An-Najah National University

Faculty of Engineering & Information Technology

Presented in partial fulfillment of the requirements for

Bachelor degree in Computer Engineering

Graduation project 1

Package4U

Students:

Abdallah Adas

Mohammad Zaied

Supervisor:

Dr. Manar Qamhieh

m.qamhieh@najah.ed

Acknowledgment

First and foremost, we'd like to express our gratitude and thanks to Dr. Manar Qamhieh who aided us in completing our Graduation project. He didn't hesitate for a moment to support us to complete our project. Furthermore, we'd also express our appreciation to our families and friends for their great support. And we will not forget everyone who assisted and emboldened us in working on our project.

Disclaimer

is report was written by students at the Computer Engineering Department, Faculty of Engineering, An-Najah National University. It has not been altered or corrected, other than editorial corrections, as a result of assessment and it may contain language as well as content errors. The views expressed in it together with any outcomes and recommendations are solely those of the students. An-Najah National University accepts no responsibility or liability for the consequences of this report being used for a purpose other than the purpose for which it was commissioned.

Abstract

Package delivery is important because it facilitates trade, provides convenience and simplifies the process of sending and receiving packages. It provides an efficient and convenient way for users to track their shipments and receive real-time updates on the status of their packages. This transparency helps build trust, provides peace of mind for both senders and recipients, creates job opportunities for a wide range of drivers, warehouse workers and employees, and greatly impacts our daily lives, the operation of businesses and economies. It continues to evolve to meet the changing needs of consumers and businesses in an interconnected world.

The package delivery company project includes building a mobile application for customers, drivers, and the manager, and a website for employees and administrators. At the end of this project, the application was able to provide many services, the most important of which are: tracking packages, creating financial reports, auditing financial accounts, and distributing packages on a daily basis to drivers according to the line. The driver's route (his work area), and determining the driver's location on the map so that the company manager can monitor the drivers while driving.

Contents

List of Figures

List of Figures

Chapter 1

Introduction

1.1 General Background

Package delivery is rapidly evolving, and the emergence of mobile applications has revolutionized the way consumers interact with package delivery services. The package delivery company's mobile application is an easy-to-use platform that connects senders and recipients, as well as managers, employees and drivers, providing a seamless and efficient package management experience. This app typically includes features such as package tracking, real-time notifications, delivery scheduling, financial reporting, and driver management.

1.2 Objectives

The primary objectives of developing a mobile application for a package delivery company are as follows:

1. Improve user experience: By creating an easy-to-use interface for both senders and recipients to simplify the process of sending, tracking, and receiving packages.

2. Enhancing operational efficiency: Streamline internal processes by providing tools for creating invoices and reviewing financial accounts within the app, reducing manual paperwork and administrative overhead.

3. Optimize package distribution: Implement a feature to efficiently distribute packages to drivers based on their daily trip routes and work areas. This helps improve delivery methods, reduce delays, and improve overall delivery speed.

4. Improving financial management: Develop in-app functionality to facilitate financial management, including invoicing, payment tracking, and reporting. This can enhance the financial transparency and accuracy of the company.

5. Real-time driver monitoring: Enabling real-time tracking of drivers' locations on the map, allowing the company manager to monitor drivers during their working hours. This ensures accountability, security, and adherence to delivery schedules.

6. Customer communication: Integrate communication features like notifications and alerts to keep customers updated on the status of their packages. Enhance overall customer satisfaction through improved communication.

1.3 Importance

The importance of mobile applications for package delivery companies lies in their ability to meet market demands. The key points:

1. Market Demand: The surge in e-commerce activities has increased the demand for efficient and reliable package delivery services. The mobile application meets this demand by providing a convenient and modern solution.

2. Competitive Advantage: In a competitive market, having a well-designed and functional mobile app sets a company apart, demonstrating a commitment to innovation and customer-focused services.

1.4 Organization of the report

The report is organized into the following sections for clarity:

1. **Introduction**: Provides an overview of the importance of package delivery in e-commerce. And determine the goals of developing the mobile application and website.
2. **Theoretical Background and Previous Work:** Mentioning what previous works lack and what distinguishes us from our application.
3. **Methodology:** Details the materials and methods employed throughout the project, including a thorough description of the experimental setup and the web and mobile application development process. Outlines the steps taken to achieve the project objectives
4. **Results and Analysis:** Showcases the outcomes obtained from the project, including results from the Package4u application.
5. **Discussion:** Delves into a comprehensive analysis of the results, highlighting features, benefits, and limitations of the Package4u. Addresses challenges faced during the project and offers recommendations for future enhancements.
6. **Conclusions and Recommendations**: Concludes the report by summarizing key findings, reaffirming the significance of the work, and highlighting its potential impact. May also include reflections on the overall project experience and suggestions for further research
7. **References**: Lists all references cited throughout the report following the conclusion.

Chapter 2
Theoretical Background and Previous Work

We have reviewed several previous projects similar to our idea, and we have developed them by adding several features, including the manager's ability to track the drivers' location on the map, the customer's ability to track the status of the package, know its condition and location, and display financial statements and accounts to the manager.

Chapter 3
Methodology

3.1 Planning

Before starting to implement the project, we carried out a planning process to complete this project. This planning process included the technology that we will use in completing this project. This stage was very important to meet the technical requirements of the project and ensure scalability and maintainability of the system. The goal of the project was also determined, which is to develop a system. Comprehensive package delivery management that meets the specific requirements of administrators, managers, employees, drivers and customers. Also included in the planning process is feature specifications. We have precisely defined the features and functionality required for each user role, ensuring a well-defined scope for the project, all of which we will discuss in this section of the report.

After completing this planning process, the UML Diagram was drawn:

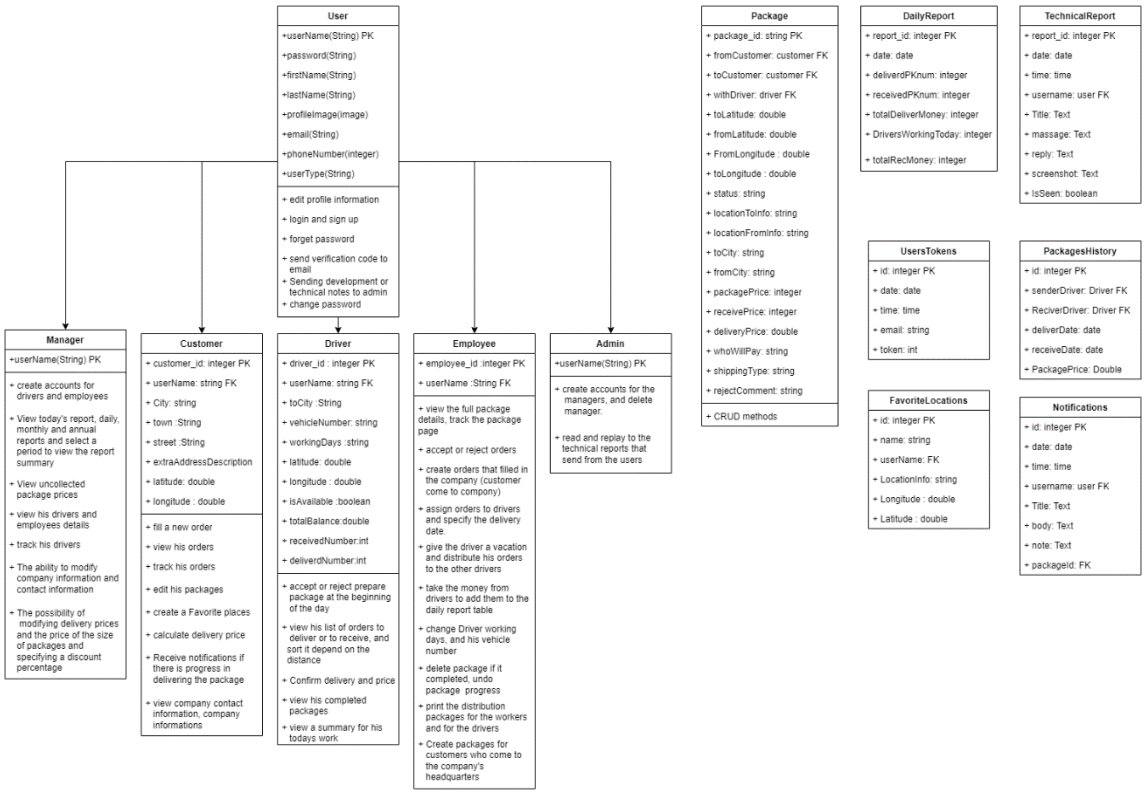

Figure 1: Package4U UML Diagram

3.2 Development Tools

1. **Visual Studio Code:**
   We have used the VSCode program to develop both the front-end and the back-end user interface, because VSCode provides us with the ability to do debugging and it has built-in Git integration, and because it contains many additions that helped us in the development process and running the program on the mobile phone.

Figure 2: Visual Studio Code

2. **GitHub**
   In our project, we used GitHub, which enables us to collaborate on the same code and have a copy of this code on the Internet. GitHub enabled us to track work on the project by writing commits on every change that occurred in the project. It also enabled us to undo some of the changes that we had dispensed with.

Figure 3: GitHub Logo

3. **Postman**
   We heavily relied on Postman as a tool for sharing requests and responses with the front-end user interface programmer, as we created a workspace in which all endpoints were published.

Figure 4: Postman

4. **MySQL Workbench**
   We used MySQL Workbench to be able to see all the changes and updates to the database we created.

Figure 5: MySQL Workbench

5. **Android Studio**
   We used Android Studio to be able to create Android devices on PC.

Figure 6: Android Studio

6. **draw.io**
   We used draw.io to draw the UML Diagram

Figure 7: draw.io

7. **Pub.dev**
   We used the pub.dev website so that we could obtain and search for the Dart libraries that they used in the development of the mobile application.

Figure 8: Pub.dev

3.3 Languages and Techniques

3.3.1 Agile Software Development Approach

By adopting an agile software development approach, our project benefited from a dynamic and iterative methodology that greatly contributed to the success of our package delivery management system. The flexible framework has allowed for incremental development of features and functions, ensuring flexible response to evolving requirements. By dividing the project into manageable sprints, each with specific objectives and deliverables, we enhanced collaboration and transparency within the development team. This iterative process facilitated continuous testing and improvement. The agile methodology not only enhanced our team's responsiveness to feedback, but also ensured a streamlined development process, which ultimately contributed to the successful and timely delivery of a robust application.

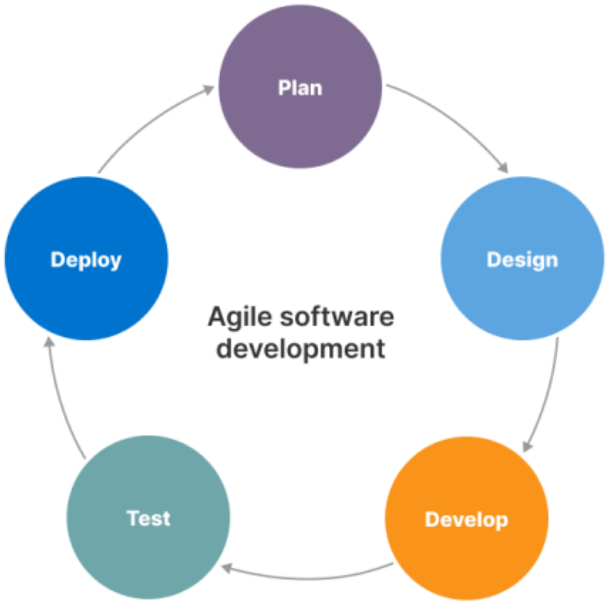

Figure 9: Agile process

3.3.2 Flutter

Flutter, developed by Google, is a cross-platform framework that enables the creation of high-performance mobile apps with a single codebase. Its hot-reload feature accelerates development, fostering faster iteration. Flutter's expressive UI toolkit, rich widget library, and consistent design across platforms streamline the development process, while its native performance ensures a smooth and responsive user experience on both iOS and Android.

Figure 10: Flutter

3.3.3 Node.js with Express.js

Node.js with Express.js is a powerful combination for building API servers due to its non-blocking, event-driven architecture, ensuring scalability and high performance. Express.js, a minimal and flexible framework, simplifies route handling, middleware integration, and provides a robust ecosystem for building RESTful APIs. The asynchronous nature of Node.js enhances responsiveness, making it well-suited for handling a large number of concurrent requests in API-centric applications.

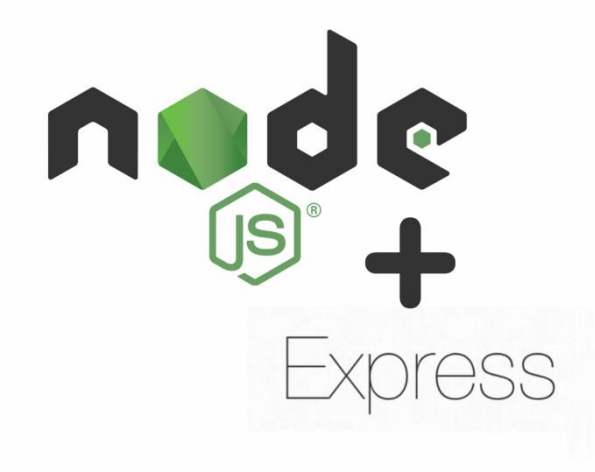

Figure 11: Node.js with Express.js

3.3.4 REST API

We used the API rest method to exchange incoming requests between the front-end and the back-end. On the frontend, we used the http library to send requests to the backend, and the backend in turn sends responses via the Express JS framework, the response is in JSON format. And this was the most suitable method for our project because it provides a unified way for mobile applications and websites to communicate over the Internet, enabling data exchange, integration, and interaction between different systems in a simple and stateless way.

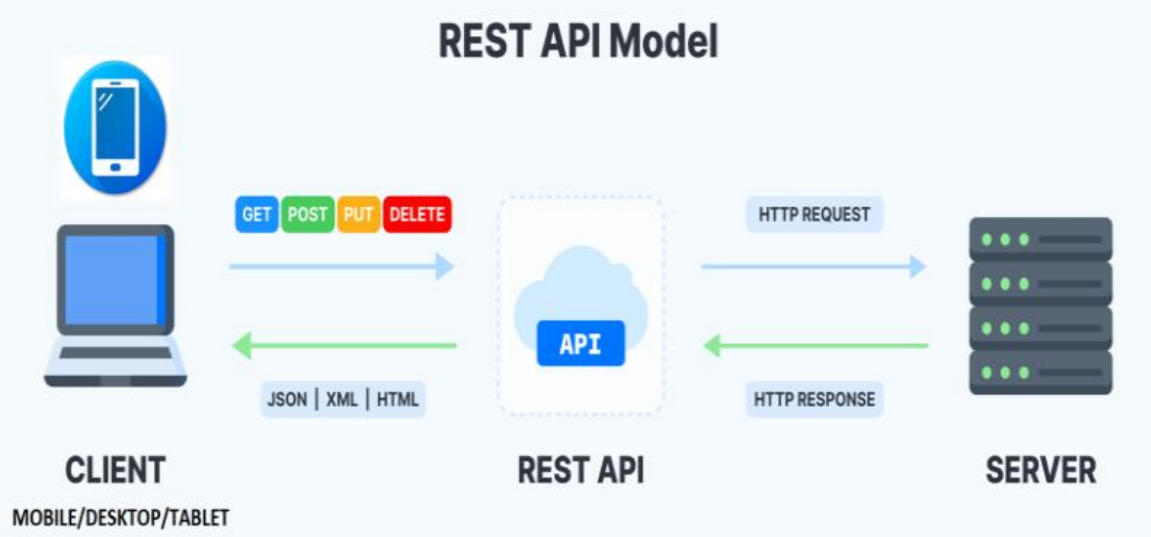

Figure 12: REST API Model

3.3.5 Google Map API

In our application, we needed to use the Google Maps API so that the customer can determine where the package is picked up and where it is delivered on the map, and this in turn helps the driver deliver the packages. He can use our application and see his location and the location of the package. We also used maps in the driver tracking feature by the manager, so the driver's location is updated periodically on the map and watching the driver move as well. Maps were also used to determine the customer's favorite places. The customer could select a point on the map and then save this location and use it more than once, such as the home or the workplace.

Figure 13: Google Map API

3.3.6 Firebase Messaging API

In our project, we used the Firebase Messaging API so that we can provide the notifications feature on mobile devices, where the notification information is sent to the API and then reaches the intended device. We also used the Firebase Messaging library so that we can receive notifications if the program is in application mode, which is in use, background mode, or in terminated mode.

Figure 14: Firebase Messaging API

3.3.7 Axios & Http Library

In our project, we used the Axios library, which enabled us to send http requests in Node.Js, specifically in the part of sending notification requests to the Firebase Messaging API for the mobile phone to receive them.
We used the http library in Flutter so that we can send requests to the backend and get responses from it.

Figure 15: Axios

3.3.8 Sequelize

Sequelize is a Node.js ORM that simplifies database interactions, offering abstraction, validation, relationships, and migrations, enhancing productivity and code maintainability. It supports multiple SQL dialects and provides a model-driven approach for working with databases in JavaScript.

Figure 16: Sequelize

3.4 User interfaces

In this section, we talk about each user of our application and all the permissions he has, and we will attach screenshots for clarification, we will also talk about the registration and sign in pages.

