

**M.S.G. Hettiarachchi**

2023175

**Packagely: Delivery Application**

BSc (Hons) Computer Science and Software Engineering

Undergraduate Thesis Report

Department of Computer Science and Technology

University of Bedfordshire

Mr.Roshan Jayawardhana

**AY October 2021**

# Abstract

Packagely is a mobile application mainly focus on overcoming the challenges when transporting goods from one place to another. This app is implemented with a vision to benefit both sellers and the customer. The contrast of this application lies in its specialty of creating an opportunity for people to earn money by couriering goods from one place to another while traveling. Commonly, many courier services provide logistics from their office to a specific location. The main downfall of this system is most courier services are operating city-based, so a sender needs to hand out the item by going into a courier service center. The main objective of Packagely is to resolve this by centralizing all courier services into a hub, which facilitates people to get all the things done using one app. The main advantage of using Packagely is that it picks the goods from the customer's location and delivers them to the selected location. Also, courier services and anyone who has a vehicle and tends to do the service while traveling can register to Packagely. They have the pleasure to decide whether to work full-time or part-time. It helps redeem their transport costs, making it a win-win situation for everyone. Also, courier services have no choice but to send a vehicle even for a single parcel destined to a rural area, making it inefficient and costly for the company. With Packagely, courier services can cooperate and send only one vehicle to remote places, drastically saving their traveling costs and reducing the delivery charges for the customer.

# Acknowledgment

I would like to express my gratitude to my supervisor, Mr. Roshan Jayawardhana, who guided me throughout this project even if the road got tough. His enthusiasm, patience, insightful comments, helpful information, practical advice, and unceasing ideas have helped me tremendously in my research. Without his support and guidance, this project would not have been possible.

Next, I would like to thank Dr. Gayana Fernando, the lecturer in charge, for teaching the module accurately and guiding me to the appropriate path to make my project a success.

Also, I am taking this chance to thank the UoB evaluators for guiding me through this project and evaluating me from the beginning to the end.

Finally, I would like to thank all my other colleagues, especially Roshana, Pasan, and Chandritha, along with the lecturers and staff members in SLIIT Academy, who supported me in this journey of success to my project.

# Dedication

In the entirety of my scholarly exercises, this paper is devoted to my caring family for their sincere good support and trust in my ability to carry forward with the course content. And I would like to mention my colleagues and whoever supported me and encouraged me throughout this project.

# Abbreviations

|  |  |
| --- | --- |
| **Abbreviation** | **Full Form** |
| API | Application Programming Interface |
| GPS | Global Possitioning System |
| UI | User Interface |
| APK | Application Programming Interface |
| SDK | Software Development KIT |
| UML | Unified Modified Language |
| Ai | Artificial Intelligence |
| IOS | Iphone Operating System |

# Glossary

|  |  |
| --- | --- |
| **Name** | **Definition** |
| Flutter | The UI development language used by the developer in order to attain the UI functionalities in the artefact |
| Dart Language | Dart implementation for asynchronous API streams development, which makes use of popular reactiveX. |
| Scoped Model | A bunch of utilities that permit you to handily pass an information Model from a parent Widget down to its descendants. Furthermore, it additionally modifies the entirety of the child classes that utilize the model when the model is refreshed. |
| Geolocation | identification of the geographic location of a user or computing device via a variety of data collection mechanisms. |

**Table of Contents**

[Abstract i](#_Toc83064514)

[Acknowledgment ii](#_Toc83064515)

[Dedication iii](#_Toc83064516)

[Abbreviations iv](#_Toc83064517)

[Glossary v](#_Toc83064518)

[1.1 Project Background 1](#_Toc83064519)

[1.2 Aims and Objectives 2](#_Toc83064520)

[1.2.1 Aims 2](#_Toc83064521)

[1.2.2 Objectives 2](#_Toc83064522)

[1.3 Research 3](#_Toc83064523)

[1.3.1 Objectives in Research 3](#_Toc83064524)

[1.3.2 How Research Helped to the Project 3](#_Toc83064525)

[1.4 Description of the Artefact 4](#_Toc83064526)

[1.5 Structure of the Report 5](#_Toc83064527)

[1.5.1 Introduction 5](#_Toc83064528)

[1.5.2 Literature Review & Market Research 5](#_Toc83064529)

[1.5.3 Methodology 5](#_Toc83064530)

[1.5.4 Artefact Design and Testing 5](#_Toc83064531)

[1.5.5 Discussion & Evaluation 5](#_Toc83064532)

[1.5.6 Conclusion & Future Work 6](#_Toc83064533)

[2 Literature Review & Market Research 7](#_Toc83064534)

[2.1 Introduction 7](#_Toc83064535)

[2.2 Courier Services and Delivery Applications 8](#_Toc83064536)

[2.2.1 Courier service 8](#_Toc83064537)

[2.2.2 Delivery applications 9](#_Toc83064538)

[2.3 Main Problems in Delivery and Courier Services 10](#_Toc83064539)

[2.4 Similar Applications 11](#_Toc83064540)

[2.4.1 Parcel2Go 11](#_Toc83064541)

[2.4.2 JNE 12](#_Toc83064542)

[2.5 Google Maps Application Programming Interface (API) 13](#_Toc83064543)

[2.5.1 Maps SDK 13](#_Toc83064544)

[2.5.2 Places SDK 13](#_Toc83064545)

[2.5.3 Global Positioning System (GPS) 14](#_Toc83064546)

[2.5.4 The Global System for Mobile Communication (GSM) 14](#_Toc83064547)

[2.5.5 Geolocation API 14](#_Toc83064548)

[2.6 Finding the Shortest and easiest route 15](#_Toc83064549)

[2.6.1 Similar Applications 15](#_Toc83064550)

[2.6.2 Saving Matrix 16](#_Toc83064551)

[2.7 Research Gap 17](#_Toc83064552)

[2.8 Market Research 18](#_Toc83064553)

[2.8.1 Questionnaire Results Summary 18](#_Toc83064554)

[2.8.2 Interview with a Courier Service Owner 23](#_Toc83064556)

[2.8.3 Interview with a Delivery Person 24](#_Toc83064557)

[3 Methodology and Artifact Design 25](#_Toc83064558)

[3.1 Software Development Methodology 25](#_Toc83064559)

[3.2 Requirement Gathering and Analysis 26](#_Toc83064560)

[3.3 Design 27](#_Toc83064561)

[3.3.1 UML Diagrams 27](#_Toc83064562)

[4 Artefact Design & Testing 31](#_Toc83064563)

[4.1 Artefact Design. 31](#_Toc83064564)

[4.1.1 User Account Management. 31](#_Toc83064565)

[4.1.2 Select Location 34](#_Toc83064566)

[4.1.3 Make a Delivery 35](#_Toc83064567)

[4.1.4 View Orders 37](#_Toc83064568)

[4.1.5 Courier Home Page 38](#_Toc83064569)

[4.1.6 Courier View Orders 39](#_Toc83064570)

[4.1.7 Find a Courier 43](#_Toc83064571)

[4.1.8 Add Destination 45](#_Toc83064572)

[4.2 Testing 46](#_Toc83064573)

[4.2.1 Unit Testing 46](#_Toc83064574)

[4.2.2 Integration Testing 46](#_Toc83064575)

[4.2.3 Acceptance Testing 46](#_Toc83064576)

[4.3 Test Cases 47](#_Toc83064577)

[4.3.1 User Application 47](#_Toc83064578)

[4.3.2 Courier Application 55](#_Toc83064579)

[5 Evaluation & Discussion 65](#_Toc83064580)

[5.1 Evaluation 65](#_Toc83064581)

[5.1.1 Questionnaire 65](#_Toc83064582)

[5.2 Discussion 71](#_Toc83064583)

[5.2.1 Accuracy and Reliability 71](#_Toc83064584)

[5.2.2 Reasons for the Levels and Objectives 71](#_Toc83064585)

[5.2.3 Overcoming Problems 72](#_Toc83064586)

[6 Conclusion and Future Work 73](#_Toc83064587)

[6.1 Conclusion 73](#_Toc83064588)

[6.1.1 Benefits of the Artefact 73](#_Toc83064589)

[6.1.2 Limitations 74](#_Toc83064590)

[6.1.3 Recommendations 74](#_Toc83064591)

[6.2 Future Work 75](#_Toc83064592)

[References 76](#_Toc83064593)

[Appendix 79](#_Toc83064594)

[Appendix A Gantt Chart 79](#_Toc83064595)

[Appendix B Work Breakdown Structure 80](#_Toc83064596)

[Appendix C Packagely Project Poster 81](#_Toc83064597)

[Appendix D Beta Evaluation Survey 82](#_Toc83064598)

[Appendix E Market Research 84](#_Toc83064600)

**Table of Figures**

[Figure 2‑1: Overall satisfaction among Zomato ,Swiggy and Uber eats (Ashish & Varinder & Dr.Arun, 2018) 9](file:///D:\SLIIT%20A\UP\Thesis%20Report%20Final%20(1).docx#_Toc83056171)

[Figure 2‑2: Interface of Parcel2Go Website 11](file:///D:\SLIIT%20A\UP\Thesis%20Report%20Final%20(1).docx#_Toc83056172)

[Figure 2‑3: Interfaces of Parcel2go Mobile Application 11](file:///D:\SLIIT%20A\UP\Thesis%20Report%20Final%20(1).docx#_Toc83056173)

[Figure 2‑4: JNE Website 12](file:///D:\SLIIT%20A\UP\Thesis%20Report%20Final%20(1).docx#_Toc83056174)

[Figure 2‑5: JNE Mobile Application Interfaces 12](file:///D:\SLIIT%20A\UP\Thesis%20Report%20Final%20(1).docx#_Toc83056175)

[Figure 2‑6:Bloc diagram of proposed methodology (Monika & Sudha, 2015) 15](#_Toc83056176)

[Figure 2‑7: image of the ground and overaid graph (Elias,2018) 15](file:///D:\SLIIT%20A\UP\Thesis%20Report%20Final%20(1).docx#_Toc83056177)

[Figure 2‑8: Image of a small town n Google Maps(Elias,2018) 15](file:///D:\SLIIT%20A\UP\Thesis%20Report%20Final%20(1).docx#_Toc83056178)

[Figure 2‑9: Saving Matrix (Timothy & Eddy & David, 2018) 16](file:///D:\SLIIT%20A\UP\Thesis%20Report%20Final%20(1).docx#_Toc83056179)

[Figure 2‑10: Flowchart Sorting & Selected Route (Timothy & Eddy & David, 2018) 16](file:///D:\SLIIT%20A\UP\Thesis%20Report%20Final%20(1).docx#_Toc83056180)

[Figure 2‑11: Courier service Usage 18](#_Toc83056181)

[Figure 2‑12: How much customers satisfied with current courier service providers 19](#_Toc83056182)

[Figure 2‑13: User's difficulties when using a courier service 19](#_Toc83056183)

[Figure 2‑14: How many users have received a package from any delivery method 20](#_Toc83056184)

[Figure 2‑15: Difficulties when receiving a package from a courier service 20](#_Toc83056185)

[Figure 2‑16:How many users like to use a mobile application 21](#_Toc83056186)

[Figure 2‑17: How many users like to deliver a package on their router 21](#_Toc83056187)

[Figure 2‑18: How many people like to deliver on their spare time 22](#_Toc83056188)

[Figure 2‑19: How many people like to work as delivery rider 22](#_Toc83056189)

[Figure 2‑20: Suggestions 23](#_Toc83056190)

[Figure 3‑1: Agile Methodology (Anon, 2019) 25](file:///D:\SLIIT%20A\UP\Thesis%20Report%20Final%20(1).docx#_Toc83056191)

[Figure 3‑2: UseCase Diagram 28](#_Toc83056192)

[Figure 3‑3: Architecture Diagram 29](file:///D:\SLIIT%20A\UP\Thesis%20Report%20Final%20(1).docx#_Toc83056193)

[Figure 3‑4: Layered Architecture Diagram 30](#_Toc83056194)

[Figure 4‑1: Login 32](file:///D:\SLIIT%20A\UP\Thesis%20Report%20Final%20(1).docx#_Toc83056195)

[Figure 4‑2: Signup 33](file:///D:\SLIIT%20A\UP\Thesis%20Report%20Final%20(1).docx#_Toc83056196)

[Figure 4‑3: Select Destination 34](file:///D:\SLIIT%20A\UP\Thesis%20Report%20Final%20(1).docx#_Toc83056197)

[Figure 4‑4: Select Location 34](#_Toc83056198)

[Figure 4‑5: Make a delivery 35](file:///D:\SLIIT%20A\UP\Thesis%20Report%20Final%20(1).docx#_Toc83056199)

[Figure 4‑6: Order Details 37](file:///D:\SLIIT%20A\UP\Thesis%20Report%20Final%20(1).docx#_Toc83056200)

[Figure 4‑7: Courier Dashboard 38](file:///D:\SLIIT%20A\UP\Thesis%20Report%20Final%20(1).docx#_Toc83056201)

[Figure 4‑8: View Orders 40](file:///D:\SLIIT%20A\UP\Thesis%20Report%20Final%20(1).docx#_Toc83056202)

[Figure 4‑9:Manage Order 41](file:///D:\SLIIT%20A\UP\Thesis%20Report%20Final%20(1).docx#_Toc83056203)

[Figure 4‑10: Shortest route and distance 42](file:///D:\SLIIT%20A\UP\Thesis%20Report%20Final%20(1).docx#_Toc83056204)

[Figure 4‑11: Find a courier 1 43](file:///D:\SLIIT%20A\UP\Thesis%20Report%20Final%20(1).docx#_Toc83056205)

[Figure 4‑12: Find a courier 2 43](file:///D:\SLIIT%20A\UP\Thesis%20Report%20Final%20(1).docx#_Toc83056206)

[Figure 4‑13: Add Destination 45](file:///D:\SLIIT%20A\UP\Thesis%20Report%20Final%20(1).docx#_Toc83056207)

[Figure 4‑14: Delivery UI 47](file:///D:\SLIIT%20A\UP\Thesis%20Report%20Final%20(1).docx#_Toc83056208)

[Figure 4‑15: Login Interface 47](file:///D:\SLIIT%20A\UP\Thesis%20Report%20Final%20(1).docx#_Toc83056209)

[Figure 4‑16: User login 48](file:///D:\SLIIT%20A\UP\Thesis%20Report%20Final%20(1).docx#_Toc83056210)

[Figure 4‑17: Login interface with empty fields 49](file:///D:\SLIIT%20A\UP\Thesis%20Report%20Final%20(1).docx#_Toc83056211)

[Figure 4‑18: Signup with Valid details 50](file:///D:\SLIIT%20A\UP\Thesis%20Report%20Final%20(1).docx#_Toc83056212)

[Figure 4‑19: Make a delivery 51](file:///D:\SLIIT%20A\UP\Thesis%20Report%20Final%20(1).docx#_Toc83056213)

[Figure 4‑20: Select a location 52](file:///D:\SLIIT%20A\UP\Thesis%20Report%20Final%20(1).docx#_Toc83056214)

[Figure 4‑21: Select a destination 53](file:///D:\SLIIT%20A\UP\Thesis%20Report%20Final%20(1).docx#_Toc83056215)

[Figure 4‑22: View Orders 54](file:///D:\SLIIT%20A\UP\Thesis%20Report%20Final%20(1).docx#_Toc83056216)

[Figure 4‑23: Courier Dashboard 55](file:///D:\SLIIT%20A\UP\Thesis%20Report%20Final%20(1).docx#_Toc83056217)

[Figure 4‑24: Courier Login 55](file:///D:\SLIIT%20A\UP\Thesis%20Report%20Final%20(1).docx#_Toc83056218)

[Figure 4‑25: Courier login with incorrect details 56](file:///D:\SLIIT%20A\UP\Thesis%20Report%20Final%20(1).docx#_Toc83056219)

[Figure 4‑26: Courier login with empty details 57](file:///D:\SLIIT%20A\UP\Thesis%20Report%20Final%20(1).docx#_Toc83056220)

[Figure 4‑27: Add destination 58](file:///D:\SLIIT%20A\UP\Thesis%20Report%20Final%20(1).docx#_Toc83056221)

[Figure 4‑28: Find a Courier - Show details 59](file:///D:\SLIIT%20A\UP\Thesis%20Report%20Final%20(1).docx#_Toc83056222)

[Figure 4‑29: Find a courier - addind details 59](file:///D:\SLIIT%20A\UP\Thesis%20Report%20Final%20(1).docx#_Toc83056223)

[Figure 4‑30: View Orders 60](file:///D:\SLIIT%20A\UP\Thesis%20Report%20Final%20(1).docx#_Toc83056224)

[Figure 4‑31: Manage Orders - Options 61](file:///D:\SLIIT%20A\UP\Thesis%20Report%20Final%20(1).docx#_Toc83056225)

[Figure 4‑32: Navigate 62](file:///D:\SLIIT%20A\UP\Thesis%20Report%20Final%20(1).docx#_Toc83056226)

[Figure 4‑33: Cancel order 63](file:///D:\SLIIT%20A\UP\Thesis%20Report%20Final%20(1).docx#_Toc83056227)

[Figure 4‑34: View Orders 64](file:///D:\SLIIT%20A\UP\Thesis%20Report%20Final%20(1).docx#_Toc83056228)

[Figure 5‑1: Beta Evaluation Question 1 results 65](#_Toc83056229)

[Figure 5‑2: Beta evaluation question 2 results 66](#_Toc83056230)

[Figure 5‑3: Beta evaluation question 3 results 67](#_Toc83056231)

[Figure 5‑4: Beta Evaluation question 4 67](#_Toc83056232)

[Figure 5‑5: Beta evaluation question 5 68](#_Toc83056233)

[Figure 5‑6: Beta evaluation question 6 results 68](#_Toc83056234)

[Figure 5‑7: Beta Evaluation question 7 part 1 69](#_Toc83056235)

[Figure 5‑8: Beta evaluation question 7 results part2 70](#_Toc83056236)

[Figure 9: Gantt Chart 79](#_Toc83056237)

[Figure 10: Work Breakdown Structure 80](#_Toc83056238)

[Figure 69: Packagely Project Poster 81](file:///D:\SLIIT%20A\UP\Thesis%20Report%20Final%20(1).docx#_Toc83056239)

[Figure 12: Beta Evaluation A 82](#_Toc83056240)

[Figure 13: Beta Evaluation B 83](#_Toc83056241)

**Table of Tables**

[Table 3: Comparison of the proposed system with existing systems 17](#_Toc83055323)

[Table 4: Log-in interface Test case 47](#_Toc83055324)

[Table 5: Login Interface 2 48](#_Toc83055325)

[Table 6: Login Interface 49](#_Toc83055326)

[Table 7: Signup Interface 50](#_Toc83055327)

[Table 8: Make a delivery 51](#_Toc83055328)

[Table 9 Make a deliver - Select a location: 52](#_Toc83055329)

[Table 10: Select Destination 53](#_Toc83055330)

[Table 11: View Orders 54](#_Toc83055331)

[Table 12: Courier Logiin 55](#_Toc83055332)

[Table 13: Courier Login 2 56](#_Toc83055333)

[Table 14: Courier login with invalid details 57](#_Toc83055334)

[Table 15: Add Destination 58](#_Toc83055335)

[Table 16: Find a Courier 59](#_Toc83055336)

[Table 17: Manage Orders 60](#_Toc83055337)

[Table 18: Manage orders - Option 61](#_Toc83055338)

[Table 19: Manage Orders - Navigate 62](#_Toc83055339)

[Table 20: Manage orders - Cancel the order 63](#_Toc83055340)

[Table 21: Manage orders - Mark order completed 64](#_Toc83055341)

**Table of Code Snippets**

[Code Snippet 1: Sign up Authentication 33](#_Toc82984863)

[Code Snippet 2: Getting users location 35](#_Toc82984864)

[Code Snippet 3: Passing data to database 36](#_Toc82984865)

[Code Snippet 4: Data access from firebase 37](#_Toc82984866)

[Code Snippet 5: Get Courier Name 39](#_Toc82984867)

[Code Snippet 6: Getting shortest route with distance 42](#_Toc82984868)

[Code Snippet 7: Finding a courier 44](#_Toc82984869)

**1 Introduction**

The introduction chapter describes the project background, aims, and objectives of the system, a brief research description about the artefact, and the structure of the thesis report.

