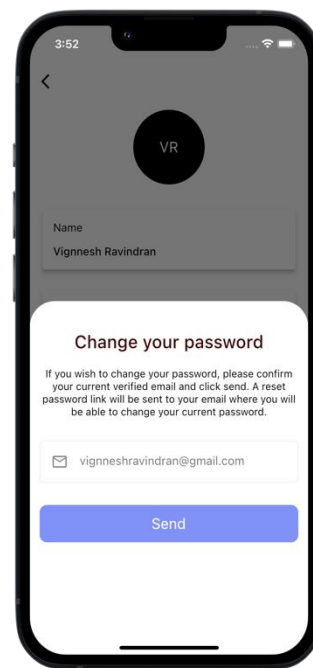


Figure 4.28: Password page

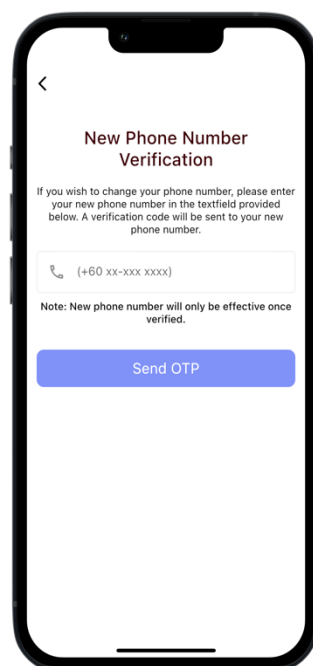


Figure 4.29: Mobile number page



Figure 4.30: Address page

User address selection page

Figure 4.31 shows the user new address selection page with required fields such as their full name, mobile number, and address for delivery purposes. Figure 4.32 shows the address selection feature that was implemented by integrating Google Maps APIs into the application to ease user address selection process. Figure 4.33 shows the delivery notice that alerts users on the terms for standard / express deliveries.

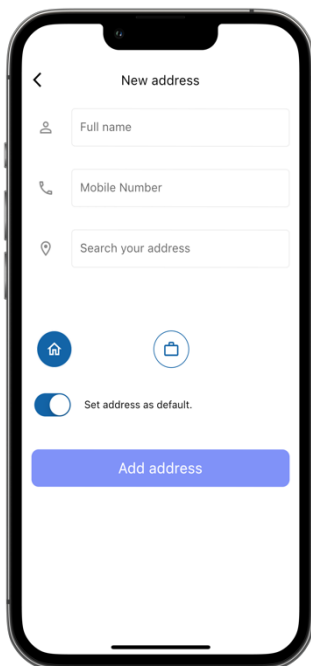


Figure 4.31: New address page

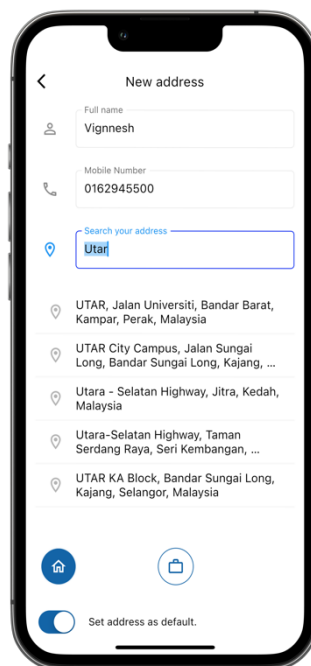


Figure 4.32: New address page – Address Selection

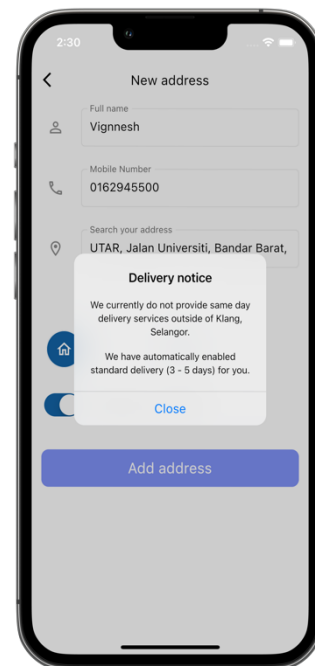


Figure 4.33: New address page – Delivery Notice alert

Checkout / Payment pages

Figures 4.34 and 4.35 shows the checkout page designed for The Quadry application. Figures 4.36 and 4.37 shows the execution of the backend Stripe payment gateway functions. Figures 4.38 and 4.39 shows the respectively redirected pages with appropriate dialog messages indicating if the payment was a success or failure.

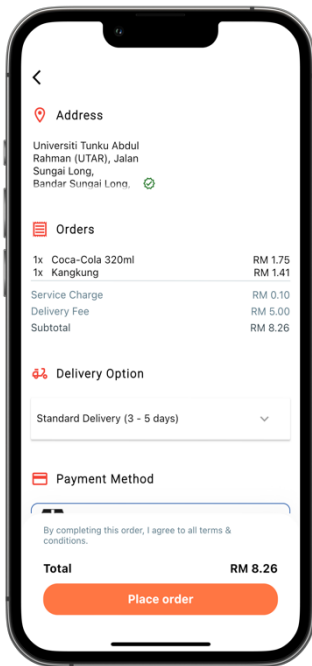


Figure 4.34: Checkout page

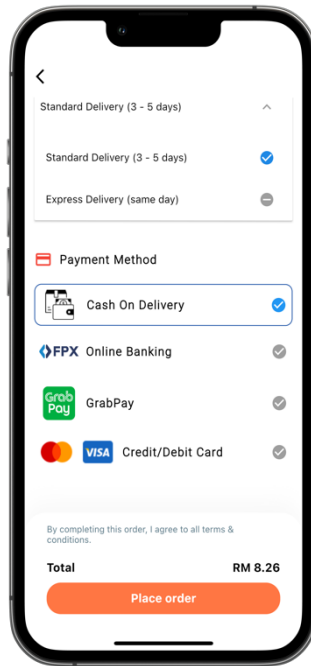


Figure 4.35: Checkout page (2)

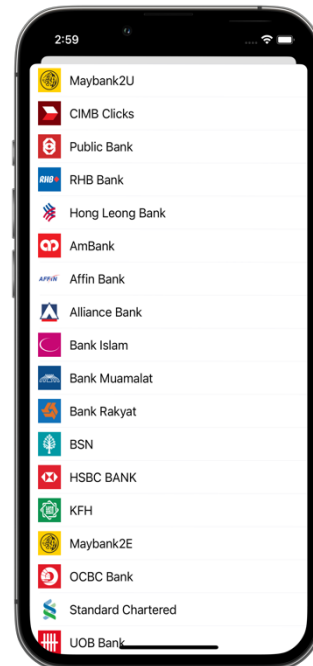


Figure 4.36: Stripe FPX payment selection

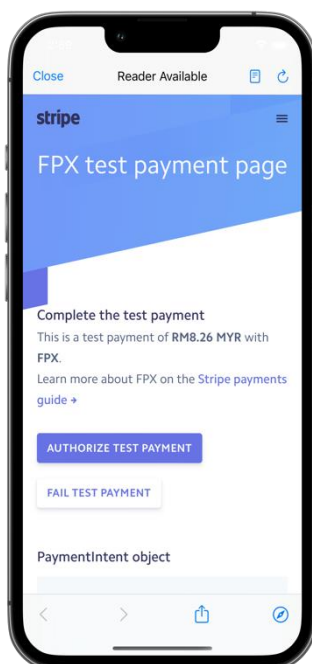


Figure 4.37: Stripe test payment page

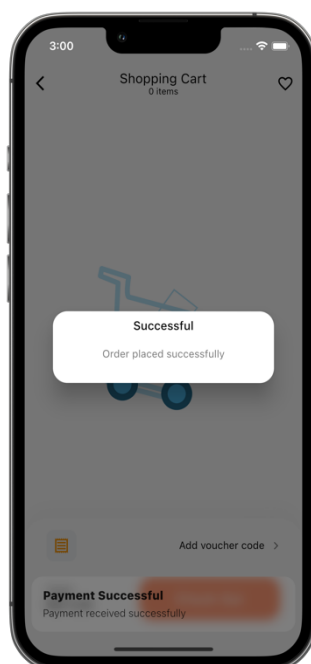


Figure 4.38: Successful test payment redirect

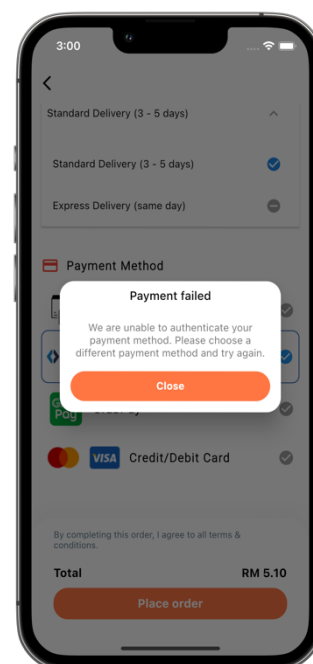


Figure 4.39: Unsuccessful test payment redirect

Order page

Figures 4.40 and 4.41 are the initial views of the orders page where past and ongoing orders are separated for a more simplistic UI. Figures 4.42 and 4.43 describes an order's details such as shop the order was made from, time of order, products purchased, delivery location and more. Figures 4.44 and 4.45 shows the pop-up that will be displayed to users when an order has been delivered.

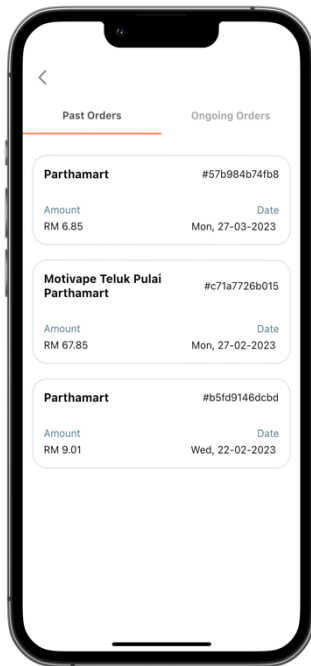


Figure 4.40: Order page
(Past Orders)

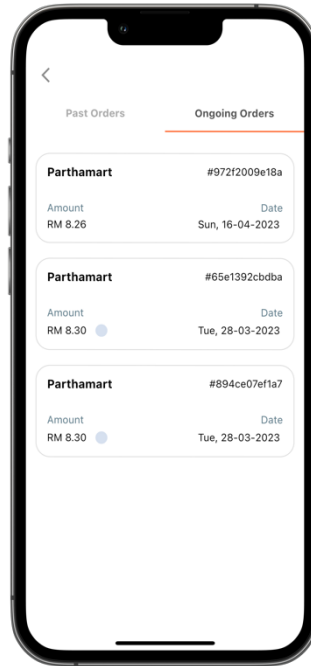


Figure 4.41: Order page
(Ongoing Orders)

