



ក្រសួងអប់រំ យុត្តិធម៌ និងកីឡា



ពិនិត្យនាលប្បទេសចរណីជាតិ

លេខាដីជីថល ទេសចរណីជាតិ និងកីឡា

ទូទាត់ការបង្កើតុះកម្មសិក្សា ពិស្វកម្មផ្លូវកិច្ច

| | |
|---------------|--|
| ប្រធានបទ | : ប្រព័ន្ធប្រព័ន្ធភោគសេវាកម្មបែកលក្ខណៈ |
| សិក្សា | : ខ្លួន បាយប្រាណីត |
| ឯកដោស | : ពិស្វកម្មបែកសេវាទិន្នន័យ និងកីឡា |
| ប្រភពទិន្នន័យ | : នោយ ហុង ីន |
| ផ្លូវកិច្ច | : ២០២២ - ២០២៣ |

MINISTERE DE L'EDUCATION,
DE LA JEUNESSE ET DES SPORTS

INSTITUT DE TECHNOLOGIE DU CAMBODGE

DEPARTEMENT DE GENIE INFORMATIQUE ET
COMMUNICATION

RAPPORT DE STAGE D'INGÉNIEUR DE QUATRIÈME ANNÉE

| | |
|-----------------|--|
| Titre | : Système de gestion des services de livraison |
| Etudiant | : ROTHA Dapravith |
| Spécialité | : Génie Informatique et Communication |
| Tuteur de stage | : Mr. Hok Tin |
| Année scolaire | : 2022-2023 |



ក្រសួងបន្ទាន់ខំ យុទ្ធសាស្ត្រ និងកីឡា



ତିଜ୍ଞାନୀବିଦ୍ୟାକାରୀଙ୍କ ପରିଚୟ

ବେଳାଣ୍ଟେତୁ କେବଳକାହିଁ ପ୍ରକଟିଶାଳା ଏବଂ କିମ୍ବା କିମ୍ବା

នហេតុវិស្សីត: នគ្គារ ចាយប្រាផីន

କୋଡ଼ିଙ୍ଗରୀ ପ୍ରେସ୍ କାର୍ଯ୍ୟାଳୟ ଲିମିଟେଡ୍: ପ୍ରେସ୍ ନଂ ୧୮ ହେ ତିବ୍ରିକା ଫ୍ଲେସି ଡ୍ରାଫ୍ଟ୍ ପାତ୍ର

ଶବ୍ଦଗୁଣାତ୍ମକାନାମଣ୍ଡରୀ

ଜୀବନ ଅନ୍ତର୍ଗତ କାହାର ଦେଖିଲାମି: _____

ପ୍ରକାଶି ହେ ପ୍ରକାଶନ

ବ୍ୟାପକ ବିଷୟ

: ប្រព័ន្ធប្រតិបត្តិកសាស្ត្រនៃជាតិខ្មែរ

សិរីប្រជាធិបតេយ្យ

ពិន្ទុសាលបច្ចេកពិន្ទុកម្ពុជា

క్రమానియాండ్ ఆస్ట్రేలియా

ଭେଟ୍ କେଣ୍ଟିଙ୍ଗ ପୋଲି

សាស្ត្រាថាមបីទេសចរណ៍

ଭେଳ ଖୁଲ୍ବ କିମ୍ବା _____

ଶ୍ରୀକୃତି ଶ୍ରୀନାଥ ପାତ୍ର ମହାନ୍ ବ୍ୟାଙ୍ଗାନ୍ତିକ ପାତ୍ର

: ເບີໂຫຼວດ ທຸກ ຂີ່ໃນ _____

ବନ୍ଦବାଣୀକୁଁଟେଙ୍ଗା ଫୁଁଲ୍‌ଟେଙ୍ଗା



MINISTERE DE L'EDUCATION,
DE LA JEUNESSE ET DES SPORTS



INSTITUT DE TECHNOLOGIE DU CAMBODGE
DEPARTEMENT DE GENIE INFORMATIQUE ET
COMMUNICATION

RAPPORT DE STAGE D'INGÉNIEUR DE QUATRIÈME ANNÉE

DE M. ROTHA Dapravith

Date de soutenance: le 17 Novembre 2023

« Autorise la soutenance du mémoire »

Directeur de l'Institut : _____

Phnom Penh, 2023

Titre : Système de gestion des services de livraison

Etablissement du stage : Institut de Technologies du Cambodge

Chef du département : Mr. LAY Heng _____

Tuteur de stage : Mr. HOK Tin _____

Responsable de l'établissement : Mr. HOK Tin _____

PHNOM PENH, 2023

ACKNOWLEDGEMENTS

This report would not have been publishing out without the guidance and assistance of several people who always provide the good solution and extended their valuable assistance in the preparation and implementation of this project.

First, I would like to express our deepest gratitude to my parents who always provide me with all kinds of support me everything not only giving me an opportunity to study, money and times but also telling me being a good human resource in the society, being person who have morality and responsibility.

Firstly, I would like to appreciate my gratitude to His Excellency **Dr. PO Kimtho**, General Director of Institute of Technology of Cambodia, for his good management of the institute and his good cooperation with the partner universities at the local, regional, and international levels, to enhance the quality of the training of engineers and senior technicians.

Secondly, I wish to express in advance thanks to **Mr. LAY Heng**, Head of the Department of Information Technology and Communication at the Institute of Technology of Cambodia, for his good management and is an integral part of the teaching.

Thirdly, I would like to express appreciation to **Mr. HOK Tin**, lecturer at ITC and as my academic supervisor, for the time he has devoted to me during this period for giving the solutions concerning internship and having answered our questions with participation in the progress of our work. We also greet of him for his remarks and corrections which allow me to write our final dissertation.