3.4.1 User

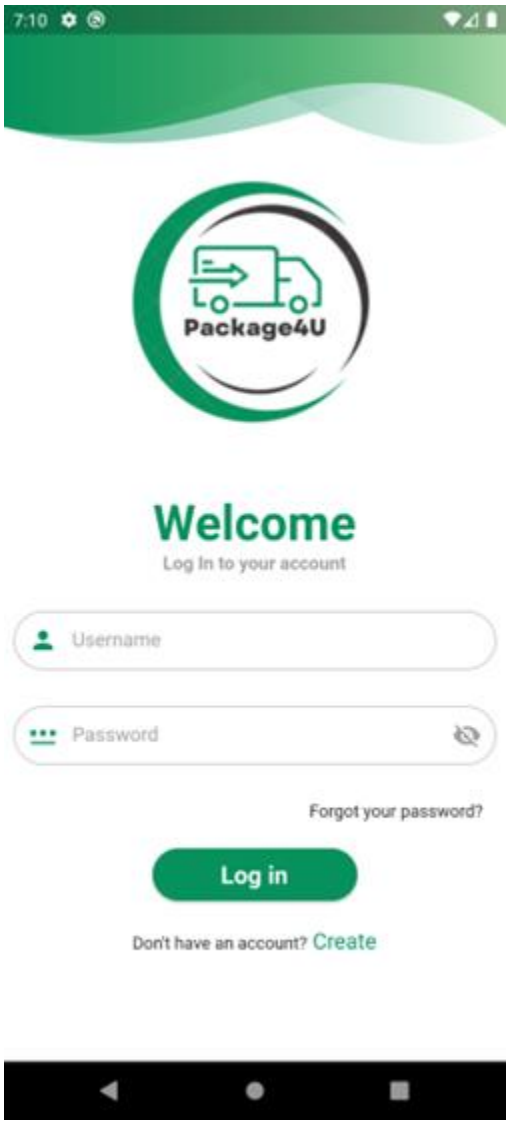

Figure 17: Sign In

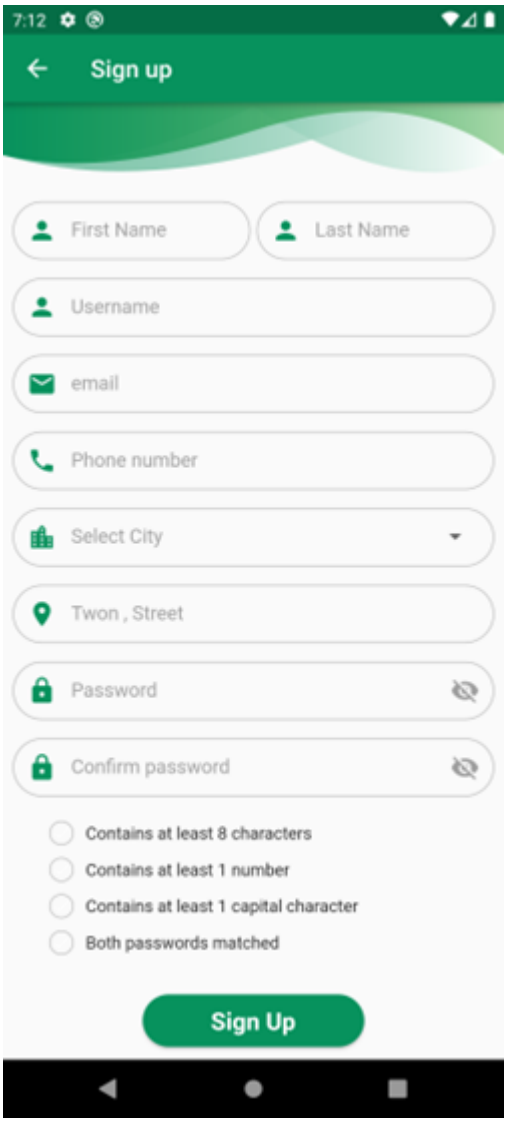

Figure 18: Customer Sign Up

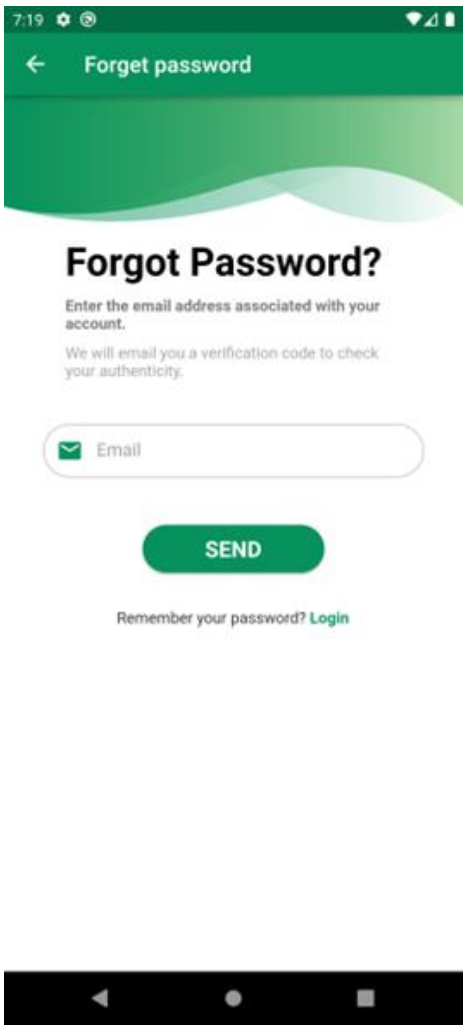

Figure 19: Forgot Password

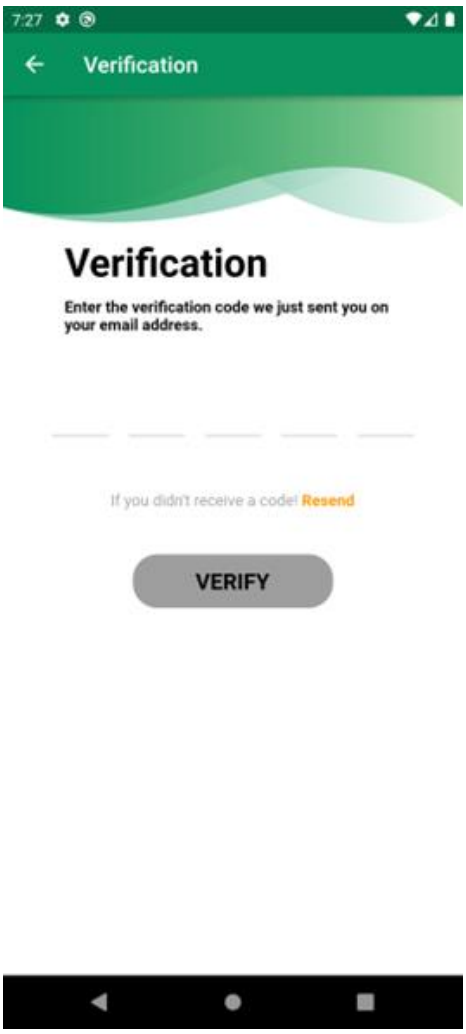

Figure 20: Verification

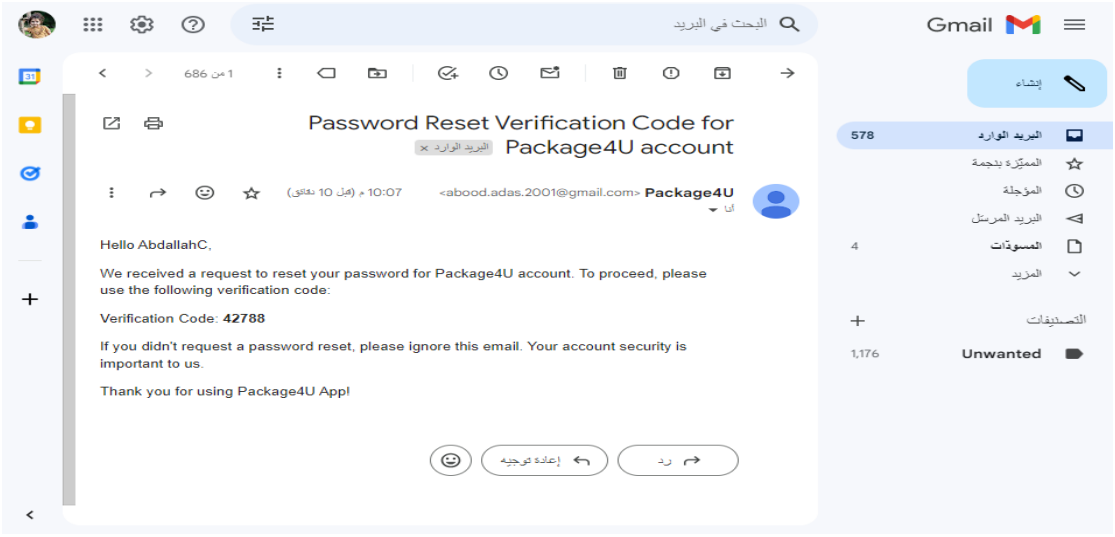

Figure 21: Password Reset Verification Code for Package4U account

3.4.2 Customer

1. Create packages.
2. View their own packages that they will receive.
3. Ability to cancel or modify the package if an hour has passed since the request for package delivery or if the employee has not approved the package yet.
4. View company information, contact information, and the company's location on the map.
5. Customers can save favorite locations where they usually receive packages, such as home or work.
6. Calculate the cost of package delivery without the need to enter the package creation page.
7. Track the status and actual location of packages.
8. Receive notifications when the status of the package changes.