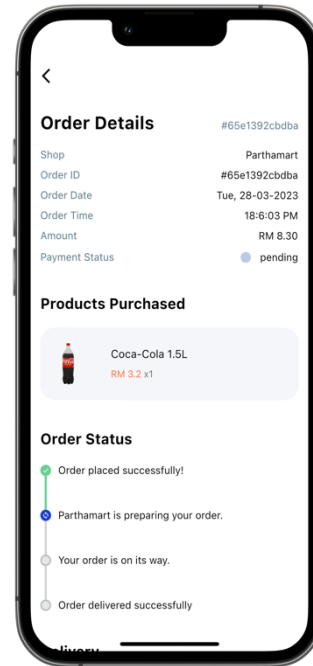


Figure 4.42: Ongoing Order
Details

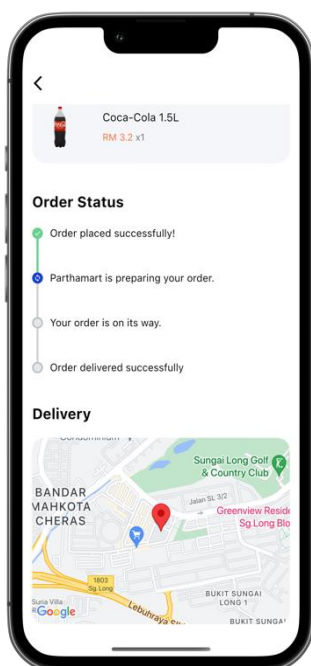


Figure 4.43: Ongoing Order
Details (2)

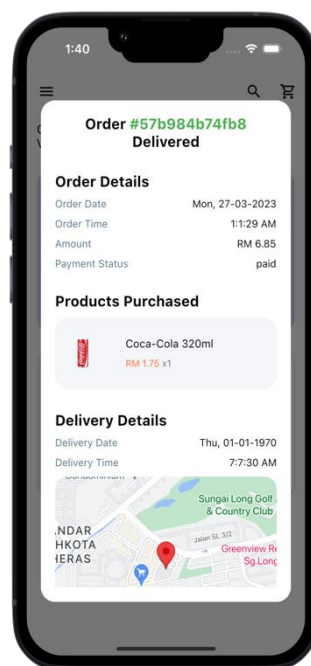


Figure 4.44: Order Delivered
Pop-up

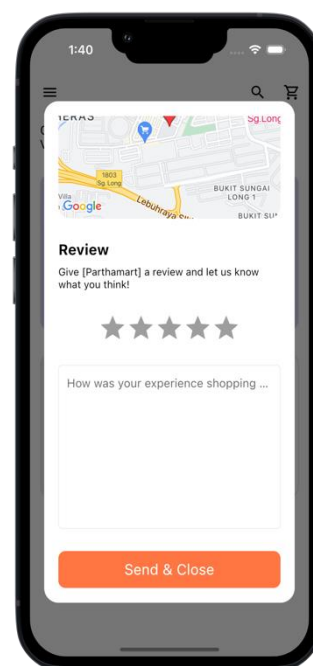


Figure 4.45: Order Delivered
Pop-up (2)

CHAPTER 5

DATA MODELLING

5.1 Introduction

This section demonstrates the database and backend designs that were implemented in the Quadry application. Among the designs that will be demonstrated are the Entity-Relationship Diagram (ERD), Firebase NoSQL Logical Design Diagram, and Data Dictionary explaining data stored and their purposes.

5.2 Entity-Relationship Diagram (ERD)

The following diagram demonstrates the Entity-Relationship Diagram (ERD) that was implemented for this project. The entities illustrated in Figure 5.1 describes the designed database tables/collections that helps with retrieving queries for the application's functionalities.

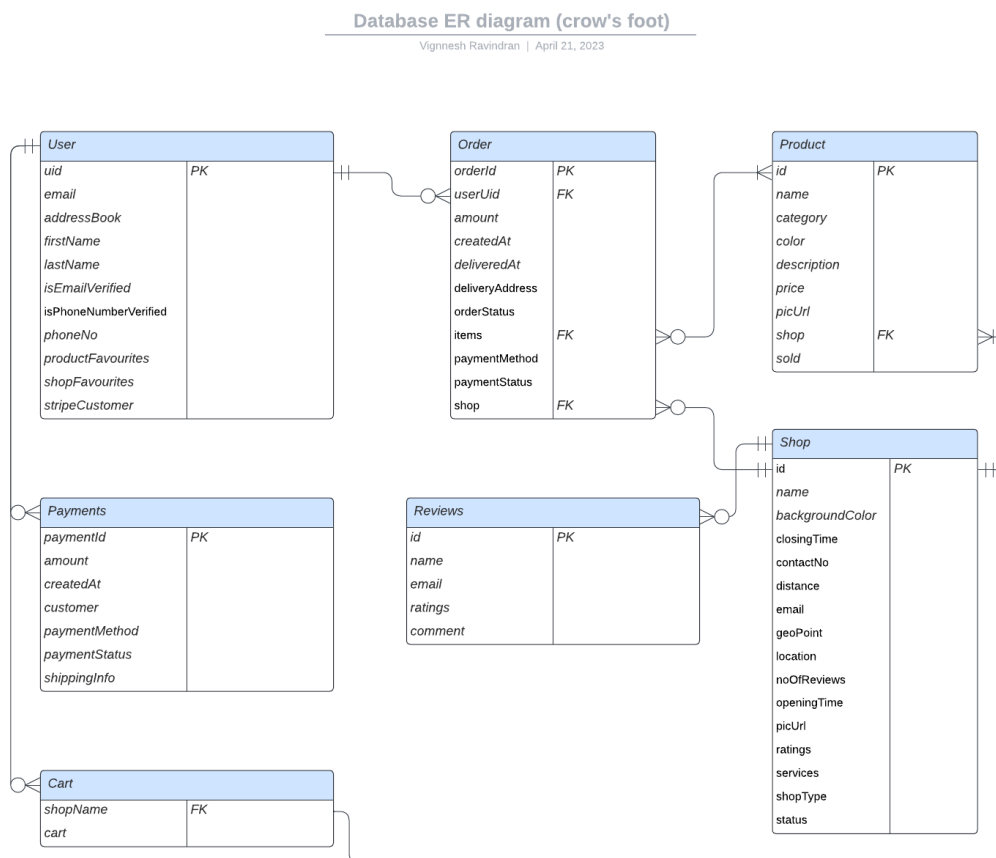


Figure 5.1: Entity-Relationship Diagram of collections in The Quadry application

5.2.1 Order Entity Relationships

The main entity in the above diagram is the order entity as it is widely connected to other entities such as the user, product, and shop. Each order can only be made by one user whereas a user can have many orders, thus the user's unique ID will be bound to the order entity as a foreign key. From the order entity perspective, their relationship will be described as the order having a one mandatory relationship with the user. From the user entity perspective however, it can be described as the user can have none or many orders. For example, if a user purchases a product, an order is placed, and it will contain the user's unique ID making it possible to identify the user that has made the purchase.

An order consists of at least one or many products whereas a product can or not exist in an order. The items attribute in the order entity is a foreign key that will be retrieved from the product entity signifying the product's unique ID to identify the different products purchased by the user. From the order entity perspective, their relationship will be described as the order having at least one to many relationships with the product entity. From the product entity perspective however, it can be described as the product exists or not in an order (depending on if the user buys the product or not). For example, a user purchases one or many products using the application.

It is mandatory for every order to include the shop ID from which the products were purchased from. The shop attribute in the order entity is a foreign key which saves the shop's unique ID as it eases the operations that will be later conducted in the merchant-side application. From the order entity perspective, their relationship will be described as an order must have one mandatory shop. From the shop entity perspective however, it can be described as the shop having none or many orders. For example, as there are many shops in the application, some of the shops may have many orders, however, some shops may not have orders at all.

5.2.2 User – Payment Entity Relationships

The user entity describes the user information stored and its relationships with other entities. It is mandatory for every payment registered be linked to only one user. In this case, from the user entity perspective, the user entity has a one mandatory to optional many relationships with payment. From the payment entity perspective however, each payment must have one user. For example, a user with a registered account may or may not choose to purchase products from the application and if product(s) were purchased, they have to be linked back to the user who purchased them.

5.2.3 Shop – Product Entity Relationships

A shop can contain many different types of products; however, it is mandatory for every product to have only one shop. This shop attribute in the product entity is a foreign key of the shop's unique ID indicating that specific product belongs to the shop. From the product entity perspective, the product entity must have one mandatory shop's unique ID, whereas from the shop entity perspective, a shop entity could have from one-to-many products. For example, when a user purchases product(s) from the application, it has to be registered to a shop to identify which shop is the user purchasing the product from.

5.2.4 Shop – Review Entity Relationships

User can optionally choose to provide their feedback regarding a shop's products or services however, each review submitted by users must be linked to a shop. Therefore, from the shop entity perspective, it can have none to many reviews whereas from the review entity perspective, it must have a shop to indicate that the review made was for that specific shop respectively. For example, upon every successful delivery of orders, users will be provided an option to rate or/and comment about the product/services received, these reviews are required to be linked to a shop which would otherwise be difficult to identify which reviews was meant for which shop.

5.3 Firebase NoSQL Logical Design Diagram

The following diagrams demonstrate the logical design of the Firebase NoSQL database that was implemented for this project. The database contains 9 collections one for products, send grid email templates, shops, shop products, shop reviews, users, user orders, user payments, and welcome new user email collection respectively. The fields stored in each document within a collection is explained by the diagram below and data dictionary.

5.3.1 Products Collection

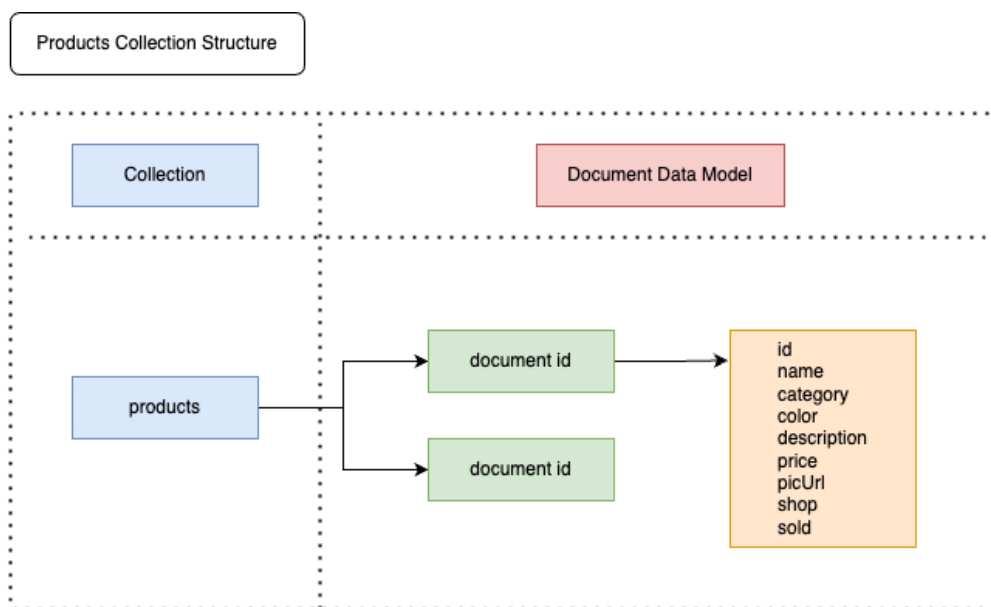


Figure 5.2: Data Structure of Products Collection

Figure 5.2 illustrates the data modelling designed and implemented for the products collection. This collection contains all products from various different merchants to serve as a centralised products collection for multiple purposes that would benefit the merchant-side application. The field attributes of the documents in this collection and their purposes will be briefly explained in the Data Dictionary section.