Fourthly, all lecturer in the department Information and Communication Engineering for knowledge they gave us during our studies at the Institute of Technology of Cambodia.

Finally, we would like to thank all the people who participated and helped us to write this report and build the Delivery Service System until we can complete successfully.

ଶ୍ରୀନାଥପୁରୀ

ជាលទ្ធផលនៃគម្រោងនេះ បច្ចុប្បន្នភាពនៃគម្រោងគេហទំនើសរបស់ក្រសួងសាធារណការអភិវឌ្ឍន៍បន្ថែមនៅឡើយ
ដោយសារនៅសល់មុខងារមួយចំនួនដែលមិនទានអភិវឌ្ឍន៍នៅឡើយនិងមានការកែតម្រូវឱ្យមុខងារធ្វើដែរ
ឡើតដើម្បីធ្វើអាយប្រព័ន្ធកេហទំនើសនៃការតែប្រសើរជាងមុន មុននឹងយានទៅដែលការដាក់ឲ្យបើត្រាស់ជាង្លើការ តាមតម្លៃការអាជីវកម្ម នាថែលអនាគត។

RÉSUMÉ

Aujourd'hui, je constate que le secteur de la livraison est en plein essor, qu'il existe de nombreuses entreprises de livraison pour toutes sortes d'articles et que les succursales de livraison se développent dans toutes les régions pour répondre aux besoins des clients en matière de services de livraison. C'est pourquoi j'ai décidé de mettre en place ce système de gestion des services de livraison afin de faciliter le travail quotidien des responsables et des propriétaires d'agences.

Dans ce système, il y a deux fonctions principales pour les administrateurs et les propriétaires d'agences. Tout d'abord, l'administrateur du système dispose d'un tableau de bord pour vérifier le résumé des données ci-dessus, telles que les colis entrants, les colis en attente, les colis livrés, le montant total des bénéfices selon chaque propriétaire d'agence, et le nombre total d'utilisateurs, tels que les gestionnaires et les propriétaires d'agence. Deuxièmement, pour les propriétaires de filiales, il existe un tableau de bord qui décrit le nombre total de marchandises non envoyées, les marchandises en cours d'envoi et, enfin, le bénéfice total du propriétaire de la filiale ainsi qu'un résumé du paquet de livraison. En outre, une facture confirme la quantité de marchandises envoyées et la date. Par ailleurs, les clients peuvent suivre les marchandises en tapant simplement le code du colis. Le système de recherche affichera les résultats des marchandises recherchées et fournira des détails ainsi qu'un code QR à scanner pour vérifier les marchandises soumises, qu'elles soient arrivées à l'endroit désigné ou non. Enfin, les clients peuvent soumettre leurs commentaires pour l'évaluation du service afin d'améliorer les lacunes et de mieux servir le service.

En raison de ce projet, l'état d'avancement de ce projet de site web est toujours en cours de développement, car certaines fonctionnalités ne sont pas encore développées et d'autres ont été améliorées pour rendre ce système de site web encore plus performant. Le lancement officiel se fera en fonction des besoins futurs de l'entreprise.

ABSTRACT

Today, I see that the shipping business is booming and there are many companies for transporting all kinds of goods of customers and have expanded branches for delivery in all regions to serve the delivery service of customers. Timely. Seeing these issues, I decided to set up this delivery service management system to facilitate the daily work of managers and branch owners.

In this system there are two main functions for managers and branch owners. First, for the system administrator, there is a dashboard to check the summary of the above data, such as incoming goods, incoming goods, sent goods and the total amount of profit according to each branch owner and the total number of users, such as Managers and Branch Owners. Second, for Branch Owners, there is a dashboard that describes the total number of unsent goods, the goods being sent, and finally, the total profit of the Branch Owner and a summary of the goods sent. In addition, there is an invoice confirming the quantity of goods sent and the date. Meanwhile, customers can track the goods by simply typing the package code, the search system will show the results of the goods searched, come out with details and QR Code for scanning to verify the submitted goods, whether the goods have arrived at the designated place or not. Finally, customers can provide feedback for evaluation of services to improve the short comings to provide better service.

As a result of this project, the current version of the website is still under development, as some features are still undeveloped and other features have been tweaked to make the site even better. Will reach the official launch according to future business needs.

LIST OF ABBREVIATION

| | |
|--------------|---|
| API | : Application Programming Interface |
| CSS | : Cascading Style Sheets |
| CRUD | : Create, Read, Update and Delete |
| DSS | : Delivery Service Management System |
| HTML | : Hyper Text Markup Language |
| HTTP | : Hypertext Transfer Protocol |
| IDE | : Integrated Development Environment |
| ITC | : Institute of Technology of Cambodia |
| JS | : JavaScript |
| JSON | : JavaScript Object Notation |
| MERN | : MongoDB, Express, React, and NodeJS |
| NOSQL | : Non-Relational Structure Query Language |
| SCSS | : Syntactically Awesome Style Sheet |
| UI | : User Interface |
| UML | : User Modeling Language |
| UX | : User Experience |