## Project Background

Transportation has become an essential need for humans over the years. Delivering packages from one location into another location is one of the main parts of transportation. People use different logistics methods, where couriering has become the primary method of them all. There are numerous courier services available domestically and internationally. Packagely mobile application centralizes all courier services, transforming it into a service that picks up and drops off to any given location. Then the customer can get an idea of how much it costs them before they courier their packages. Currently, no application has surpassed these limitations and connects all the courier services as a hub. Packagely will help customers and the courier service owners, creating more job opportunities to work as part-time delivery personnel for the public.

Most courier services are only operational at their respective company's premises or branches. With specific terms and conditions only, customers can enjoy the door-to-door delivery service. The pricing information is hidden quite often, making the customers contact the courier company via telephone/email or visit their respective websites or location to get a quote. Only then can the customer decide on the service they will proceed with after receiving all the quotations. This process could be very time-consuming, mainly due to the inflexibility of services. And in most cases, many courier services don't afford customer package pickups, leaving the customers to hand over the packages to one of their warehouses, which becomes a timely and annoying process. Usually, courier services are stuck with sending a vehicle to remote areas to deliver just one package, which is costly for the company.

Packagely surpasses all these flaws. It acts as a hub for courier services so customers can choose from what courier service they proceed to courier their package. Not only users but courier services can profit by communicating with other courier services with this application, which enables them to cooperate when delivering packages to rural areas.

## Aims and Objectives

### Aims

* This project's primary aim is to develop an Android-based application that connects customers to courier services and upholding customer satisfaction by providing the service for an unbelievable price.
* Aimed to create a courier service hub by centralizing all courier services and letting them contact each other, they can reduce the cost and reduce the price of delivery charge.

### Objectives

* To identify the problems around delivery.
* To identify the solutions for these problems.
* To identify a suitable platform for development and testing.
* To develop an algorithm that can find the shortest delivery route.
* Connect courier services.
* Find another courier service that is going to the same area.
* Implement to find courier services and delivery persons.
* To identify solutions for the problems identified.
* To identify people's opinions on this app.
* To compare the application with other similar applications identified.
* Make a user-friendly application.

## Research

### Objectives in Research

* Find related research articles.
* Find newest research articles.
* Find problems around delivery.
* Find solutions implemented up to now.
* Find that it is worth it to implement this type of application.
* Find what the problems in courier services are.
* The audience for the market research.
* Find a courier service for market research.
* Find a delivery rider for market research.

### How Research Helped to the Project

Because of the research, the proposed application has shifted. There are several reasons for it,

* Research articles
* Problems around delivery
* Audience opinions in market research
* Interviews in market research

The research helped find similar applications to Packagely, and most of the functions in Packagely are non-existent in these applications.

Market Research helped identify the user's opinions about Packagely mobile application, what they expect from this kind of application, what they need to focus more on, and the opinion of Packagely application in the eyes of courier services and delivery riders.

Research Gap led to finding what the similarities are and which functions highlight Packagely from other existing delivery applications

## Description of the Artefact

Packagely is a centralized hub for courier services, and customers can choose from a variant of courier services and proceed with the one that fulfills their expectations. Not only users but courier services can profit by communicating with other courier services with this application, which enables them to cooperate when delivering packages to rural areas. The application is based on geolocation and courier services. The geolocation allows the users to find out the location of drivers, for the driver's application helps find the easiest route.

The other main function of this application is that courier services can also deliver their packages by partnering with another courier service. And courier services can add their destinations then that other courier services can make partnership with them and send their packages also with this application. It is a win-win situation for both courier services because the courier service that sends a vehicle can profit from the additional packages. Other courier services can profit by saving their delivery costs.

Other than that, to cater to both users and courier services, the application is integrated with two account types. Namely, User accounts and Courier accounts. The two accounts present different functionalities for the registered users of the respective type. For users: it allows them to make an order, manually find a courier service or select a courier service suggested by the app, view the location, and edit their user profiles. For the courier services: it allows them to showcase their prices, check their orders, check their balance, enable them to add a destination and date to check if any other courier services are likely to make a partnership and deliver their package for a reasonable price. Courier services can add an area with time to contact other courier services for a partnership.

The artefact was developed in an environment with constant updates and integrations. JAVA language was mainly used, which integrates best-in-class libraries and dependencies, providing a complete, clean outlook and unique functionality to the artefact. The artefact database is maintained using Google Firebase Realtime Database, allowing the developer to access Google Authentication.

The artifact includes a user manual, installation guide, source code, and project poster.

## Structure of the Report

The document comprises six main chapters: introduction, Literature review, Methodology, Discussion and Testing, Evaluation of the project, and conclusion and future works in this project.

### Introduction

The introduction contains a rundown of the project's history, followed by a brief listing priority in development, describes how research helped for the project, and finally comes to an end with the report layout, which will allow the reader to identify the content yet to come.

### Literature Review & Market Research

The literature chapter describes all the research done by the researcher to get an idea of what approach is suitable for the proposed application and what other researchers have done in the same field with different techniques. Furthermore, this chapter focuses on the questionnaire, interviews, and the results obtained.

### Methodology

This chapter is explained with UML diagrams following up with processes and procedures related to them. Ultimately, the chapter will conclude with a test schedule, conducted tests, and their outcomes. Here the developer will present the UML-related workings related to each and every step of development, allowing the readers to understand the build-up of the project.

### Artefact Design and Testing

This chapter will include photographs of UI elements and implementation of the application with backend codes segments. Further, this chapter will present the tests and the test outcomes achieved at the end of the testing phase of the artefact.

### Discussion & Evaluation

This chapter discusses the functionalities accomplished during the project's life cycle. The evaluation illustrates how the application is evaluated by presenting it to real users to get the application's feedback. The results will be summarized and presented along with the discussions which explain the difficulties faced by the researcher when developing the application.

### Conclusion & Future Work

Conclusion and future work chapter describes the project benefits, limitations of the project, recommendations, and the future work.

# Literature Review & Market Research

This literature review focuses on similar exciting systems, technologies, methodologies, and algorithms used in other systems alongside describing the perks and flaws of those systems. This literature review will give an idea of whether the proposed approach is technically feasible and market research which the researcher does, explained with figures and descriptions.

## Introduction

The primary purpose of this literature review is to give a basic idea of how far the delivery applications solve the delivery problems up to date, similar applications to the proposed system, issues that need resolving in the delivery service area, methods to overcome the issues mentioned earlier. Moreover, this review gives vast knowledge about whether the proposed system is technically feasible. Therefore, this literature review covers delivery applications, courier applications, google maps application programming (API), and market research separately. Facts on research papers and articles are used frequently throughout this review to supplement the reader's knowledge on this subject. These are described clearly with information, images, and diagrams with a brief outline of similar systems. Finally, with research gap can identify what the similarities are and which functions highlighting Packagely application.

The following literature review will provide a brief description of each technological aspect while giving the reader an insight into its adaptation to the proposed application.

## Courier Services and Delivery Applications

### Courier service

Couriers are distinguished from ordinary mail services by features such as speed, security, tracking, signature, specialization, and individualization of express services, and swift delivery times, which are optional for most everyday mail services. As a premium service, couriers are usually more expensive than standard mail services. Their use is normally limited to packages where one or more of these features are considered important enough to warrant the cost. Various factors must be considered when running a courier service in an urban area because a city's infrastructure differs from those in suburban or countryside areas. Of note, population density is higher, and vehicles encounter greater restrictions. Moreover, air pollution from fossil fuel combustion is more severe. As tailpipe emissions are becoming costly to both corporations and the environment, researchers are increasingly exploring more appealing transportation options. (Keiju & Junjae & Jinwoo, 2019).

The business landscape in the courier services industry has a very significant change due to its high and rapid growth of e-commerce. "The tight competition between companies in the courier service industry can greatly affect employees' work-life balance, especially for "Y" Generation, those born between 1979 and 1994." (Setyo\*& Eny & Lukertina, 2019). Recently, the courier, express, and parcel market (CEP) has developed dynamically worldwide. "For example, in Poland, in the period 2000–2015, the number of registered courier operators increased by 1400%" (Aleksandra Gulc,2017)

#### FedEx

FedEx is one of the best and famous courier services in the world. On April 19, 1973, FedEx was born with a small group of people who stood around a makeshift system and sorted 186 packages. Then they loaded onto 14 small planes that promptly took off for 25 cities across the United States (Maden Birla, 2016). FedEx mainly delivers packages from one country to another. Figure 2-1- Overall Satisfaction (Ashish & Varinder & Dr. Arun Singh, 2018). They have dealers stationed whom we have to hand over packages that we need to deliver.

There are many courier services similar to FedEx like DHL, UPS, etc. And when we need to provide our package, we'll have to use a local delivery service most of the time.

### Delivery applications

Developments in technology and reliance on delivery have pitched a new pathway for marketing through mobile applications (Ashish & Varinder & Dr. Arun Singh, 2018). Delivery is the process of transporting goods from a source location to a predefined destination. This service has leaped to vast development over the years, and as a result, we can order various products to our doorstep or a given location today by using our mobile phone. Delivery services mainly use roads and railroads on land and use shipping lanes on the sea and airline networks in the air to deliver things from one country to another. Delivery of goods to the consumer is known as delivery service, postal, courier, and relocation services, also delivering of goods for commercial and private interests.

#### Zomato

Zomato is the most popular food delivery application in Ludhiana among the Swiggy, Uber eats. Considering customer satisfaction, most of them have chosen Zomato from their survey (Ashish & Varinder & Dr. Arun Singh, 2018). They did their survey targeting the most active service, the service availing maximum offers, and which service has the best customer service, etc. "Service sector in India is dependent on the Hospitality industry as it has emerged as one of the main industries in India in 21st century. The needs and desires of the people have been taken care of by the hospitality industry over time" (Prajogo & Sohal, 2006). Below figure shows overall satisfaction among Zomato,Swiggy and Uber Eats in Ludhiana.



Figure 2‑1: Overall satisfaction among Zomato ,Swiggy and Uber eats (Ashish & Varinder & Dr.Arun, 2018)

## Main Problems in Delivery and Courier Services

There are numerous courier services available locally or worldwide. However, most services are only applicable at their respective company's premises or branch. The pricing information is hidden often, so customers usually need to contact the company via telephone/email or visit their respective websites to get a quote. The customer can then decide on the service they will proceed, only after receiving all the quotations. This process could be very time-consuming, mainly due to the inflexibility of services. With specific terms and conditions only, customers can enjoy the door-to-door delivery service. .(Saipunidzam Mahamad & Suziah Sulaiman & Wong Yi Leng, 2018).Alexandra Gulc has identified ten major problems in courier services which are;

1. Time of delivery.
2. Service price.
3. Service availability and information (the availability of information about the company and the proposal, fixed working hours, the location of the collection points, ease of contact, etc.
4. Flexibility to handle the delivery in terms of time and place.
5. Ability to take advantage of additional service (notification of receipt by text message or email, charges on delivery, written proof of delivery, etc.).
6. Comprehensive service – the ability to use the accompanying services (a receipt of the waste, packaging, package pickup, full logistics service, etc.).
7. Service individualization (proposal matching to the needs of a specific client or industry.
8. Modern technology (individual PUDO – pick up drop off point near houses or apartments instead of boxes, drones or robots as couriers, etc.
9. Tele-technologies (ICT, mobile apps, the ability to manage shipment tracking, the ability to generate reports, constant contact with the client, etc.
10. Modern packaging solutions (the variety of packaging, packaging availability, the possibility of packaging by courier, packaging aesthetics, packaging customization, etc.

(Alexandra Gulc,2017).

"The problem deals with a combination of multiples variants of the vehicle routing problem as capacity, multiple periods, time windows, due dates, and distance as constraints. Since in the courier services the demands could be of hundreds or thousands of packages to be delivered, the problem is computationally unmanageable." ([Eduyn](http://m.growingscience.com/beta/authors/Eduyn+L%C3%B3pez-Santana/) & [William](http://m.growingscience.com/beta/authors/William+Camilo+Rodr%C3%ADguez-V%C3%A1squez/) & [Germán Méndez-Giro](http://m.growingscience.com/beta/authors/Germ%C3%A1n+M%C3%A9ndez-Giraldo/), 2017).

## Similar Applications

### Parcel2Go

Parcel2Go is one of the most popular courier services in the UK. Parcel2Go Mobile Application is a giant leap towards resolving the conflicts of inconvenience recurring to customers and the inflexibility of the current courier services. This application will ease out the lengthy process of package delivery for the people. (Saipunidzam Mahamad & Suziah Sulaiman & Wong Yi Leng, 2018).

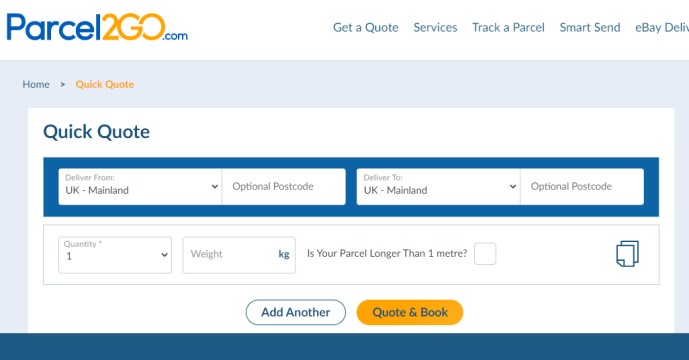


Figure 2‑2: Interface of Parcel2Go Website

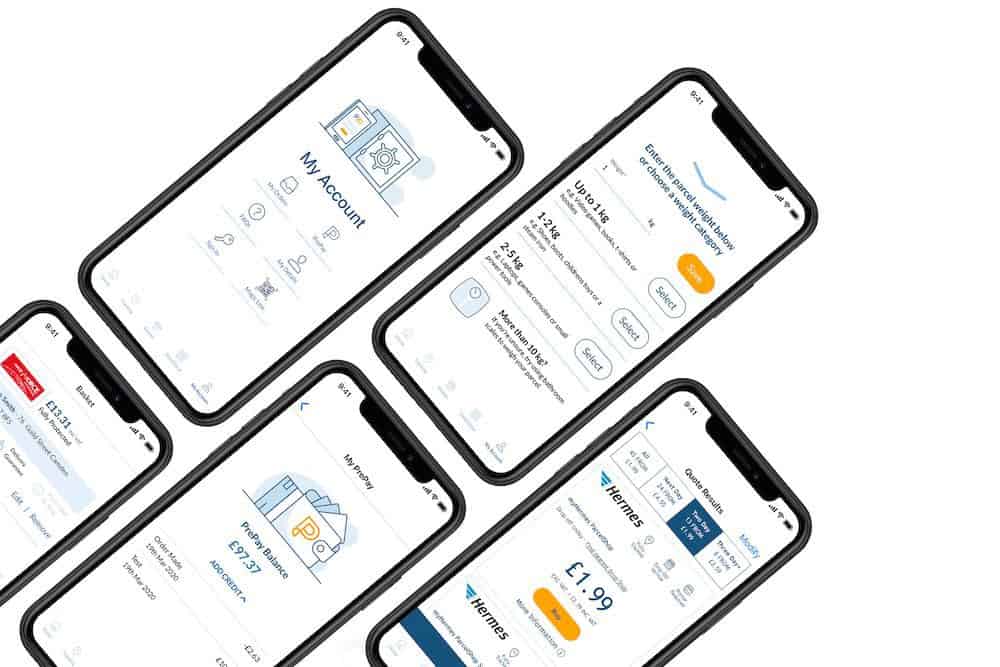


Figure 2‑3: Interfaces of Parcel2go Mobile Application

### JNE

PT Tiki Jalur Nugraha Ekakurir (JNE) is Indonesia's most popular courier service and ranked first in the category of courier service companies in Indonesia Top Brand 2018 (R. Hendayani & M.C. Dharmawan, 2018). PT Tiki Jalur Nugraha Ekakurir, or JNE, is an express and logistics courier services company originating from Indonesia. Its official name is Tiki Jalur Nugraha Ekakurir. Nowadays, Tiki JNE is better known as JNE because JNE has a separate mobile application that can track the package.



Figure 2‑4: JNE Website



Figure 2‑5: JNE Mobile Application Interfaces

## Google Maps Application Programming Interface (API)

Google Maps is a web mapping service application and technology provided by Google. It launched in 2005 has revolutionized online mapping service applications on the World Wide Web. Based on Asynchronous JavaScript and XML (AJAX), a new type of client/server interaction was introduced in Google Maps to maintain a continuous connection between the client and the server to download additional map information immediately. In addition, Google also provides programmers its extensive sources of code called the Application Programming Interface (API). The API consists of a set of data structures, object classes, or functions that can be used by a programmer using JavaScript, PHP, or another scripting language (Shunfu Hu & Ting Dai, 2013). Google Maps provides an API for developers to develop the application by integrating Google Maps, displaying the location maps, and navigating different routes. Google Maps is a free online map provided by Google. Google also provides APIs for developers who develop applications that enable them to use Google Maps in their applications. Google Maps API is a library which is written in JavaScript (Gunawan & Purnama, 2015). There are two types of Google APIs for Android Applications.

1. Maps SDK
2. Places SDK

### Maps SDK

Maps SDK allows developers to add Google Maps to their applications. The API automatically controls the access to Google servers, data downloading, map display, and reposed to map gestures. Developers can also add API calls to add markers, polygons, and covering the basic map. Additionally, it allows changing the user's view of a map.

### Places SDK

The Places SDK for Android allows developers to build location-aware apps that respond contextually to local businesses and other places near the user's device. Hence, a developer can build rich apps based on places that mean something to the user to complement the specific geographic-based services.

### Global Positioning System (GPS)

The GPS is a space-based satellite navigation system that provides location and time information in all weather conditions, anywhere on or near the earth, with an unobstructed line of sight to four or more GPS satellites. The system provides critical capabilities to military, civil, and commercial users around the world. The United States government created this system, maintains it, and makes it freely accessible to anyone with a GPS receiver (Monika & Sudha, 2015).

### The Global System for Mobile Communication (GSM)

GSM uses cell identification techniques to provide the location of the mobile phones. Cell Identification identifies the cells of the wireless networks which the device is using. The location of the base station is also the location of the mobile user. The accuracy of this technique strictly depends upon the size of the cell, and it can increase by adding time advance and signal strength. This technology is used only if a mobile user is 550m or more away from the base station (Monika & Sudha, 2015).

### Geolocation API

Geolocation is a process that identifies the geographic location of a user or hosting device via a variety of data collection mechanisms. Most geolocation services use network routing addresses or internal GPS devices to determine this location (Monika & Sudha, 2015). Monika and Sudha used the Geolocation API to get the geographical details of a location, like an attitude and longitude associating with the hosting device. Also, they have used Google Geolocation API to identify the current address (Monika & Sudha, 2015). Monika and Sudha proposed a methodology 10 for obtaining the current location of the user. This system provides the current location of the user or the hosting device. They firstly calculate the geographical coordinates, i.e., latitude and longitude of the current location of the hosting device, then based on latitude and longitude values, the address will be calculated. Reverse Geocoding will be performed on the set of geographical coordinates to convert the latitude and longitude into the formatted address. This system will use the geolocation API to provide the user's address (Monika & Sudha, 2015).

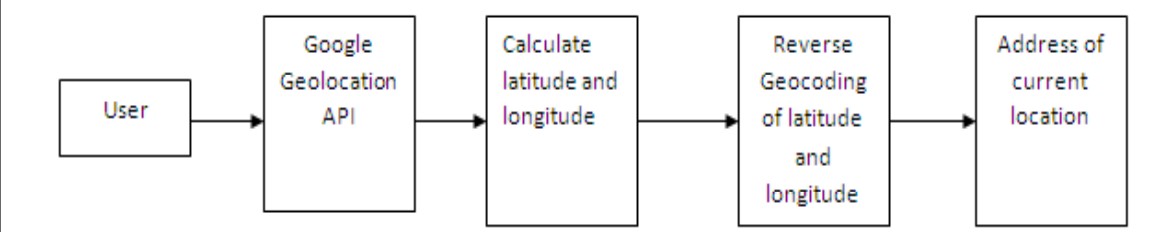


Figure 2‑6:Bloc diagram of proposed methodology (Monika & Sudha, 2015)

## Finding the Shortest and easiest route

People use many algorithms and calculations to calculate the distance of a route and track locations with the aid of Google Maps APIs. Some researchers use their own newly invented algorithms to find the shortest path possible.