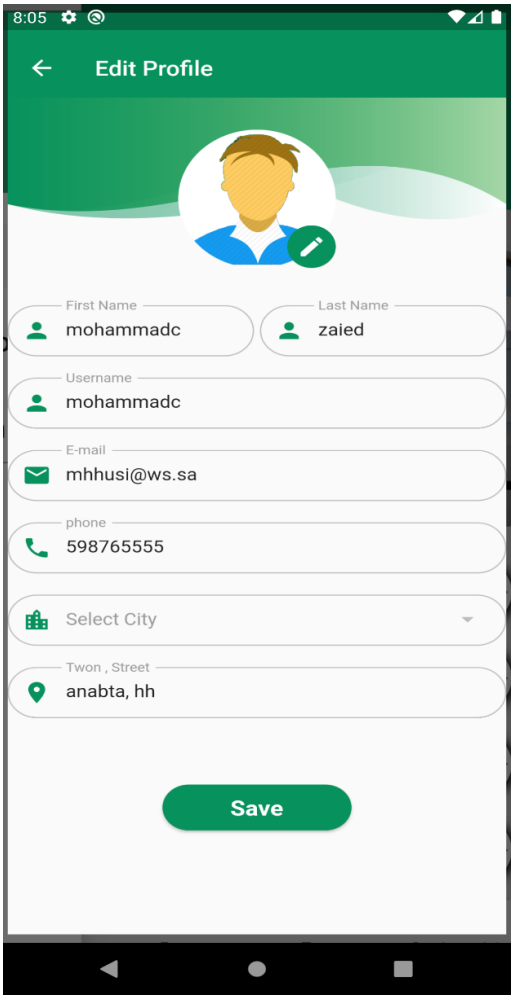

Figure 22: Edit Profile for customer, driver, Manager, employee

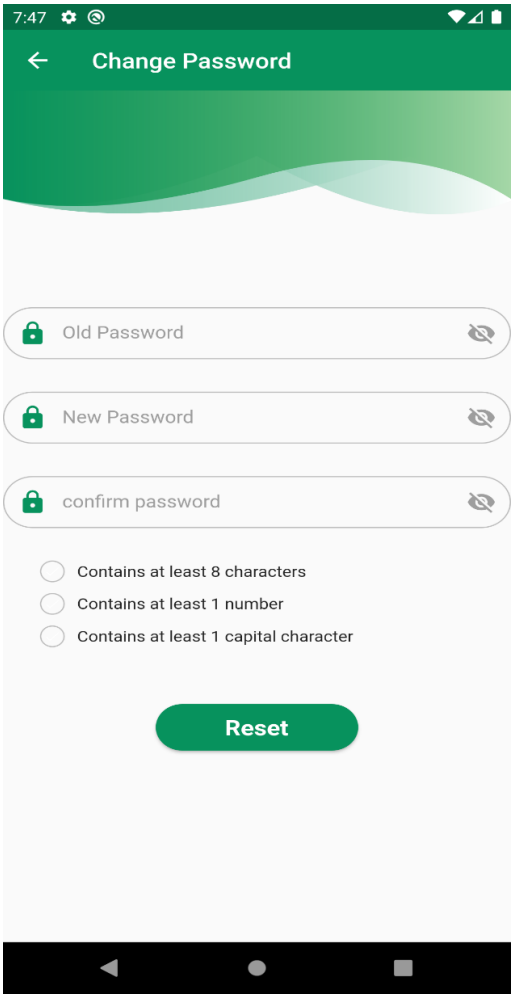

Figure 23: Reset Password

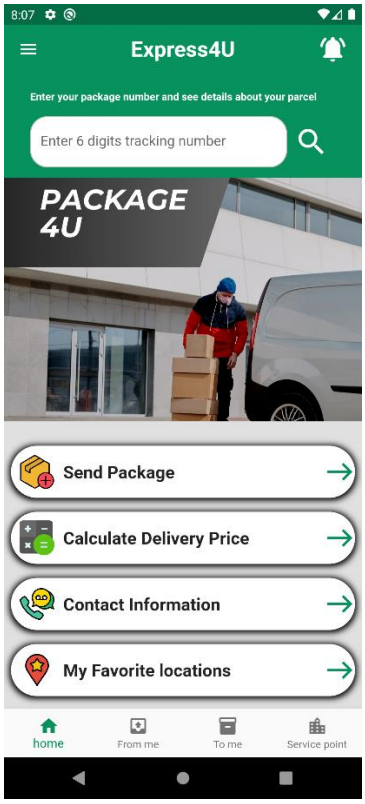

Figure 24: Home Page Customer

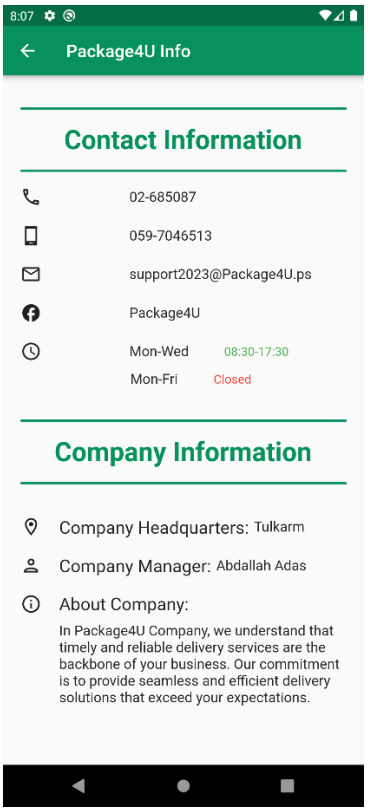

Figure 25: Company Information

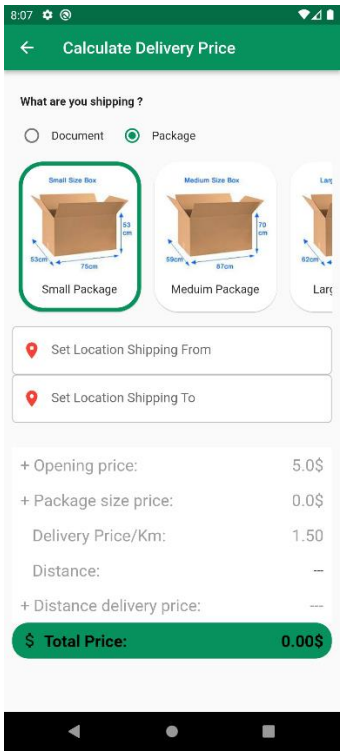

Figure 26: Calculate Delivery Cost

Figure 27: Service Point

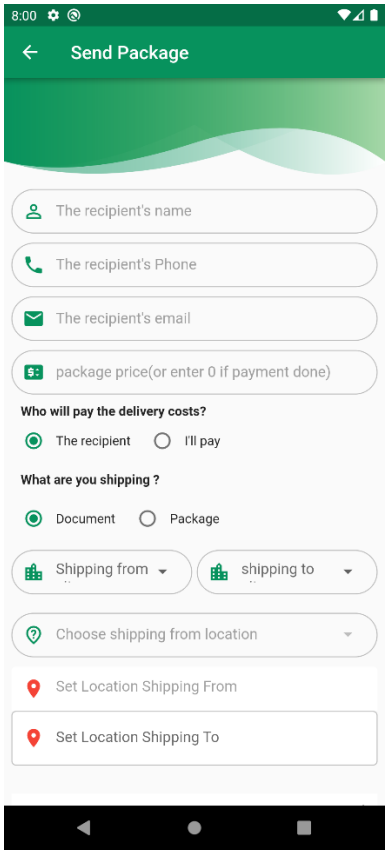

Figure 28: Create package 1

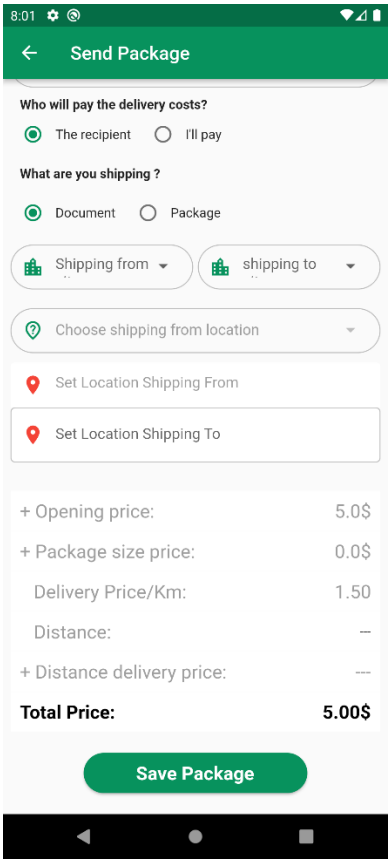

Figure 29: Create package 2

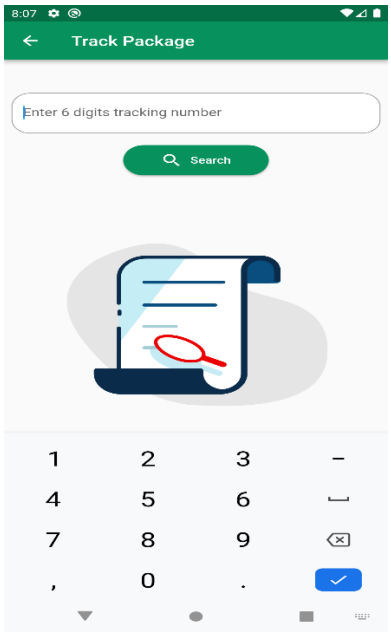

Figure 30: search Package

Figure 31: Track Package Completed

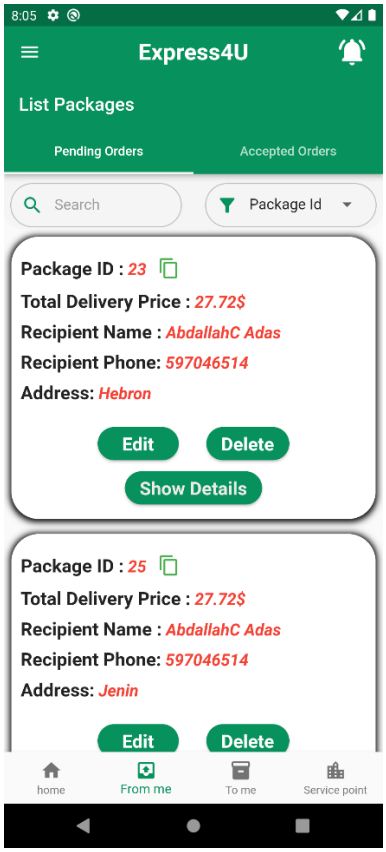

Figure 32: Packages From Me Pending

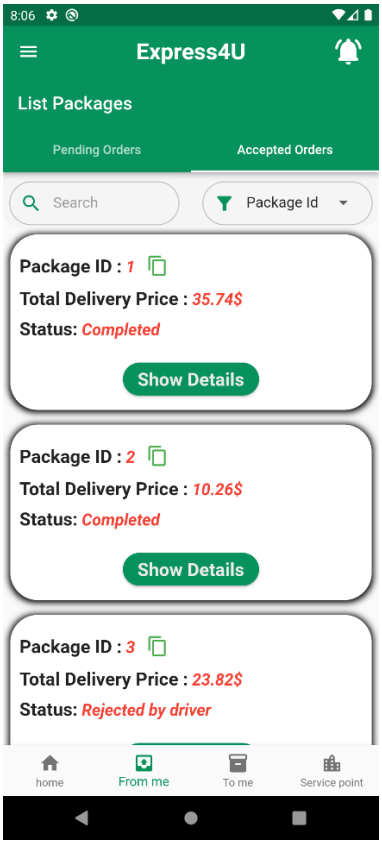

Figure 33: Packages From Me Accepted

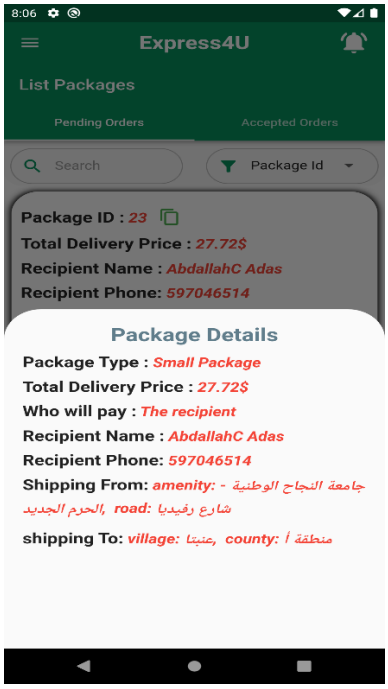

Figure 34: Package Details

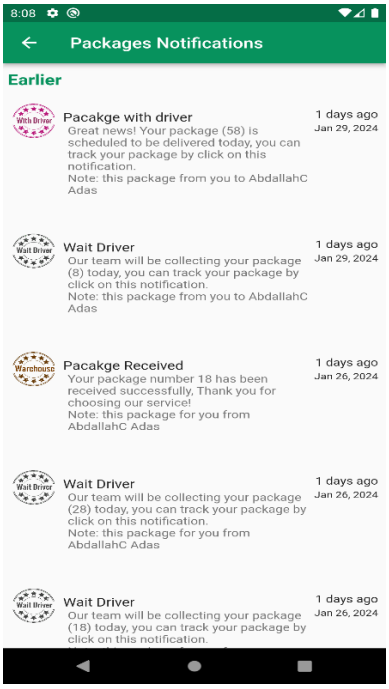

Figure 35: Notifications

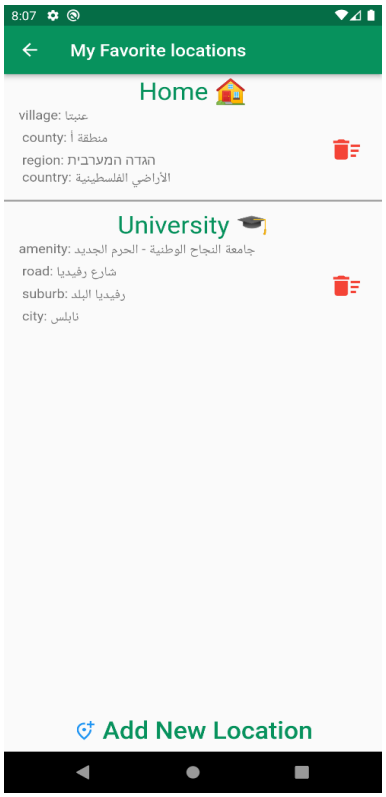

Figure 36: My Favorite Locations

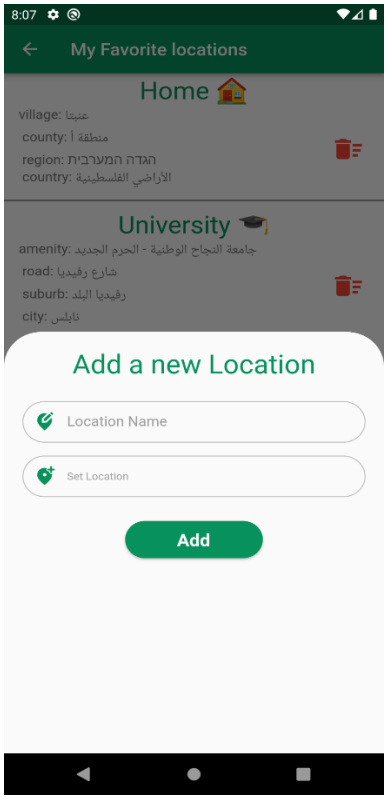

Figure 37: Add new Location

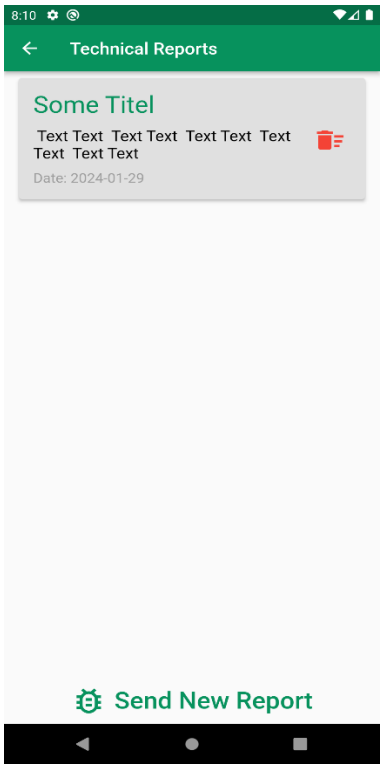

Figure 38: Technical & Improvement Reports

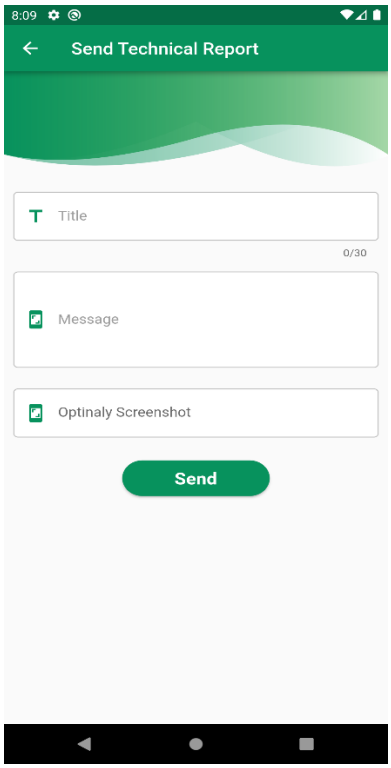

Figure 39: Send New Report

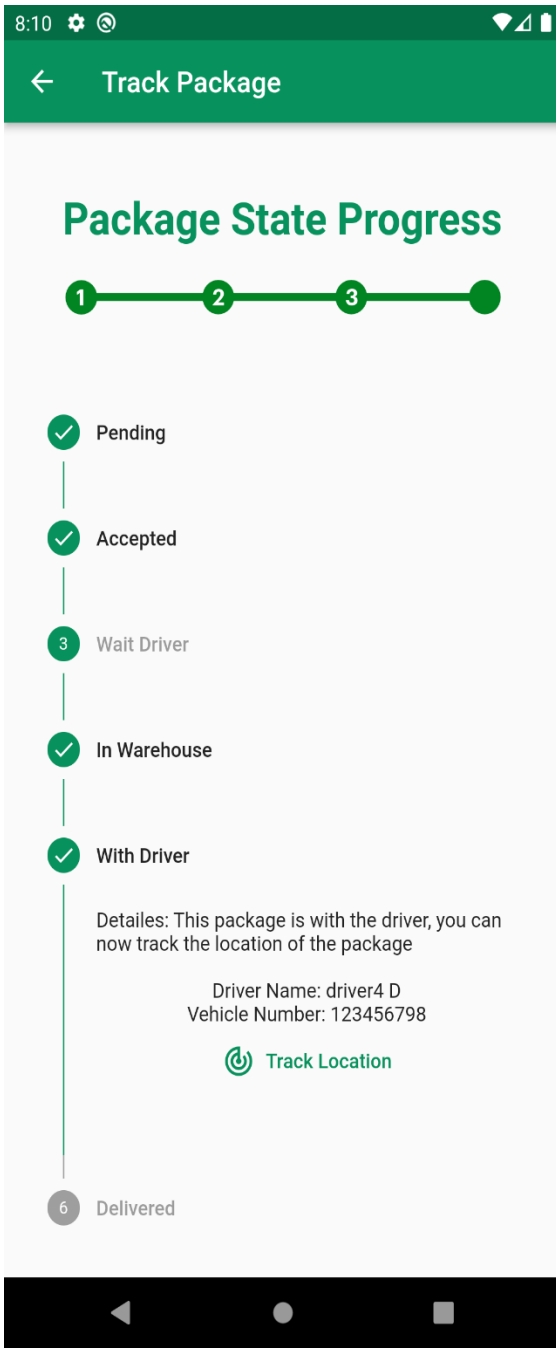

Figure 40: Track Package Status (With Driver)

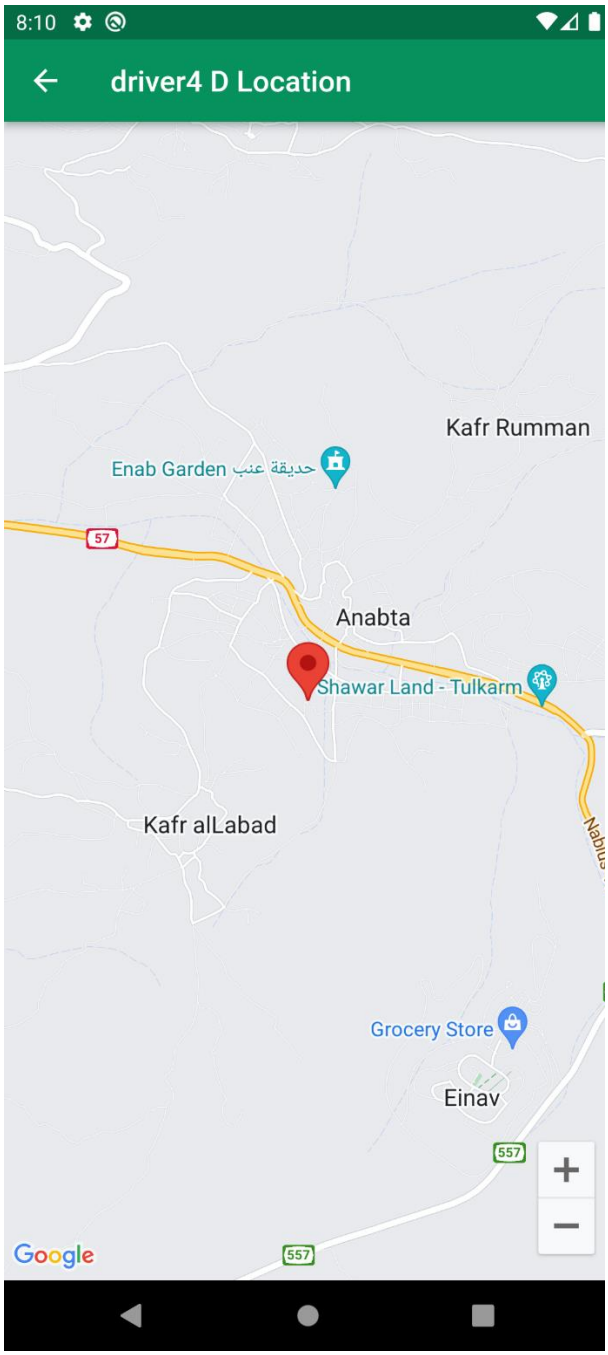

Figure 41: Track Driver Location

3.4.3 Employee

1. Ability to modify driver information such as workdays, change vehicle numbers, and grant leave to drivers.
2. Print and distribute papers to warehouse workers and drivers. These papers contain information about packages to be worked on that day.
3. Collect money from drivers.
4. Ability to accept or reject new packages.
5. Distribute packages to drivers on the same day or distribute packages over multiple days.
6. Ability to view all package information and retract progress in case of a mistake by the driver or employee. Also, the ability to delete completed or rejected packages.
7. Ability to provide customers with any information about the package's status if the customer has not installed our application.