LIST OF FIGURES

| | |
|---|----|
| Figure 1: ITC logo | 2 |
| Figure 2: ITC organization | 3 |
| Figure 3: ITC location map | 4 |
| Figure 4: GIC department logo..... | 5 |
| Figure 5: Scrum events | 8 |
| Figure 6: Physical architecture of system..... | 13 |
| Figure 7: Logical architecture of system | 15 |
| Figure 8: Use case diagram | 17 |
| Figure 9: Database Schema | 20 |
| Figure 10: Activity diagram of add new admin account | 22 |
| Figure 11: Activity of Add new branch owner account | 23 |
| Figure 12: Activity diagram of add package for delivery | 24 |
| Figure 13: Activity diagram of QR Code scan | 25 |
| Figure 14: JavaScript logo..... | 26 |
| Figure 15: Mern Stack logo | 27 |
| Figure 16: Html,css,javascript logo | 27 |
| Figure 17:Scss logo | 27 |
| Figure 18: ReactJS logo..... | 28 |
| Figure 19: React Native logo..... | 28 |
| Figure 20: ExpressJS logo | 29 |
| Figure 21: NodeJS logo | 29 |
| Figure 22: MongoDB..... | 30 |
| Figure 23: Docker logo..... | 30 |
| Figure 24:Gitlab logo..... | 31 |
| Figure 25: SourceTree logo | 31 |
| Figure 26: Visual studio code logo..... | 32 |

| | |
|---|----|
| Figure 27:StarUML logo | 32 |
| Figure 28: Draw.io..... | 32 |
| Figure 29: Postman logo..... | 33 |
| Figure 30: Trello logo..... | 33 |
| Figure 31: Figma logo | 33 |
| Figure 32: NodeJS 's project structure..... | 35 |
| Figure 33: ReactJS 's project structure | 36 |
| Figure 34: React Native 's project strcture | 38 |
| Figure 35: Generate token flowchart | 39 |
| Figure 36: Generate token | 39 |
| Figure 37: Flowchart of verify token..... | 40 |
| Figure 38: Implement verify token..... | 41 |
| Figure 39: Flowchart of QRCode scanner..... | 42 |
| Figure 40: QRcode scanner implement code..... | 42 |
| Figure 41: Admin's login page | 52 |
| Figure 42: Admin dashboard | 52 |
| Figure 43: Details information of branch owner account..... | 60 |
| Figure 44: List of branch owner account..... | 60 |
| Figure 45: Create branch owner account..... | 61 |
| Figure 46: Create admin account..... | 61 |
| Figure 47: List of admin account..... | 62 |
| Figure 48: Edit admin information | 62 |
| Figure 49: Admin's profile..... | 62 |
| Figure 50:Change password admin | 63 |
| Figure 51: Change admin's profile | 63 |
| Figure 52:Homepage | 64 |
| Figure 53: Add new feedbacks | 64 |

| | |
|---|----|
| Figure 54:Welcome screen | 65 |
| Figure 55:Login screen..... | 65 |
| Figure 56: Scanner screen..... | 66 |
| Figure 57: Result of scan..... | 66 |
| Figure 58: View delivered package | 67 |

LIST OF TABLES

| | |
|---|----|
| Table 1: Planning table | 10 |
| Table 2: Functional project requirement | 12 |
| Table 3: Admin role summary task | 44 |
| Table 4: Branch-owner role summary task..... | 45 |
| Table 5: Customer role summary task..... | 46 |
| Table 6: Back-end summary tasks..... | 47 |

TABLE OF CONTENT

| | |
|---|-----|
| ACKNOWLEDGEMENTS | i |
| មុលនឹងយសដោប | ii |
| RÉSUMÉ | iii |
| ABSTRACT | iv |
| LIST OF ABBREVIATION | v |
| LIST OF FIGURES..... | vi |
| LIST OF TABLES..... | ix |
| TABLE OF CONTENT..... | x |
| INTRODUCTION | 1 |
| CHAPTER I: GENERAL PRESENTATION OF INTERNSHIP..... | 2 |
| 1.1. Introduction | 2 |
| 1.2. Vision, Mission, and Objective | 2 |
| 1.2.1. Vision..... | 2 |
| 1.2.2. Mission..... | 3 |
| 1.2.3. Objective | 3 |
| 1.3. ITC Organization Chart..... | 3 |
| 1.4. Address and Contact..... | 4 |
| 1.5. Department of Information and Communication Engineering | 5 |
| CHAPTER II: THE INTERNSHIP PROJECT PRESENTATION..... | 6 |
| 2.1. Project Structure | 6 |
| 2.2. Duration..... | 6 |
| 2.3. Problematic..... | 7 |
| 2.4. Objective | 7 |
| 2.5. Software development methodology..... | 8 |
| 2.5.1. Scrum Framework..... | 8 |
| 2.5.2. Team Structure | 8 |
| 2.5.3. Team Rule..... | 9 |
| 2.6. Planning Project Table | 10 |
| CHAPTER III: PROJECT ANALYSIS, AND DESIGN..... | 12 |
| 3.3. Case Study of Requirement..... | 12 |
| 3.1.1. Functional Requirement..... | 12 |

| | |
|---|-----------|
| 3.1.2. Non-Functional requirement..... | 13 |
| 3.2. System and Design | 13 |
| 3.2.1. Physical Architecture | 13 |
| 3.2.2. Logical Architecture | 15 |
| 3.3. Project Analysis..... | 16 |
| 3.3.1. Use Case Diagram..... | 16 |
| 3.3.2. Database Schema | 19 |
| 3.4. Activity Diagram..... | 22 |
| 3.4.1. Add New Admin Account Diagram | 22 |
| 3.4.2. Add New Branch Owner Account Diagram | 23 |
| 3.4.3. Add new Item for Delivery | 24 |
| 3.4.4. Scan QR Code for Tracking Package | 25 |
| 3.5. Programming Language | 26 |
| 3.6. Frameworks and Technology | 26 |
| 3.7. Version Control System | 31 |
| 3.8. Choice of Tools | 32 |
| CHAPTER IV: PROJECT IMPLEMENTATION | 34 |
| 4.1. Project setup | 34 |
| 4.1.1. Backend project setup..... | 34 |
| 4.1.2. Frontend Project setup..... | 34 |
| 4.1.3. Mobile Project setup | 34 |
| 4.2. Project Structure..... | 35 |
| 4.3. Project Implementation flowchart..... | 39 |
| 4.3.1. Generate token flowchart..... | 39 |
| 4.3.2. Verify token flowchart..... | 40 |
| 4.3.3. QR Code Scanner flow chart | 42 |
| CHAPTER V: CONCLUSION | 44 |
| 5.1. Completed and Uncompleted Tasks..... | 44 |
| 5.1.1. Admin Roles Table | 44 |
| 5.1.2. Branch Owner Role Table | 45 |
| 5.1.3. Customer Role Table | 46 |
| 5.1.4. Backend Task Summary Table | 46 |
| 5.2. Strong Point..... | 48 |