### Similar Applications

#### Dijkstra Algorithm

Google Maps use the Dijkstra Algorithm to find the shortest path. Edger Dijkstra formulated Dijkstra's Algorithm as a graph search algorithm that will seek the shortest route. Dijkstra's Algorithm is a k-search algorithm that will visit each graph node to find the shortest path from start to finish. With the help of computers, the algorithm can be computed in a timely matter. From the pseudo-codes of Dijkstra's Algorithm, personal navigation systems were born. Dijkstra's Algorithm can only calculate the shortest path based on the path weight, and only weights (lengths) must be nonnegative (Daniel & Gregory & Jin, 2014). Figure2-8 and 2-9 show how google maps calculate the shortest path of the same town by applying the Dijkstra Algorithm.

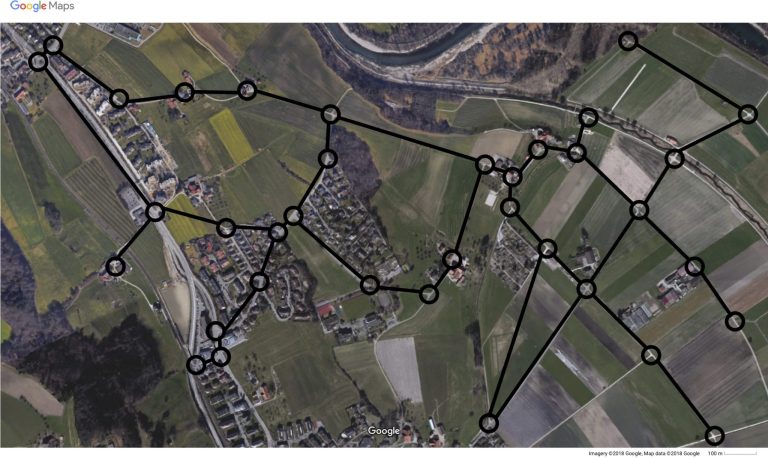


Figure 2‑7: image of the ground and overaid graph (Elias,2018)

Figure 2‑8: Image of a small town n Google Maps(Elias,2018)

### Saving Matrix

Saving Matrix is a method to minimize distance, time, and cost to produce an efficient delivery route. The Saving Matrix method is expected to help companies in their distribution network. This method requires the distance between the warehouse owned by a company and the destination of goods delivery, originating from the warehouse company (Timothy & Eddy & David, 2018).

Figure 2‑9: Saving Matrix (Timothy & Eddy & David, 2018)

S(x,y) = J(G,x) + J (G,y)- (x,y)

where: S(x,y) = Saving Matrix value or the combined distance

J(G,x) = distance from warehouse to destination x

J(G,y) = distance from warehouse to destination y

J(x,y) = distance from destination x to destination y

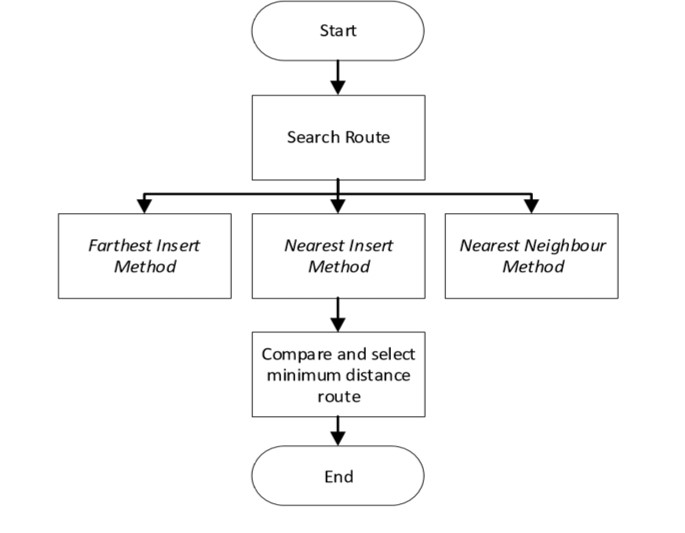


Figure 2‑10: Flowchart Sorting & Selected Route (Timothy & Eddy & David, 2018)

The above figure depicts the methodology of the saving matrix and how it calculates the minimum distance route from the given routes.

## Research Gap

The business landscape in the courier services industry has a very significant change due to its high and rapid growth of e-commerce. This literature review was carried out to find what other researchers have accomplished in the field with new technologies and the achievements and new approaches they have proposed. Many technological advancements came to light with the vast growth in courier and delivery services in recent years. Still, there is no common ground in conquering some basic limitations, though they are found in a few applications. For instance, many courier services don't afford customer package pickups, leaving the customers to hand over the packages to one of their warehouses. Some delivery applications pick up from a given location, but those delivery applications only deliver one specific category, Ex: Uber Eats only delivers food items. Figure 2-10-Flowchart Sorting & Selected Route (Timothy & Eddy & David, 2018). As my research concluded, I found out that no mobile application or service connects local courier services as a hub, resulting in prominent and cheap courier service for the users if the conceptual application can be developed.

Table 1: Comparison of the proposed system with existing systems

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Pronto** | **Uber**  **Eats** | **Parcel2Go** | **JNE** | **FedEx** | **Proposed**  **System** |
| Have a mobile application | × | ✓ | ✓ | ✓ | ✓ | ✓ |
| Track Package | × | ✓ | ✓ | ✓ | ✓ | ✓ |
| Picking up the package | × | ✓ | v | × | × | ✓ |
| Connected with other delivery method or courier service | × | × | × | × | × | ✓ |
| Finding the easiest route | × | ✓ | ✓ | ✓ | × | ✓ |
| Courier to abroad | On  Request | × | ✓ | ✓ | ✓ | × |
| Deliver Any Category | ✓ | × | ✓ | ✓ | × | ✓ |

## Market Research

Market research was conducted by sending a questionnaire to a group of people to get their feedback and interviewing a courier service sales, operations executive, and delivery person. Through the questionnaire, the main target is to get to know the major problems that occur when using a courier service, what they expect from courier service, suggestions for the proposed system, and many more. The questionnaire was finalized with the help of 103 people who represent the general public. 14 Through the interview with courier service sales and operations executive, the main target was to get an idea about the issues that courier services have, the opinion about the proposed system, and whether they would like to work with it. Through the interview with the delivery personnel, the main target was to get an idea about the issues that delivery persons have, their opinion about the proposed system, and whether they would like to work with it. The questionnaire and the results summary are as follows.

### Questionnaire Results Summary

### 

Figure 2‑11: Courier service Usage

Above "Figure 2-11" represents how often using a courier service, according to this many users, uses a courier service rarely or once a month. Some users still haven't used a courier service.

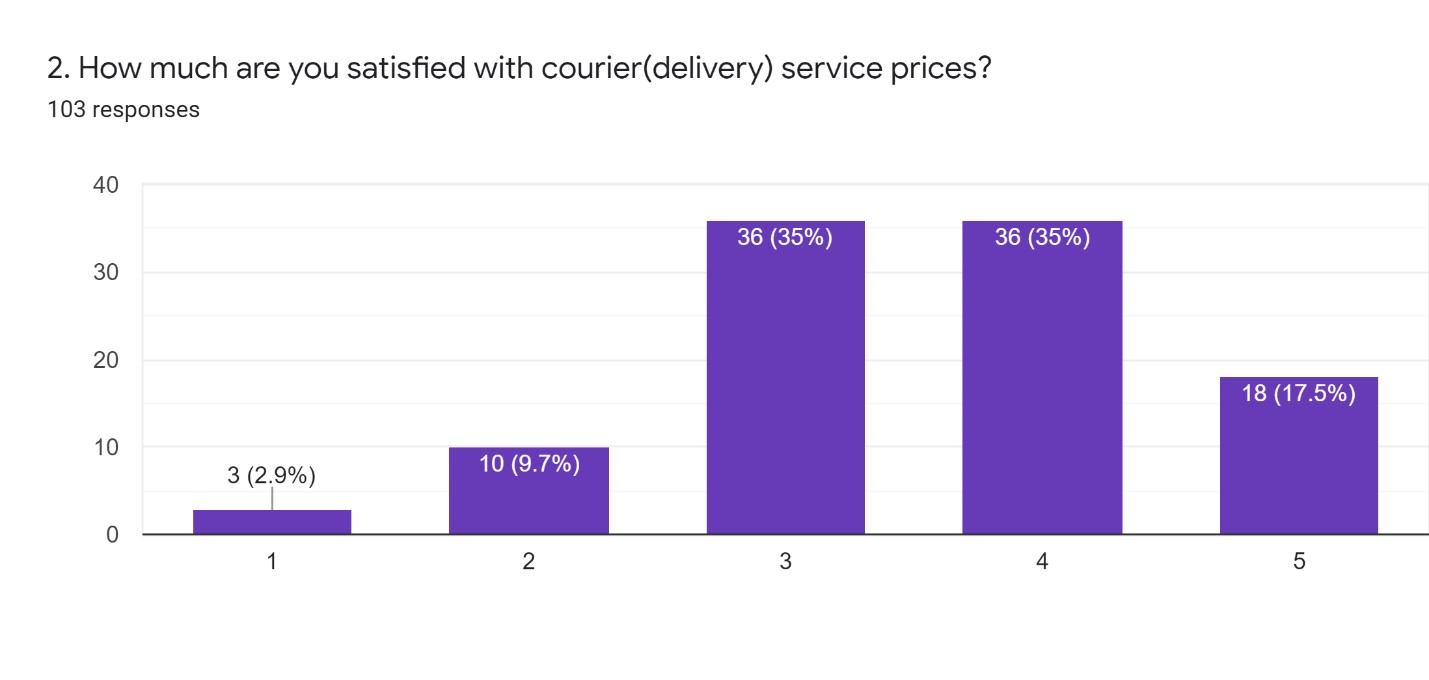


Figure 2‑12: How much customers satisfied with current courier service providers

The above "Figure 2-12" graph represents the customers' satisfaction with current courier service prices. According to these findings, the majority are happy with the current pricing on couriering. Very few people do not agree with the current prices.

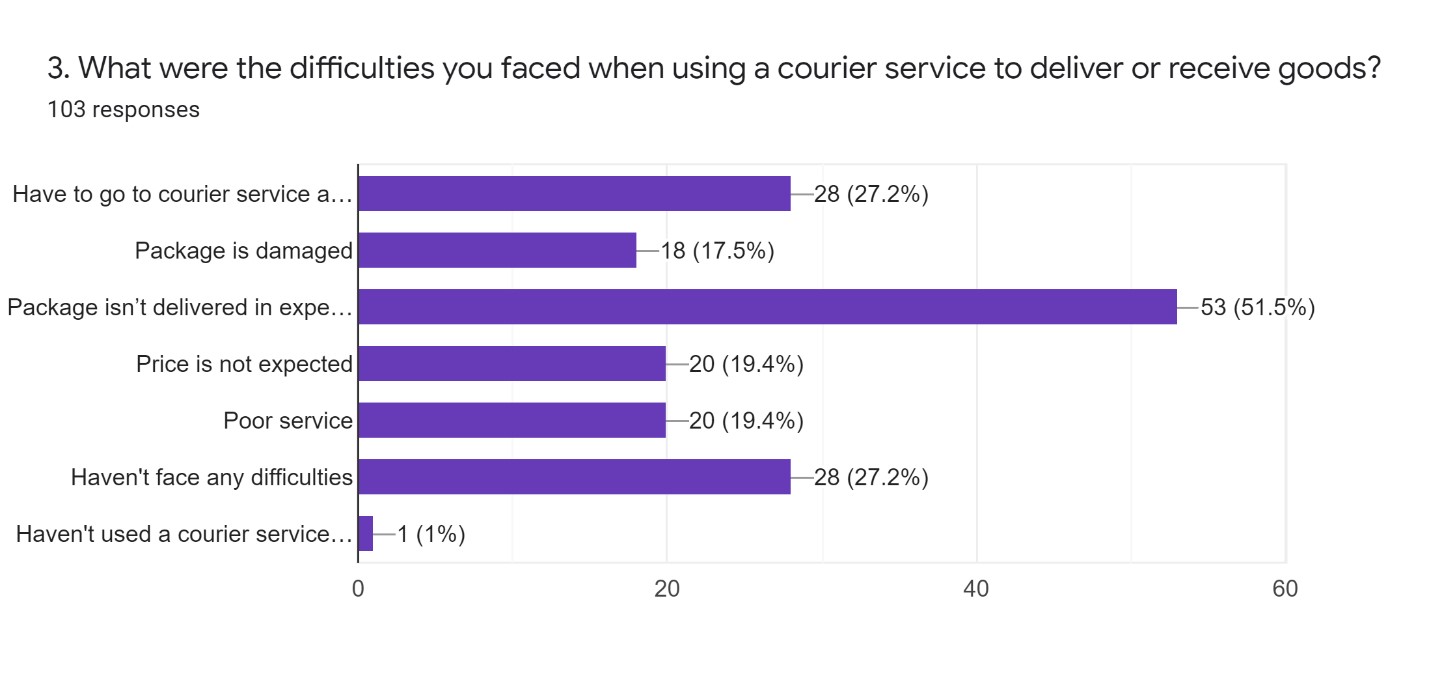


Figure 2‑13: User's difficulties when using a courier service

The above "Figure 2-13" graph represents users' problems when using a courier service. Based on these findings, most of the user's package was not delivered at the expected time. A very few haven't faced any difficulties.

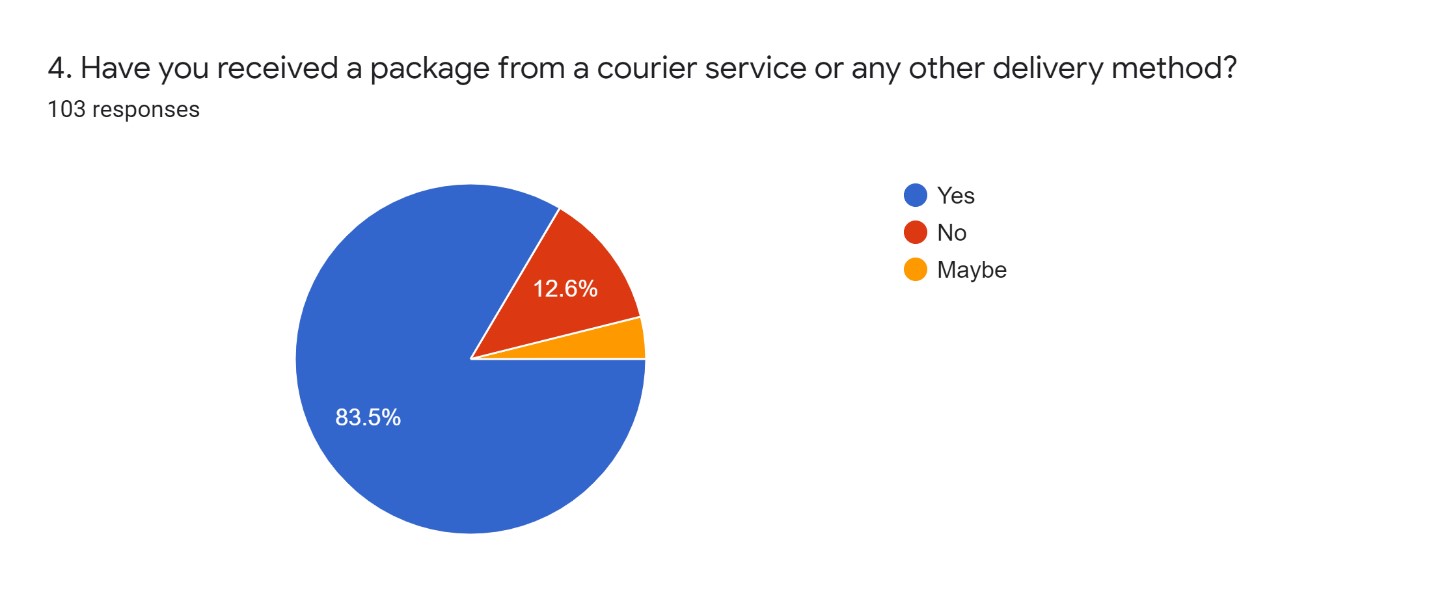


Figure 2‑14: How many users have received a package from any delivery method

The above "Figure 2-14" graph represents how many users have received a package from a courier service or any delivery method. According to this graph majority of users have received a package. Few haven't received a package at all, and very few don't have an exact idea whether they have received it or not.

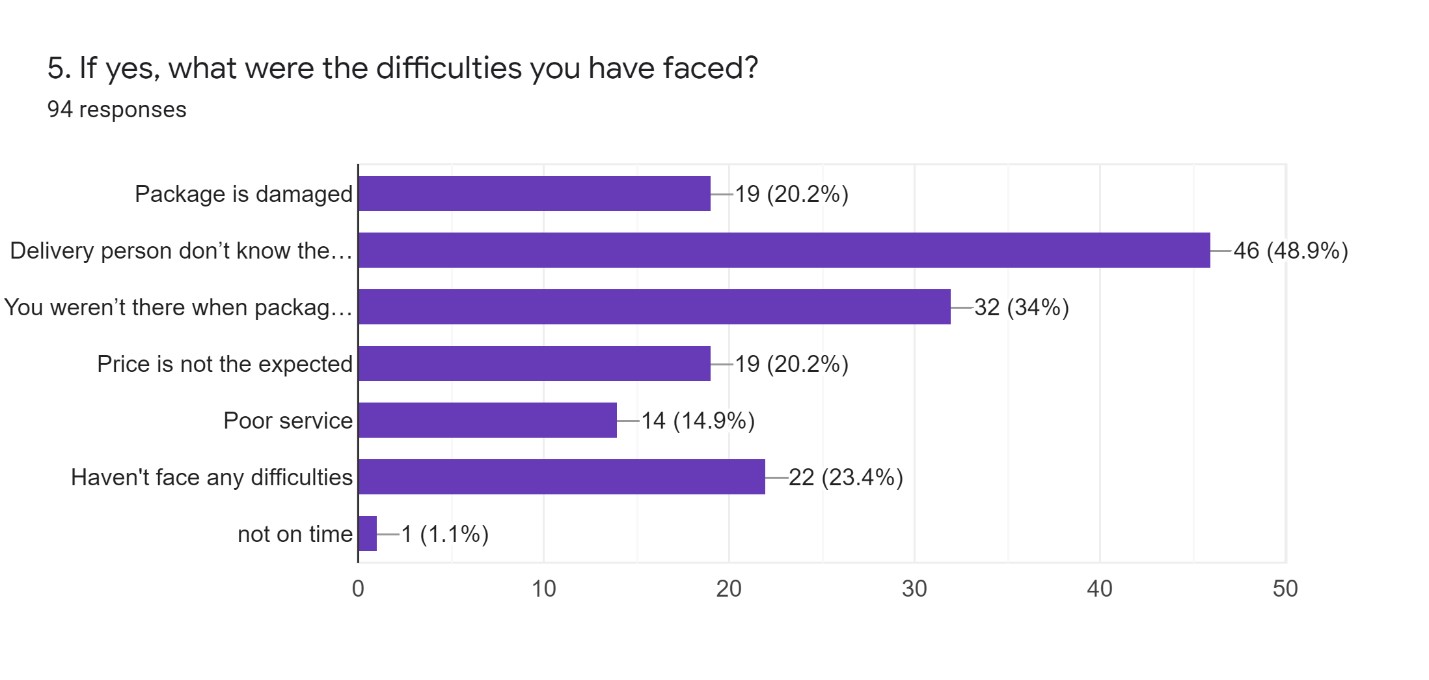


Figure 2‑15: Difficulties when receiving a package from a courier service

The above "Figure 2-15" graph represents users' difficulties when receiving a package from a courier service or any delivery method. As per this graph, most users have claimed that the delivery person didn't know the route. Few haven't faced any difficulties and very few claims that the package didn't arrive on time.