5.3.2 SendGrid Email Templates Collection

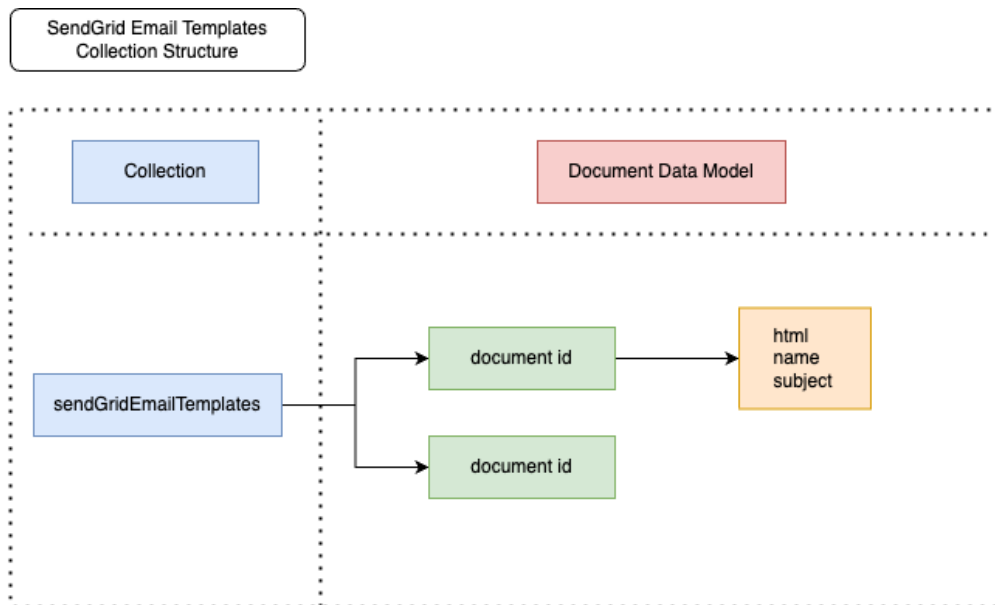


Figure 5.3: Data Structure of SendGrid Email Templates Collection

Figure 5.3 demonstrates the data model that was designed for the SendGrid email templates collection. The purpose of this document is to store the various types of email templates to be filled and sent to users. This feature works by installing the Trigger Email service in Firebase Extension and specifying the name of this collection as its template repository. Once that is done, whenever there arises a need to send users an email, the template name will be specified (document id) and Trigger Email will automatically embed the mentioned template's html attribute with provided values such as the name of the user before sending users the email with the subject attribute. This collection was created for the sole purpose of storing the templates that will be needed for the automation of sending users an email using the aforementioned services.

5.3.3 Shops Collection

The next illustration, Figure 5.4, outlines the collection structure of the shops collection. This collection serves the purpose of saving all shops' details such as the name of the merchant, opening and closing times, GeoPoint (coordinates) of the merchant, contact info such as email, and other relevant information. This collection, however, contains two other subcollections named 'products' and 'reviews' which helps with the overall flow of the application. Describing the first subcollection, 'products', contains all the products the organisation offers in their store. Unlike the centralised products collection viewed in Figure 5.2, this collection only contains products that are sold by the merchant. This helps when a user navigates to a specific merchant's page on the application, allowing them to view all types of products offered. The second subcollection, 'reviews' is used to store all reviews provided by customers/users towards that specific merchant's product or service. The documents of this 'reviews' collection consists of the comment field for user review or comment and the ratings given by the user which will be used to calculate the overall ratings for that merchant.

5.3.4 Users Collection

The collection that stores all users' information is called the users collection as illustrated in Figure 5.5 below. This collection consists of the user's unique ID as the document ID and within each document contains the user's details such as first name, last name, their addresses in the address book, the default address selected, their phone number, etc. Every user document is the parent of three other subcollection similar to the shops collection but for a different purpose. The first subcollection is called 'orders' where all of the user's orders will be stored with detailed information regarding each order such as the order ID, amount that was paid, payment method used to pay for that order, delivery address for the order, items purchased, order status, and more. The second subcollection is called 'payments' where the payment details for each order placed or products purchased will be logged. These documents contain the customer's payment ID, Stripe customer ID, payment status, shipping information, and more. This subcollection exists to help troubleshoot issues in the event of any errors faced by the user regarding their payment or order. The last subcollection is called cart which consists of the products that were added

to the cart by users. All documents in the collection are named after the shops' ID of which the products are from. This is to organise items in the cart by the shop they belong to. The map of maps field called 'cart' in each of these documents holds a map of product ID as the key and the product's attributes such as name, category, etc. as the value.