| | | |
|---|-------------------|-----------|
| 5.3. | Weak Point | 48 |
| 5.4. | Difficulties..... | 48 |
| 5.5. | Experiences | 49 |
| 5.6. | Perspectives..... | 49 |
| 5.7. | Summary | 50 |
| REFERENCES | | 51 |
| APPENDIX..... | | 52 |
| APPENDIX A: Screenshot of Delivery Service System..... | | 52 |
| APPENDIX B: QR Code Scanner App | | 65 |

INTRODUCTION

As engineering students of the Department of Information and Communication Engineering at ITC, the students are required to take an internship for thesis defending before graduation, so they can apply the knowledge that they have learned from school as well as acquire new knowledge both soft and hard skill from a company, organization or somewhere else.

Due to the year fourth internship required students to hold an acceptable project considered by their own adviser. Meanwhile, management ITC needs a system to help the delivery service system manage their delivery faster and easier than usual. Delivery Service System to manage all products needed to deliver and track location when product arrival update status. Moreover, this application is built for a convenient branch Owner team to manage their own branch for customer requests to delivery.

As a result, my team has proposed a project on the Web application “Delivery Service Management System” to meet these constraints to facilitate item tracking to manage their system faster as usual.

This is the final internship report such a thesis divided into 5 chapters. The first is the general presentation of the internship project and information about place where I took in order to build this application. The second focuses on the internship project presentation that I work on for two months internship. Third is the analysis of project ideas, conception and design that talk about main functional requirement and optional functional requirement. Fourth talk about technology and tool that I use. Fifth is focuses on project implement. The last point is screenshot about overall figures that I have done in this project.

CHAPTER I: GENERAL PRESENTATION OF INTERNSHIP

This chapter will discuss details about the project that working on, its purpose, Activities, and service. Adding that members also find and solve the problem and object of this project. In addition, I arranged the timetable as a schedule of the activities during this internship. In the meantime, this chapter also covers the scope and limitations of this project, and I have discovered what I am able to do and what I have learned during this journey.

1.1.Introduction



Figure 1: ITC logo

The Institute of Technology of Cambodia (ITC) is the organization in charge of educating people in technology and engineering in Cambodia. It provides undergraduate and graduate degrees in a range of engineering, scientific, and technology disciplines. The school is committed to boosting technological research and education while also advancing the nation's industries and infrastructure.

1.2.Vision, Mission, and Objective

1.2.1. Vision

The Institute will expand into an international public higher education institution with multidisciplinary training, providing scientific research and innovation as a priority, continuing to produce high-quality human resources in science and technology as a hub for technology transfer. And skills to the national community and continue to expand cooperation with national and international partners to contribute to the development of the Cambodian economy.

1.2.2. Mission

Provide quality and equitable education in science and technology to become technicians with high professional ethics, entrepreneurial ability, creativity, innovation, virtue, morality, conscience, patriotism, social responsibility, and high awareness. Assignment to contribute to the development of the country and develop scientific research, transfer technology to the national community to enhance production capacity and harmonization.

1.2.3. Objective

- Continue to improve the quality of training.
- Continue to strengthen and expand scientific research, technology transfer, and expand cooperation.
- Continue to strengthen good governance.
- Continue to develop physical materials.
- Continue to strengthen financial management.
- Continue to strengthen and expand human resources.