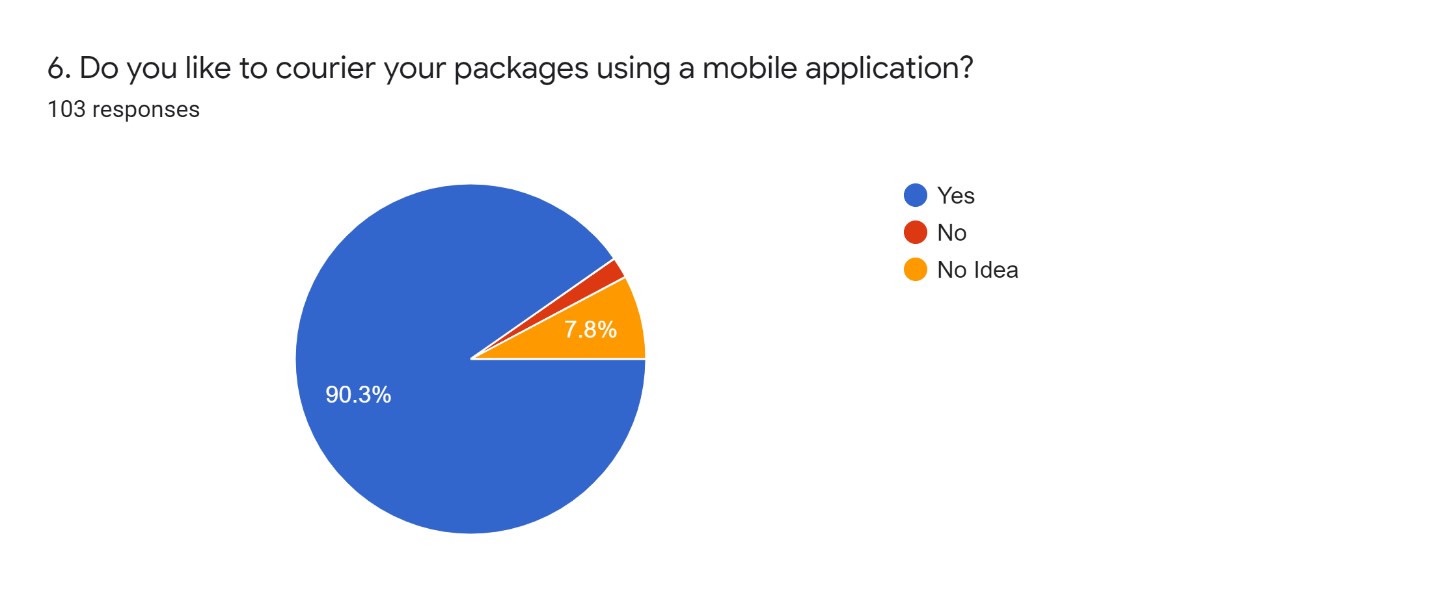


Figure 2‑16:How many users like to use a mobile application

The above "Figure 2-16" graph represents how many like to courier their package using a mobile application. According to this graph, many users want to use a mobile application to courier their packages. Very few users don't like using a mobile application to send their packages, and some don't have an exact idea.

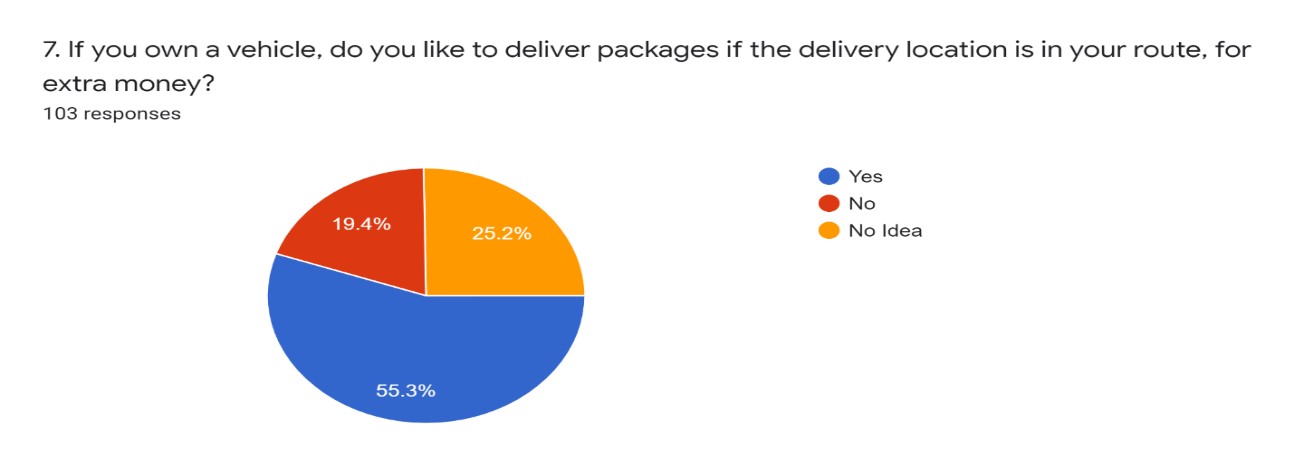


Figure 2‑17: How many users like to deliver a package on their router

The above "Figure 2-17" graph represents the favorable numbers who'd like to deliver a package if it's on their route. According to this graph, most users favour delivering packages if the delivery address is within their bounds. A quarter of people don't have an exact idea about it, and few people don't like to deliver.

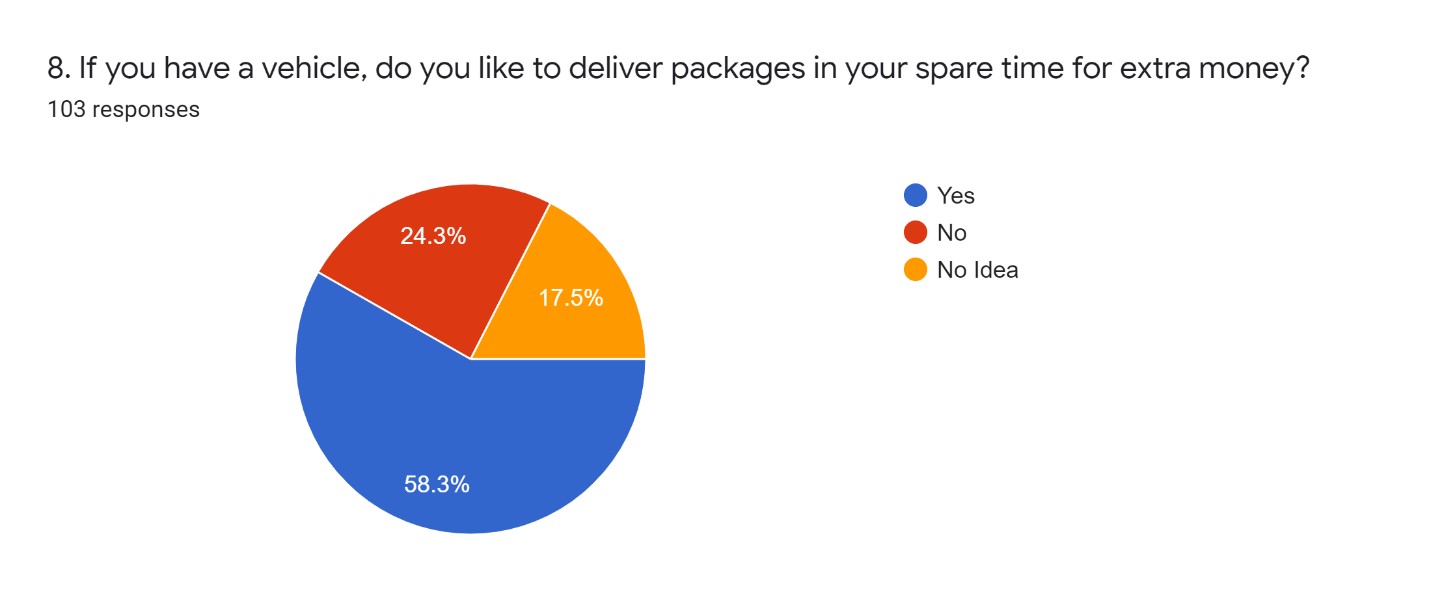


Figure 2‑18: How many people like to deliver on their spare time

The above "Figure 2-18" graph represents how many like to deliver packages in their spare time if they have a vehicle. According to this graph majority of users like to deliver packages in their spare time. Few people don't like to deliver packages in their spare time, and very few don't have an exact idea whether they could deliver or not in their spare time.

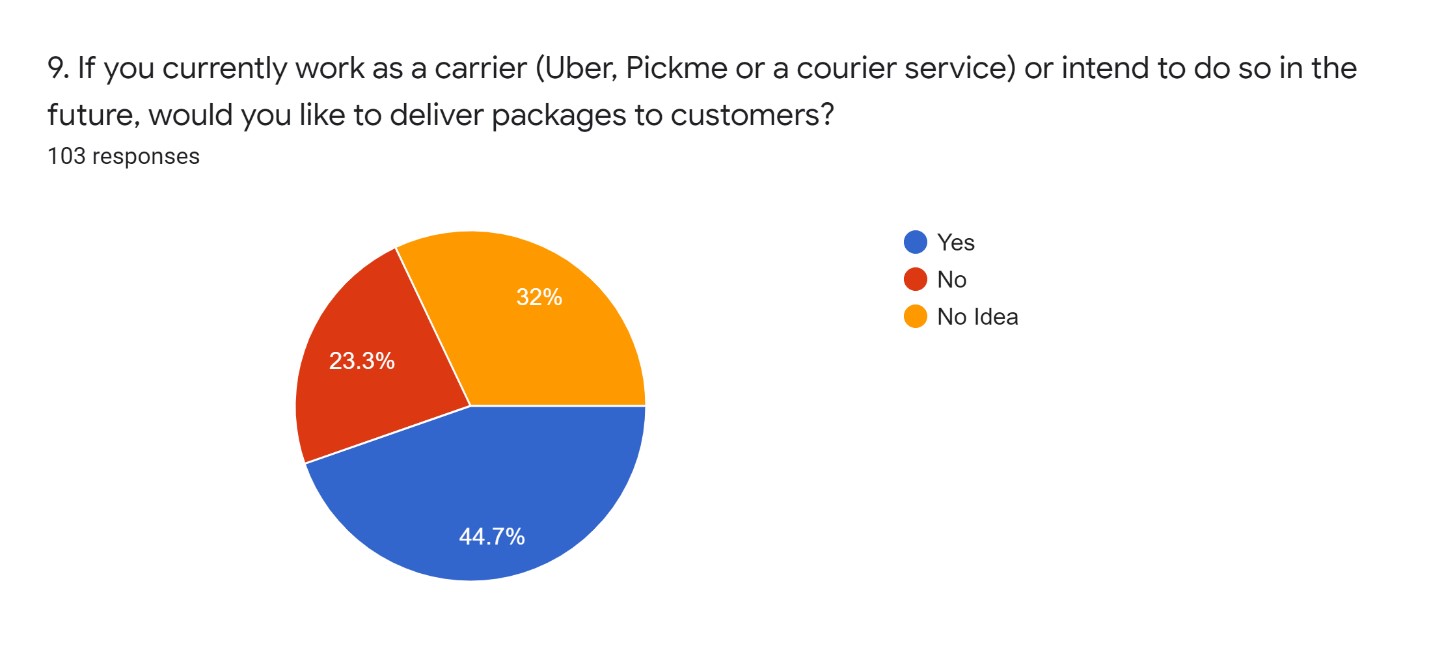


Figure 2‑19: How many people like to work as delivery rider

The above "Figure 2-16" graph represents the likeness towards delivering packages of people that currently operate in the delivery service or intend to do so in the future. According to this graph majority of users like to deliver packages. Few people don't like to deliver packages, and a few don't have an exact idea whether to deliver or not.

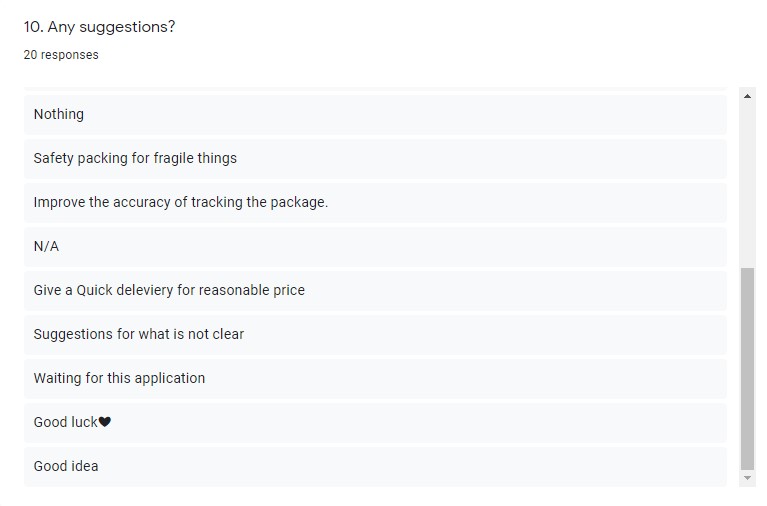
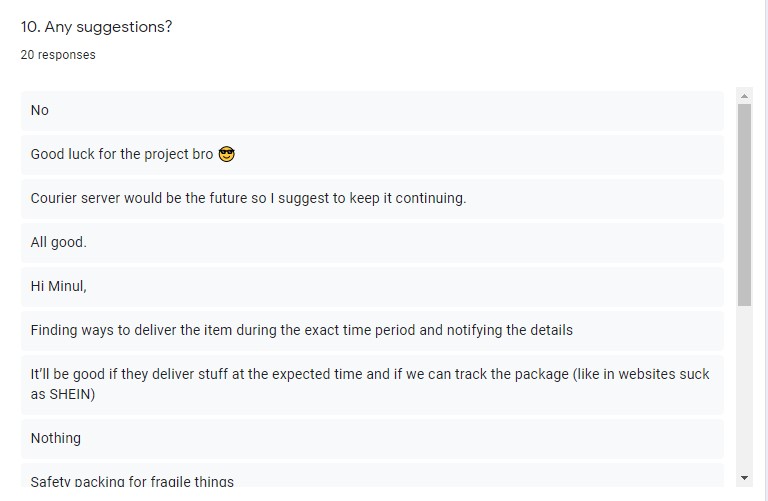


Figure 2‑20: Suggestions

"Figure 2-20" briefs the public's suggestions for the proposed system.

### Interview with a Courier Service Owner

In the proposed application, one of the primary functions is connecting many courier services. It is essential to get opinions from experienced personnel like courier service owners and get to know the problems they have and their recommendations on this concept. To accomplish the above procedure, I interviewed Mr. Danushka Udayanga, Sales and operations executive at Domex courier service.

First, I described the proposed system to Mr. Danushka, and he very much liked it. He also raised his current concerns about losing customers due to neglecting long-distance packages because they are unprofitable since they cannot charge the customer with their standard rate for such items. They also proved in favor of reducing the price of their courier services. One of his suggestions was to let the customer decide the courier service they intend to proceed with rather than an auto selection when the application is running. If only the customer skips that step and goes with a regular delivery guy, display a pop-up message to the customer stating if the package is of valuable itineraries, then it's better to go with a courier service.

He very much appreciated the function of showing the simplest route to the delivery person on the proposed application. He stated that delivery riders spend a lot of time finding the exact location, and they have to make many calls to the customers, and some customers don't know how to tell the route.

He applauded the idea of letting the courier service know if the customer is within the location or not when delivering the package because usually, they have to go multiple times for the delivery upon confirmation from the customer. Finally, he stated that they like to work with this application and would be able to increase their number of deliveries with this application.

### Interview with a Delivery Person

Delivery persons are playing a leading role in this proposed application. Because of that, it's a must to get a statement from delivery personnel about the current problems they face and their opinion on the proposed system. To accomplish this virtue, I interviewed Mr. Hashan Botheju, delivery personnel at Domex courier service. Initially, I described the proposed system to Mr. Hashan, and he very much liked it. He stated that they felt the need for an application of sort, mainly to find the simplest route, as they are sometimes stuck with a single package for the whole day, unable to find the customer's location.

He suggested adding an option, whereas when delivering a package, if another customer places an order in the same route as the initial package, open a pop-up asking the delivery personnel whether they can deliver the second package. Also, suppose a package needs to be delivered to a different location, and the pickup is enrooting to the current delivery route. In that case, the application can inform the delivery personnel to pick up the package before delivering his/her initial package.

He also said that this application could be the cue for many opportunities, and they don't need to rely on one company. He was very excited to work with this application and said he'd be looking forward to the launch.

# Methodology and Artifact Design

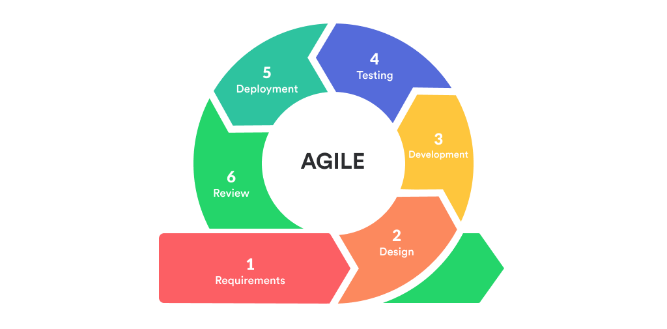
This chapter is explained with the aid of UML diagrams, following up with processes and procedures related to them. Here the developer will present the UML related workings related to each and every step of development allowing the readers to understand the build-up of the project. Ultimately, the chapter will conclude with a test schedule, conducted tests, and their outcomes.

## Software Development Methodology

Agile methodology is using for the proposed system because of the flexible nature of this proposed system. The main reason for using agile methodology for this proposed system is the potential for the developer and the client to add new features and improve the design at any phase of the process. Moreover, it helps the developer to test the system throughout the development from start to end. The below figure shows how the agile methodology works. Requirement gathering is done as the first phase to check whether the system is feasible. Designing, developing, testing, deployment and review are all done together in a recurring cycle. Since the Agile methodology was used to develop the application researcher maintained the time, tools, documentation and developed an error-free application that met all the components. A bug-free application can be created using this process.

The Gantt chart and the Work Breakdown structures are attached in Appendix A and B, respectively, as references, to get a clear idea of how the methodology was affected for the application development,. Since the work was divided step by step, it was easy for the researcher to work according to a timeline.

Figure 3‑1: Agile Methodology (Anon, 2019)



The above figure shows how the agile methodology works. Requirements gathering is done as the first phase to check whether the application is feasible. For this, the researcher did a primary and secondary data gathering so that accurate data will be gathered to develop the application. Designing was done by creating UML diagrams to ensure and encounter the external and internal hardware and software which is required for the application. Developing was done by engaging with tools and technologies which are familiar to the researcher. Testing was done regularly in every Sprint while developing the application since the application did according to the agile methodology. Deployment and review are done together as in a circle in agile methodology.

## Requirement Gathering and Analysis

Requirements were gathered in two ways to get an idea about the requirements needed to develop the proposed system: primary data gathering and secondary data gathering. Primary data is collected by sending a questionnaire to a targeted audience and finding results. I also conducted two interviews with people working in the delivery sector and included their opinions and suggestions for the proposed system. A broad scoped literature review was done by referring to journals, websites, paper articles, etc., to gather secondary data, a. This requirement gathering ensured that the proposed system is technically feasible and essential to people's daily lives.

To get a proper idea of constructing the artefact, the researcher needs to analyze what he gathered. As the first step, the researcher analyzed data from the market research questionnaire by surveying the most common opinions the audience has entered. As the second step of analyzing the data gathered, the researcher thoroughly examined the researches already done by checking journals, research papers, and websites. By analyzing data in the literature review, the researcher analyzed the existing similar applications, which functions highlighting Packagely mobile application.

## Design

The proposed project developed android studio as the development environment and the emulator for testing. The coding language for the Packagely application is JAVA language which is one of the most common languages among programmers. Google Firebase is used for the database of this application. And Google Maps API contributes to finding the easiest route and map requirements. The proposed project will be designing user-friendly as much as possible— the interface design, in particular. Users can register in the application by adding a brief set of information to the application. Users can easily navigate through the functions and choose a required delivery option. After adding a relevant destination, the user can select the best delivery method. The application automatically suggests several courier services for the user to pick the one that best suits them. The proposed application offers a variant experience for the courier services and the delivery riders. While selecting an area to deliver a package by a courier service, the system shows other courier services scheduled for delivery in the selected area. Upon confirmation, they can come to an alliance when sending packages that will cut their costs drastically.

This phase demonstrates how the application was designed and built by UML diagrams.

### UML Diagrams

The Unified Modeling Language (UML) diagrams have been created for the application since it gives a clearer understanding of how the application will interact with the user and what activities and processes will happen inside the application.

#### Use Case Diagram

Use case diagram in the figure below describes how the actors interact with the Packagely application. An unregistered user or courier service can register with the application. After an unregistered user gets registered, that user can log in to their personal account, find a courier service, track packages, place an order, track orders, add current location and destination, and edit their profile as well. After an unregistered courier service gets registered, that courier service can do all the things: login and contact other courier services, add a destination, and edit their profiles. The application admin manages the application, changes prices number, and contains users and courier service details saved in the database.