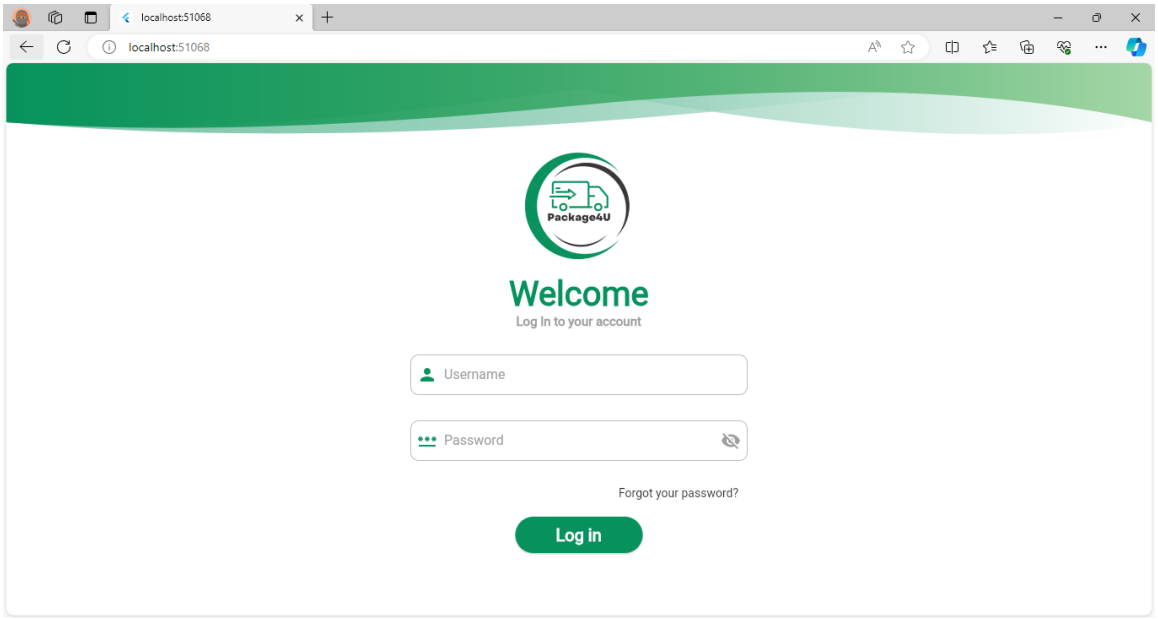

Figure 42: Employee Sign in

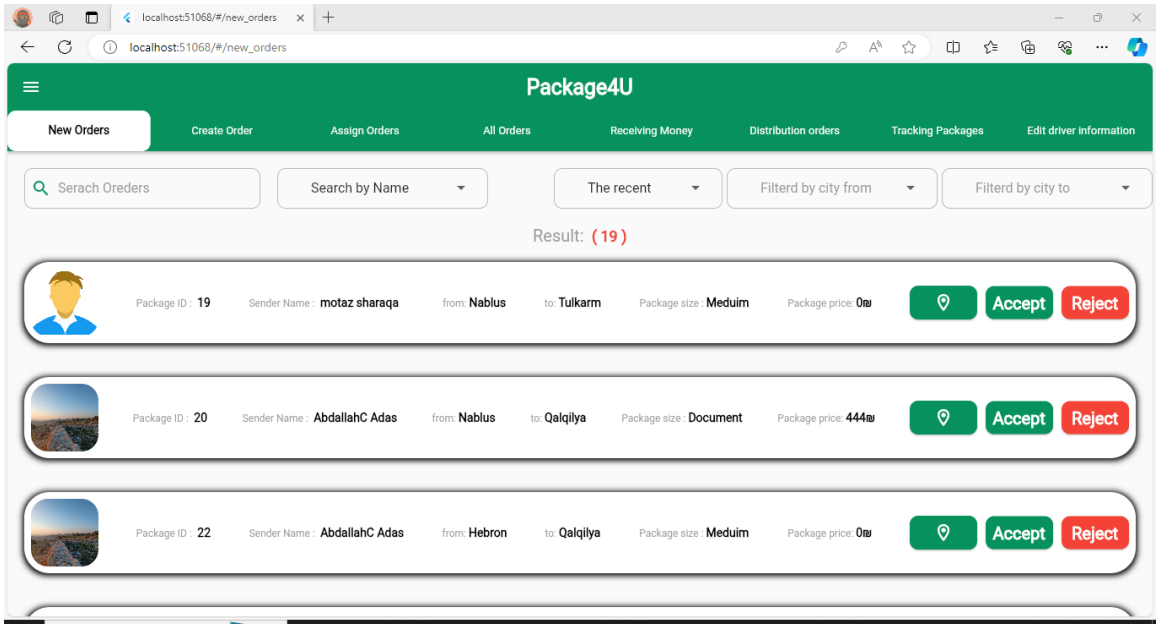

Figure 43: New Packages

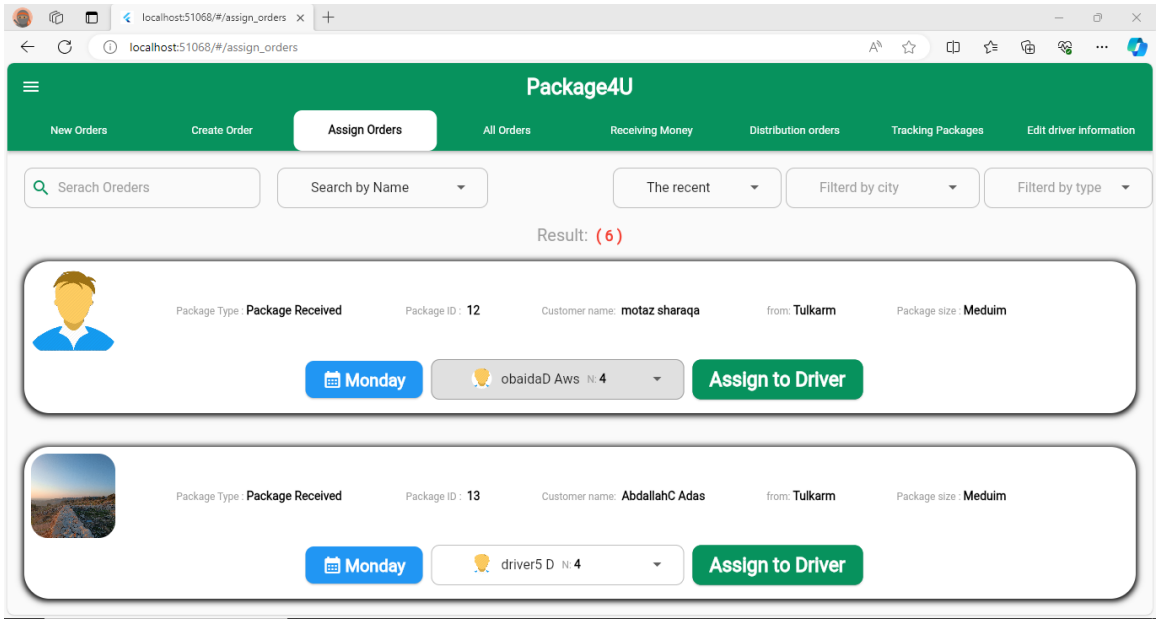

Figure 44: Assign Packages to Drivers

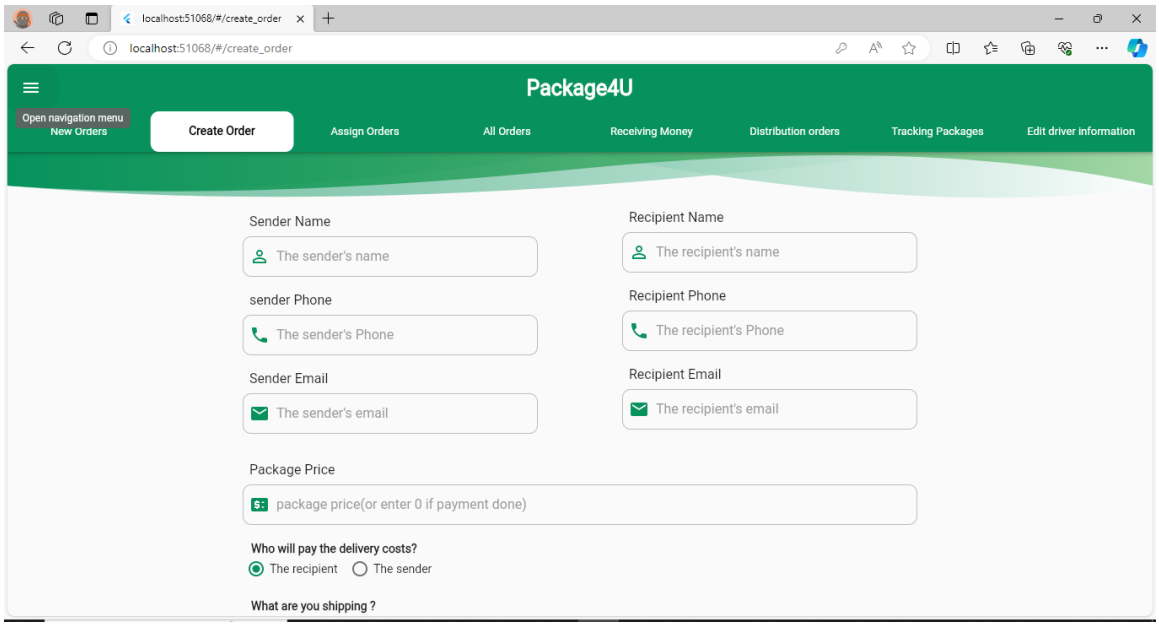

Figure 45: Create new package for customers how's come to the company

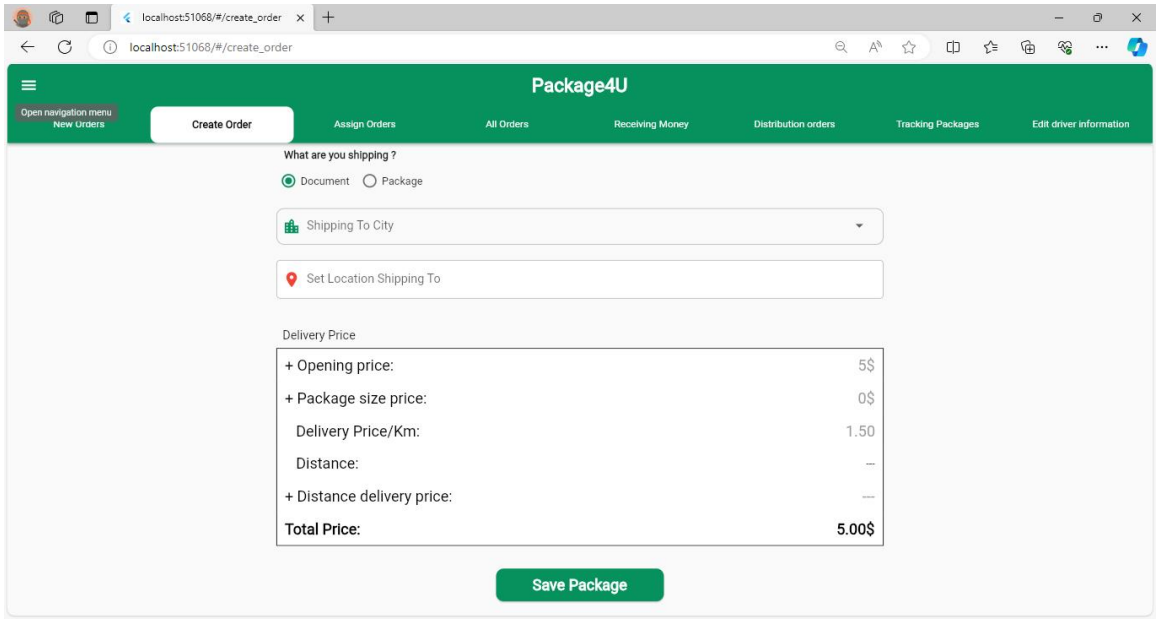

Figure 46: Create new package for customers how's come to the company 2

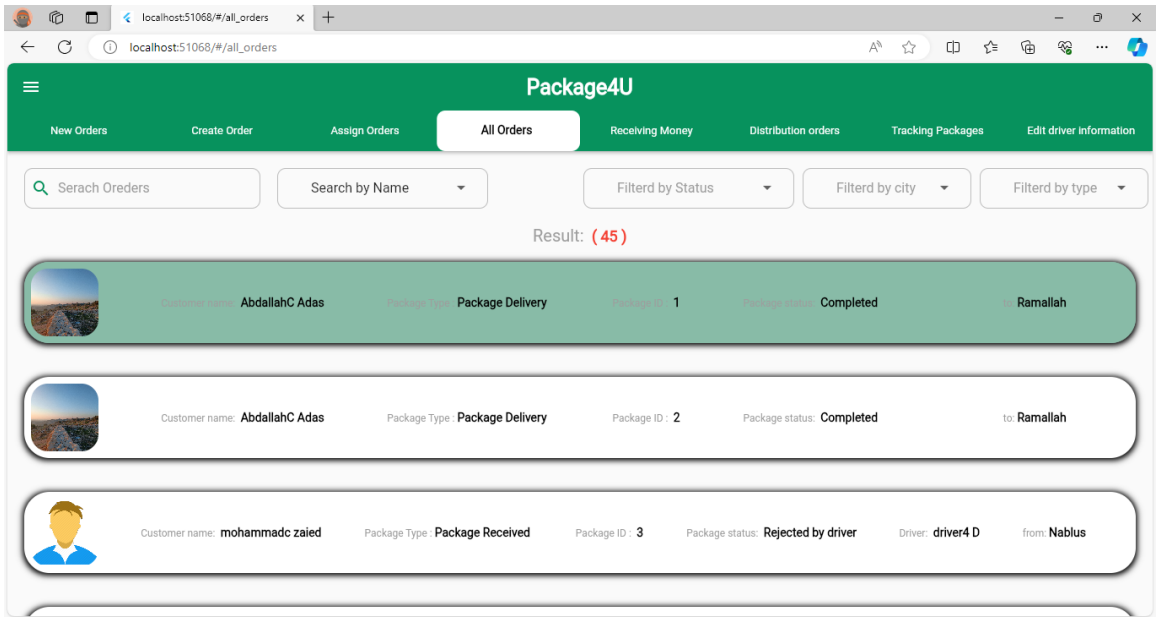

Figure 47: View all order in the system

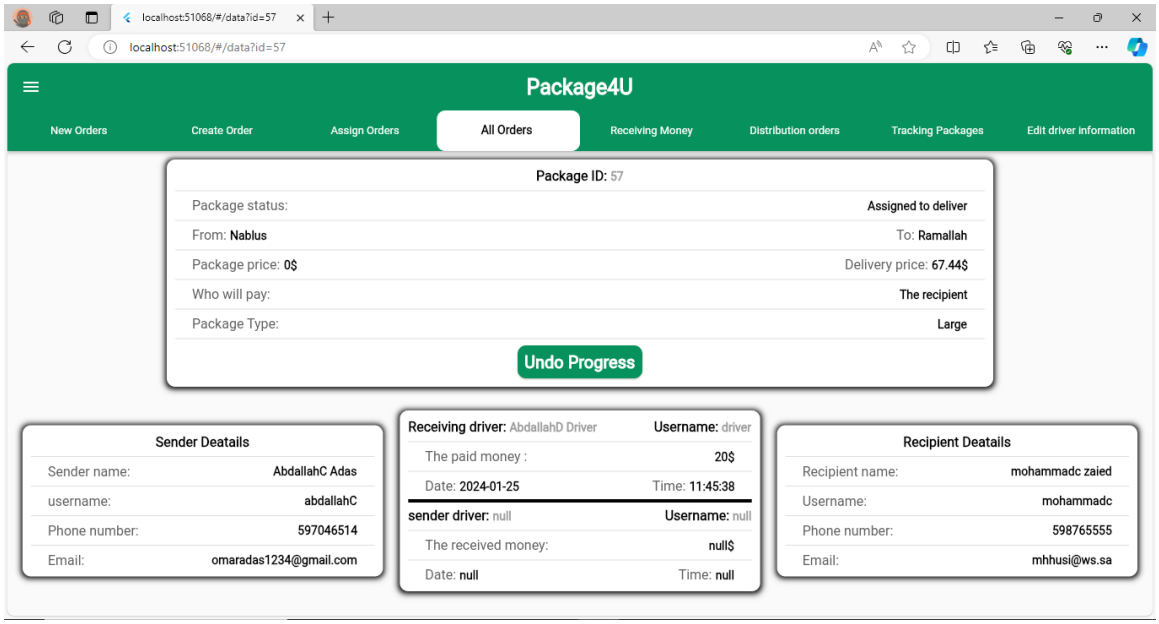

Figure 48: View specific package details

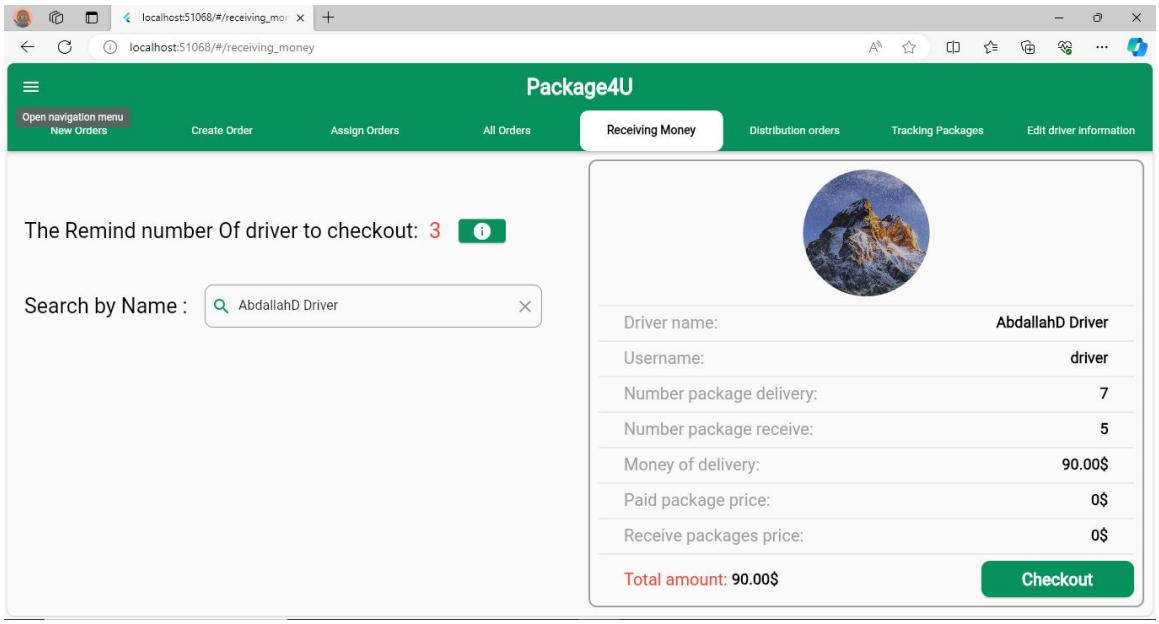

Figure 49: Receiving Money from drivers

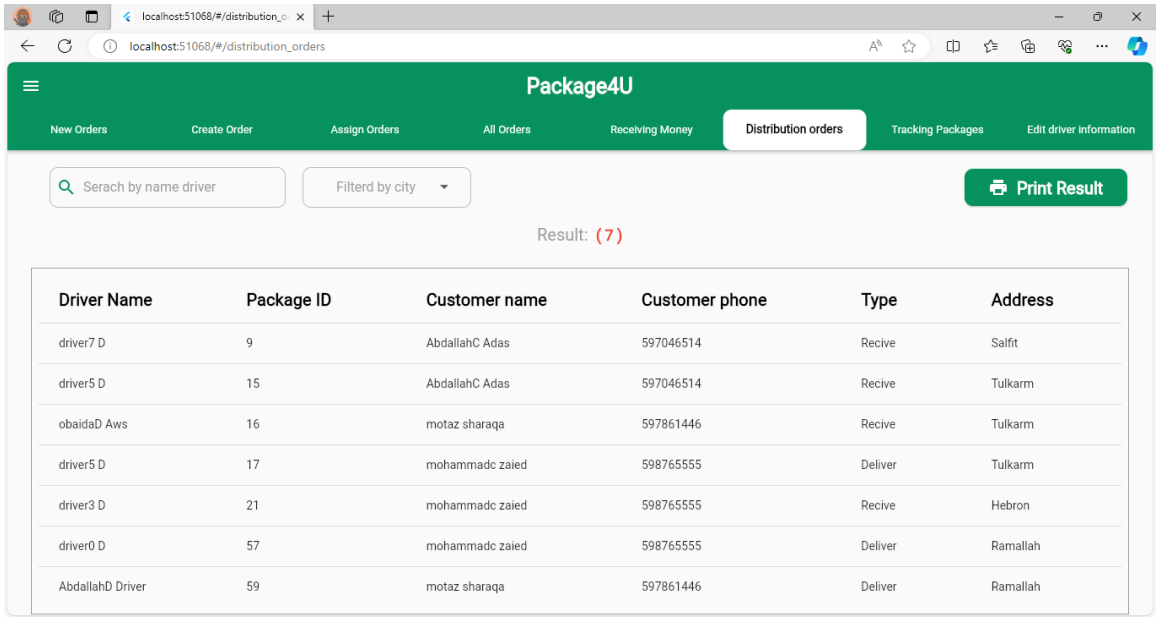

Figure 50: Print list of packages to give it to drivers and warehouse workers