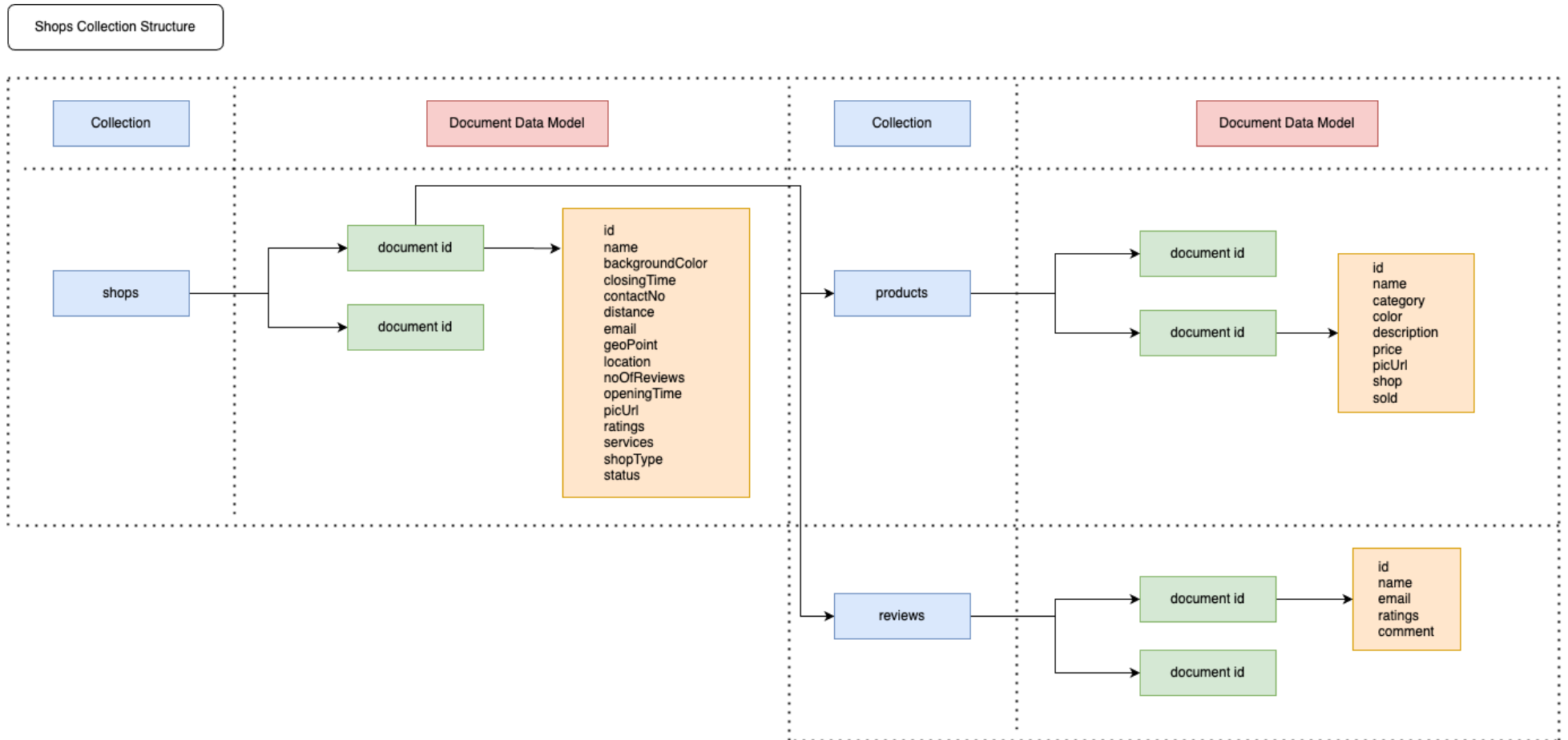


Figure 5.4: Data Structure of Shops Collection

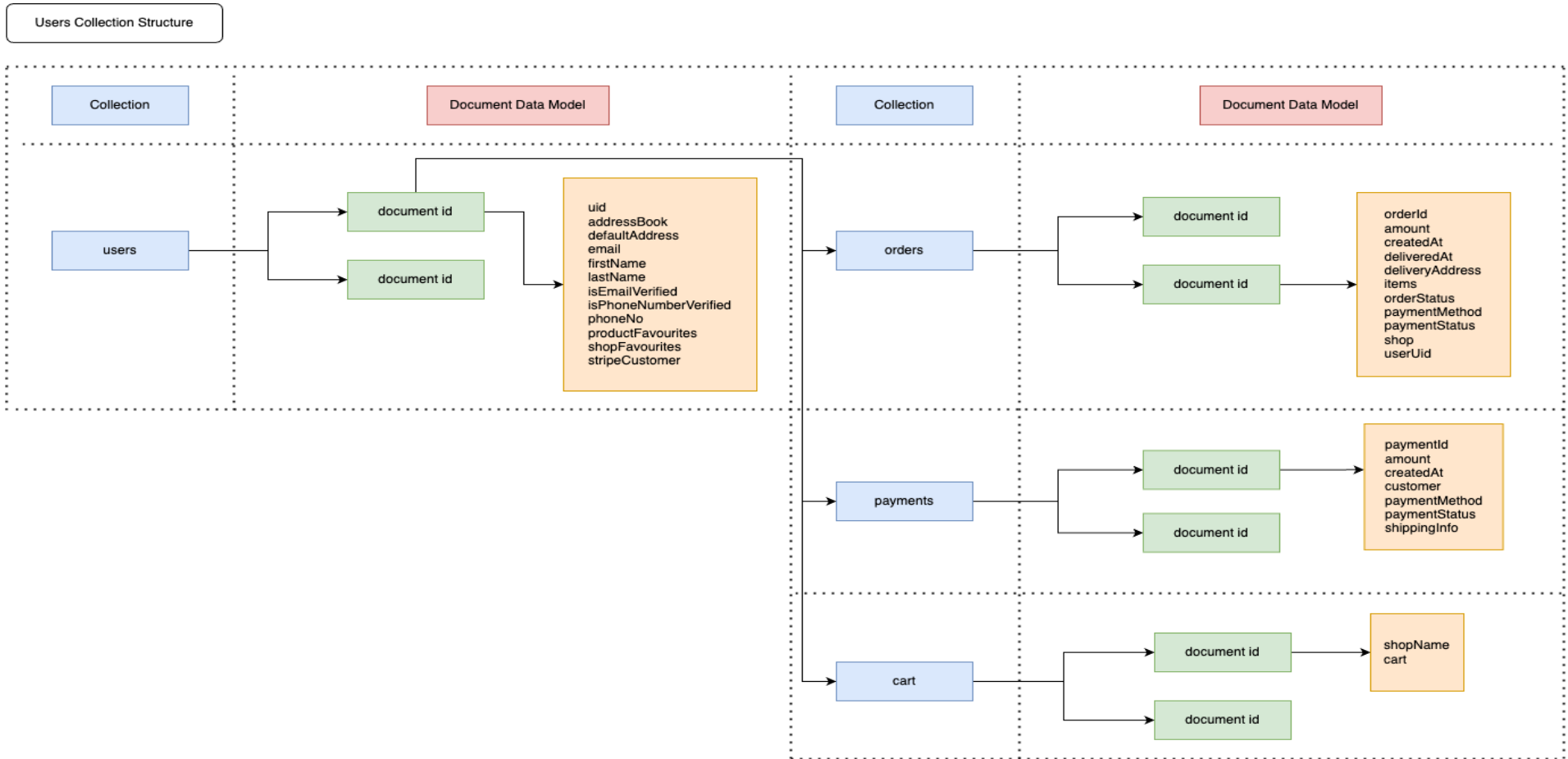


Figure 5.5: Data Structure of Users Collection

5.3.5 Welcome New Users Mail Collection

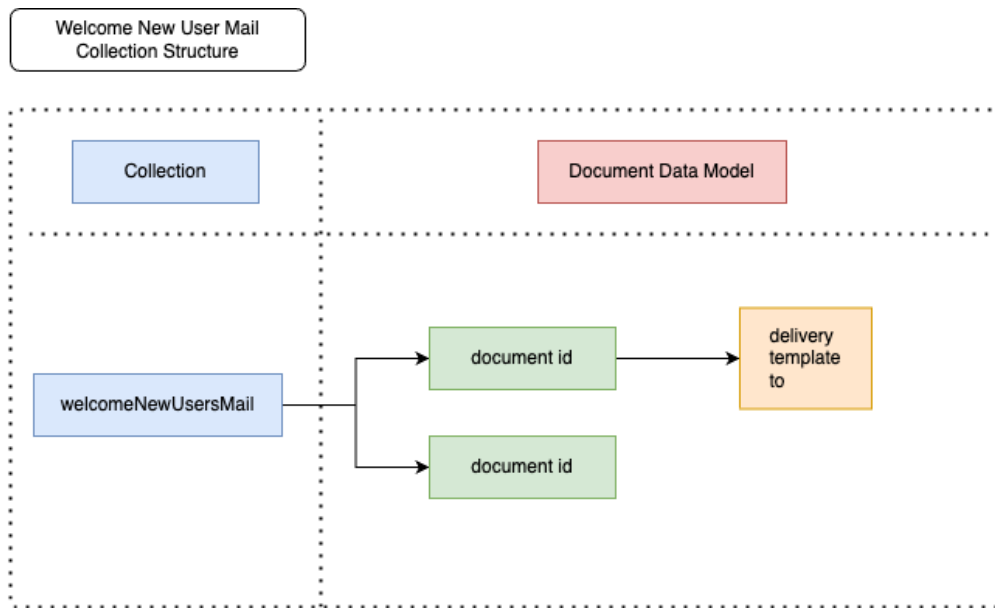


Figure 5.6: Data Structure of Welcome New User Mail Collection

The welcome new user mail collection in Figure 5.6 was modelled for the purpose of sending automated emails to users upon successfully registering into the application for the first time. This was implemented by using Firebase Extension and Tigger Email which was then configured to use any newly created document in the `welcomeNewUsersMail` collection. Every newly created document consists of the delivery status attribute, which is dynamically updated by the Trigger Email extension, the template attribute to be used from the templates collection called `sendGridEmailTemplates`, and the `to` attribute which specifies the user's email address in which the automated email will be sent to. The template is an attribute of type map which can contain attribute `data` which specifies the data that will be embedded into the template to be used and must contain the `name` attribute which holds the value of the name of template from the `sendGridEmailTemplates` collection.

5.4 Data Dictionary

The following tables explain the attributes in each of the eight database collections that was defined in the above subsections. The ‘products’ subcollection of the ‘shops’ collection will not be tabled as its attributes, usage, and data types are similar to the centralised ‘products’ collection. Columns with the name of the attribute, its data type, a detailed explanation in terms of usage and purpose, and an example value of the attribute will be briefly outlined.

5.4.1 Products Data Dictionary

Field Name	Data Type	Description	Examples
id	String	An auto-generated unique ID of the product.	5m96YYK09S03cQ39Q6ML
name	String	The name of the product.	Coca-Cola 1.5L
category	String	The category that the product belongs to.	Beverage
color	Number (int)	The most dominant colour of the product’s picture to illuminate the background colour of the UI in the application.	4281150251
description	String	A short description regarding the product.	Lorem Ipsum is simply dummy text of the printing and typesetting industry.
Price	Number (Float)	The price of the product.	3.19
picurl	String	The hosted URL of the product’s picture.	https://firebasestorage.googleapis.com/v0/b/quadry-455e5.appspot.com/o/products%2F5m96YYK09S03cQ39Q6ML.png?alt=media&token=9ed86388-5967-4ff9-acbd-b09903cb8b2a

shop	String	The name of the shop that the product belongs to.	Parthamart
sold	Number (int)	The number of items users have purchased this product.	400

Table 5.1: Products Data Dictionary

5.4.2 SendGrid Email Templates Data Dictionary

Field Name	Data Type	Description	Examples
html	String	The html code of which will be used by the template.	<!DOCTYPE html>
name	String	The name of the template.	newUserWelcomeV1
subject	String	The subject of the email when sending users an email with this template.	Welcome to The Quadry!

Table 5.2: SendGrid Email Templates Data Dictionary

5.4.3 Welcome New User Mail Data Dictionary

Field Name	Data Type	Description	Examples
delivery	Map	An automated map field containing the delivery status of the mail, among keys included are if the email was delivered, pending, rejected state, start time, attempts, etc.	{ attempts: 1, error: null, response: 250, state: "success", ... }
template	Map	A map containing the name of template to be used when sending this mail and optionally the data to be embedded into said template.	{ name: "newUserWelcomeV1", data: { user: "Vig" } }
to	String	The email of user to which this mail will be sent to.	test@gmail.com

Table 5.3: Welcome New User Mail Data Dictionary

5.4.4 Shops Data Dictionary

Field Name	Data Type	Description	Examples
id	String	An auto-generated unique ID of the shop.	3ipPoxVsNNbHn6MOAmml
name	String	The name of the shop.	Parthamart
backgroundColour	Number (int)	The most dominant colour of the shop's picture to illuminate the background colour of the UI in the application.	4293398807
closingTime	String	Closing time of the shop to reflect on the application on whether or not users can place an order.	23:30
contactNo	String	Contact number of the shop for enquiry purposes.	+60162999999
distance	Number (Float)	Distance of the shop from the user's location.	5.5
email	String	Contact email of the shop for enquiry purposes.	parthamart@gmail.com
geoPoint	GeoPoint	GeoPoint (coordinates) of the shop to track distance and shipment information.	[2.9949472619 3865° N, 101.435352629 61936° E]
location	String	Plain text address of the shop for user's reference.	Bandar Bukit Tinggi 2, 41200 Klang, Selangor
noOfReviews	Number (int)	Number of reviews this shop has.	7
openingTime	String	Opening time of the shop to reflect on the application on whether or not users can place an order.	09:00

picUrl	String	The hosted URL of the product's picture.	https://firebases-storage.googleapis.com/v0/b/quady-455e5.appspot.com/o/shopPictures%2Fjaya_grocer.png?alt=media&token=a265f6cf-311a-4e3c-a02f-a25bd428462e
ratings	Number (double)	The average ratings of the shop which is calculated using an automated listener hosted in Firebase Cloud Functions.	4.285714285714286
services	Array	An array of type String which stores the types of services the shop provides.	["Grocery", "Delivery"]
shopType	String	The category of shop this merchant belongs to ensure the shop is queried and displayed to the user when they select the shop category on the home page.	Grocery
status	String	The status of the shop indicating if the shop is 'coming soon' to the platform, 'temporarily close' or 'active'.	active

Table 5.4: Shops Data Dictionary

5.4.5 Reviews Subcollection Data Dictionary

Field Name	Data Type	Description	Examples
id	String	An auto-generated unique ID of the review.	0UVJZHh9utOIV BbVbIIv
name	String	Name of user to who submitted this review.	Vignnesh
email	String	Email of user to who submitted this review.	test@gmail.com
ratings	Number (float)	The ratings given by the user regarding the shop's products or services	4.5
comment	String	An optional comment that users can provide upon successfully receiving their purchased product(s).	Very good service

Table 5.5: Reviews Subcollection Data Dictionary

5.4.6 Users Data Dictionary

Field Name	Data Type	Description	Examples
uid	String	An auto-generated unique ID of the user.	7ZhCqwAaPXXm 4HnGHMuXzfJQ4 6f2
addressBook	Map	A map containing the user's addresses with a unique ID as the key and address components such as address line 1, line 2, city, etc. for values.	{ asbdjabs: { city: ..., line1: ..., line2: ..., } }
defaultAddress	String	The address selected as the default using the address unique ID.	Asbdjabs
email	String	Email address of the user.	test@gmail.com
firstName	String	First name of the user.	Vignnesh

lastName	String	Last name of the user.	Ravindran
isEmailVerified	Boolean	Boolean value to detect if the user's email address has been verified.	true/false
isPhoneNumberVerified	Boolean	Boolean value to detect if the user's phone number has been verified.	True/false
phoneNo	String	Phone number of the user.	+60162999999
productFavourites	Array	An array of maps storing the products favoured by the user.	[{product 1}, {product 2}, ...]
shopFavourites	Array	An array of maps storing the shop favoured by the user.	[{shop 1}, {shop 2}, ...]
stripeCustomer	String	User's unique Stripe ID which is created when a payment is made.	cus_NL45PhM0hd tKNy

Table 5.6: Users Data Dictionary

5.4.7 Orders Subcollection Data Dictionary

Field Name	Data Type	Description	Examples
orderId	String	An auto-generated unique ID of the order.	b9956d80-b2c6-11ed-9276-b5fd9146dcbd
amount	String	Total amount of the order that was paid by the user.	9.01
createdAt	Number (int)	Timestamp in milliseconds since epoch of when the order was created.	1677080210522

deliveredAt	Number (int)	Timestamp in milliseconds since epoch of when the order was delivered to the user.	1677479746537
deliveryAddresses	Map	Map containing components of the user's address such as address line 1, line 2, city, etc.	{ line1: ..., line2: ..., city: ..., }
items	Array	An array of maps storing the product(s) that were purchased by the user.	[{product 1}, {product 2}, ...]
orderStatus	String	Status of the order indicating if the order is preparing, delivering, or completed.	completed
paymentMethod	String	Payment method that was chosen by the user when purchasing the product(s).	FPX
paymentStatus	String	Status of the payment performed by user indicating if its paid or pending (for Cash on Delivery orders)	paid
shop	String	Unique ID of the shop of which the purchased product(s) in this order belong to.	3ipPoxVsNNbHn6MOAmml
userId	String	Unique ID of the user that created this order.	7ZhCqwAaPXXm4HnGHMuXzfJQ46f2