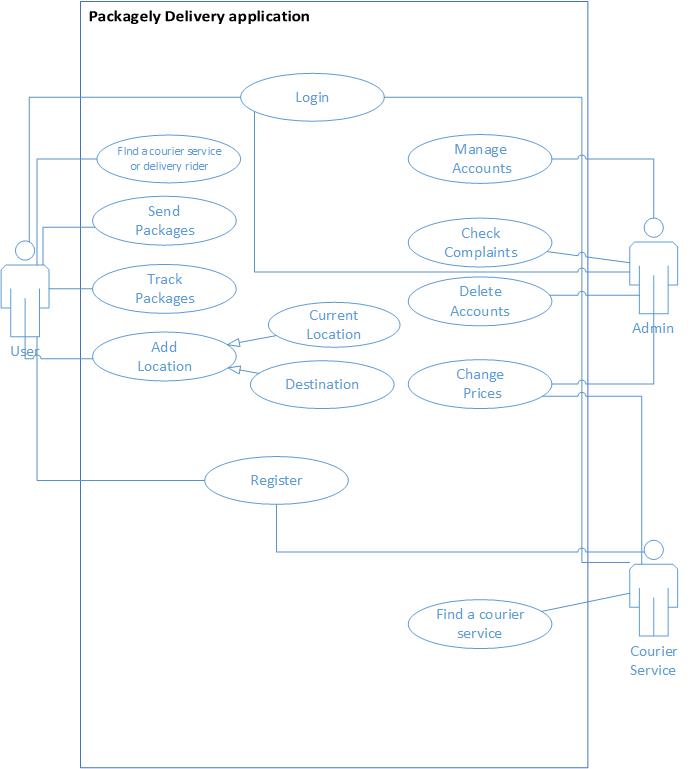


Figure 3‑2: UseCase Diagram

#### Architecture Diagram

The architecture diagram demonstrated below explains the overall outline of the proposed artefact. The artefact will contain a frontend created using JAVA Language and Google Firebase as the database. Other than that, the application uses Google Maps to implement geolocation. Finally, the application uses the mobile device's local storage as the cache storage for the application.

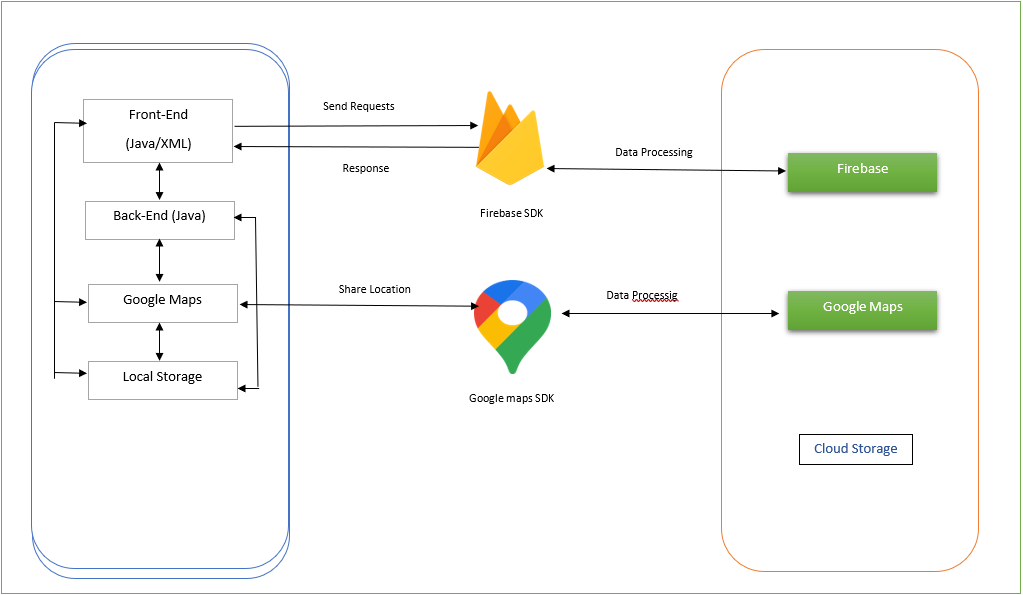


Figure 3‑3: Architecture Diagram

#### Layered Architecture Diagram

The artefact contains an android application at the base application layer. The Access layer of the artefact includes the outer APIs and SDKs utilized for the application: the Google Maps API and Firebase SDK. The Service layer of the artefact contains the Google Maps and Additional Systems for the application. Finally, the Storage Layer of the artefact includes the Cloud Storage and the Local Storage of the application needed for the information base and the reserving of the application.

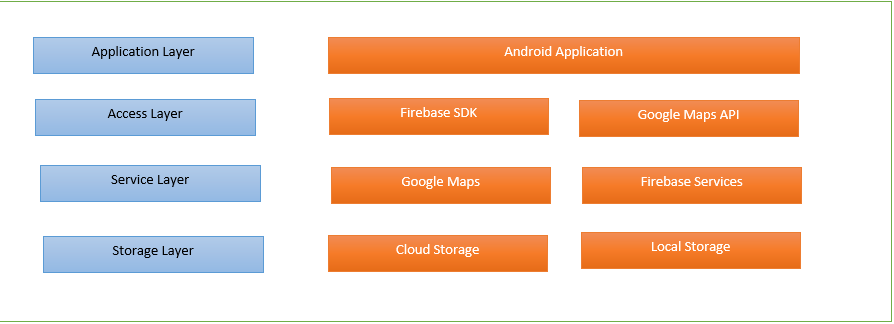


Figure 3‑4: Layered Architecture Diagram

# Artefact Design & Testing

This section demonstrates the development of Packagely application in detail with the relevant tools, technologies, and examples of code snippets. Moreover, the testing of each unit is also briefly discussed.

## Artefact Design.

Packagely application is mainly composed with Java language in Android Studio as IDE. Google Maps API is used to get the locations, maps, and find the shortest route. Google Firebase is used as the database for this application to store user data and all database functionalities. UIs were mainly designed using Android UI and Adobe Photoshop. Android emulator and an Android Smartphone was used to test Packagely application.

### User Account Management.

Every user must sign up for a user account to access all functionalities. This proposed application has two users; courier and customer. Those users have separate accounts to log into the system. If the user does not have an account, they must create an account by entering the relevant details. All inputs are validated before the login and register the user.

The developer has set up Firebase authentication and a real-time database to store information. The database has been initiated by using the Firebase console and the Firebase Google platform. Other than that, the developer integrated Firebase into the Android Studio using the Firebase tool. During the integration, the developer added several dependencies and plugins.

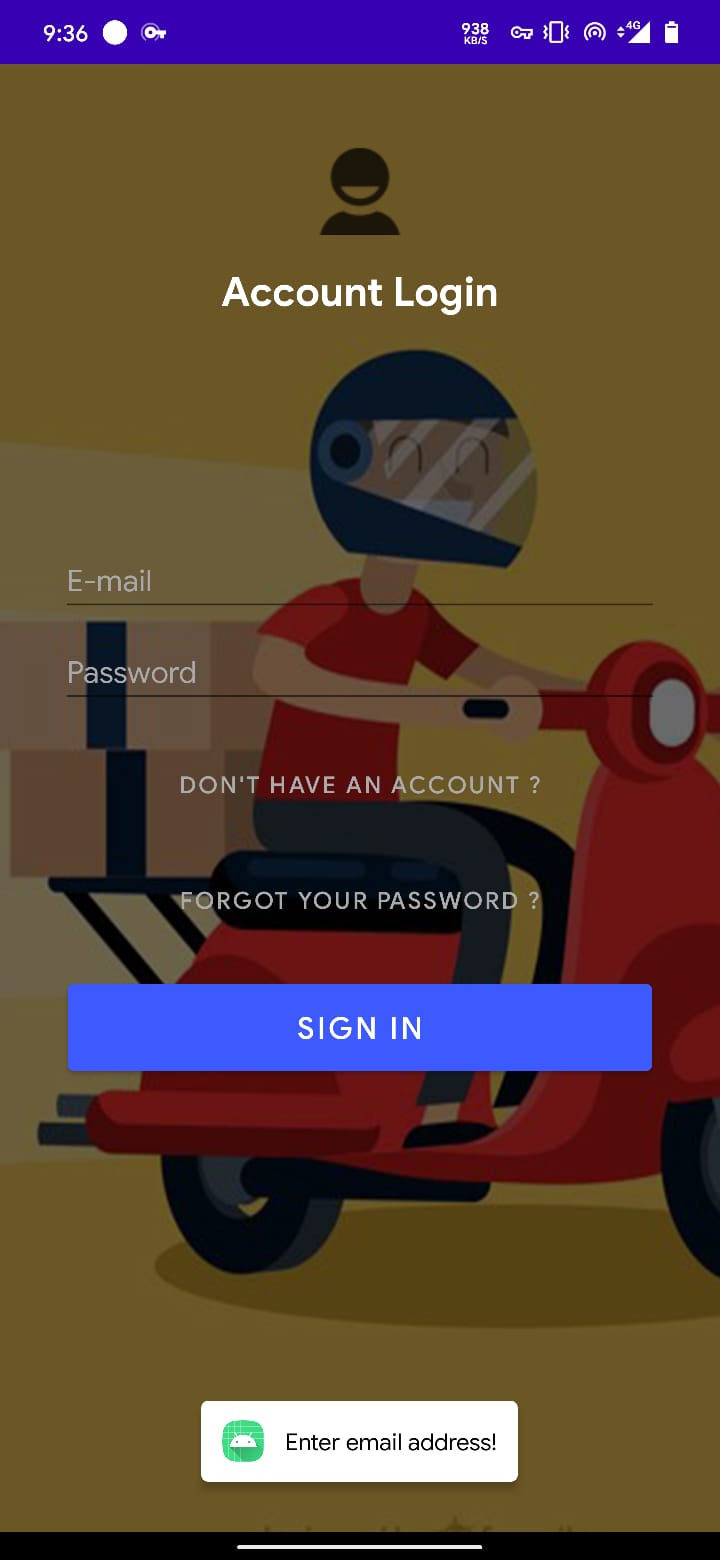


Figure 4‑1: Login

The below code snippet depicts the implementation of the signup process in the proposed application. The developer was using the Firebase Authentication and Realtime database. Once the user has entered the details, the email and password will run through Firebase Verification, and the remaining information will be stored in the Fire store database.

mAuth.createUserWithEmailAndPassword(email, password)

.addOnCompleteListener(new OnCompleteListener<AuthResult>() {

@Override

public void onComplete(@NonNull Task<AuthResult> task) {

if (task.isSuccessful()) {

User user = new User(

name,

email,

phone,

rate

); FirebaseDatabase.getInstance().getReference("CourierService")

.child(FirebaseAuth.getInstance().getCurrentUser().getUid())

.setValue(user).addOnCompleteListener(new OnCompleteListener<Void>()

Code Snippet 1: Sign up Authentication

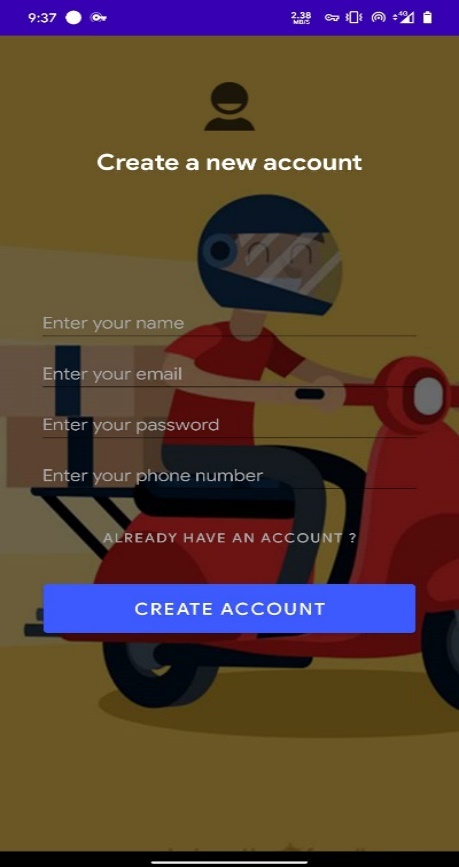
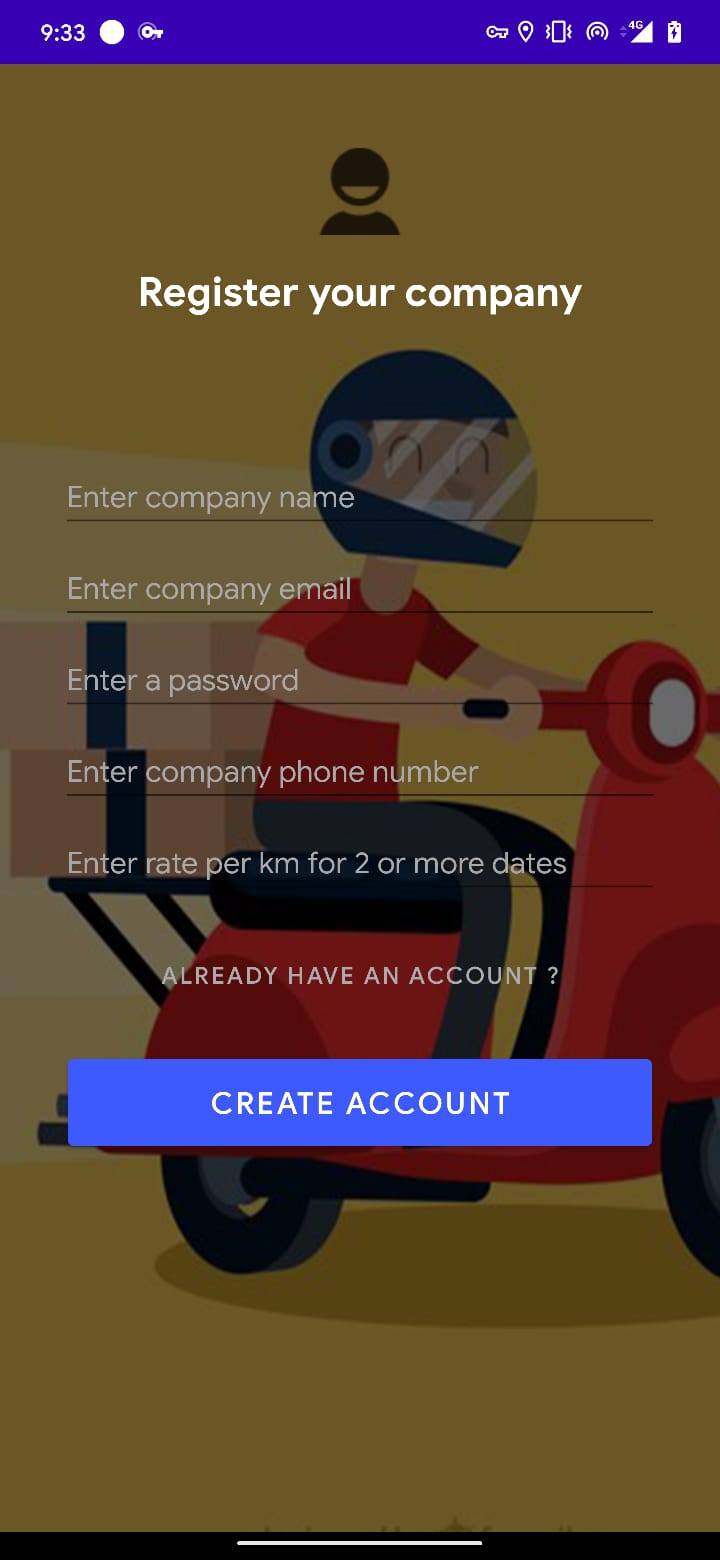


Figure 4‑2: Signup

### Select Location

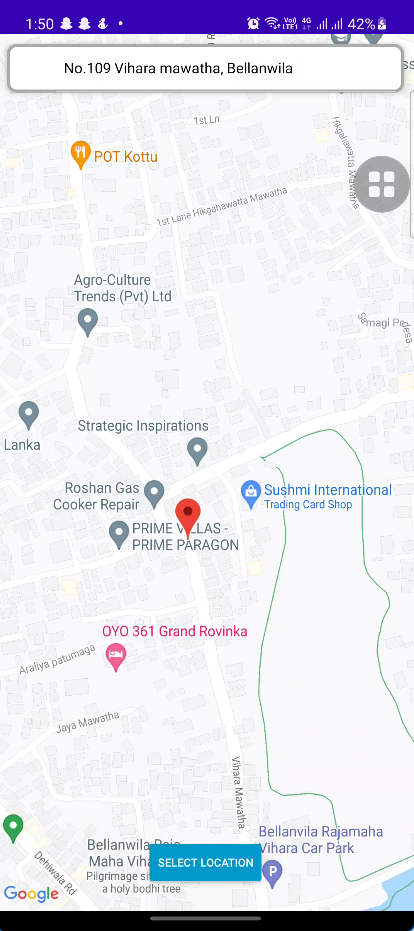
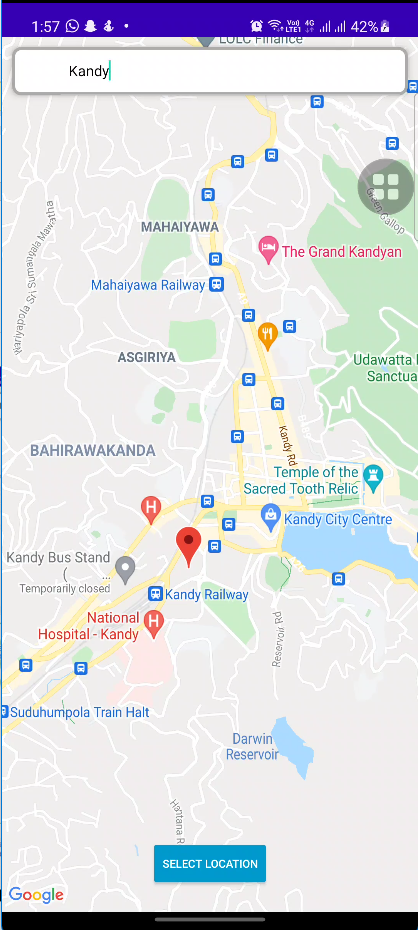


Figure 4‑3: Select Destination

Figure 4‑4: Select Location

To start the process, the delivery user should give the start and end destinations. Users can provide locations via the Google Map application as in the above figures.

The following code snippet explains how the application records the location instructions entered by the user.

Google map location codes are retrieved from Google Maps API documentation for this project.

mapFragment.getMapAsync(this);

Code Snippet 2: Getting users location

selectLocation.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

if (!searchView.getQuery().toString().equals("")){

saveFromLocation();

startActivity(new Intent(MapSearchFromActivity.this, MainActivity.class));

finish();

}

else

{

Toast.makeText(MapSearchFromActivity.this, "Please enter a location", Toast.LENGTH\_SHORT).show();

}

### Make a Delivery

After adding the appropriate locations, the user needs to select the package size, their choice of courier service and click the "Place Delivery" button. Make a delivery UI is shown in the below figure. Upon making a purchase, the user receives a message stating whether the placed order was successful from the application.

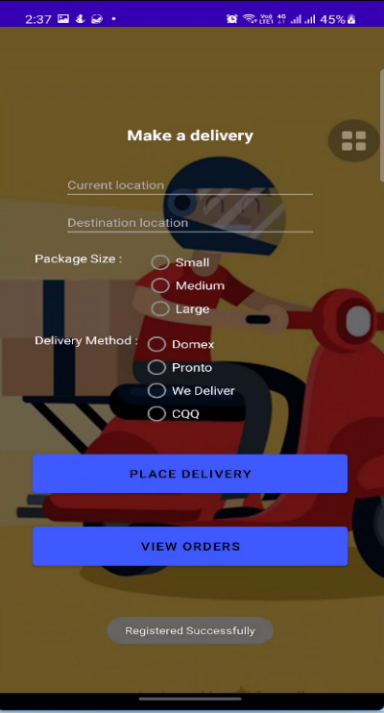


Figure 4‑5: Make a delivery

Delivery delivery = new Delivery(

FirebaseAuth.getInstance().getCurrentUser().getUid(),

fromLocation,

fromLat,

fromLong,

toLocation,

toLat,

toLong,

PackageSize,

DeliveryMethod,

DeliveryDate

);

FirebaseDatabase.getInstance("https://packagelydelivery-default-rtdb.asia-southeast1.firebasedatabase.app").getReference("Delivery")

.child(uniqueID)

.setValue(delivery).addOnCompleteListener(new OnCompleteListener<Void>()

Code Snippet 3: Passing data to database

When the user enters all details and clicks the "Place delivery" button, data goes to the database, and the courier service can check the order details as mentioned in the above code snippet.