1.3. ITC Organization Chart

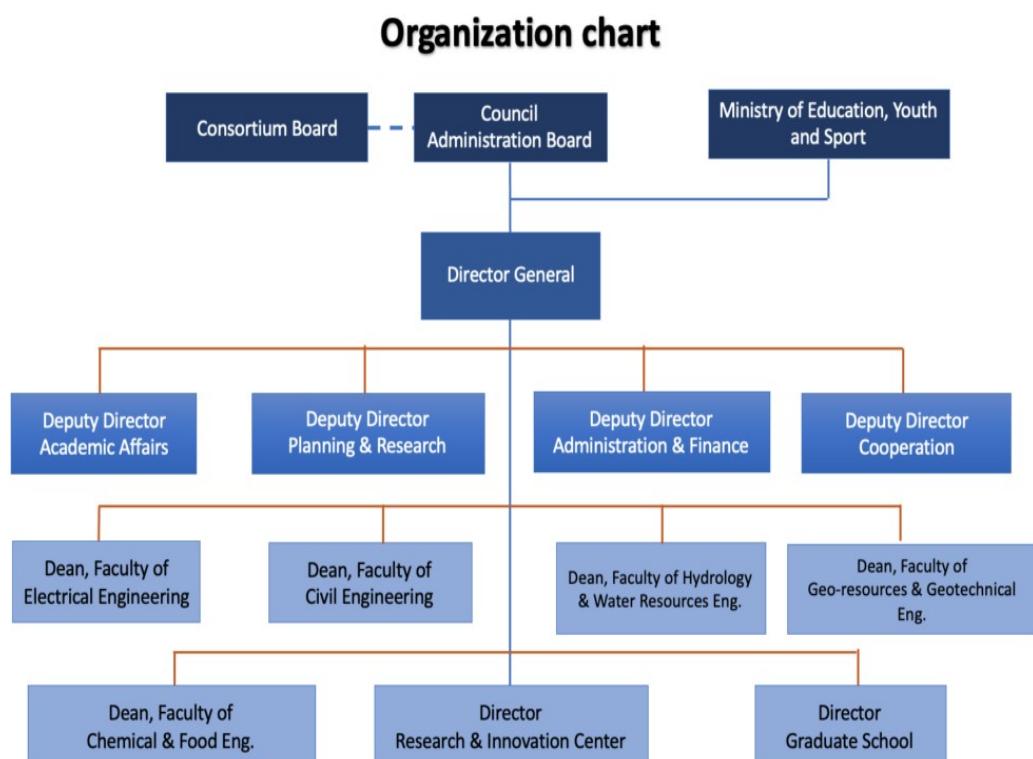


Figure 2: ITC organization

1.4. Address and Contact

- Address : PO Box 86, Russian Conf. Blvd. Phnom Penh, Cambodia
- Tel : (+885) 23 880 370 / 982 404
- Email : info@itc.edu.kh
- Website : <https://www.itc.gov.kh/>
- Facebook : <https://www.facebook.com/itckh>
- Youtube : <https://www.youtube.com/c/itcedukhcambodia>
- Telegram : <https://t.me/itckh>
- Location : Shown in *Figure 3*

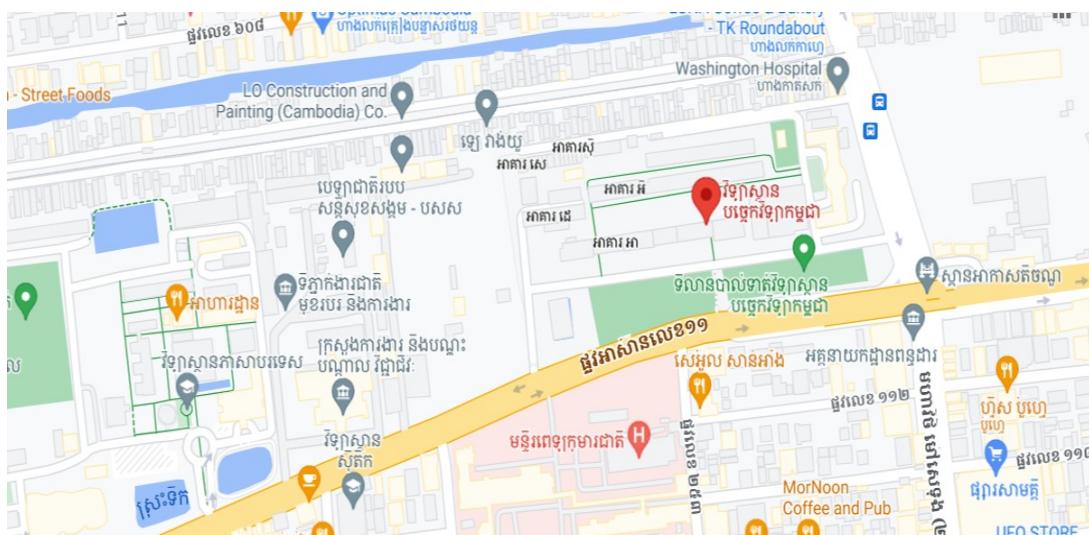


Figure 3: ITC location map

1.5. Department of Information and Communication Engineering



Figure 4: GIC department logo

The Institute of Technology of Cambodia (ITC) is a leading institution in Cambodia that offers a range of programs in information and communication technology engineering. The Department of Information and Communication Technology Engineering at ITC is responsible for providing training and education in the fields of technology, engineering, and computer science. In terms of services, ITC provides the following:

- Software Development
- Web Development
- Mobile App Development
- Database Management
- Network
- Cybersecurity Services
- IT Consultancy
- Research and Development
- Training and Education

These services are designed to meet the needs of various organizations and businesses seeking to leverage technology to improve their operations and services.

CHAPTER II: THE INTERNSHIP PROJECT PRESENTATION

For the internship during two months at the Institute of Technology of Cambodia, I worked on a project called “Delivery Service Management System”, which is a web application developed to manage delivery products and monitor all effective delivery. The goal of the Delivery Service System project is a web-based digital platform that connect providers with customers to efficiently order and deliver goods and service. it allows users to browse products, place orders, make payments, and track deliveries in real-time.

2.1. Project Structure

The internship and project development were guided and assisted by:

- Academic Supervisor: **Mr. HOK Tin**, Lecturer in the Department of Information and Communication Engineering.
- Project Advisor: **Mr. HOK Tin**
- Frontend Developer: **Mr. Sok Pagnavath**
- Backend Developer: **Mr. Rotha Dapravith**
- DevOps: **Ms. Toun Sreynit**

2.2. Duration

For defense in the four years at the semester II, my internship starts only two months that I statedinternship from 7 August 2023 to 7 October 2023, and I work with Team and spend time to learnnew technology for mobile App React-Native is an exciting framework that enables web developers to create robust mobile applications using their existing JavaScript knowledge. Furthermore, I try to understand about my requirement of the project and responsibility, task, process, and flow of the project.