Table 5.7: Orders Subcollection Data Dictionary

5.4.8 Payments Subcollection Data Dictionary

Field Name	Data Type	Description	Examples
paymentId	String	A Stripe auto-generated payment intent ID of the payment.	pi_3Mb52SAycqP IWvxN0qyzQsgA
amount	Number (float)	Total amount of the purchase that done by the user.	2068
createdAt	Number (int)	Timestamp in milliseconds since epoch of when the payment was performed.	1676306280
customer	String	User's unique Stripe ID which is created when a payment is made.	cus_NL45PhM0hd tKNy
paymentMethod	String	Payment method that was chosen by the user when purchasing the product(s).	Card
paymentStatus	String	Status of the payment performed by user indicating if the payment was processed by Stripe.	succeeded
shippingInfo	Map	Map containing components of the user's address such as address line 1, line 2, city, etc., user's name, phone number, delivery carrier, and tracking number.	{ address: { line1: ..., line2: ..., city: ..., }, name: ..., phone: ..., trackingNo: ..., carrier: ... }

Table 5.8: Payments Subcollection Data Dictionary

5.4.9 Cart Subcollection Data Dictionary

Field Name	Data Type	Description	Examples
shopName	String	Unique ID of the shop of which products in this document belongs to.	3ipPoxVsNNbHn6 MOAmml
cart	Map of Maps	A map with the product ID as the key and its attributes as the value.	{ 5m96YYK09S03c Q39Q6ML: { id: ..., cost: ..., name: ..., price: ..., quantity: ..., picUrl: ..., shopName: ..., }, }

Table 5.9: Cart Subcollection Data Dictionary

5.5 Conclusion

This section summarises the data modelling for database and backend designed to help with the implementation and development of The Quadry application. The section starts off by describing the entity-relationship models between each collection/table that will exist in the database. The Firebase NoSQL Logical Design diagram is then presented and briefed, showing its architecture and workflow between each collection, documents, and fields. Ending the section on a detailed note, the data dictionary is presented tabling the attributes or fields of every collection in the Firestore with their respective data type, description, and an example for further clarification.

CHAPTER 6

IMPLEMENTATION

6.1 Introduction

This section outlines the implementation performed for the development of The Quadry application where each design towards the implementation will be displayed with their respective brief explanation and workflow. This project mainly focuses on the user side of the entire system and therefore all modules and designs outlined in this section refers to the user side of the application.

6.2 User Authentication Design

Upon reaching the authentication page, users will be presented with two options which are to login or register into the application. This is the authentication designed to handle user registration and login.

6.2.1 Registration

- a. User registration is performed via Firebase Authentication. Users can register themselves into the application as a new user by using their email and password, if they do not already have an existing account. They will also be required to provide additional details such as their first name, last name, and mobile number.
- b. The users upon providing all required information and submitting, their info will be saved in the Firebase Firestore collection.
- c. The creation of a new user document triggers a backend function hosted in Firebase Cloud Functions to enable the Firebase Extension and SendGrid API to send the user a welcome email via SMTP service.
- d. The user will then be navigated to a verification page where they will be required to verify their mobile number with an OTP which they will receive to the provided mobile number.
- e. Successful verification of their mobile number will navigate the user to another verification page where they will be required to verify their email with a verification link that will be sent to their registered email account.

- f. Successful verification of both mobile number and email will allow users into the home page of the application while marking their profile as verified.

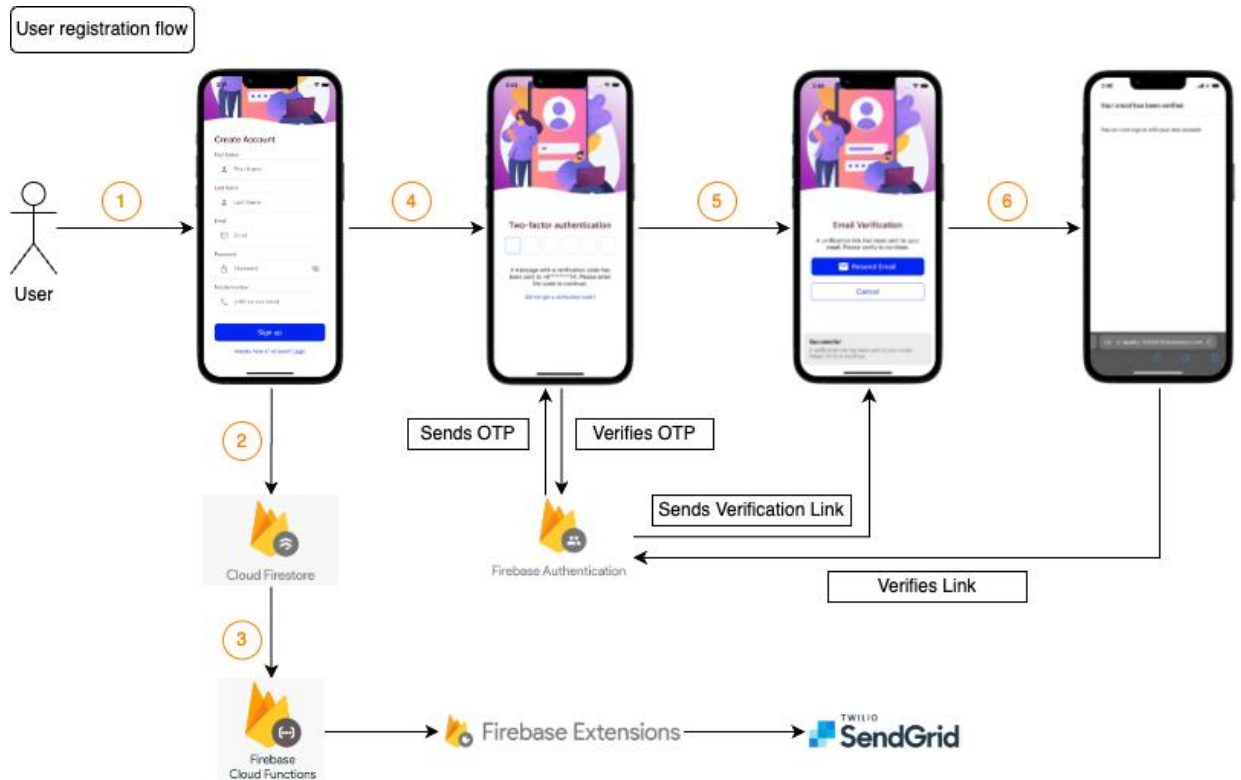


Figure 6.1: User Registration Architecture

6.2.2 Login

- User login is done via Firebase Authentication. Users can log into the application via their registered email and password.
- Email and password are validated to ensure they are registered, this process will be performed by Firebase Authentication.
- Upon successful verification the user details will be retrieved to be used in the application.

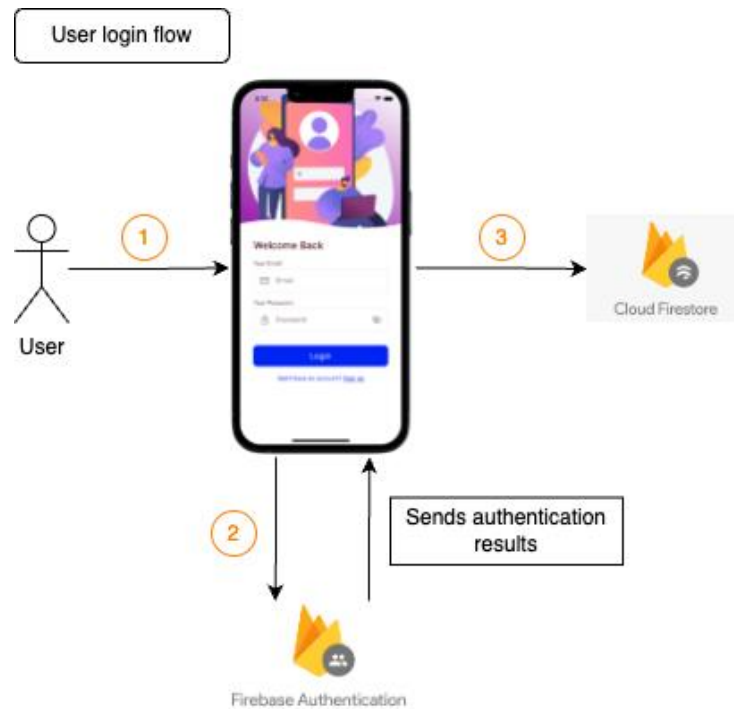


Figure 6.2: User Login Architecture

6.3 Add Products to Cart Design

The following steps and diagram describe and illustrates the process of user adding products to their cart upon successful authentication.

- a. User authenticates into the application using Firebase Authentication, which in return will allow user access into the Firebase Firestore to retrieve user their respective details using their unique id.
- b. Once navigated to the homepage, the user will then be required to select the type of shop they would like to purchase products from. The type of shops includes grocery shops, pet shops, and vape (e-cigarette) shops.
- c. Upon shop type selection, best-selling products for the selected shop type will be retrieved from Firebase Firestore and all shops that fall under that type will be rendered.
- d. User will then be able to add any product(s) to their cart from the best - selling section or by selecting a specific shop, and then adding product(s) to their cart.
- e. The added products are then stored in Firebase Firestore under the specific user's document.