### View Orders

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_view\_orders);

databaseReference = FirebaseDatabase.getInstance("https://packagelydelivery-default-rtdb.asia-southeast1.firebasedatabase.app").getReference("Delivery");

view = findViewById(R.id.recycler1);

view.setLayoutManager(new LinearLayoutManager(this));

FirebaseRecyclerOptions<Delivery> options = new FirebaseRecyclerOptions.Builder<Delivery>()

.setQuery(databaseReference, Delivery.class)

.build();

Code Snippet 4: Data access from firebase

When the user wants to check details about their order history, they can go to the "View Orders" section that briefs about their recent order history. The above code depicts its process.

The following UI can be seen when the user clicks the "View orders" button.



Figure 4‑6: Order Details

### Courier Home Page

When a courier signs in to their account, they are redirected to the home page where all application functionalities could be accessed; check orders, view remote travel patterns of other courier services, and add a destination for the different courier services to partner. The below figure shows the UI of the courier home page.

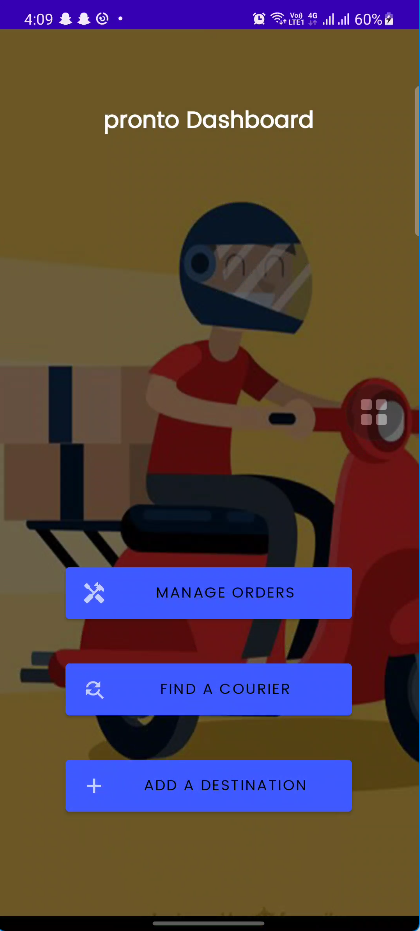


Figure 4‑7: Courier Dashboard

The above figure courier dashboard UI shows courier name with "dashboard," It transforms to the relevant courier's name upon registering. The code snippet for this process can be seen below.

private void getCompanyName()

Code Snippet 5: Get Courier Name

{ FirebaseDatabase.getInstance().getReference("CourierService").child(FirebaseAuth.getInstance().getCurrentUser().getUid())

.child("companyName").get().addOnCompleteListener(new OnCompleteListener<DataSnapshot>() {

@Override

public void onComplete(@NonNull Task<DataSnapshot> task) {

if (!task.isSuccessful()) {

Log.e("firebase", "Error getting data", task.getException());

}

else {

Log.d("firebase", String.valueOf(task.getResult().getValue()));

tTitle.setText(String.valueOf(task.getResult().getValue())+" Dashboard");

### Courier View Orders

As shown in the figure, the courier service can check the orders by clicking the "Manage Orders" button.



Figure 4‑8: View Orders

#### Manage Order

Courier can manage orders by clicking options and marking whether the order is complete or canceled, and they can find the shortest route and distance from a given location to the drop-off location. When the courier service clicks cancel or successfully delivers the package, they receive a pop-up notification.

These functions are performed by connecting to the Firebase.

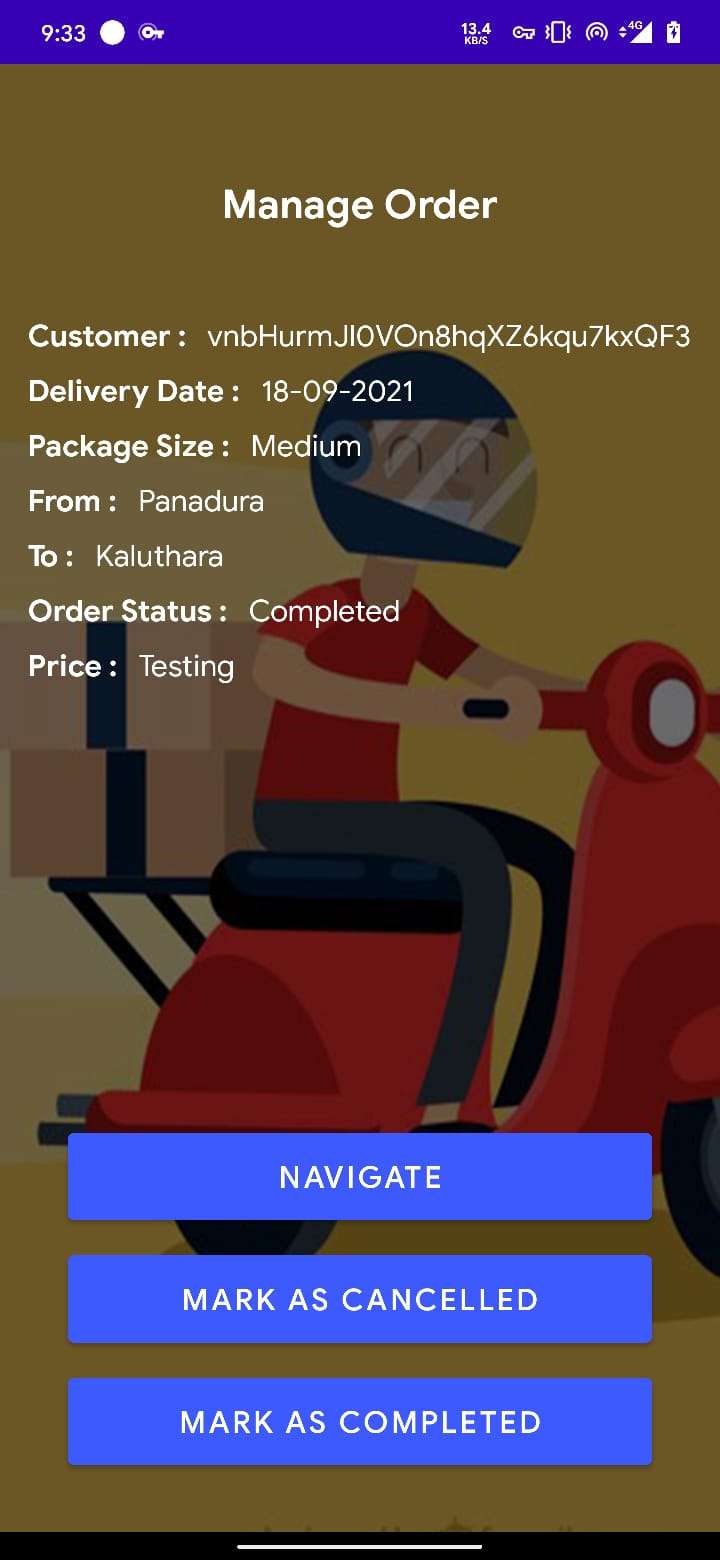


Figure 4‑9:Manage Order

#### Navigate

When the courier service clicks "Navigate," they can check the shortest route and distance from the given location to the destination, as shown below.

The navigate function works as the code snippet below, getting the Longitude and Lattitude from Firebase and sending it to Google Maps.

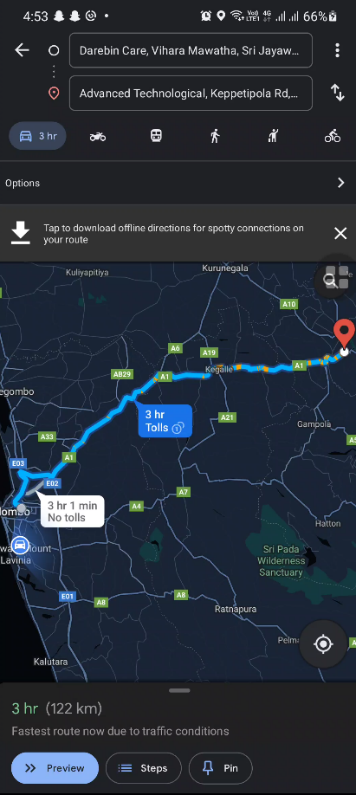


Figure 4‑10: Shortest route and distance

Code Snippet 6: Getting shortest route with distance

@Override

public void onClick(View v) {

String uri = "https://www.google.com/maps/dir/?api=1&origin="+fromLat+","+fromLong+"&destination="+toLat+","+toLong+"&travelmode=driving&dir\_action=navigate";

Intent intent = new Intent(android.content.Intent.ACTION\_VIEW, Uri.parse(uri));

intent.setClassName("com.google.android.apps.maps", "com.google.android.maps.MapsActivity");

startActivity(intent);

}

### Find a Courier

Courier services can access requests and destination patterns of other courier services to deliver packages to remote areas by partnering with them.

As in Figure 4-11, the courier service needs to add the pickup drop-off locations and the due dates for the delivery to take place. As in Figure 4-12, a courier service can check if other courier services are visiting a remote location to which they intend to go.

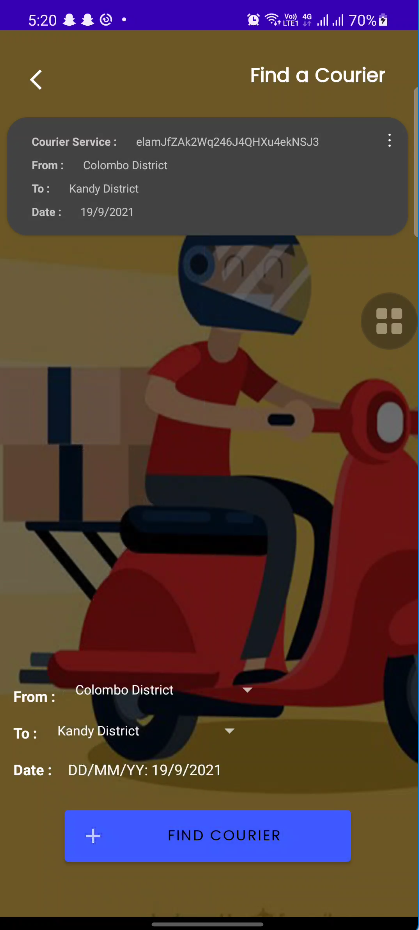




Figure 4‑11: Find a courier 1

Figure 4‑12: Find a courier 2

The code snippet below depicts the connection to the Firebase to check the same pickup and drop-off locations on a specific date.

Code Snippet 7: Finding a courier

public static class FindDestinationViewHolder extends RecyclerView.ViewHolder

{

TextView courier, from, to, date, buttonViewOption;

public FindDestinationViewHolder(@NonNull View itemView) {

super(itemView);

courier = itemView.findViewById(R.id.txtCourier);

from = itemView.findViewById(R.id.txtDFrom);

to = itemView.findViewById(R.id.txtDTo);

date = itemView.findViewById(R.id.txtDDate);

buttonViewOption = itemView.findViewById(R.id.textDViewOptions);

}

}

### Add Destination

When a courier service wants to get packages from other courier services, the courier service needs to add the destination credentials as below. All the data is sent to Firebase.

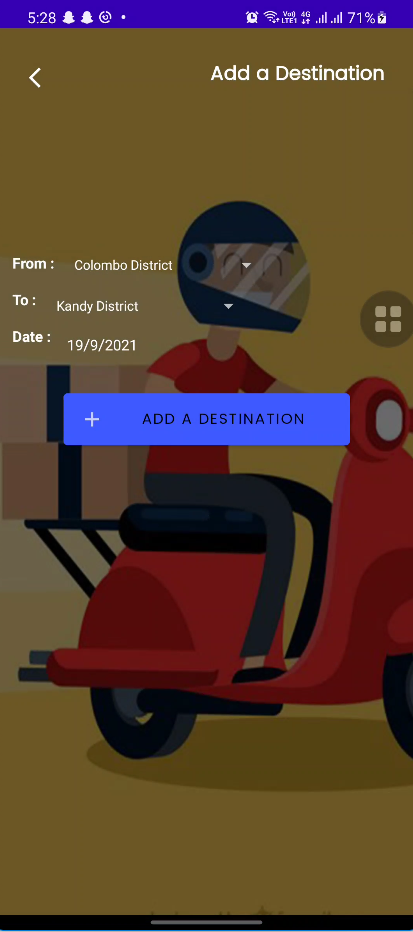


Figure 4‑13: Add Destination

## Testing

Testing is the foremost phase that must be considered serious when developing an application. As the researcher selected Agile Methodology, testing should be done parallelly to ensure that the application runs error-free when a certain phase or component is developed. Unit Tests, Integration Testing, System Testing, and Acceptance Testing are done accordingly. All the mentioned testing is discussed separately in this phase with results.

Packagely application has been tested at many levels of development. Moreover, it is manually tested by the developer.

### Unit Testing

Unit test is done by conducting test cases for every interface and function to ensure that they work correctly without bugs. All the interfaces are tested by writing test cases.

### Integration Testing

An integration test is conducted after all the functions are integrated into the system. It ensures that Packagely is an error-free application.

### Acceptance Testing

Acceptance testing was done after the development of the application has concluded to ensure that the outcome was achived on par with the project's aim.

## Test Cases

### User Application

Table 2: Log-in interface Test case

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Login Interface | | | |  |
| **Test No.** | **Task** | **Purpose of test** | **Expected result** | **Achieved** |
| 01 | Login with valid credentials | Verify whether user can login with correct credentials.. | User should be able to log in and go to make a delivery UI | YES |
| **Evidence:**                                    Figure 4‑14: Delivery UI  Figure 4‑15: Login Interface | | |  |

Table 3: Login Interface 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Login Interface | | | |  |
| **Test No.** | **Task** | **Purpose of test** | **Expected result** | **Achieved** |
| 02 | Login with invalid credentials | Verify whether user can't login with incorrect credentials.. | Show a error message | YES |
| **Evidence:**                                  Figure 4‑16: User login | | |  |

Table 4: Login Interface

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Login Interface | | | |  |
| **Test No.** | **Task** | **Purpose of test** | **Expected result** | **Achieved** |
| 03 | Login with empty fields | Verify whether user can't login with emptyt fields. | Show a error message | YES |
| **Evidence:**                                  Figure 4‑17: Login interface with empty fields | | |  |

Table 5: Signup Interface

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| signup Interface | | | |  |
| **Test No.** | **Task** | **Purpose of test** | **Expected result** | **Achieved** |
| 01 | Signup with valid credentials | Verify user can signup the application. | User should be able to signup. | YES |
| **Evidence:**                                    Figure 4‑18: Signup with Valid details | | |  |

Table 6: Make a delivery

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Make a delivery Interface | | | |  |
| **Test No.** | **Task** | **Purpose of test** | **Expected result** | **Achieved** |
| 01 | Make a order | Verify whether user make a order. | User should be able to make a order | YES |
| **Evidence:**                                    Figure 4‑19: Make a delivery | | |  |

Table 7 Make a deliver - Select a location:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Make a delivery | | | |  |
| **Test No.** | **Task** | **Purpose of test** | **Expected result** | **Achieved** |
| 02 | Select the current location | Verify whether can select the location on Google Map. | User should be able select the location on map. | YES |
| **Evidence:**                                    Figure 4‑20: Select a location | | |  |

Table 8: Select Destination

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Make a delivery | | | |  |
| **Test No.** | **Task** | **Purpose of test** | **Expected result** | **Achieved** |
| 03 | Select the destination location | Verify whether can select the location on Google Map. | User should be able select the location on map. | YES |
| **Evidence:**                                    Figure 4‑21: Select a destination | | |  |

Table 9: View Orders

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| View Orders | | | |  |
| **Test No.** | **Task** | **Purpose of test** | **Expected result** | **Achieved** |
| 01 | View users orders | Verify whether user can check their orders. | User should be able check their orders. | YES |
| **Evidence:**                                    Figure 4‑22: View Orders | | |  |

### Courier Application

Table 10: Courier Logiin

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Login Interface | | | |  |
| **Test No.** | **Task** | **Purpose of test** | **Expected result** | **Achieved** |
| 01 | Login with valid credentials | Verify whether user can login with correct credentials.. | User should be able to log in and go to courier dashboard | YES |
| **Evidence:**                                    Figure 4‑23: Courier Dashboard  Figure 4‑24: Courier Login | | |  |

Table 11: Courier Login 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Login Interface | | | |  |
| **Test No.** | **Task** | **Purpose of test** | **Expected result** | **Achieved** |
| 02 | Login with invalid credentials | Verify whether user can't login with incorrect credentials. | Show a error message | YES |
| **Evidence:**                                  Figure 4‑25: Courier login with incorrect details | | |  |

Table 12: Courier login with invalid details

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Login Interface | | | |  |
| **Test No.** | **Task** | **Purpose of test** | **Expected result** | **Achieved** |
| 03 | Login with empty fields | Verify whether user can't login with empty fields. | Show a error message | YES |
| **Evidence:**                                    Figure 4‑26: Courier login with empty details | | |  |

Table 13: Add Destination

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Add destination Interface | | | |  |
| **Test No.** | **Task** | **Purpose of test** | **Expected result** | **Achieved** |
| 01 | Add destination with correct details | Verify whether can add destination with details. | Adding the database to the database and get a popped a notification | YES |
| **Evidence:**                                    Figure 4‑27: Add destination | | |  |

Table 14: Find a Courier

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Find a courier | | | |  |
| **Test No.** | **Task** | **Purpose of test** | **Expected result** | **Achieved** |
| 01 | Find a courier to given destination. | Verify whether user can find a courier with given location. | User should be able to check cvourier services that going to that destination | YES |
| **Evidence:**                                  Figure 4‑28: Find a Courier - Show details  Figure 4‑29: Find a courier - addind details | | |  |

Table 15: Manage Orders

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Manage Orders | | | |  |
| **Test No.** | **Task** | **Purpose of test** | **Expected result** | **Achieved** |
| 01 | Manage orders that courier have | Verify whether courier can open their orders | User should be able open orders | YES |
| **Evidence:**                                    Figure 4‑30: View Orders | | |  |

Table 16: Manage orders - Option

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Manage Orders | | | |  |
| **Test No.** | **Task** | **Purpose of test** | **Expected result** | **Achieved** |
| 02 | Manage the orders | Verify whether courier can manage orders | User should be able open manage order UI | YES |
| **Evidence:**                                    Figure 4‑31: Manage Orders - Options | | |  |

Table 17: Manage Orders - Navigate

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Manage Orders | | | |  |
| **Test No.** | **Task** | **Purpose of test** | **Expected result** | **Achieved** |
| 03 | Navigata to given location to the destination | Verify whether courier can Navigate given location to the destination | User should be able open navigate easiest route and see the distance | YES |
| **Evidence:**                                    Figure 4‑32: Navigate | | |  |

Table 18: Manage orders - Cancel the order

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Manage Orders | | | |  |
| **Test No.** | **Task** | **Purpose of test** | **Expected result** | **Achieved** |
| 04 | Cancel the order | Verify whether courier can cancel their order | User should be able cancel the order and getting a popped up notification | YES |
| **Evidence:**                                    Figure 4‑33: Cancel order | | |  |

Table 19: Manage orders - Mark order completed

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Manage Orders | | | |  |
| **Test No.** | **Task** | **Purpose of test** | **Expected result** | **Achieved** |
| 05 | Mark order is completed | Verify whether courier can mark order is cancelled | User should be able mark order is completed and getting a popped up notification and order status should change. | YES |
| **Evidence:**                                    Figure 4‑34: View Orders | | |  |

# Evaluation & Discussion

This chapter discusses the functionalities accomplished during the project's life cycle. The evaluation illustrates how the application is evaluated by presenting it to third-party users to get honest feedback for the application developed. The results will be summarized and presented, and the discussions will be added by explaining the developer's achievements and difficulties faced by the researcher throughout the project. Moreover, discussing reasons which led to deviate from the original plans.

## Evaluation

### Questionnaire

To measure the potential for the artefact in the future, the developer has shared the application via a google drive link among 25 participants and sent a google form to share their experience about Packagely application. Then developer gathered an end-user evaluation from them based on their experience with the artefact. Under this chapter, the developer presents the results of the said end-user evaluation to the user and will discuss the achieved results. The full questionnaire of the beta evaluation form is attached in the appendix.