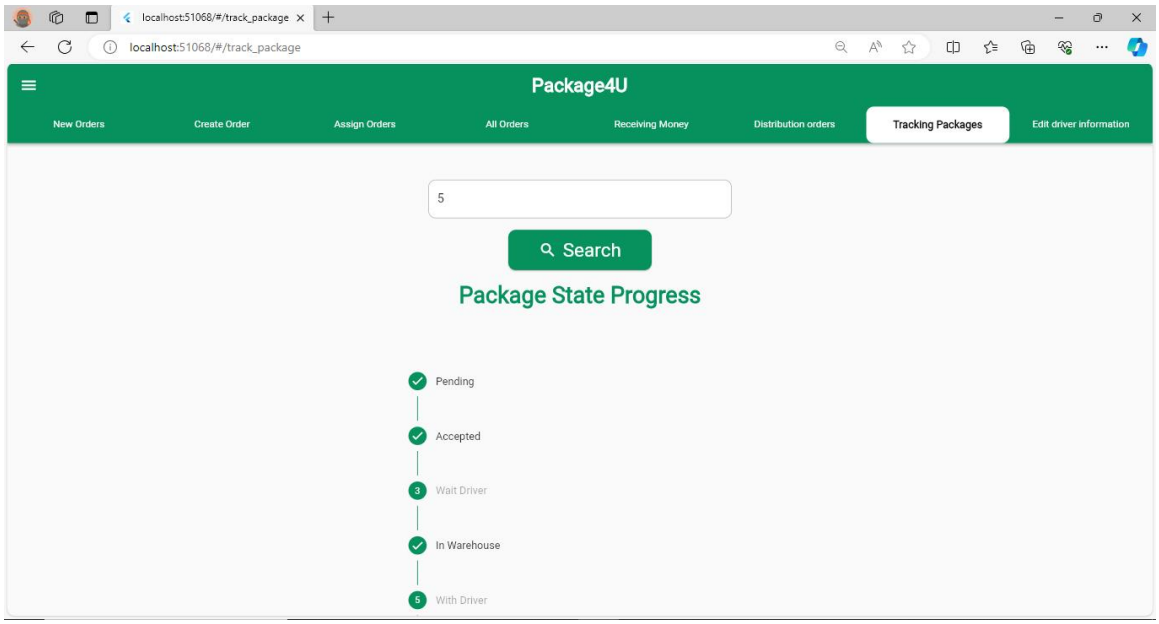

Figure 51: Give the customers details about his packages status

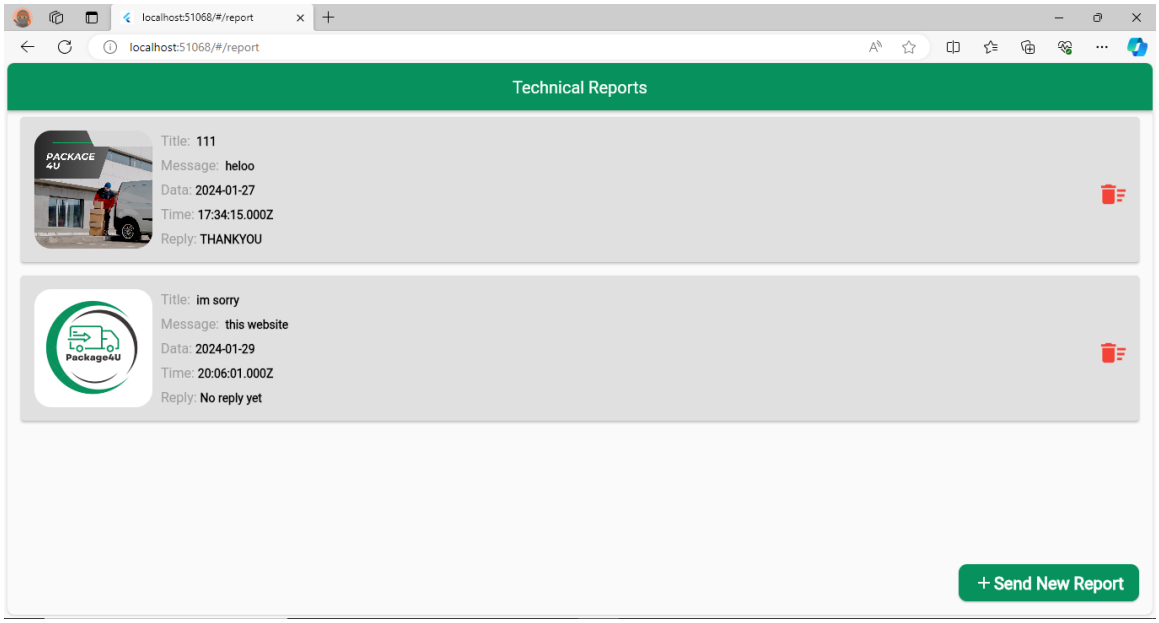

Figure 52: View the Technical reports and send technical or improvement reports

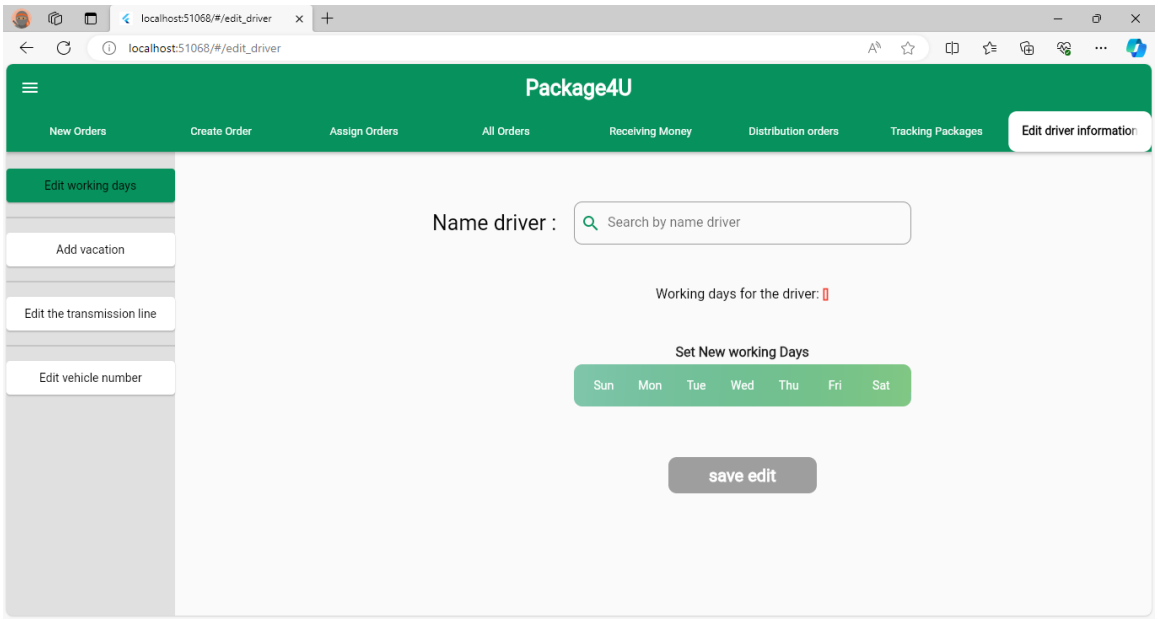

Figure 53: Edit drivers working days

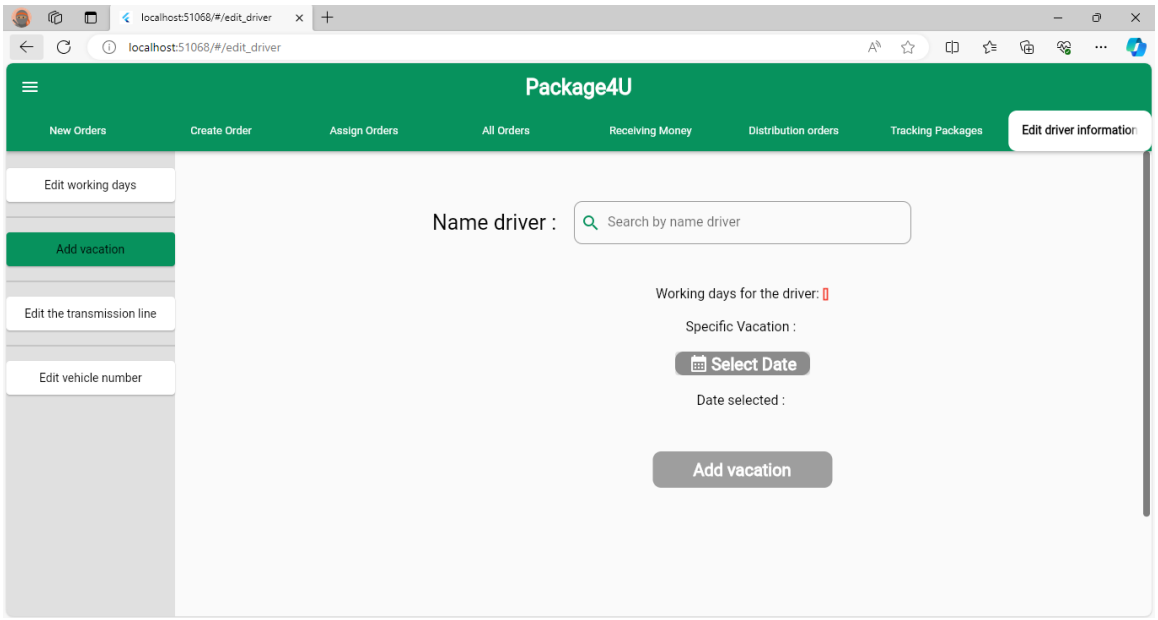

Figure 54: Add a vacations for drivers

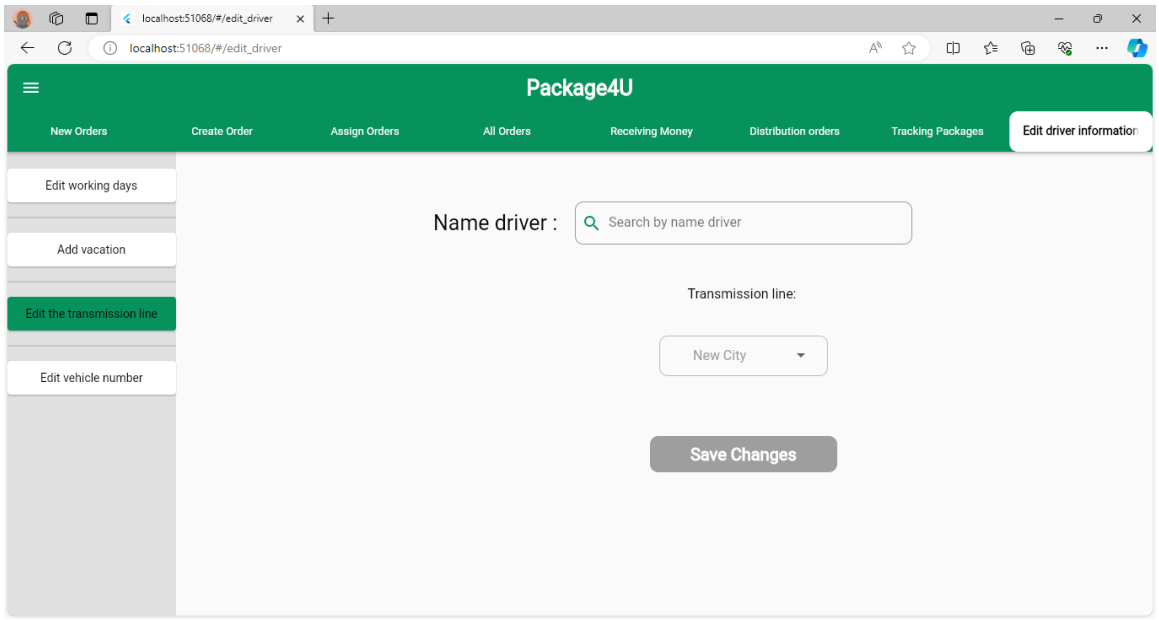

Figure 55: Edit driver transmission line

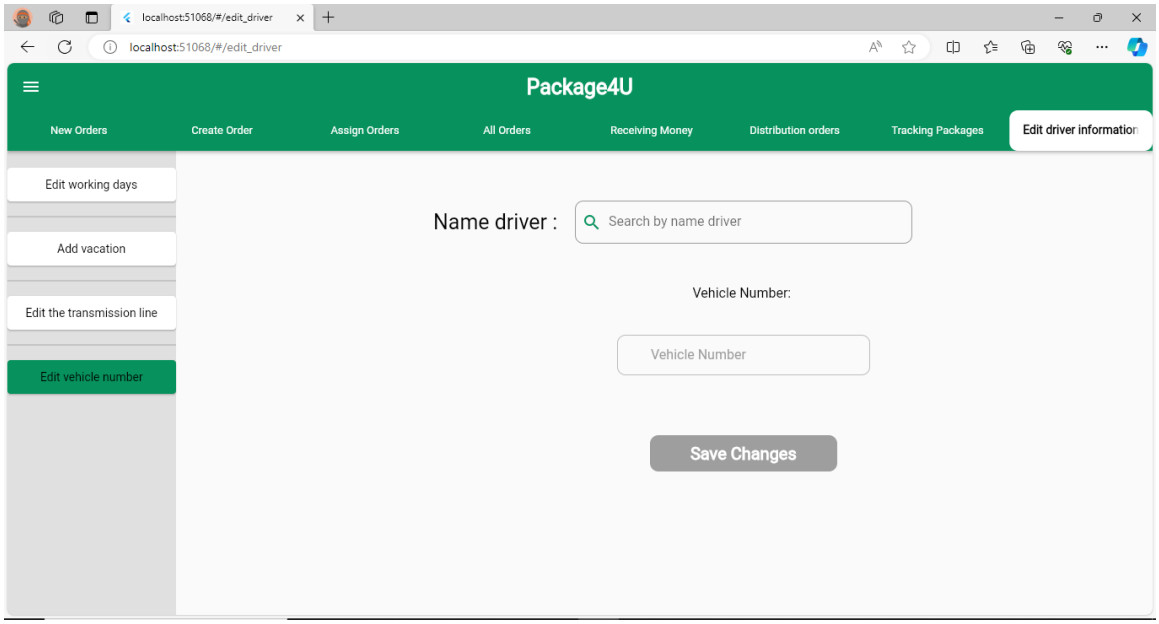

Figure 56: Edit driver's vehicle number

3.4.4 Admin

1. Ability to add managers to the company on the application, view their details, and remove them.
2. Ability to read error reports or developmental suggestions sent by all application users and respond to them.