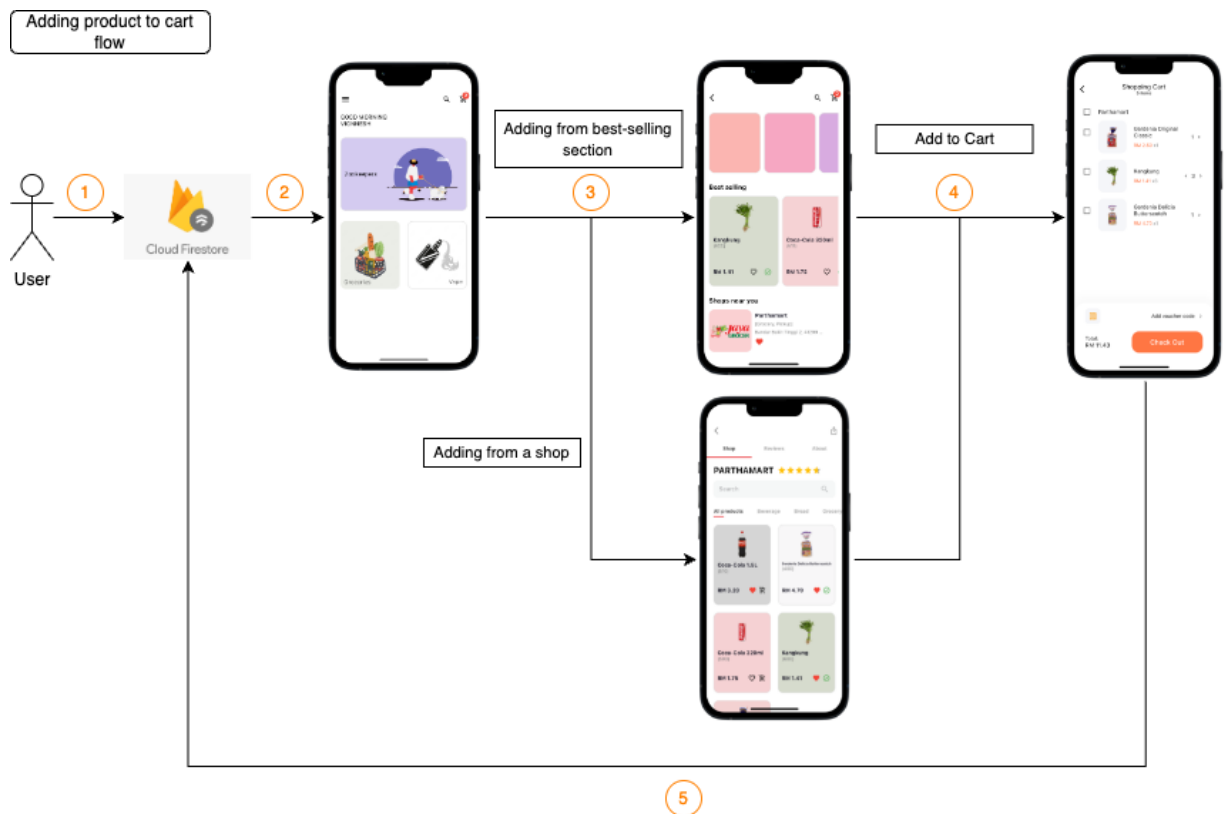


Figure 6.3: Adding Products to Cart Architecture

6.4 Purchasing Products via Stripe Payment Gateway

The following steps and diagram describe and illustrates the process of user purchasing products from their cart via the Stripe Payment gateway and the validations involved along the process.

- a. Once all desired products have been added to the cart by user, they will be required to checkout with the products intended to be purchased. This can be done by selecting certain products from the list or all products under a shop.
- b. On checkout button clicked, a validation is performed to ensure user has at least one address provided in their address book before proceeding to the checkout page. If there are no addresses present, a dialog box will be popped indicating the user that no address is provided.
- c. If an address is provided, the application navigates to the checkout page displaying several details such as delivery address, products selected with their respective quantity, total price, service and delivery charges and payment method for user selection.

- d. In order for a user to place order by clicking the 'Place Order' button at the bottom bar, they will be required to select a delivery address, delivery option ('Standard delivery' by default) and payment method ('Cash on delivery' by default). Clicking the 'Place Order' button will validate these requirements.
- e. If the user would like to switch their delivery option to 'same day delivery', they must select a delivery address within the vicinity of Klang city. Otherwise, a dialog will be prompted notifying user that same day delivery is only available within Klang city.
- f. If all required details are selected, the user will be redirected to the Stripe Payment Gateway to perform their transaction on a secure and authenticated page upon clicking the 'Place Order' button. This redirection is based on the type of payment method chosen, and the functions invoked will be hosted in the Firebase Cloud Function backend for security purposes.
- g. Upon successful payment, the user will be redirected back to the application (cart page), where they will be able to view that the selected products for purchase are no longer in the cart. A snackbar and dialog box will be displayed to notify user that the payment was successful, and the order was placed.



Figure 6.4: Purchasing products architecture.

6.5 User Address Selection

The following steps and diagram describe and illustrates the process of user selecting and saving their address in the application's address book and all the validation that is implemented throughout the process.

- There are two ways where the application prompts a user to insert their address into the address book; user can navigate to the drawer section then select profile or by checking out from the cart page with no address inserted in the address book.
- Once the user is in the address book page, they can add an address by clicking the 'Add Address' floating button on the bottom right corner.

- c. This navigates the user to a new page where they will be required to provide all necessary details such as their name, mobile number, and address.
- d. Clicking the 'Add Address' button without filling all fields will result in a validation error as all fields in this section are required to ensure the delivery process can be done without any confusion or trouble.
- e. As the user starts filling the address field, the Google Places Autocomplete API integrated with the application will be used to retrieve places that match the query of the user.
- f. Upon selection of user's desired address, the Google Place Details API will be used to retrieve the place's details such as address component (consists of street number, sub-locality, locality, postal code, etc.), geometry (longitude and latitude), and name (for organisations), using the selected place's id.
- g. A validation performed to ensure the address selected is within the vicinity of Klang city, otherwise a dialog box will be prompted to notify user of the delivery option constraint; same day delivery is only available within Klang.
- h. Once all details are retrieved, they will be saved in Firebase Firestore when user selects the 'Add address' button at the bottom of the page. The new address can then be viewed at the address book page.



Figure 6.5: User address selection architecture

6.6 Order Status Automation & User Notification

The following steps and diagram describe and illustrates the process of user placing an order upon successful payment via Stripe Payment Gateway. This section will also describe on the automated order status monitoring and user notification.

- a. Upon successfully completing the payment process via Stripe Payment Gateway, a new order will be created with its own unique ID and stored in Firebase Firestore. The user will then be redirected to the cart page where they will see that the product(s) that was purchase will no longer be in the cart. From there, the user can go to the drawer section to view their orders.
- b. Once users click the 'Orders' tile in the drawer section, they will be navigated to an orders page where users can view ongoing and past orders (set to be the default view).
- c. For products that are recently purchased, a new order will be created under the status of 'preparing' which essentially implies that the order has been accepted and is currently being prepared by the merchant.
- d. Once the order is prepared and ready to be delivered, the order status changes to 'delivering' where Firebase Cloud Functions listens to the changes and notifies the user that their order status had been changed.
- e. When the order has been successfully delivered to the user, the user will be notified again via Firebase Cloud Functions and an overlay widget will be displayed to the user regarding the order that was delivered with details such as the order ID, shop purchased from, products purchased, total amount, delivered time, etc. and a rating from that the user can submit to rate the merchant and optionally provide any comment regarding their service/product.
- f. If the user rates the merchant or optionally provides any comment regarding the service/product of the merchant, a Firebase Cloud Function is triggered to automatically update the average ratings of the merchant based on the pervious ratings and the number of reviews provided in total. These details are updated in Firebase Firestore.

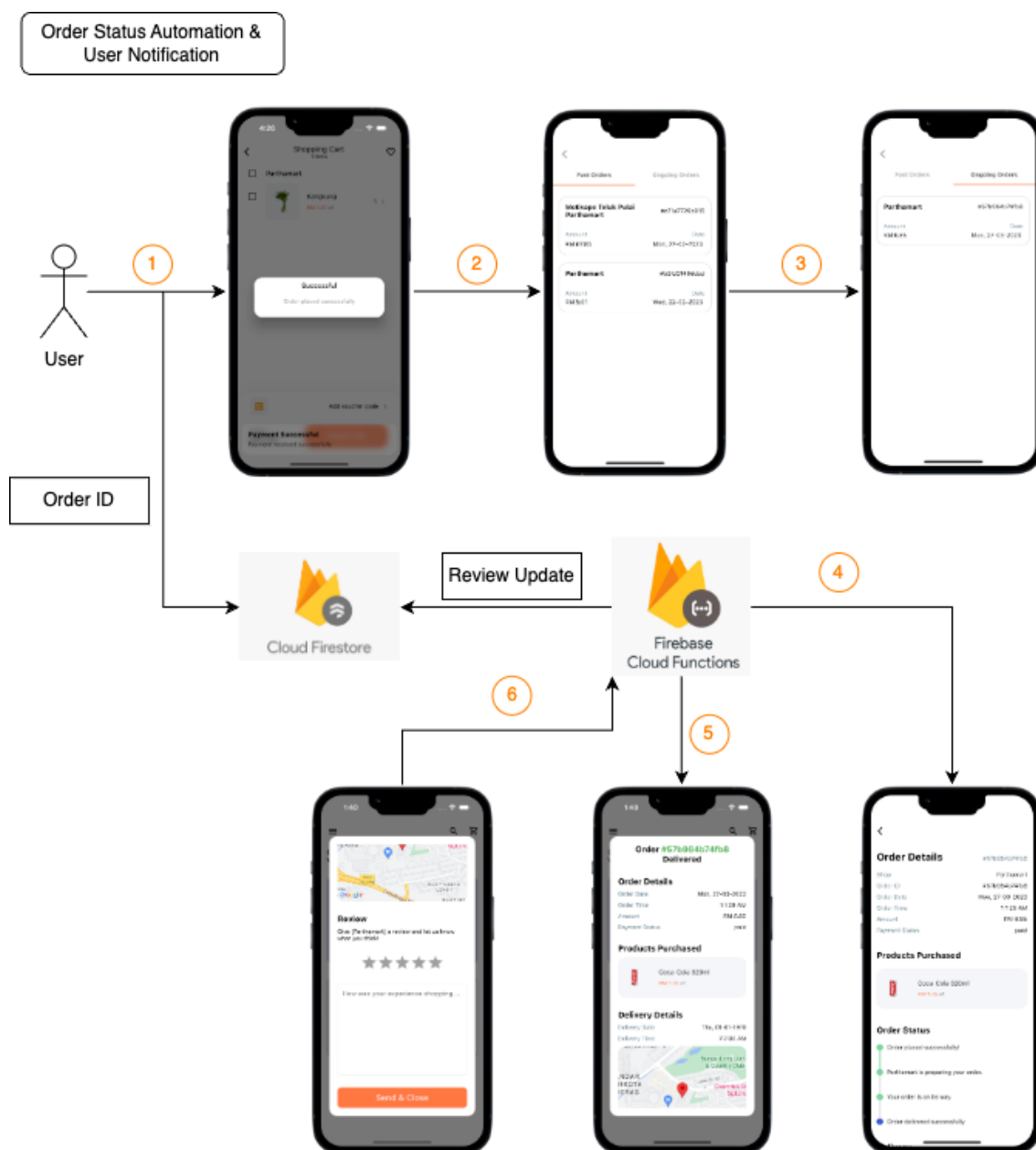


Figure 6.6: Order status automation architecture

6.7 Summary

To recap, this chapter summarises the implementation of the development of The Quadry mobile application. Among the topics discussed in this chapter were the overall flows of the application from when a user registers/logs into the application, through the process of purchasing product(s), and receiving order status notifications. It covers the main aspects and implementations of the application detailing the architecture of how each process or user functionalities were designed and integrated with Firebase services such as Firestore, Cloud Functions, etc. and third-party services/APIs such as Google APIs and Stripe.

CHAPTER 7

SYSTEM TESTING

7.1 Introduction

Testing is a mandatory phase when developing any software systems to ensure the application components behave as how they are intended to according to the requirements. In this chapter, the testing objectives, testing scope, and types of testing will be outlined in detail.