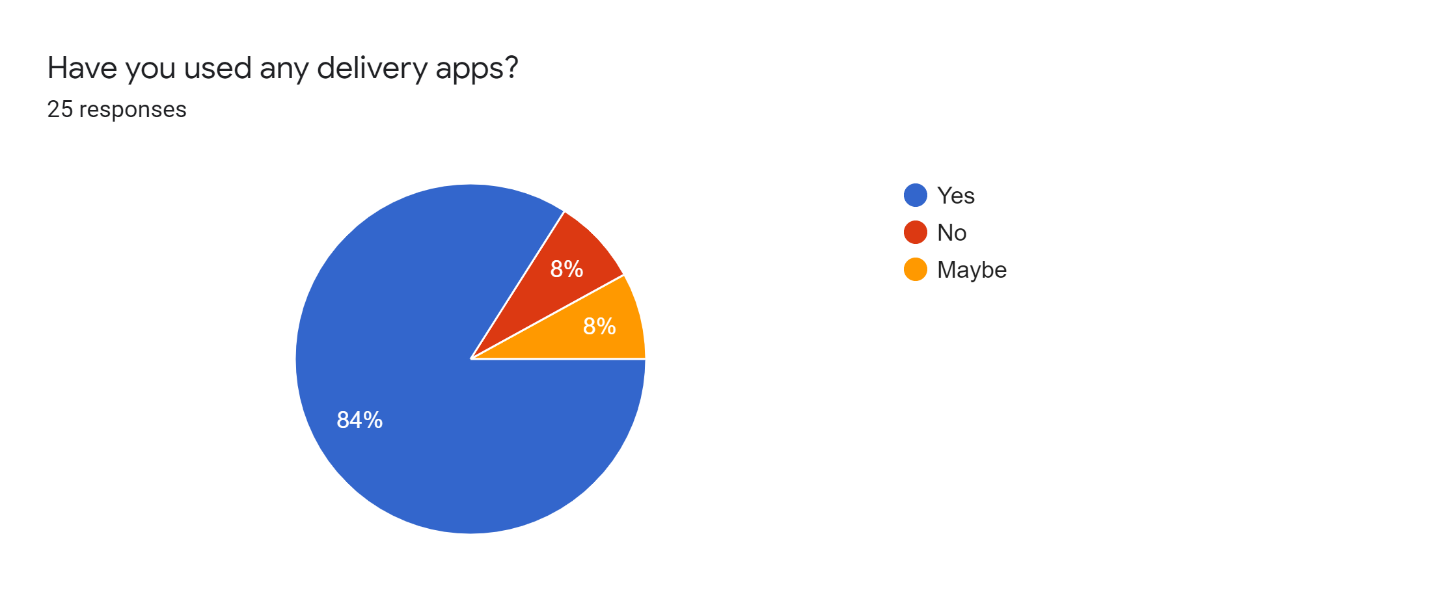


Figure 5‑1: Beta Evaluation Question 1 results

The above figure depicts the variety of replies from the users about their previous usage of similar kinds of applications. According to that, most users have a decent experience with delivery applications.

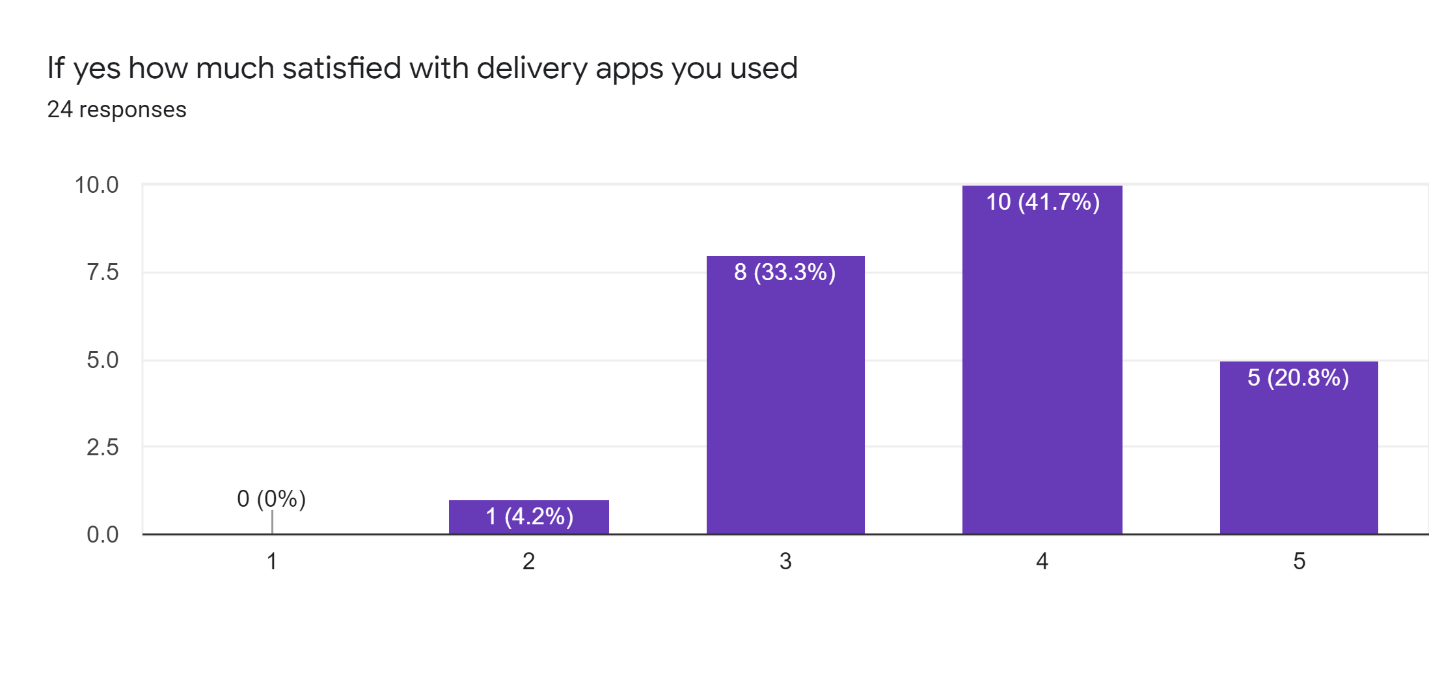


Figure 5‑2: Beta evaluation question 2 results

The above figure illustrates the satisfaction of delivery applications they have used before. According to this, most users have a neutral idea about the delivery application they have used.

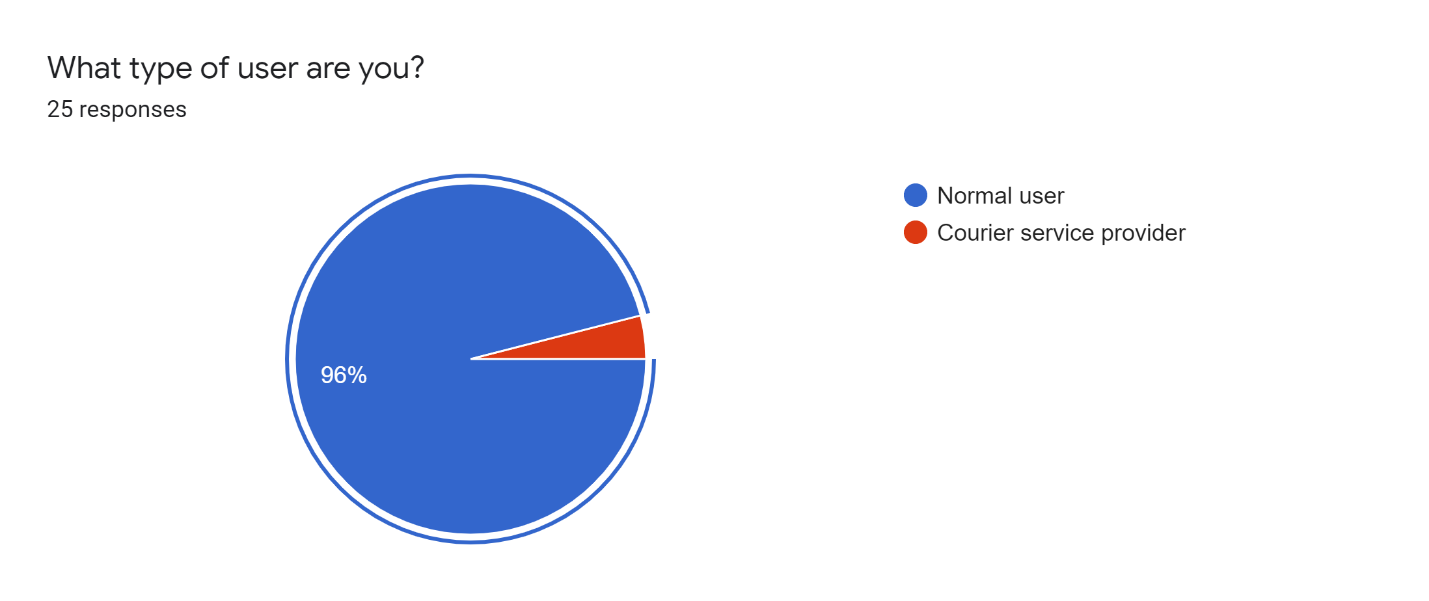


Figure 5‑3: Beta evaluation question 3 results

Since Packagely application targets two types of users', the developer needs to identify what kind of users participated for the questionnaire and gave feedback. The above figure shows what kinds of users have completed the questionnaire. According to the figure, these results show that most users who participated in the questionnaire were customers.

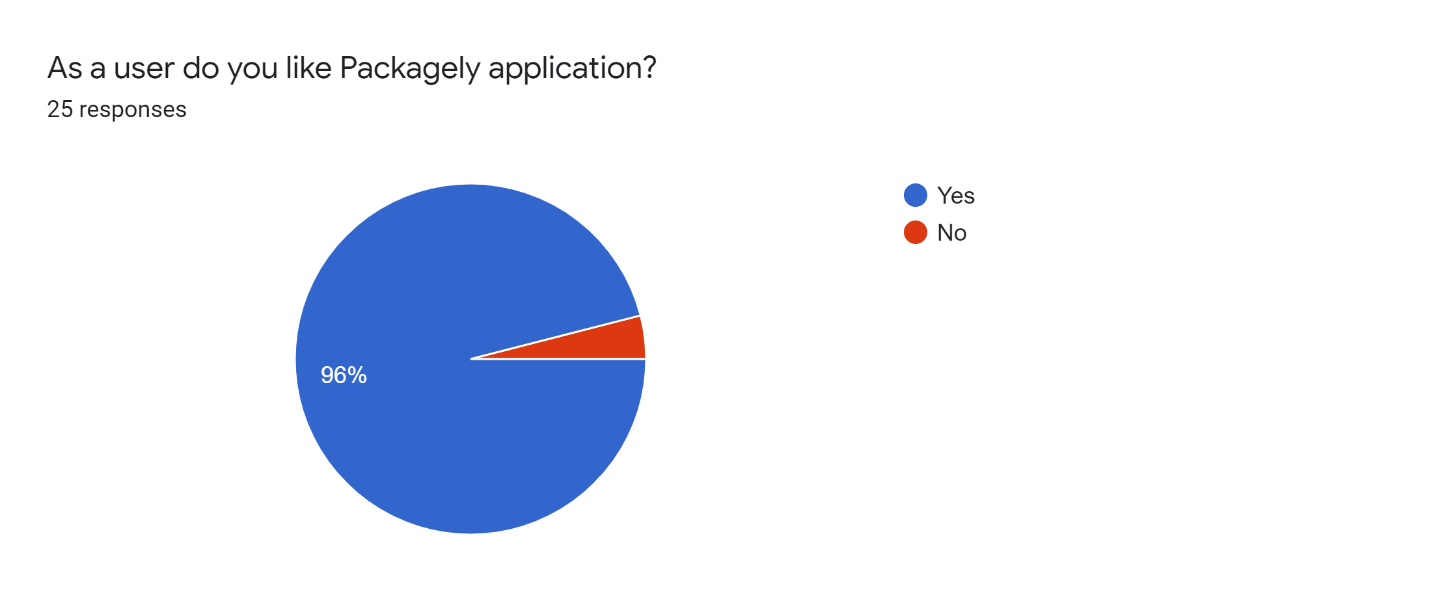


Figure 5‑4: Beta Evaluation question 4

The above figure illustrates the user's likability towards Packagely application. According to this, most users like Packagely application.

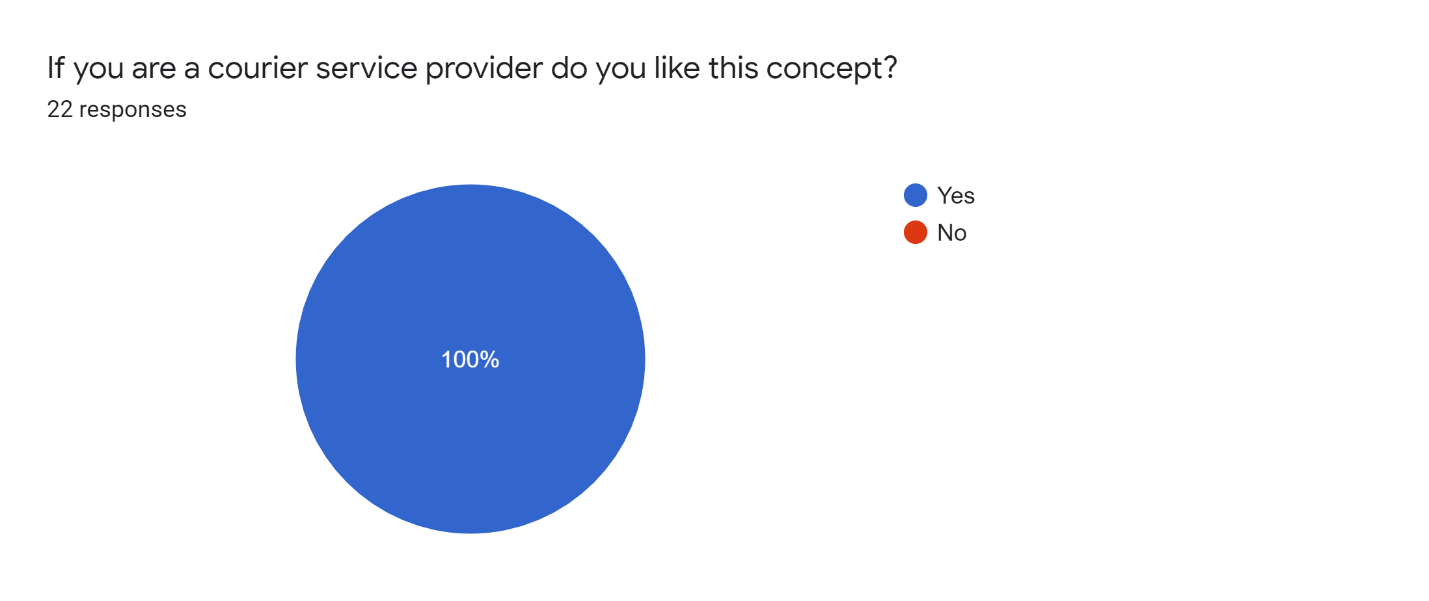


Figure 5‑5: Beta evaluation question 5

The above figure illustrates how much they like this concept if the end-user is a courier service provider. According to this evaluation, all of the users like this concept.

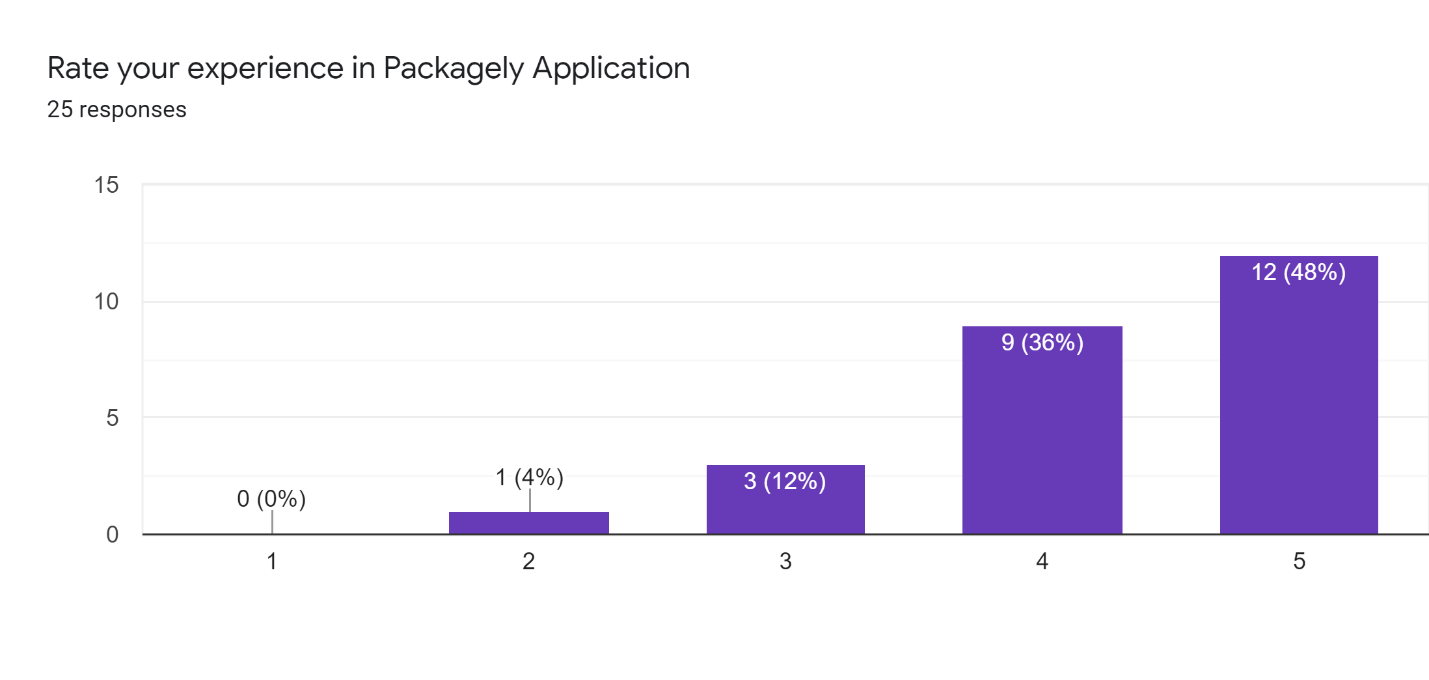


Figure 5‑6: Beta evaluation question 6 results

When considering the results obtained by the developer, most of the closed evaluation users had high satisfaction with the application rating at a staggering amount of 12 upvotes, and 9 users mostly like this. In contrast, only one of the users rated the app at a slightly lower amount. These findings proved that the developer had tackled the needs of the users via the application, and the goals set prior to development on user satisfaction were achieved.

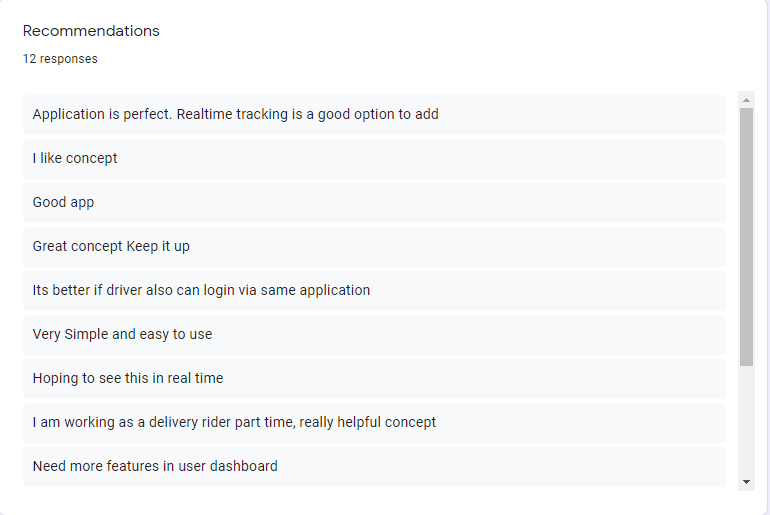


Figure 5‑7: Beta Evaluation question 7 part 1



Figure 5‑8: Beta evaluation question 7 results part2

Abobe "Figure-5-7" and "Figure 5-8" show the suggestions of users. Out of 25 participants, 12 have presented their recommendations. The majority of participants have given positive feedback on the application and future enhancements. Users have mentioned that the Packagely application is simple and easy to use.

The participant with the negative on the application has suggested adding the driver's interface to the same application. Moreover, users have mentioned seeing more functionalities on the user dashboard in future updates. In summary, most users have said that it's a good application and has given positive feedback.