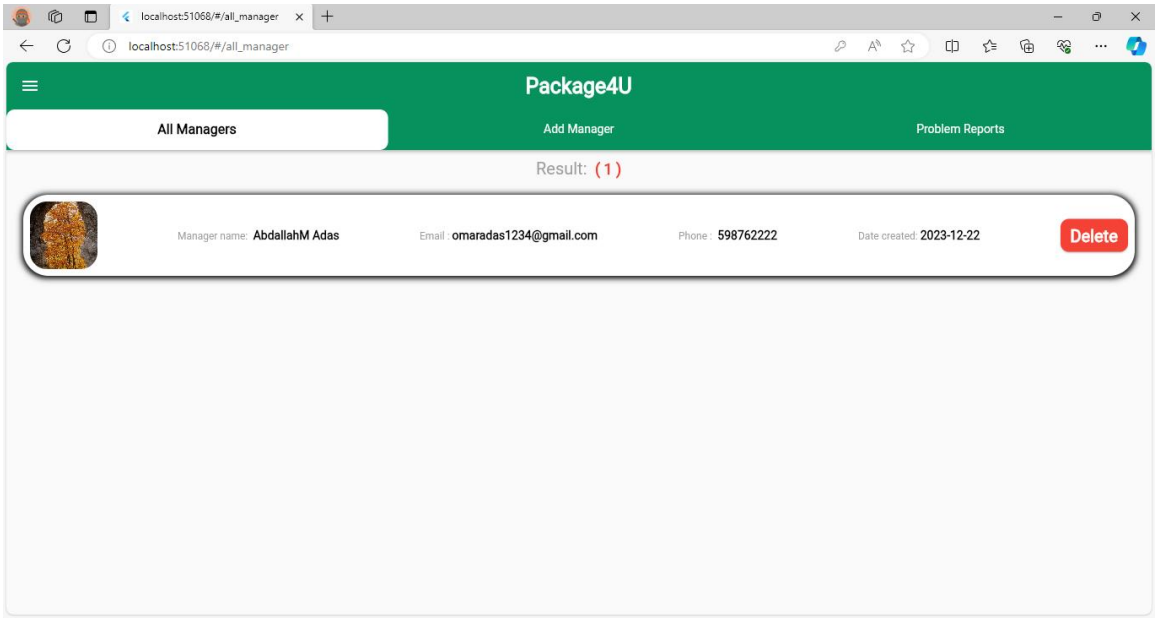

Figure 57: View company manager

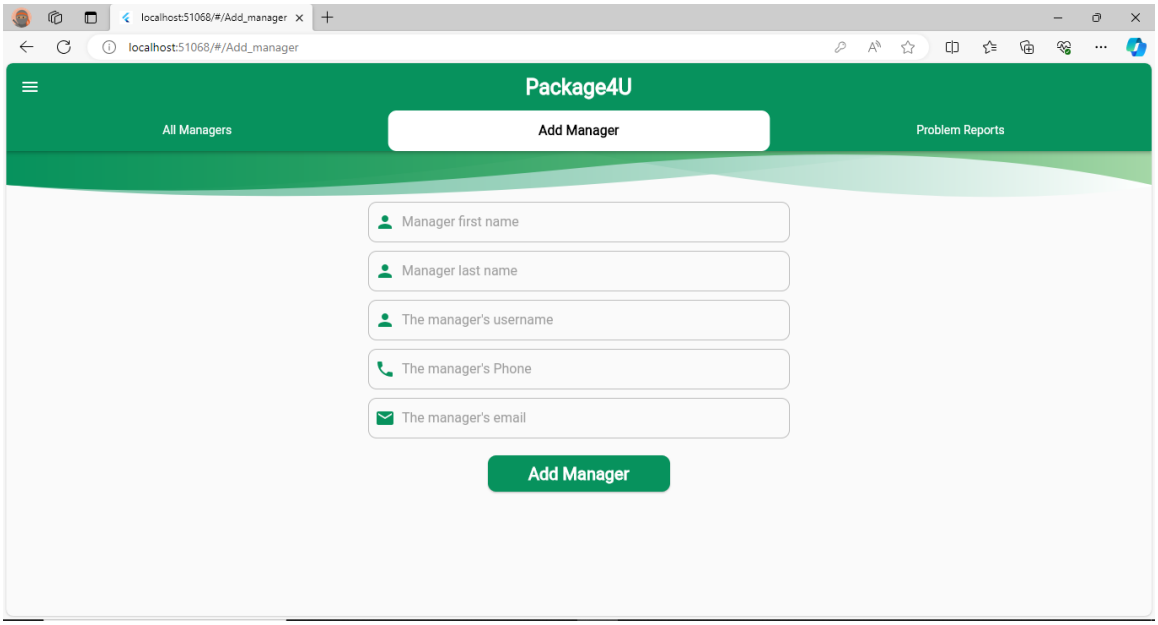

Figure 58: Add new manager to the company

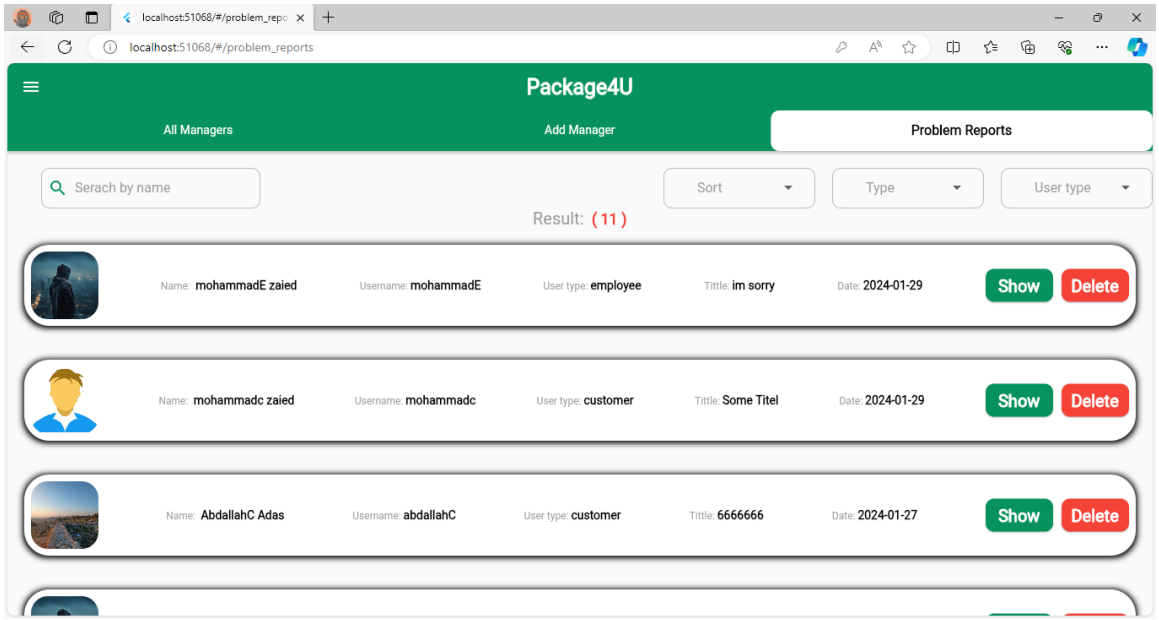

Figure 59: View all technical or improvement suggestions reports

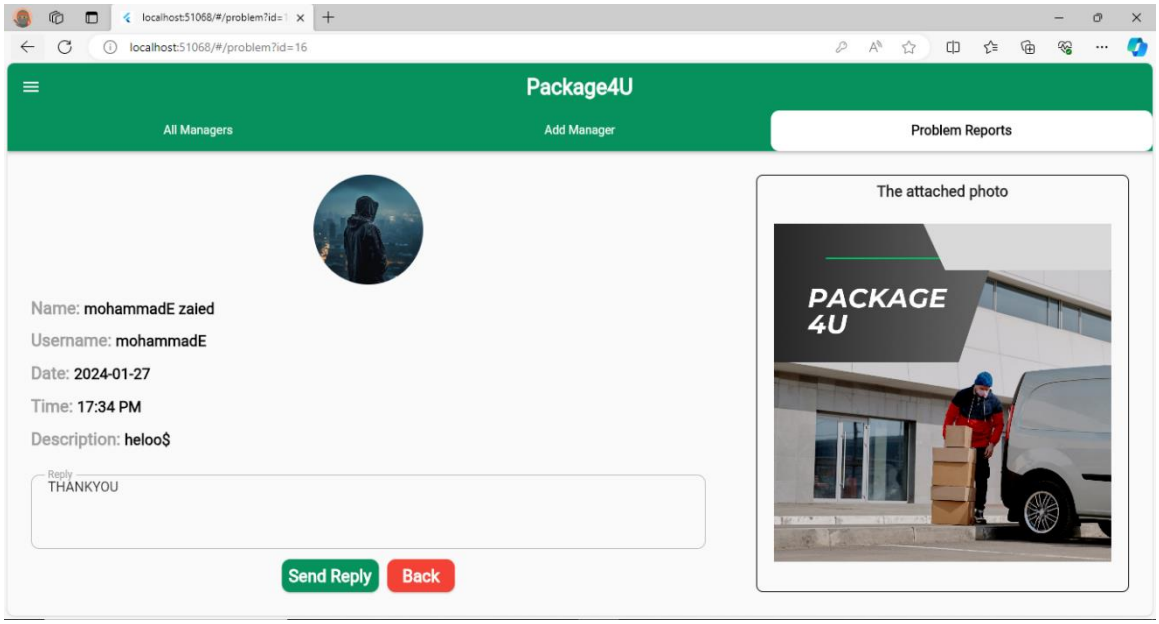

Figure 60: View reports details and reply

3.4.5 Driver

1. The driver can accept packages and prepare them before starting work, contact customers to ensure their readiness to receive packages, and reject delivering a package based on several reasons explained in a comment about the rejection reason.
2. Work on delivering packages or receiving them from customers. The program arranges packages by distance, and the driver can also use the map provided by our program to reach the package.
3. Confirm the receipt of cash from customers.
4. Ability to refuse to receive the package from the customer for any reason explained by the driver, such as a disturbance at the package delivery location or the customer's refusal to pay.
5. Ability to postpone package delivery if there is no time left for delivery or in case of an emergency.
6. View the packages worked on that day.

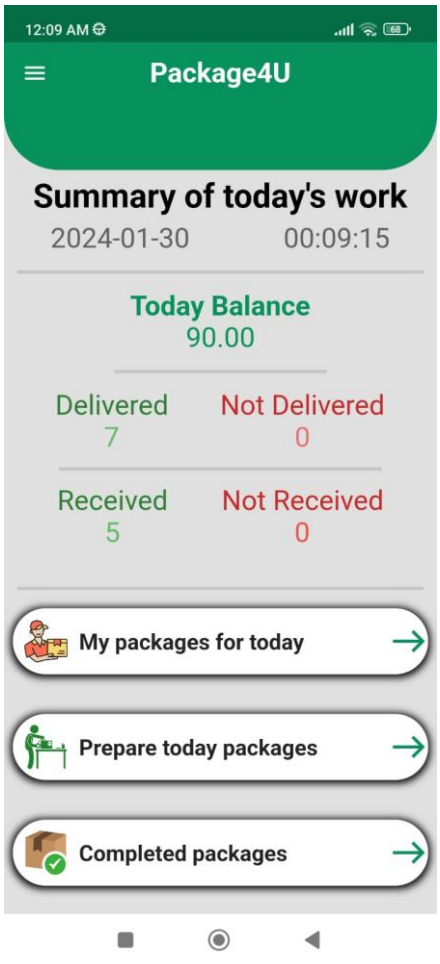

Figure 61: Driver Home Page

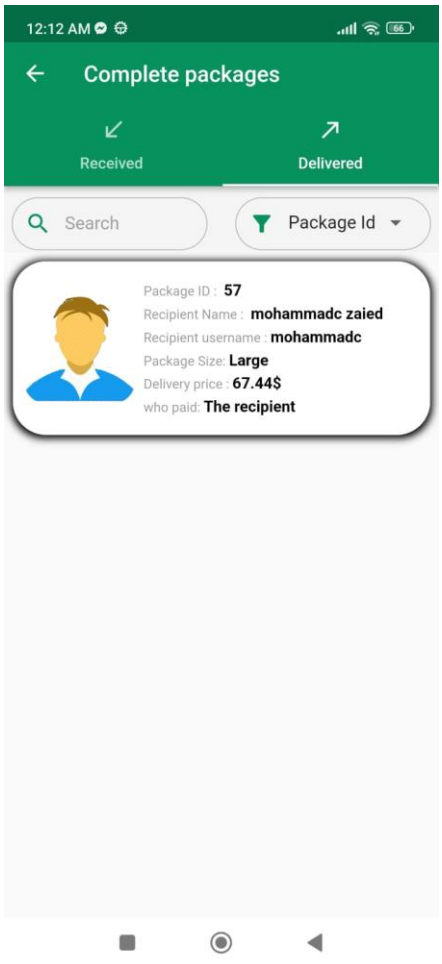

Figure 62: View the completed packages

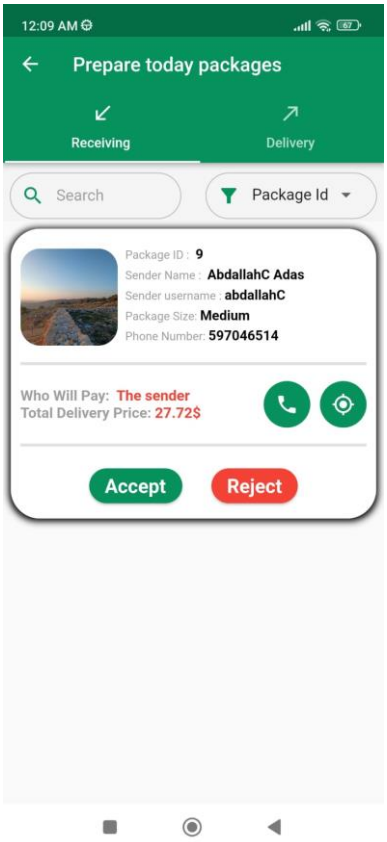

Figure 63: Prepare today packages

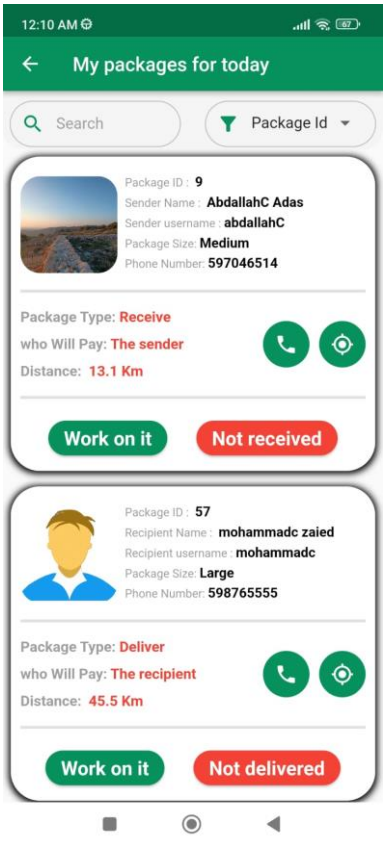

Figure 64: View all today packages and select one to work on it

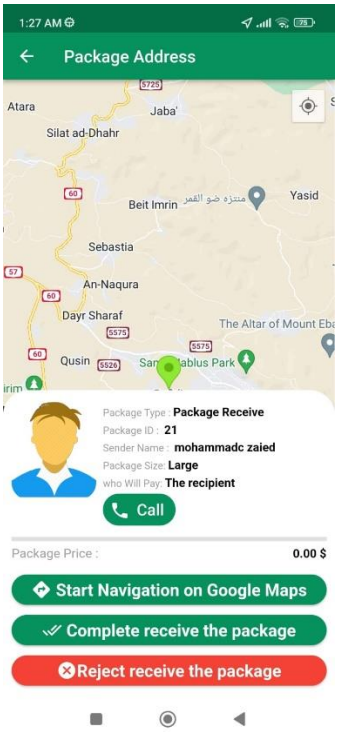

Figure 65: Example of work on receive package

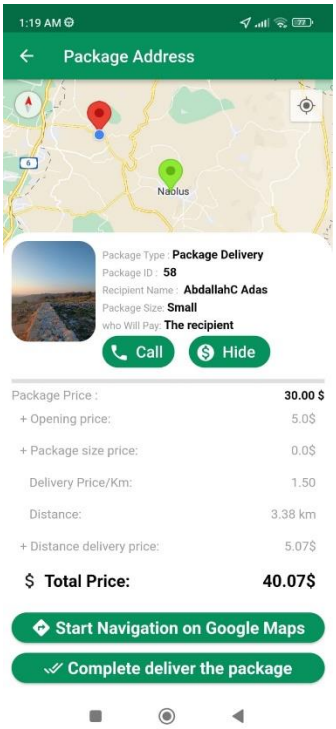

Figure 66: Example of work on delivery package

3.4.6 Manager

1. Ability to view all driver information, add drivers to the system, or remove them if a driver fails to deliver a package.
2. Ability to view employee information, add or remove them.
3. Ability to modify company information, edit delivery prices, and add a discount percentage on delivery fees.
4. Display financial reports for the manager, including the number of delivered and received packages, total earnings from package deliveries, amounts paid by the company for packages, amounts received, and the number of drivers working on that day.
5. Ability to view daily work reports and monthly and yearly reports. The manager can also specify a time period for report display.
6. Display the amounts paid by the company for packages and provide a detailed breakdown of this amount for packages whose fees have not been collected.
7. Ability to track all drivers working for the company on the map and view their latest appearance time on the program.