7.2 Testing Objectives

The objectives aimed to accomplish through testing is explained as below:

- a. Ensure the user is able to register into the application.
- b. Ensure the user is able to verify their phone number via TAC.
- c. Ensure the user is able to verify their email via verification link.
- d. Ensure the user is able to login into the application.
- e. Ensure the use is able to add products to cart.
- f. Ensure the user is able to place an order successfully.
- g. Ensure the user is notified upon order completion.
- h. Ensure the application can communicate with Google Places Autocomplete API for user address selection.
- i. Ensure the application can communication with Stripe Payment Gateway for payment methods.
- j. Ensure the user is able to receive a welcome email after successful registration via automated SendGrid API.

7.3 Testing Scope

For this application, the components that are tested are listed as follows:

- a. Requirements listed on user registration and login.
- b. Requirements listed on the functionality of the application.

Components that are not tested are listed as follows:

- a. Performance testing
- b. Database testing

7.4 Testing Type

Testing that has been conducted on the application are as follows:

- a. Unit Testing on the major functionalities of the application.
- b. Integration Testing between the application and integrated third-part services.
- c. User Acceptance Testing (UAT) which is conducted together with Maurchand Ent.

7.5 Testing Results

This section describes the results for the different types of testing that were conducted on the application. Tests case for testing objectives will be provided with detailed descriptions on the summary, test steps, test data, expected result, actual result, and the status of the test indicating if the test passed or failed. Unit and User Acceptance Testing (UAT) were conducted on all the test objectives; however, Integration Testing was conducted on objectives (g), (h), and (i) as they involve the application interacting with their-party applications.

7.5.1 Unit Testing

Test Case No	001	Test Case Name	Create Account		
Designed by	Vignnesh Ravindran	Design date	30/03/2023		
Role	User				
Executed by	Vignnesh Ravindran	Executed date	30/03/2023		
Pre-conditions	User has no registered account prior to creating account.				
Summary	Test steps	Test data	Expected results	Actual Results	Status
Enter first name, last name, email, phone number, and password.	1. Enter first name.	Vignnesh	A new user created in the database and a TAC is sent to the user's phone number.	A new user was created in the database and a TAC was sent to the user's phone number.	Pass
	2. Enter last name.	Ravindran			
	3. Enter email.	ravin.vignnesh@gmail.com			

	4. Enter password	123456	User is navigated to the TAC verification page.	User was navigated to the TAC verification page.	
	5. Enter phone number.	+60162945500			
	6. Click 'Sign up' button.	-			
Enter first name	1. Enter first name.	Test	Error messages will prompt requesting user to fill in all other fields.	Error messages prompted requesting user to fill in all other fields.	Pass

Table 7.1: Unit Test Case for Create Account

Test Case No	002		Test Case Name	Verify Phone Number	
Designed by	Vignnesh Ravindran		Design date	30/03/2023	
Role	User				
Executed by	Vignnesh Ravindran		Executed date	30/03/2023	
Pre-conditions	User has registered an account and TAC was sent to user's phone number.				
Summary	Test steps	Test data	Expected results	Actual Results	Status
Enter received TAC.	1. Enter the TAC sent to phone number.	53706	Verification is successful and user is navigated to the email verification page with a verification link sent to the provided email.	Verification was successful and user was navigated to the email verification page with a verification link sent to the provided email.	Pass

Enter incorrect TAC	1. Enter a random TAC code, other than the one sent.	12345	A snackbar is displayed with “Verification Failed”.	A snackbar was displayed with “Verification Failed”.	Pass
---------------------	--	-------	---	--	------

Table 7.2: Unit Test Case for Verify Phone Number

Test Case No	003		Test Case Name	Verify Email Address	
Designed by	Vignnesh Ravindran		Design date	30/03/2023	
Role	User				
Executed by	Vignnesh Ravindran		Executed date	30/03/2023	
Pre-conditions	User has registered an account and verified TAC sent to their phone number.				
Summary	Test steps	Test data	Expected results	Actual Results	Status
Verify email address with link sent to email.	1. Verify email address using the verification link sent to the provided email.	-	Verification is successful and when user opens the application again, they are navigated to the home page.	Verification was successful and when user opens the application again, they were navigated to the home page.	Pass
Open app without verifying email.	1. Open the application without verifying email address.	-	User is navigated to the email verification page and unable to proceed unless email is verified.	User was navigated to the email verification page and unable to proceed unless email is verified.	Pass

Table 7.3: Unit Test Case for Verify Email Address

Test Case No	004	Test Case Name	Login to Account		
Designed by	Vignnesh Ravindran	Design date	30/03/2023		
Role	User				
Executed by	Vignnesh Ravindran	Executed date	30/03/2023		
Pre-conditions	User has registered an account.				
Summary	Test steps	Test data	Expected results	Actual Results	Status
Enter email and password.	1. Enter email.	ravin.vignnesh@gmail.com	Authentication is successful and user is navigated to the home page.	Authentication is successful and user is navigated to the home page.	Pass
	2. Enter password	123456			
Enter incorrect password	1. Enter random password	test	A snackbar is displayed indicating incorrect password.	A snackbar is displayed indicating incorrect password.	Pass

Table 7.4: Unit Test Case for Login to Account

Test Case No	005	Test Case Name	Add Products to Cart		
Designed by	Vignnesh Ravindran	Design date	30/03/2023		
Role	User				
Executed by	Vignnesh Ravindran	Executed date	30/03/2023		
Pre-conditions	User has successfully logged into the account.				
Summary	Test steps	Test data	Expected results	Actual Results	Status
Successfully adding products to cart	1. Select a shop type.	Groceries	Selected product(s) is added to cart.	Selected product(s) is added to cart.	Pass

from any shop in the app.	2. Select a shop from the shop list.	Parthamar			
	3. Add any product from the shop to cart by clicking the shopping cart icon.	Coca-Cola 1.5L			

Table 7.5: Unit Test Case for Add Products to Cart

Test Case No	006	Test Case Name	Place Order		
Designed by	Vignnesh Ravindran	Design date	30/03/2023		
Role	User				
Executed by	Vignnesh Ravindran	Executed date	30/03/2023		
Pre-conditions	User has successfully logged into the account and has product(s) in cart.				
Summary	Test steps	Test data	Expected results	Actual Results	Status
Successfully placing order by checking out from cart page.	1. Click “Checkout” button.	-	A new order will be created in Firebase Firestore and user will be navigated to cart page where purchased products will be deleted from cart.	A new order was created in Firebase Firestore and user was navigated to cart page where purchased products was deleted from cart.	Pass
	2. Select a delivery address.	Universiti, Tunku Abdul Rahman			
	3. Select ‘Cash on Delivery’.	Cash on Delivery			
	4. Click ‘Place Order’ button	-			

Do not select a delivery address when placing order.	1. Click 'Place Order' without selecting an address.	-	A snackbar is displayed with message "No delivery address selected".	A snackbar was displayed with message "No delivery address selected".	Pass
--	--	---	--	---	------

Table 7.6: Unit Test Case for Place Order

Test Case No	007		Test Case Name	Order Completion Notification	
Designed by	Vignnesh Ravindran		Design date	30/03/2023	
Role	User				
Executed by	Vignnesh Ravindran		Executed date	30/03/2023	
Pre-conditions	User has successfully logged into the account and has an order that is ongoing.				
Summary	Test steps	Test data	Expected results	Actual Results	Status
User receiving push notification upon order completion.	1. Keep application running in foreground.	-	An overlay dialog will be displayed to user notifying that the order was delivered with information regarding the order.	An overlay dialog was displayed to user notifying that the order was delivered with information regarding the order.	Pass
	2. Change order status value in database to 'completed'.	completed			

Table 7.7: Unit Test Case for Order Completed Notification

7.5.2 Integration Testing

Test Case No	008	Test Case Name	User Address Selection		
Designed by	Vignnesh Ravindran	Design date	30/03/2023		
Role	User				
Executed by	Vignnesh Ravindran	Executed date	30/03/2023		
Pre-conditions	User has successfully logged into the account.				
Summary	Test steps	Test data	Expected results	Actual Results	Status
User address selection using Google Places Autocomplete API.	1. Select 'Address Book' at the bottom of Profile page.	-	The provided name, mobile number, and selected address should be saved in Firebase Firestore under the user's document. Google Places Autocomplete API should function without any query errors.	The provided name, mobile number, and selected address was saved in Firebase Firestore under the user's document. Google Places Autocomplete API functioned without any query errors.	Pass
	2. Select 'Add Address'	-			
	3. Enter name	Vignnesh			
	4. Enter mobile number	+6016299999			
	5. Enter address and select relevant address.	Universiti Tunku Abdul Rahman			
	6. Click 'Add Address'	-			
Add address without filling any fields.	1. Click 'Add Address'	-	Error messages will appear under fields.	Error messages appeared under fields.	Pass

Table 7.8: Integration Test Case for User Address Selection

Test Case No	009	Test Case Name	Stripe Payment Gateway		
Designed by	Vignnesh Ravindran	Design date	30/03/2023		
Role	User				
Executed by	Vignnesh Ravindran	Executed date	30/03/2023		
Pre-conditions	User has successfully logged into the account and has product(s) in cart.				
Summary	Test steps	Test data	Expected results	Actual Results	Status
Successfully purchasing product(s) using Stripe payment gateway.	1. Select a delivery address in checkout page.	-	The application should be redirected to Stripe Payment Gateway where user can authorise the payment. Stripe API hosted in the backend should function without any errors on the console.	The application was redirected to Stripe Payment Gateway where user authorised the payment. Stripe API hosted in the backend functioned without any errors displayed on the console.	Pass
	2. Select 'FPX' as payment method.	FPX			
	3. Click 'Place Order' button	-			
Fail the test payment by not authorising it.	1. Follow above test steps.	-	An error dialog will be popped with a description about the error. User will be redirected back to the checkout page.	An error dialog was popped with a description about the error. User was redirected back to the checkout page.	Pass
	2. Click 'Fail the test payment' once redirected.	Fail test payment			

Table 7.9: Integration Test Case for Stripe Payment Gateway

Test Case No	010	Test Case Name	Email Automation with SendGrid		
Designed by	Vignesh Ravindran	Design date	30/03/2023		
Role	User				
Executed by	Vignesh Ravindran	Executed date	30/03/2023		
Pre-conditions	User has successfully logged into the account and has product(s) in cart.				
Summary	Test steps	Test data	Expected results	Actual Results	Status
Automated email sent to new user upon successful registration via Firebase Extension and SendGrid API.	1. Enter first name.	Vignesh	A new user created in the Firebase Firestore and an automated email is sent to user's email to welcome them into The Quadry application.	A new user was created in the Firebase Firestore and an automated email was sent to user's email to welcome them into The Quadry application.	Pass
	2. Enter last name.	Ravindran			
	3. Enter email.	vigneshravindran@gmail.com			
	4. Enter password	123456			
	5. Enter phone number.	+60162945500			
	6. Click 'Sign up' button.	-			

Table 7.10: Integration Test Case for Email Automation with SendGrid

7.5.3 User Acceptance Testing (UAT)

The tests listed in the table below were designed by Vignnesh Ravindran on 30/03/2023 in accordance with the testing objectives described in 7.2. They were performed by the higher management of Maurchand Ent. on the 2nd of April 2023. Results obtained for each test are marked with an 'X' symbolising if said test was a success or failure. Remarks will be given for tests that fail to indicate cause of failure.