2.3. Problematic

We are building this delivery service system because we noticed businesses struggling with their delivery operations business. This project will make things smoother and more efficient. There are some problems that need to solve including:

- **Delivery process management is difficult to handle:** Without advanced logistics technologies, delivery companies struggle to efficiently manage services and track packages, leading to delays that frustrate customers, drive up costs, and threaten competitiveness. To keep pace in a fast-moving market, these companies must collaborate with partners to enhance their strategic solutions and logistics technology.
- **Delivery package challenge:** The issue at hand for administrators and branch owners are facing have difficulty managing package deliveries promptly, particularly during periods of high customer demand. This leads to challenges in managing time effectively for deliveries.
- **Branch owners have trouble control package:** Branch owners are crucial in handling deliveries and customer concerns, but managing package delivery after transactions with admin can be challenging due to their individual work.
- **Customer have trouble finding package code for tracking:** Branch owners struggle with managing customer orders, missing packages, and incorrect delivery addresses without a proper delivery system, as they struggle to track package locations.

2.4. Objective

The Objective of developing an all-inclusive delivery service system is to tackle obstacles that businesses often encounter and to meet certain targets. In this section, I will dive deep into the main reasons and goals for establishing such a web application system include:

- **Enhance faster delivery process:** The goal of the system is to handle packages smoothly, especially when customers are demanding a lot. It has dashboards for administrators and branch owners with summarized data on branch owner accounts, package statuses, profits as well as details of recent deliveries.
- **User Convenience use the system website:** This system involves three main roles, admin, branch owner and customer. However, customer does not have an account to login, just locate package code and check to verify delivered package in the homepage.

- **Package and location tracking:** Branch owners prepare the packages with labels containing QR code, and customer schedules delivery with a QR reader which verifies all the details. The invoices will be made in form of PDF for easy, the branch owner dashboard displays the longitude and latitude values of the account holders and provides a comprehensive summary of the outstanding deliveries. a specific date format is used for the package history sidebar indicating the arrival date of each package.
- **QR Code scanning for package arrival status verification:** When a package arrives at its destination branch, the branch owner scans a QR code on the package using their mobile device. This scan confirms delivery and helps ensure accuracy by preventing the acceptance of fake or incorrect packages. Upon the package's arrival, the branch owner scans its unique QR code. They then contact the sender and receiver to arrange delivery and provide a PDF invoice to verify the package.

2.5. Software development methodology

2.5.1. Scrum Framework

Scrum is a framework for agile project management. It is particularly popular in software development because it is simple, flexible, and quickly provides usable results. It works with multidisciplinary teams that create working software in short iterations (sprints). Close cooperation with the customer, feedback, and team spirit ensures an effective work process.

My internship project at ITC used the SCRUM methodology because it is the best methodology for application development. In the scrum framework, the scrum team consists of a Product Owner, a Development Team, and a Scrum Master. There are four events Sprint Planning, Daily Scrum, Sprint Review, and Sprint Retrospective.

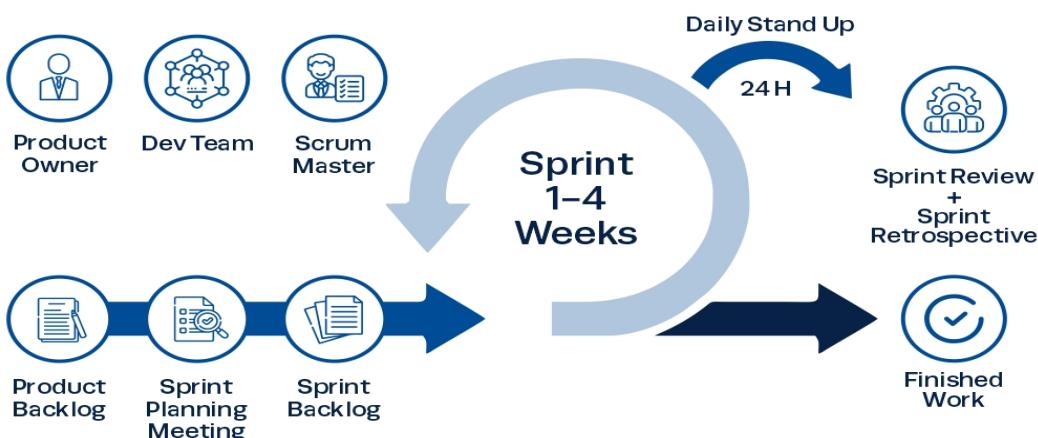


Figure 5: Scrum events

2.5.2. Team Structure

Following to Scrum Framework, our team has THREE roles: Product Owner (PO), Scrum Master(SM), and Development Team such as:

- **Product owner, Mr. Hok Tin**, plays an important role in maximizing the value of the product and the work of the development team by creating and maintaining a clear and feasible product backlog.
- **Scrum Master, Ms. Toun Sreynit**, a DevOps member is responsible for ensuring that the team follows Scrum theories, rules, and practices. He also creates and prepares the documentation of each sprint including the Trello project management board and the reports of sprint planning, sprint reviews, and sprint retrospectives.
- **The development team** consists of TWO members: (1) **Mr. Rotha Dapravith** (Backend)and (2) **Mr. Sok Pagnavath** (Frontend). We work together to deliver a releasable increment of Done at the end of each sprint. development team can add new tasks to the board at any time during the sprint.

2.5.3. Team Rule

According to Scrum, the team prepares specific time slots for Scrum events. All events are conducted at the following schedules:

- Daily Scrum will happen every working day from 10:00 a.m. to 10:15 a.m .
- Sprint Review will happen when each sprint is finished. It will gather actionable feedbackon what the Team has completed.
- Sprint Retrospective will happen after Sprint Review. This meeting is facilitated by the Scrum Master at which the team discusses the just-concluded sprint and determines what could be changed that might make the next sprint more productive.
- Sprint Planning will happen before the implementation new sprint. It is an event in the Scrum framework where the team determines the product backlog item they will work onduring that sprint and discusses their initial plan for completing those product backlog items.