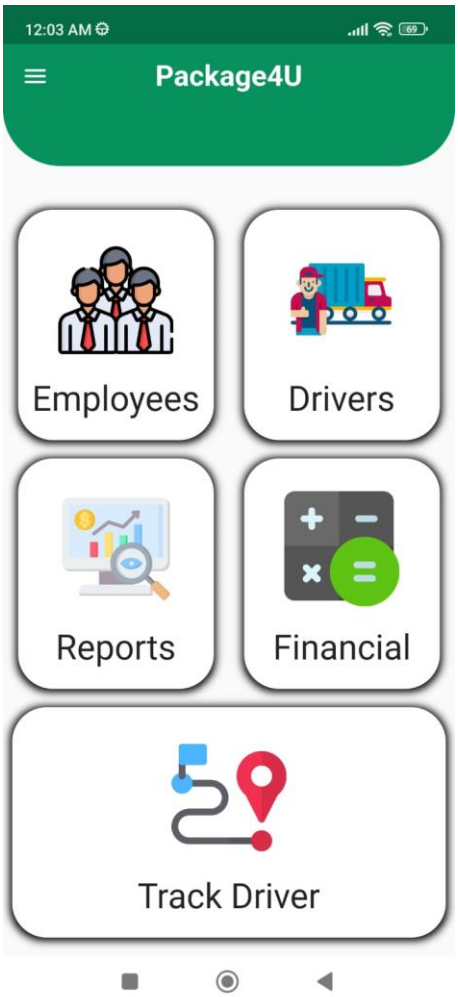

Figure 67: Manager Home Page

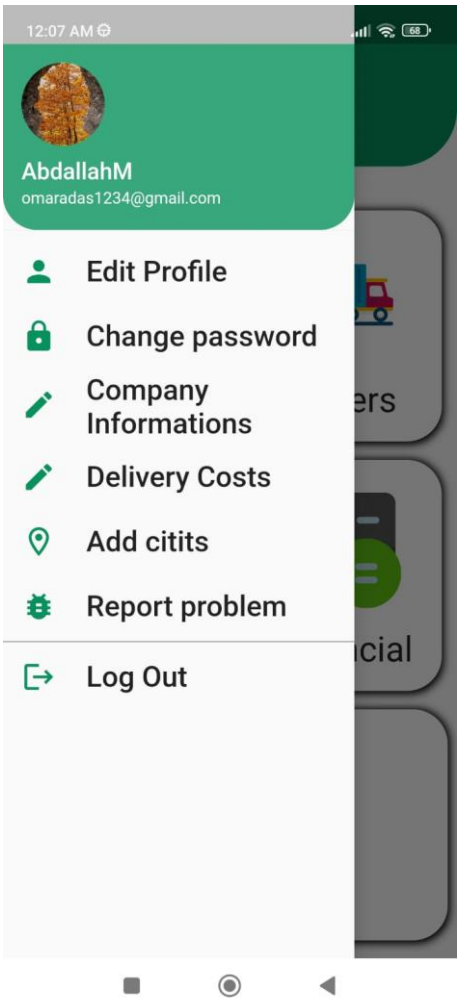

Figure 68: Drawer Content

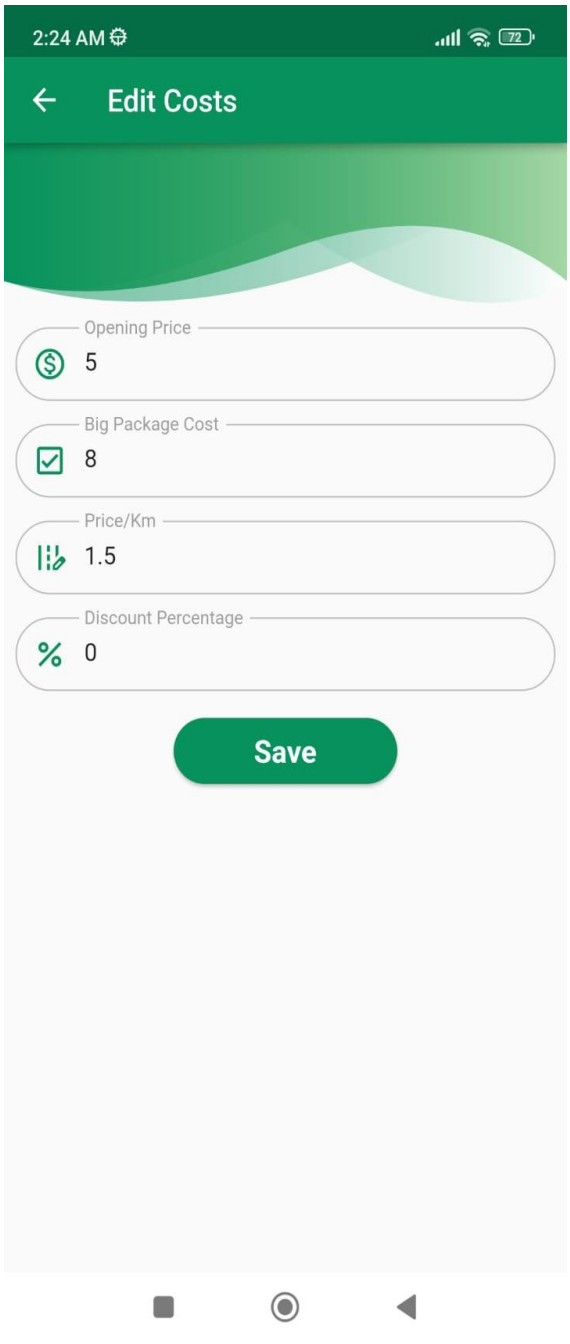

Figure 69: Edit Delivery Costs

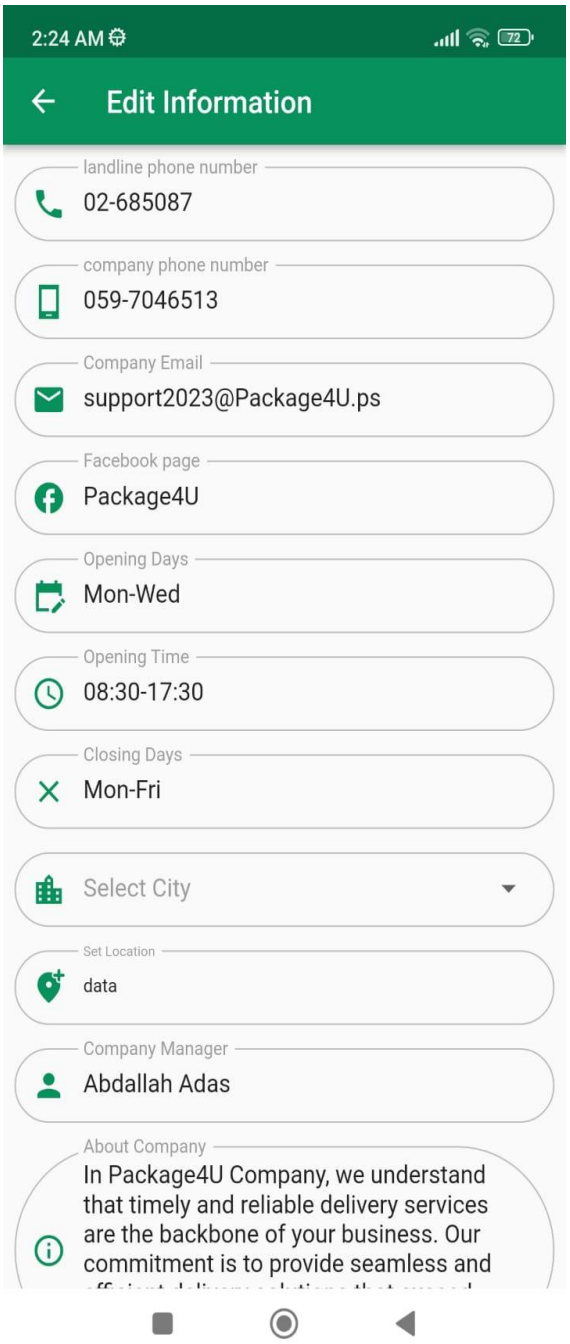

Figure 70: Edit Company Informations

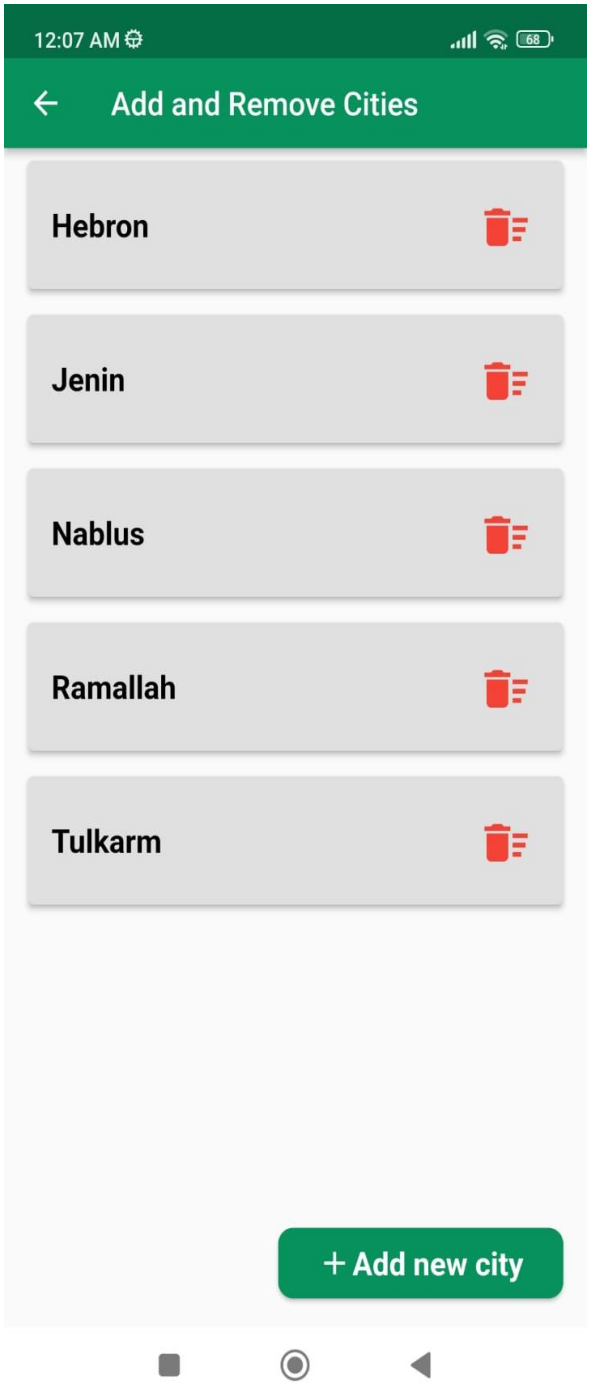

Figure 71: Add or Remove Cites

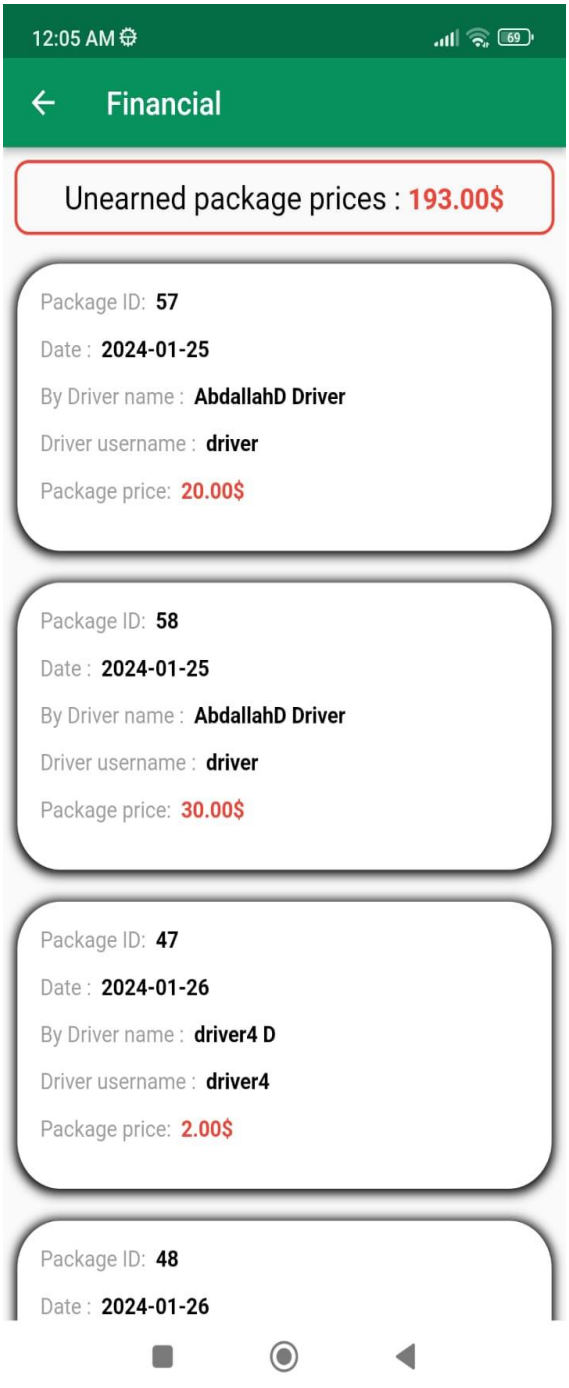

Figure 72: Unearned Package Prices

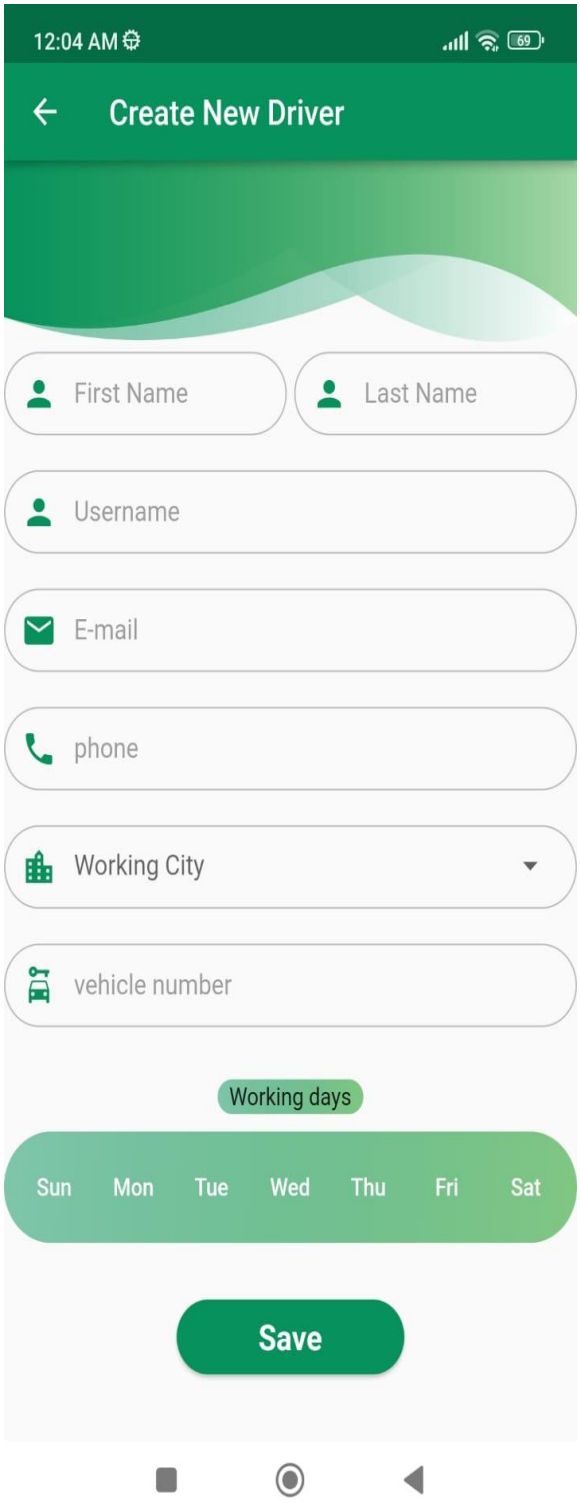

Figure 73: Add New Driver

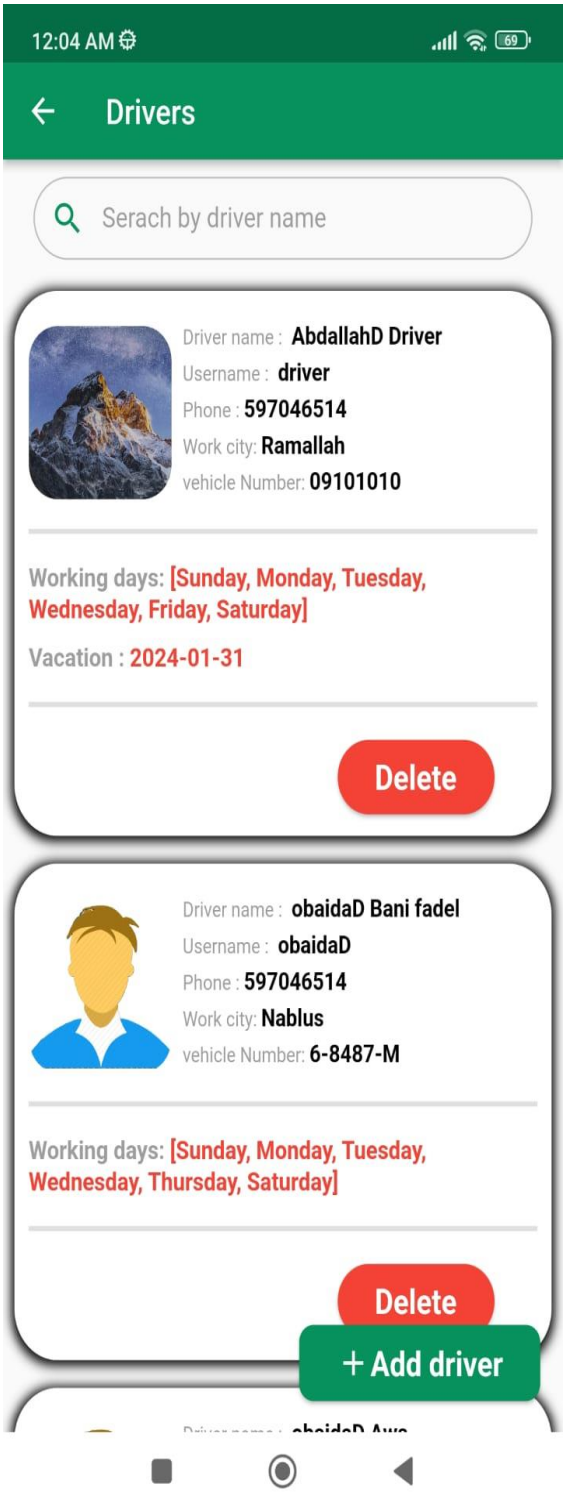

Figure 74: View All Drivers in the System

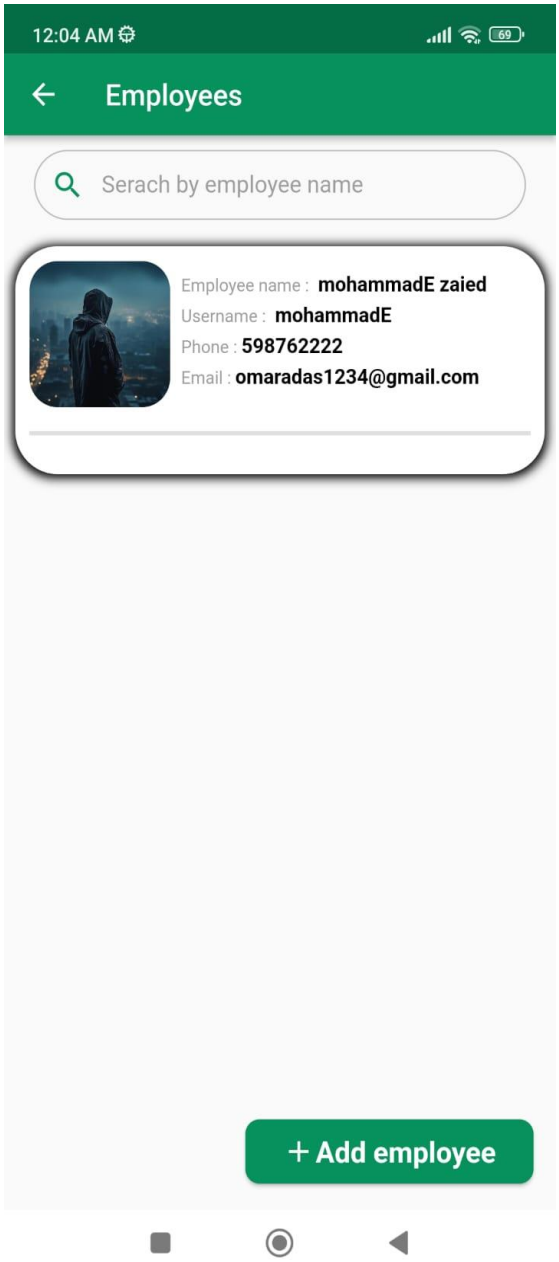

Figure 75: View All Employees

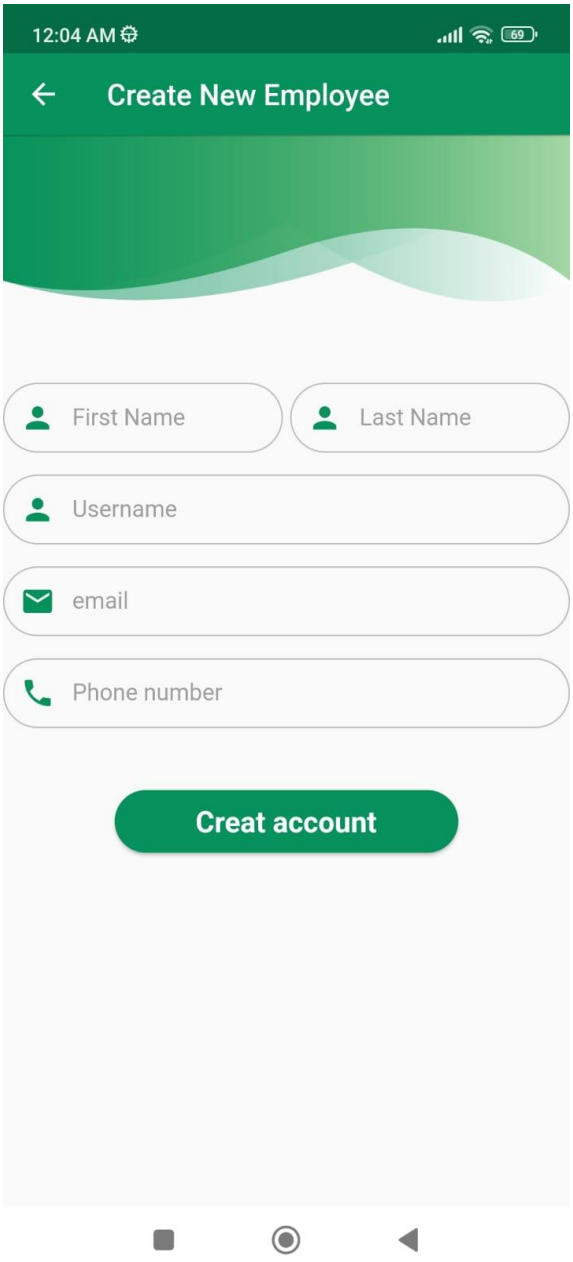

Figure 76: Create Employee

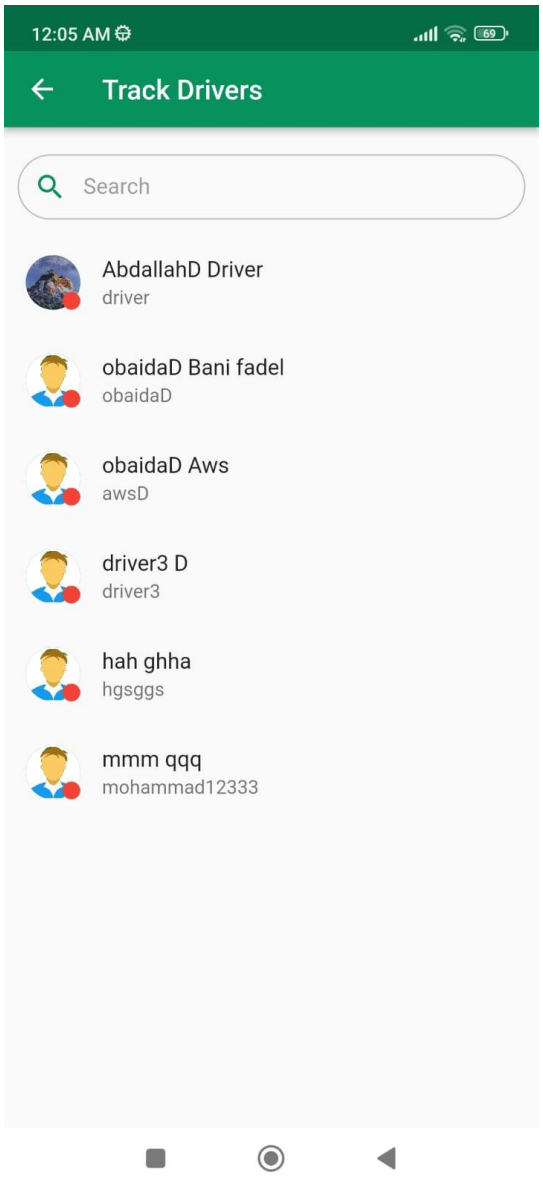

Figure 77: Track Drivers

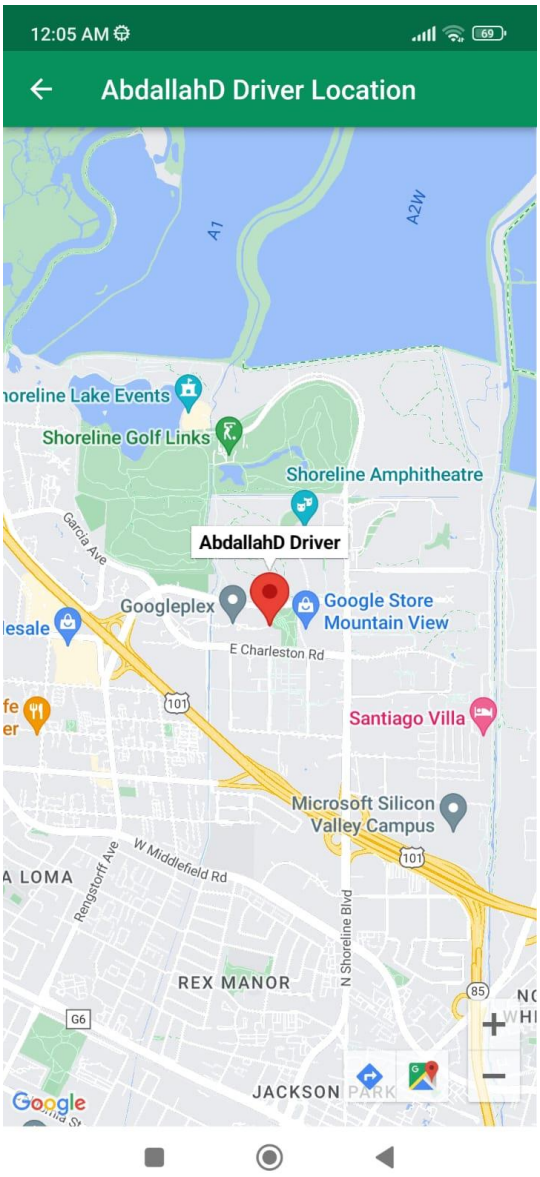

Figure 78: Driver Location

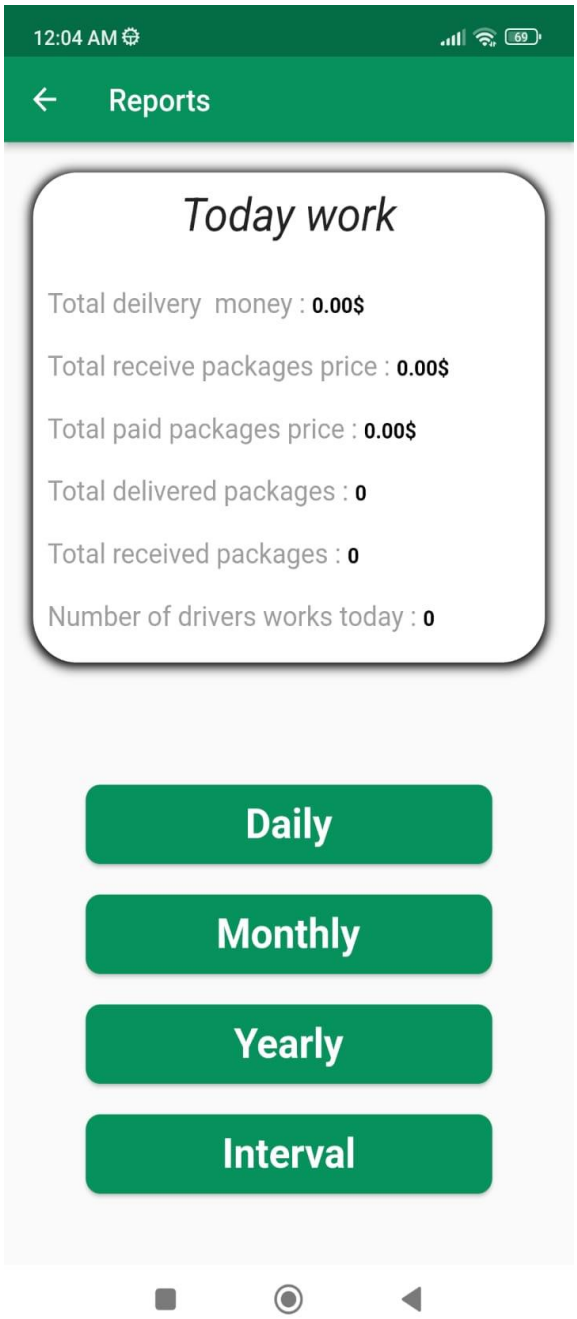

Figure 79: Manager Reports

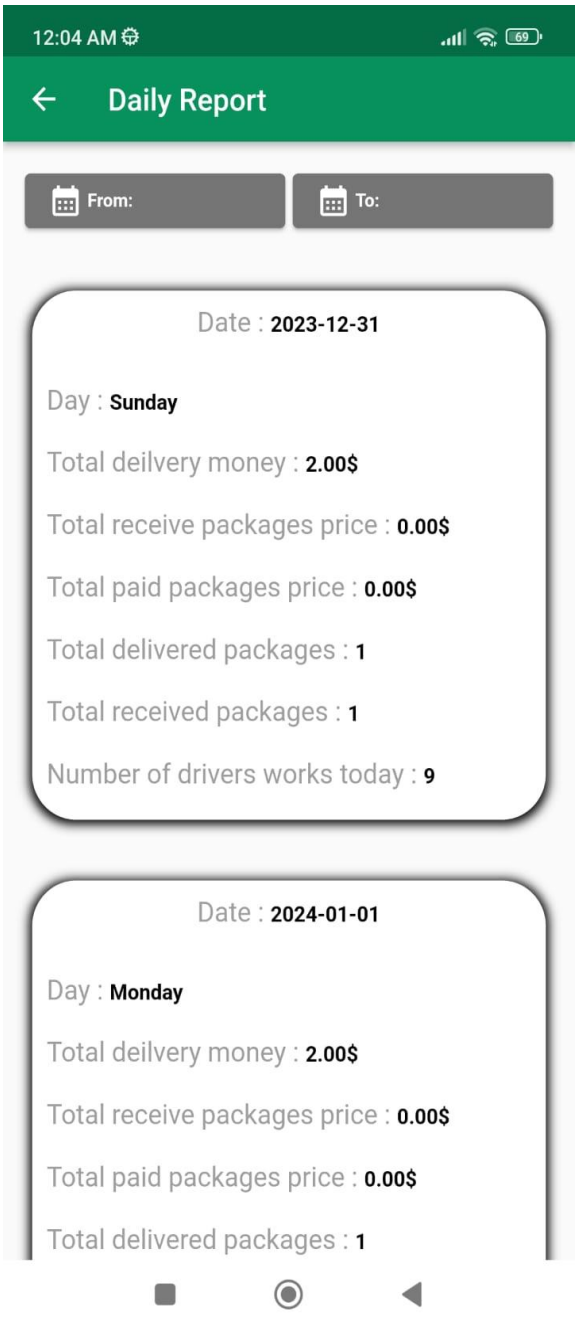

Figure 80: Daily Reports

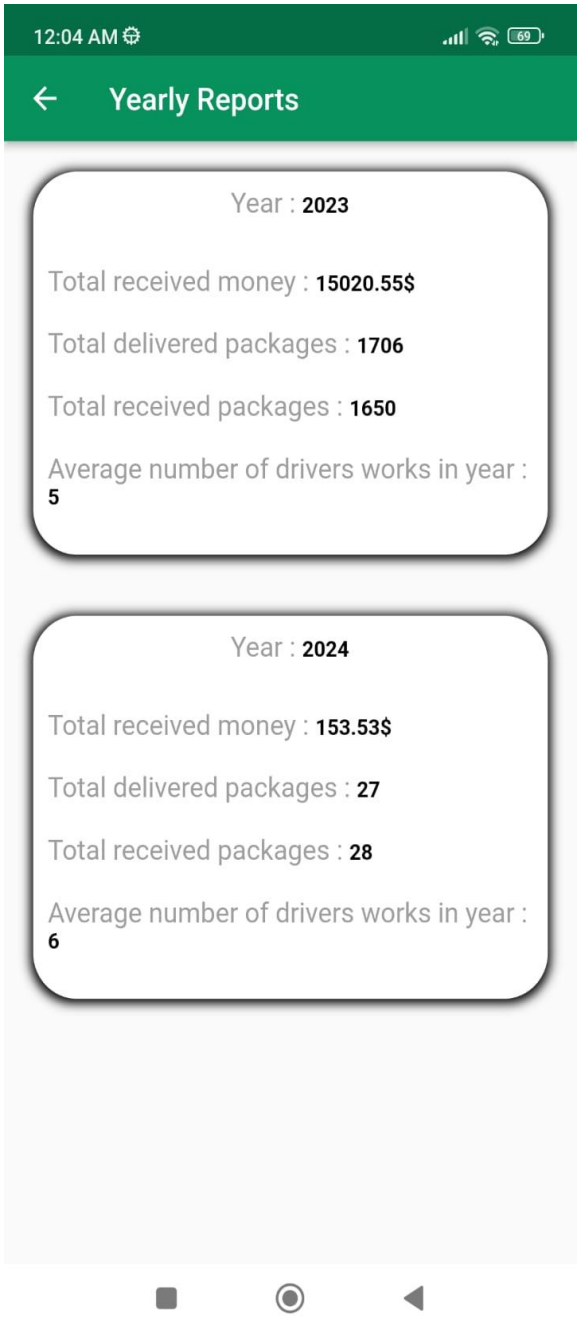

Figure 81: Yearly Reports

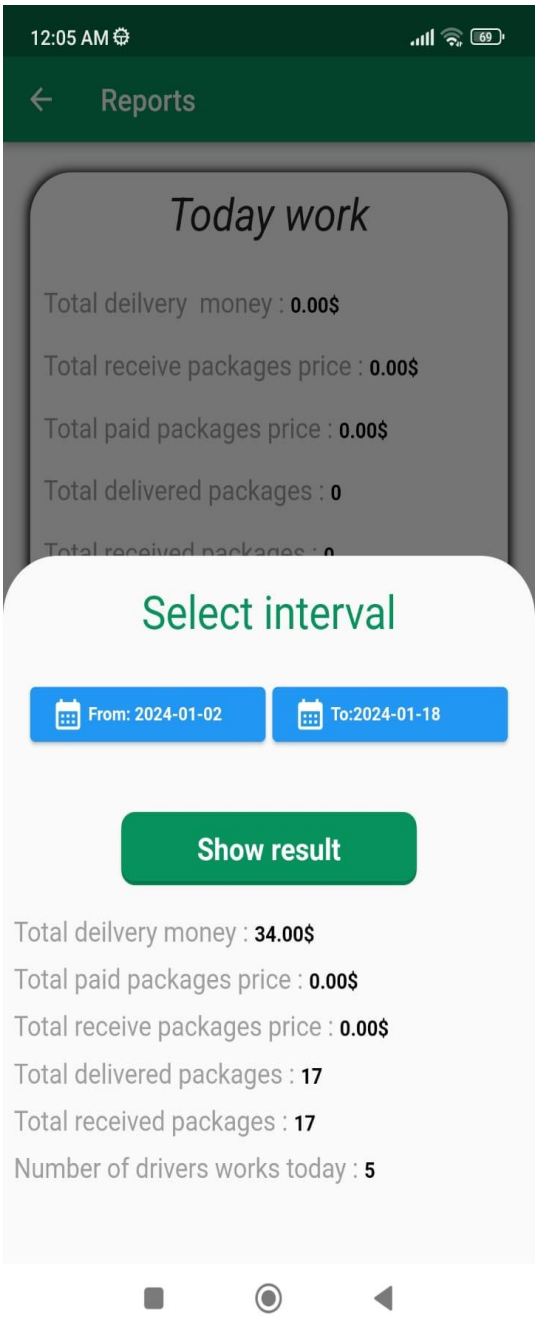

Figure 82: Get Report in the Selected Interval

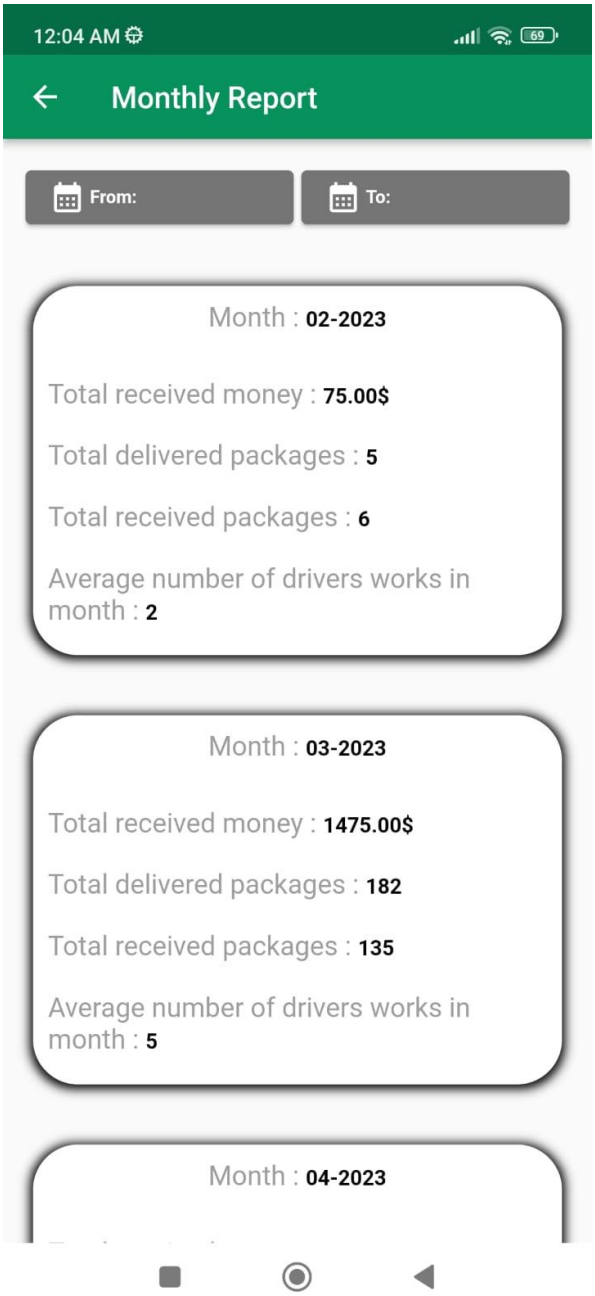

Figure 83: Monthly Report

Chapter 4
Results and Analysis

At the end of this project, we were able to provide an integrated program that links each of the customers, employees, drivers, and managers to deliver and track orders. Our application also allowed us to provide ease of use for customers, order packages from one region to another, and track the status of the package from the time the package was created until it was delivered. We were also able to save time and money for the drivers by arranging the packages entrusted to them according to the distance closest to their location and giving each package accurate location data that makes it easier for the driver to reach it or deliver it. We also created an electronic website for the employee through which he can fully control the packages and distribute them to the drivers and manage the drivers' work days. We also succeeded in providing different reports like Daily, monthly and annual finances for the manager.

Chapter 5

Discussion

5.1 Features

Customers can track the status and location of their packages, receiving real-time notifications for these changes. Customers can also save their favorite locations for package delivery. Employees can distribute packages to drivers, accept or reject packages before working on them, and modify some driver's information. The drivers can work on packages assigned to them based on distance, delivering them by viewing the package on the map. Also, the manager can access financial reports for the company, monitor the work of drivers on the map, and know their latest appearance time on the application. Finally, the administrator can receive development suggestions or technical or problem reports from all application users.