Test Subject	Pass	Fail	Remarks
Create Account	X		
Verify Phone Number	X		
Verify Email Address	X		
Login into Account	X		
Add Products to Cart	X		
Place Order	X		
Order Completion Notification	X		
User Address Selection	X		
Stripe Payment Gateway	X		
Email Automation with SendGrid	X		

Table 7.11: User Acceptance Testing Results

7.6 Summary

To recap, this chapter summarises the testing phase of The Quadry mobile application. Among the topics discussed in this chapter were the objectives wished to be accomplished, testing scope, types of testing performed, and the results for each testing performed. It covers the test cases indicating the steps taken for testing for each unit and integration testing respectively. User Acceptance Testing was also performed by the higher management of Maurchand Ent. and the results were tabulated.

CHAPTER 8

CONCLUSION & FUTURE DEVELOPMENTS

8.1 Introduction

With solutions implemented, application developed, and testing performed, this chapter describes how the project objectives mentioned in [section 1.4](#) were attained and the possible future developments that could be performed to further improve and integrate the application with latest technologies and features.

8.2 Achieving Objectives

Objectives defined in [section 1.4](#) have been successfully achieved with studies conducted on requirements and similar applications. E-commerce aspects of mobile application development have also been covered to match the need of the project.

The first objective to analyse the project requirements and compare existing e-commerce applications was achieved by performing a thorough analysis on similar existing applications in [section 2.4](#). In this analysis, the two most used/downloaded applications (Grab Superapp & FoodPanda) were used for comparison to The Quadry in terms of architecture and features.

A model was constructed to utilise the features of an e-commerce application such as authentication, payment, order status, etc. With that, a cross-platform application called The Quadry was developed within a period of 14 weeks using the Flutter framework with Firebase as the backend service provider and Stripe as the payment gateway to manage the application's payment features. This satisfies the second and third objectives to develop a model of the e-commerce application by integrating a specific payment gateway and an e-commerce application in 14 weeks respectively.

A complete system testing was conducted on the developed application to evaluate its functionalities and all results obtained were tabulated. This fulfils the last requirement.

8.3 Future Developments

An application is never fully completed. It can be, in terms of implemented features, however, there will always be bugs to solve during maintenance and further upgrades for enhancement purposes. The following are the future developments of The Quadry application, making it a full scaled e-commerce application equipped with all the necessary technologies and features:

- a. A Frequently Asked Questions (FAQ) page. This will be implemented in the near future to ensure that users can refer or get the answers to any questions that they may have. The benefit of this implemented includes reducing the burden placed on the support team as the answers already exists in this page.
- b. A support chatbot. Chatbots serve the purpose of assisting with users' queries which ultimately also reduces the burden of the support team as well as eliminating the organisation's effort to gather/hire a large support team. Time and resources spent by the organisation can be saved with this implementation.
- c. Settings feature. Every e-commerce application consists of the settings feature where users will be able to control various types of functionalities within the application itself. For example, a user frustrated with the constant notifications of promotions by merchant in the application, can choose to turn off the push notifications in the settings page.
- d. Continuous Integration / Continuous Delivery (CI/CD) workflow. Currently there is no CI/CD set up for the application which helps with the continuous upgrades or implementation of new features. Setting up a CI/CD workflow helps the development team to push new features and change in a more frequent, efficient, and optimised way.

8.4 Summary

This chapter summarises the implementation of The Quadry application and how it resonates with the well achieved project objectives. Besides that, future development plans of the project are also briefly specified.

REFERENCES

- Alvi, S., Laila, U., Khan, K. and Hussainy, S.K., 2016. Intention to Adopt M-Commerce over E-Commerce. *KASBIT Business Journal*, 9(1), pp.154-175.
- Bakos, Y., 2001. The emerging landscape for retail e-commerce. *Journal of economic perspectives*, 15(1), pp.69-80.
- Blair, I., 2022. 10 Mobile Commerce Trends that will dominate 2022 - BuildFire Tech talks. *BuildFire*. Available at: <https://buildfire.com/mobile-commerce-trends/> [Accessed August 28, 2022].
- Bolilia, J., Basseville, R. & Kue, K., 2019. Making grab's everyday app super. *Grab Tech*. Available at: <https://engineering.grab.com/grab-everyday-super-app> [Accessed July 25, 2022].
- Chan, M., 2017. [update] does Foodpanda Msia secretly hike up prices of their food? read this before you order. *CILISOS*. Available at: <https://cilisos.my/foodpandas-menu-prices-are-higher-than-the-actual-restaurant-menu-prices-but-why/#update> [Accessed July 22, 2022].
- Chien, C., 2020. What is rapid application development (RAD)? *Codebots*. Available at: <https://codebots.com/app-development/what-is-rapid-application-development-rad> [Accessed July 25, 2022].
- Digital Marketing Institute. (2018, July 17). What's the cost of not going digital for a business? Retrieved May 17, 2023, from <https://digitalmarketinginstitute.com/blog/what-is-the-cost-of-not-going-digital-for-a-business> [Accessed May 17, 2023].
- Dominic, B., 2021. The evolution of digitalization and its impact on the ecommerce industry. *Cogneesol Blog*. Available at: <https://www.cogneesol.com/blog/how-digitalization-impact-on-ecommerce-industry/> [Accessed July 22, 2022].
- Grab, n.d.. Request to update menu item/price. *Grab HelpCenter*. Available at: <https://help.grab.com/merchant/en-my/360027910431-Request-to-update-menu-itemprice> [Accessed July 22, 2022].
- Grzmił, P., Skublewska-Paszkowska, M., Łukasik, E. and Smółka, J., 2017. Performance analysis of native and cross-platform mobile applications. *Informatyka, Automatyka, Pomiary w Gospodarce i Ochronie Środowiska*, 7.
- Koetsier, J., 2021. Covid-19 accelerated e-commerce growth '4 to 6 years'. *Forbes*. Available at: <https://www.forbes.com/sites/johnkoetsier/2020/06/12/covid-19-accelerated-e-commerce-growth-4-to-6-years/?sh=72452c4d600f> [Accessed July 24, 2022].
- Loesche, D. & Richter, F., 2018. Infographic: Mobile e-commerce is up and poised for further growth. *Statista Infographics*. Available at:

<https://www.statista.com/chart/13139/estimated-worldwide-mobile-e-commerce-sales/> [Accessed July 24, 2022].

LucidContentTeam, 2018. 4 phases of Rapid Application Development Methodology. *4 Phases of Rapid Application Development Methodology / Lucidchart Blog*. Available at: <https://www.lucidchart.com/blog/rapid-application-development-methodology> [Accessed July 25, 2022].

McCallum, A. et al., 2022. How grab built a scalable, high-performance AD server. *Grab Tech*. Available at: <https://engineering.grab.com/scalable-ads-server> [Accessed July 25, 2022].

Nawrocki, P., Wrona, K., Marczak, M. and Sniezynski, B., 2021. A comparison of native and cross-platform frameworks for mobile applications. *Computer*, 54(3), pp.18-27.

Nguyen, D.D. and Kira, D., 2000. Trust and quality assurance in business-to-consumer electronic commerce: Enhancing consumer acceptance and participation.

Nikolaieva, A., 8 best software development methodologies. *8 Best Software Development Methodologies*. Available at: <https://www.uptech.team/blog/software-development-methodologies#:~:text=Software%20development%20methodology%20refers%20to,systematic%20approach%20to%20software%20development.> [Accessed August 21, 2022].

Niranjanamurthy, M., Kavyashree, N., Jagannath, S. and Chahar, D., 2013. Analysis of e-commerce and m-commerce: advantages, limitations and security issues. *International Journal of Advanced Research in Computer and Communication Engineering*, 2(6), pp.2360-2370.

Omonedo, P. and Bocij, P., 2014. E-commerce versus m-commerce: Where is the dividing line. *International Journal of Social, Behavioural, Educational, Business and Industrial Engineering*, 8(11), pp.3610-3615.

Payaro, A. and Papa, A.R., 2021. Online Grocery Brick and Mortar Retailers. State of the Art in Italy, But Is It a Profitable Strategy?. *Management*, 9(1), pp.14-23.

Pinto, C.M. and Coutinho, C., 2018, September. From native to cross-platform hybrid development. In 2018 International Conference on Intelligent Systems (IS) (pp. 669- 676). IEEE.

Richter, F., 2017. Infographic: Mobile Payment Volume to increase tenfold by 2021. *Statista Infographics*. Available at: <https://www.statista.com/chart/7793/mobile-payment-transaction-volume/> [Accessed August 28, 2022].

Samsukha, A., 2022. Ecommerce Mobile App Development Guide – Cost and features. *Tech Blog / Mobile App, eCommerce, Salesforce Insights*. Available at: <https://www.emizentech.com/blog/e-commerce-mobile-app-development.html> [Accessed July 24, 2022].

Shopify, n.d.. Ecommerce definition - what is Ecommerce. *Shopify*. Available at: <https://www.shopify.com/encyclopedia/what-is-ecommerce> [Accessed July 24, 2022].

Steinfeld, C., Adelaar, T. and Lai, Y.J., 2002, January. Integrating brick and mortar locations with e-commerce: understanding synergy opportunities. In *Proceedings of the 35th Annual Hawaii International Conference on System Sciences* (pp. 2920-2929). IEEE.

Subramaniam, P., 2021. Delivery apps: Foodpanda wants to use drones to expand on-demand delivery to rural Malaysia. *The Edge Markets*. Available at: <https://www.theedgemarkets.com/article/delivery-apps-foodpanda-wants-use-drones-expand-ondemand-delivery-rural-malaysia> [Accessed July 25, 2022].

Surendran, S., 2018. Cover story: Online retailers see healthy growth. *The Edge Markets*. Available at: <https://www.theedgemarkets.com/article/cover-story-online-retailers-see-healthy-growth> [Accessed July 23, 2022].

TheFutureofCommerce, 2022. Consumer behavior is changing drastically: See the e-commerce data. *The Future of Customer Engagement and Experience*. Available at: <https://www.the-future-of-commerce.com/2020/05/13/covid-19-e-commerce-data/> [Accessed July 24, 2022].

Tutorialspoint, n.d.. SDLC - Rad Model. *Tutorials Point*. Available at: https://www.tutorialspoint.com/sdlc/sdlc_rad_model.htm [Accessed July 23, 2022].

Wu, D., Ray, G., Geng, X. and Whinston, A., 2004. Implications of reduced search cost and free riding in e-commerce. *Marketing Science*, 23(2), pp.255-262.

Wu, J.H., Li, Q. and Kee Wei, K., 2016. Alibaba's IT platform and electronic commerce synergy in driving "Singles' Day". *Journal of Organizational Computing and Electronic Commerce*, 26(3), pp.193-202.