## Discussion

The discussion section reveals the level of accuracy & reliability of the proposed artefact with reasons, how developer faced the objectives, and reasons for deviating from the original plan.

### Accuracy and Reliability

The developer checked the accuracy and reliability of this artefact before sharing the bet version with 25 users, where most users complemented with positive feedback. The developer has used Google Maps API and Google Firebase when composing this project. They are the most accurate map system API and cloud database, respectively, making Packagely more precise and reliable for the user.

### Reasons for the Levels and Objectives

The researcher proposed the artefact with three major applications: user application, courier service application, and driver application. Researcher started to implement this artefact with a flutter framework with dart language. Since it's not a familiar language to the developer and it's a new application, there was a lack of tutorials to study flutter and dart language. Even though there are tutorials for flutter and dart, when flutter is updated to a new version, the old tutorials don't help the developer much. Because of that developer decided to switch implementation of artefact with Java language, which is more familiar to the developer, and used Android Studio as the IDE.

The developer had to recreate the functions all over with the Java language.

Moreover, the developer was not familiar with developing an artefact using Google Map API. So the developer had to study Google Map API while integrating the feature, which led to more time waste., Also, the developer hadn't allocated much time for studying in the Gantt chart, making the developer work on other sections with a tight deadline.

Since this application needs details about courier service, the developer couldn't find the specific details physically due to the pandemic situation in the country.

### Overcoming Problems

* To study new theories and apply them to artefact, developers mostly used Github codes and youtube tutorials.
* To finish this artefact within the given timeframe, the developer worked more than the allocated time in the planning stage.
* Because of the pandemic in the country, the developer contacted courier services over the phone.
* Moreover, the developer was advised by the supervisor frequently. It would've been a rough path to the developer without the advice and motivation of the supervisor.

# Conclusion and Future Work

Conclusion and future work chapter describes the project benefits, limitations of the project, recommendations, and the future work carried out.

## Conclusion

The developer has successfully concluded most functions as proposed. The benefits of Packagely application, limitations, and recommendations will be discussed in this section. Any researcher who works in the same area and field could briefly refer to the conclusion to get an idea about the developed application.

### Benefits of the Artefact

Developer have developed this artefact mainly focussing to centralize all the courier services to a hub. By implementing this application, both users and courier services get many advantages. It's a win-win situation for both user and courier service both.

#### Benefits for the Users

* Since this is a hub for courier services, users can check their prices and select a courier service they like or select the courier service that the application suggests itself.
* Since most courier services won't pick up from the given location, users have to go to their warehouse and hand over the package. But with Packagely, users can select their location, and then courier service comes to pick their package to the given location.
* Can check their delivery process by Packagely application.
* Since Packagely gives the shortest route, users will benefit as they only need to pay for the distance traveled by courier service.

#### Benefits for the Courier Service Provider

* With Packagely, courier service providers can obtain a larger number of customers.
* Since this Packagely application connects courier services, they can connect with other courier services and send packages together. They don't need to send a vehicle to remote areas for a single package with this function. Moreover, with this function, if a courier service sends a vehicle containing space for more packages, they can collobarate with another courier service and deliver their package. It will be a win-win situation for both courier service providers.
* They will get the easiest route and can charge for the distance that they are traveling.

### Limitations

Since artefact was developed within a limited time frame, there are several limitations in this project.

Mainly this application only supports the English language, and all the UI have the English language. Because of that, it is hard to use this application for users who lack English knowledge.

And since this application is made with Google Firebase and Google Maps API, users must have an internet connection to work with this application. Also, the real-time location will not work if the user decides not to allow permission to access their mobile phone location.

### Recommendations

The developer recommends users use the Packagely Application with an Android amartphone that supports Android Version 6.0 (Marshmellow) and upwards. Before getting started with the application for the first-time user must have a proper internet connection. To get the real-time location user needs to turn on the GPS on their mobile.

## Future Work

* Since the researcher proposed making a complete artifact with three aplplications, the developer has to finish the driver application with the proposed functions.
* Develop an AI algorithm to suggest a suitable courier service,
* Find a courier service that intends to gon on the same route using an AI algorithm.
* Since this only supports the English language, the developer plans to make this application with multi languages that most locals understand.
* Packagely is only developed for the android platform, so the developer plans to implement the application with all functions for the iOS platform.
* Give a more user-friendly, clean UI with every update.

# References

* Raina, A., Rana, V.S. and Thakur, A.S., 2018. Popularity of online food ordering and delivery services–a comparative study between Zomato, Swiggy and Uber Eats in Ludhiana. International Journal of Management, Technology and Engineering, 8(3), pp.350-355.
* Riyanto, S., Ariyanto, E. and Lukertina, L., 2019. Work life balance and its influence on employee engagement "Y" generation in courier service industry. International Review of Management and Marketing, 9(6), p.25.
* Gulc, A., 2017. Courier service quality from the clients' perspective. Engineering Management in Production and Services, 9(1).
* Birla, M., 2013. FedEx Delivers: How the World's Leading Shipping Company Keeps Innovating and Outperforming the Competition. John Wiley & Sons.
* Melgarejo, P.A., Laborie, P. and Solnon, C., 2015, May. A time-dependent no-overlap constraint: Application to urban delivery problems. In International Conference on AI and OR Techniques in Constriant Programming for Combinatorial Optimization Problems (pp. 1-17). Springer, Cham.
* Brandeau, M.L., Sainfort, F. and Pierskalla, W.P., 2005. Health care delivery: current problems and future challenges. In Operations Research and Health Care (pp. 1-14). Springer, Boston, MA.
* Alshamrani, A., Mathur, K. and Ballou, R.H., 2007. Reverse logistics: simultaneous design of delivery routes and returns strategies. Computers & Operations Research, 34(2), pp.595619.
* Parragh, S.N., Doerner, K.F. and Hartl, R.F., 2008. A survey on pickup and delivery problems. Journal für Betriebswirtschaft, 58(2), pp.81-117.
* Berbeglia, G., Cordeau, J.F., Gribkovskaia, I. and Laporte, G., 2007. Static pickup and delivery problems: a classification scheme and survey. Top, 15(1), pp.1-31.
* Mahamad, S., Sulaiman, S. and Leng, W.Y., 2018, November. An integrated courier services application: a new user experience. In 2018 IEEE Conference on e-Learning, eManagement and e-Services (IC3e) (pp. 161-166). IEEE.
* Hendayani, R. and Dharmawan, M.C., 2020, July. Strategies for Improving the Quality of

Logistics Courier Services Through Priority Problem-solving Based on Multiclass

Classification. In IOP Conference Series: Materials Science and Engineering (Vol. 879, No. 1, p. 012051). IOP Publishing.

* Hu, S. and Dai, T., 2013. Online map application development using Google Maps API, SQL database, and ASP .NET. International Journal of Information and Communication Technology Research, 3(3).
* Hu, W., Kaabouch, N., Yang, H. and Wang, X., 2014. Essential android technologies and google maps APIs for location-based services. In 47th Annual Midwest Instruction and Computing Symposium.
* Google Developers. 2021. Maps SDK for Android overview | Google Developers. [online] Available at: <https://developers.google.com/maps/documentation/androidsdk/overview> [Accessed 11 May 2021].
* Google Developers. 2021. Overview | Places SDK for Android | Google Developers.[online]Availableat: <https://developers.google.com/maps/documentation/places/android-sdk/overview> [Accessed 11 May 2021].
* Purnama, B.E., 2015. Implementation of Location Base Service on Tourism Places in West Nusa Tenggara by using Smartphone.
* Sharma, M. and Morwal, S., 2015. Location Tracking using Google Geolocation API. Int. J. of Sci. Tech. & Eng, 1(11), pp.29-32.
* Singh, Y., Sharma, S., Sutton, R. and Hatton, D., 2018. Towards use of dijkstra algorithm for optimal navigation of an unmanned surface vehicle in a real-time marine environment with results from artificial potential field. TransNav: International Journal on Marine Navigation and Safety of Sea Transportation, 12.
* Teresco, J.D., 2010. A Dijkstra's algorithm shortest path assignment using the Google Maps API. Journal of Computing Sciences in Colleges, 25(6), pp.253-255.
* Lanning, D.R., Harrell, G.K. and Wang, J., 2014, March. Dijkstra's algorithm and Google maps. In Proceedings of the 2014 ACM Southeast Regional Conference (pp. 1-3).

* Hettiarachchi, M., 2021. Packagely application and current courier service problems.
* Hettiarachchi, M., 2021. Packagely application and current Delivery problems.
* Google Docs. 2021. Delivery Application. [online] Available at:

<https://forms.gle/pnFAzF7UxQKgzBLe9> [Accessed 17 May 2021].

* 2021. [online] Available at: <https://www.softwaretestinghelp.com/testers-in-tdd-bddatdd-techniques/> [Accessed 18 May 2021].
* Lee, K., Chae, J. and Kim, J., 2019. A courier service with electric bicycles in an Urban Area: The case in Seoul. *Sustainability*, *11*(5), p.1255.
* Lopez-Santana, E., Rodríguez-Vásquez, W. and Méndez-Giraldo, G., 2018. A hybrid expert system, clustering and ant colony optimization approach for scheduling and routing problem in courier services. *International Journal of Industrial Engineering Computations*, *9*(3), pp.369-396.
* Martínez-Tobón, D.I., Gul, M., Elias, A.L. and Sauvageau, D., 2018. Polyhydroxybutyrate (PHB) biodegradation using bacterial strains with demonstrated and predicted PHB depolymerase activity. *Applied microbiology and biotechnology*, *102*(18), pp.8049-8067.

# Appendix

## Appendix A Gantt Chart

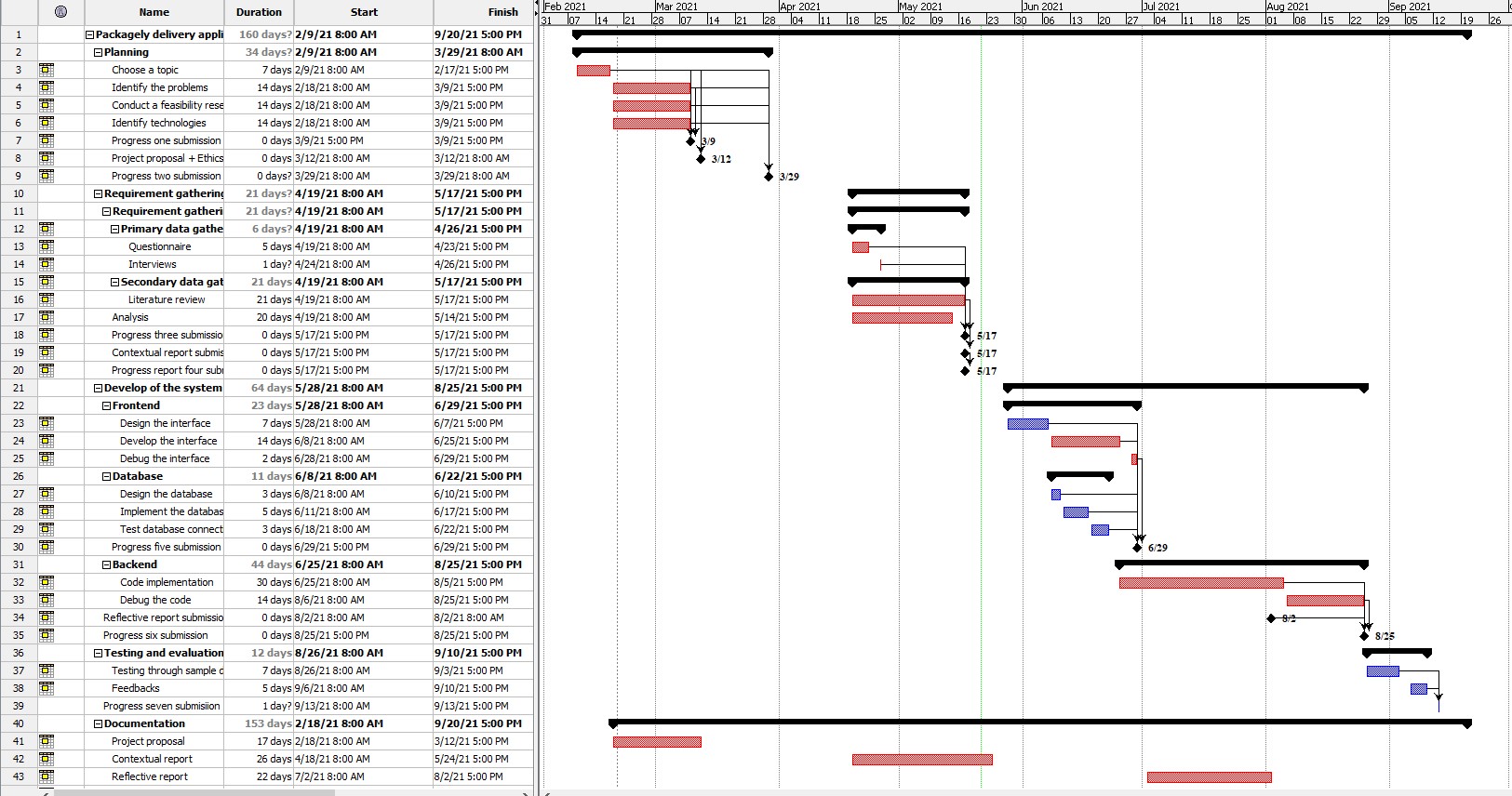


Figure 9: Gantt Chart

## Appendix B Work Breakdown Structure

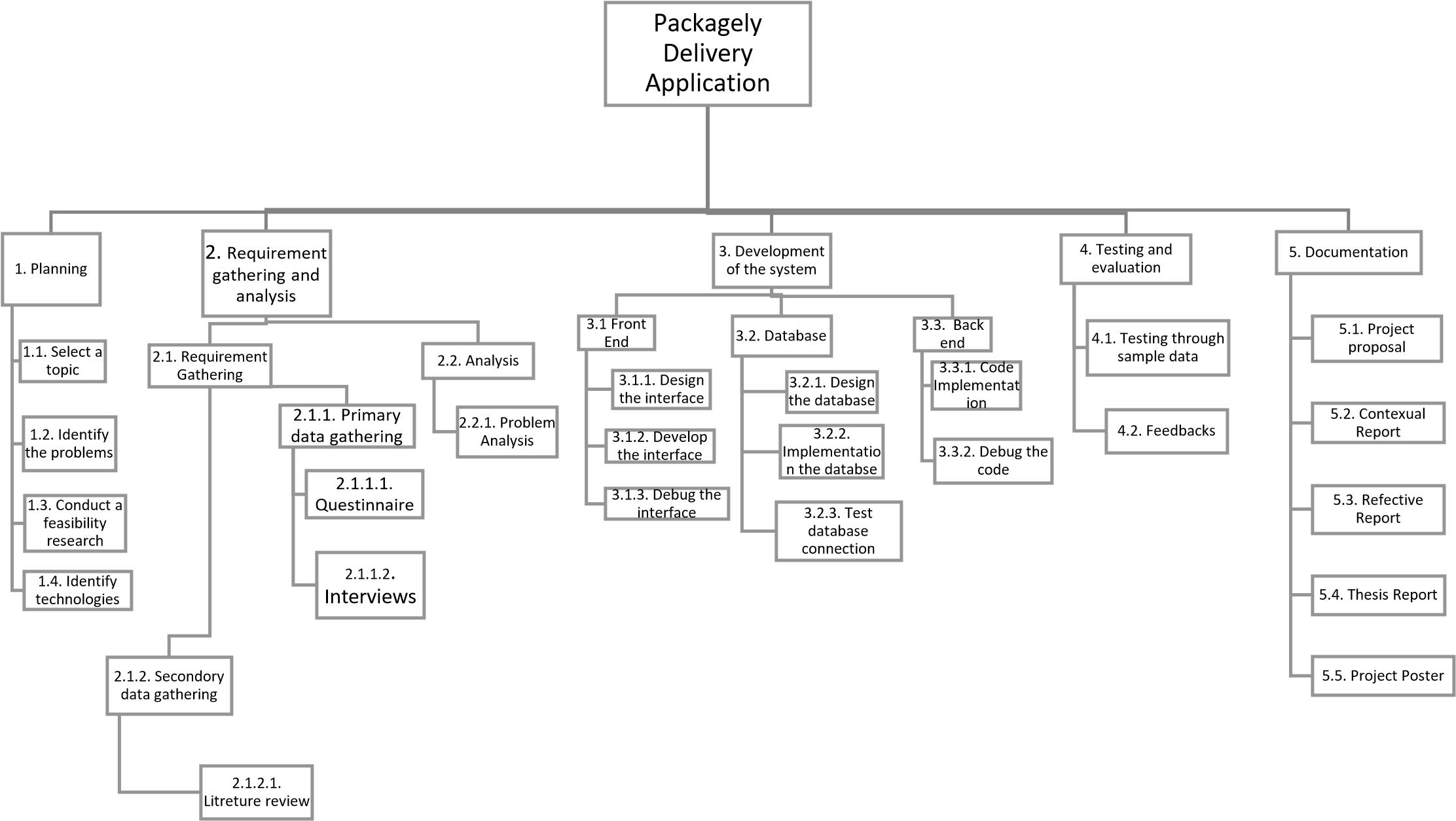


Figure 10: Work Breakdown Structure

## Appendix C Packagely Project Poster

Figure 69: Packagely Project Poster

## Appendix D Beta Evaluation Survey

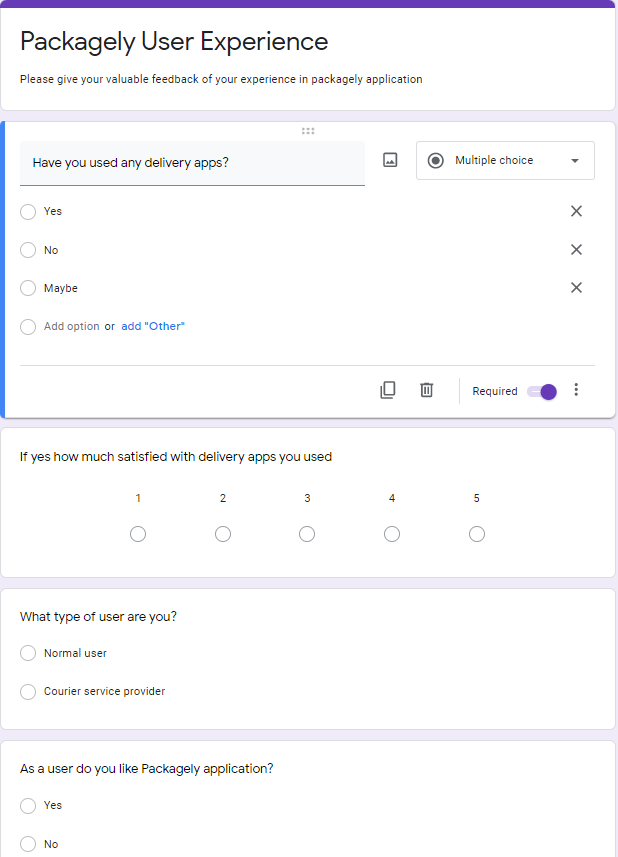


Figure 12: Beta Evaluation A

# 

Figure 13: Beta Evaluation B

## Appendix E Market Research

1. How often do you use a courier service?
   * Daily
   * Once a week
   * Once a month
   * Rarely
   * Haven't used
2. How much are you satisfied with courier(delivery) service prices?
   * Fully satisfied
   * Satisfied
   * No idea
   * Not satisfied
3. What were the difficulties you faced when using a courier service to deliver or receive goods?
   * Have to go to courier service and give the package
   * Package is damaged
   * Package isn't delivered in expected time
   * Price is not expected
   * Poor service
   * Other
4. Have you received a package from a courier service or any other delivery method?
   * Yes
   * No
   * Not Remember

1. If yes, what were the difficulties you have faced?
   * Package is damaged
   * Delivery person don't know the route
   * You weren't there when package is delivered
   * Price is not the expected
   * Poor service
   * Other
2. Do you like to courier your packages using a mobile application?
   * Yes
   * No
   * No idea
3. If you own a vehicle, do you like to deliver packages if the delivery location is in your route, for extra money?
   * Yes
   * No, I don't like
   * Depends
4. If you have a vehicle, do you like to deliver packages in your spare time for extra money?
   * Yes
   * No, I don't like
   * Depends
5. If you currently work as a carrier (Uber, Pickme or a courier service) or intend to do so in the future, would you like to deliver packages to customers?
   * Yes
   * No, I don't like
   * Depends
6. Any suggestions?