2.6. Planning Project Table

| Sprint | WEEKS | | | | | | | | |
|----------|-------|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| Sprint 1 | | | | | | | | | |
| Sprint 2 | | | | | | | | | |
| Sprint 3 | | | | | | | | | |
| Sprint 4 | | | | | | | | | |
| Sprint 5 | | | | | | | | | |
| Sprint 6 | | | | | | | | | |

Table 1: Planning table

As I shown in *Table 1* During this internship period, I have spent the first two weeks to learn about new technology and understanding the nature of the project. The work for each sprint is described below:

1. Sprint 1:

- Get the required project scope, goals, and objectives.
- Analysis of the project scope.
- Start Design User Interface in Figma.
- Set up the development environment and learn to use the version control system.
- Set up the database and define the data model.

2. Sprint 2:

- Learn new technology frameworks.
- Learn Scrum Methodology and Software development life cycle.

- Implement with full authentication.
- Divide task responsibilities both frontend and backend.
- Design User Interface and improve User Experience.
- Admin login authentication.

3. Sprint 3:

- User management functionalities.
- Item management for delivery.
- Fetch API from the backend integrates with the front end.
- Generate QR code for scan to track products.

4. Sprint 4:

- Implement source destination to final branch destination.
- Admin dashboard management.
- Branch owner dashboard management.

5. Sprint 5:

- Pending package management.
- Conduct invoice for package history.
- Build QRcode for Scan update arrival status.
- Customers homepage and customer's feedback.

6. Sprint 6:

- Debug and maintain if there are any issues.
- Review and implement check code.
- Maintenance and Testing.
- Prepare document for writing report deliver to department.

CHAPTER III: PROJECT ANALYSIS, AND DESIGN

In this section, I begin to identify all the main functional requirement and optional functional requirement of the system. then I will describe my design, and the process of building the web application system.

3.3.Case Study of Requirement

3.1.1. Functional Requirement

Delivery Service System creates with three differences role of users. For any user role, they can work on their functions as shown in the Table 2. Hence the requirement is divided into two parts which is functional and non-functional requirement.

| Roles | Module | Feature in module |
|---------------|---------------------|--|
| Admin | Authentication | Login |
| | | Logout |
| | Dashboard | Total users, total package, total profit |
| | Account Management | Manage admin account |
| | | Manage branch owner account |
| Branch Owners | Authentication | Login |
| | | Logout |
| | Dashboard | Total pending, incoming, delivered package, total profit |
| | Package Management | Manage Package |
| | | Package Status |
| | | Package History |
| | QRCode Management | Generate QRCode |
| | | Scan QRCode |
| | | Display QRCode |
| | Location Management | Manage Branch owner Location |
| | | Manage Source Destination |
| Customers | Homepage | Find Package Code |
| | | View Details Package |
| | | Leave Feedbacks |

Table 2: Functional project requirement

3.1.2. Non-Functional requirement

Non-functional requirements focus on enhancing the system's quality rather than its visible features. Key non-functional aspects we've prioritized include:

- **Clean Code:** Our team ensures our code is well-organized, properly indented, and thoroughly commented, detailing functions, parameters, and return variables.
- **Performance:** We prioritize a seamless user experience, ensuring the website operates without glitches.
- **Stability:** Our goal is a consistent website performance that prevents errors, slowdowns, or data loss.
- **Security:** We've fortified our website's defenses to deter potential hacking attempts.
- **Maintenance:** Regular checks help us identify and rectify any issues, enhancing the application's quality.
- **User Interface:** We have designed a straightforward, intuitive, which make user seemly to experience with it.

3.2. System and Design

3.2.1. Physical Architecture

As shown in the *Figure 6*, the physical architecture of the system is described as below through the figure and the explanation of each component. It defines the distribution and arrangement of the system's components, including the client web browser, web server, database server, and file system, among others. The physical architecture is essential because it addresses the practical aspects of deploying and running the system in a real-world environment.