5.2 limitations of the work

We faced a challenge in determining the delivery and pickup locations of packages, relying entirely on Google Maps. This was due to the lack of accurate and consistent information for Palestinian cities and villages, primarily caused by the presence of the Israeli occupation and the existence of multiple names in different languages for Palestinian cities and villages. We managed to overcome this issue by requesting customers to specify the name of the city from which they will send and receive the package.

Chapter 6

Conclusion and Recommendations

In conclusion, our project offers a powerful and versatile package delivery management system that meets the distinct needs of administrators, managers, employees, drivers and customers. The app facilitates seamless communication and efficient workflow across different roles. Administrators benefit from the ability to manage administrative roles and address user feedback instantly. Managers, in turn, have access to tools that enable them to supervise drivers, manage company information, and analyze financial reports to make informed decisions. For employees, the app simplifies tasks such as editing driver information, distributing paperwork, and handling package-related activities. Drivers can efficiently manage package deliveries, communicate with customers, and adapt to changes in real-time, ensuring a smooth delivery process. Customers enjoy an intuitive interface that allows for easy creation, tracking of packages and receiving real-time updates on package status and location. Additional features such as cost calculation, saved delivery locations, and notifications improve the customer experience. Our project seeks to provide a comprehensive and integrated solution for package delivery management and enhance efficiency, transparency and customer satisfaction across the entire delivery process. The future work on the project may involve integrating QR code scanning functionality and establishing a connection with the program for reading package details. Additionally, expanding the project to accommodate a sizable company with multiple branches is under consideration.

References:

- **Flutter:** Available: https://flutter.dev/
- **Google Maps:** Available: https://www.google.com/maps/
- **Firebase:** Available: https://firebase.google.com/docs/cloud-messaging
- **Node Js:** Available: https://nodejs.org/en
- **Axios:** Available: https://www.axios.com/
- **Sequelize:** Available: https://sequelize.org/
- **Pub.dev:** Available: https://pub.dev/
- **Android Package4U:** GitHub: https://github.com/AbdallahOmarAdas/Package4U
- **Web Package4U:** GitHub: https://github.com/Mohammadzaied/Web_flutter_P4