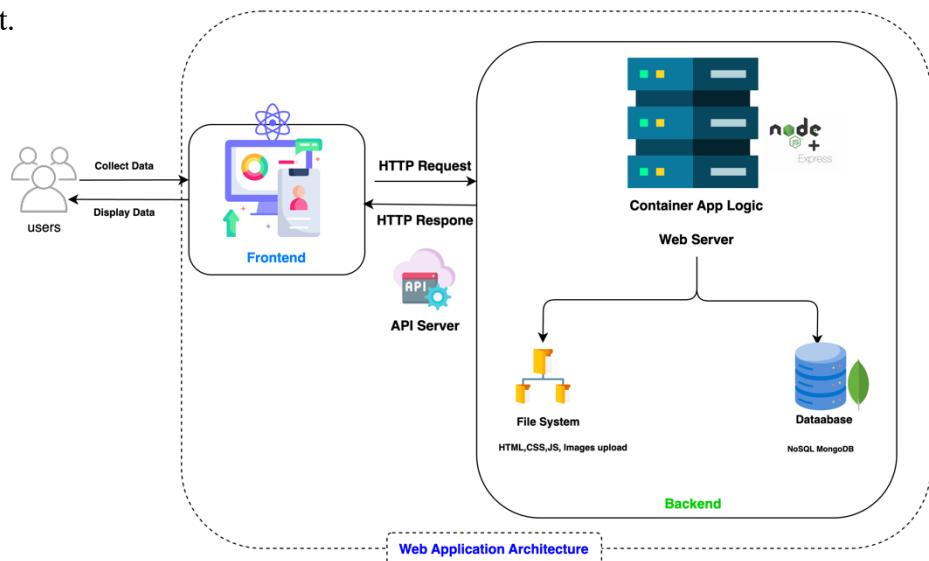


Figure 6: Physical architecture of system

- **Users:** User can access by request for client then Client will response back to User.
- **Frontend:** This represents the front-end of the application, which is built using React.js or a similar JavaScript framework/library.
- **Client Web Browser:** The client web browser is the user's interface to interact with the delivery service system. It can be any modern web browser such as Chrome, Firefox, Safari, or Edge. The web browser renders the user interface components, handles user inputs, and communicates with the web server to request and receive data.
- **Web Browser:** The web server acts as the central hub for processing client requests and serving responses. It receives requests from the client web browser and handles them by retrieving or updating data from the database server. It also performs business logic and applies any necessary processing before sending the response back to the client. The web server utilizes technologies such as NodeJS and React JS to handle the server-side processing and generate dynamic web pages.
- **Database Server:** The database server stores and manages the system's data. It uses a database management system (DBMS) such as MySQL and NoSQL to store structured data related to menus, user information, reservations, and other entities. The database server ensures data integrity, provides efficient data retrieval and storage, and supports query operations.
- **File System:** The system may require storing files such as images or documents related to menus or user profiles. A file storage component, such as a local file system can be used to manage and store these files.
- **API Server:** It is used to handle contain logical code with routes and controller. API servers act as data bridges, enforcing rules for seamless data exchange between different systems. It is enabling diverse applications and services to integrate, playing a vital role in today's digital ecosystem.

3.2.2. Logical Architecture

The logical architecture as shown in Figure 7 is describes about the idea of how Structureframework using MERN Stack structure to build the web application with Node JS and Express JS implement with ReactJS.

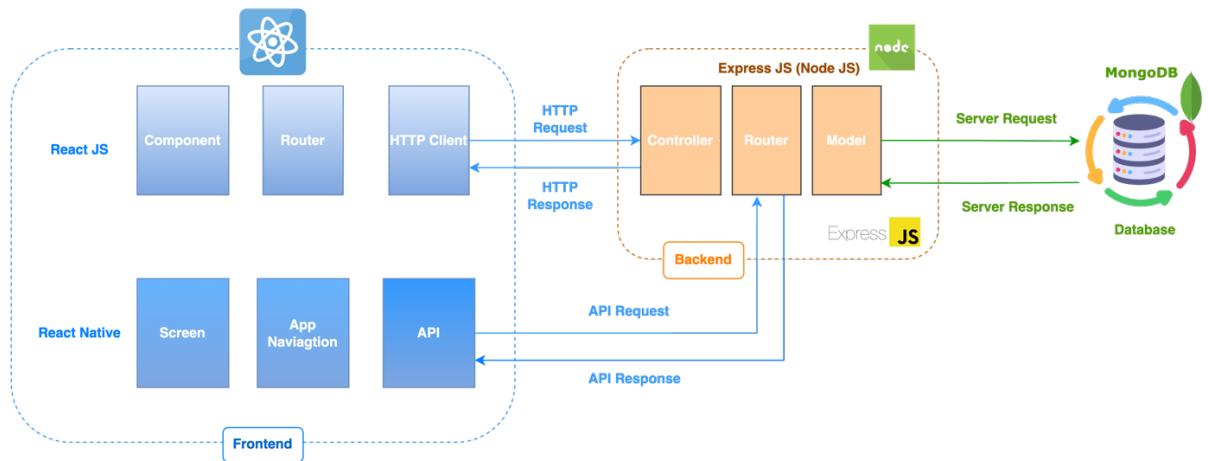


Figure 7: Logical architecture of system

➤ **ReactJS** is a JavaScript library for creating user interfaces. It focuses on the display layerof the application and offers tools for creating component-based and reactive user interfaces. The following elements are commonly found in a React.js application's logicalarchitecture:

- **Components:** are reusable UI elements that provide state management and render logic capabilities. They serve as the foundation for React applications. A component might be as basic as a button or as sophisticated as a whole page. They may also include additional parts, creating a layered structure.

- **Routes:** Specify the page's endpoint and choose the component that should be displayed based on the current path.

➤ **NodeJS** is a runtime environment built on the V8 JavaScript engine, which allows you to run JavaScript on the server side. It follows a single-threaded, event-driven, and non-blocking I/O model, making it highly scalable and efficient for building server-side applications. The logical architecture of a Node.js application typically consists of the following components:

- **Controllers:** This layer houses the core business logic, processing incoming requests and formulating the appropriate responses.

- **Router:** These serve as the API gateways, directing incoming HTTP requests to the appropriate handlers.

- **Models:** Represent the data structure and interact with the database.

➤ **React Native** provides high-performance UI components that are platform-specific and can be used for both web and mobile applications. By rendering native APIs, it can replicate the look of native apps.

Here is the logical architecture diagram of a React Native which include the following components:

- **User Interface:** This component, which consists of screens, views, buttons, and interactive components for user interaction, represents the mobile application's visible elements.
- **App Navigation:** Implement user navigation screen between sections or features, navigation components manage the flow between various screens or views within the application.
- **API:** This component is important for connecting with the backend to retrieve data from the API server implemented by MongoDB. It is used to handle data retrieval, storage, and synchronization.

3.3. Project Analysis

In this section, I begin to use UML Diagram that refer to a standard language. I used it to visually model processes or designs in software bases system.

3.3.1. Use Case Diagram

The use case diagram is important for defining and validating the required features in our application. Each use case box represents an important module that is implemented and contains other features within it. There are three types of users: admins, branch-owners and Customer. Branch Owner are requiring logging in, whereas admins must log in before accessing all functionalities, But for Customers cannot login because they don't have any account to login and only find the package code for view their tracking package on delivered branch-Owner's location, then they can go to pick up their package if their product has delivered on location. The following figure is the use case diagram of a “Delivery Service Management System”.

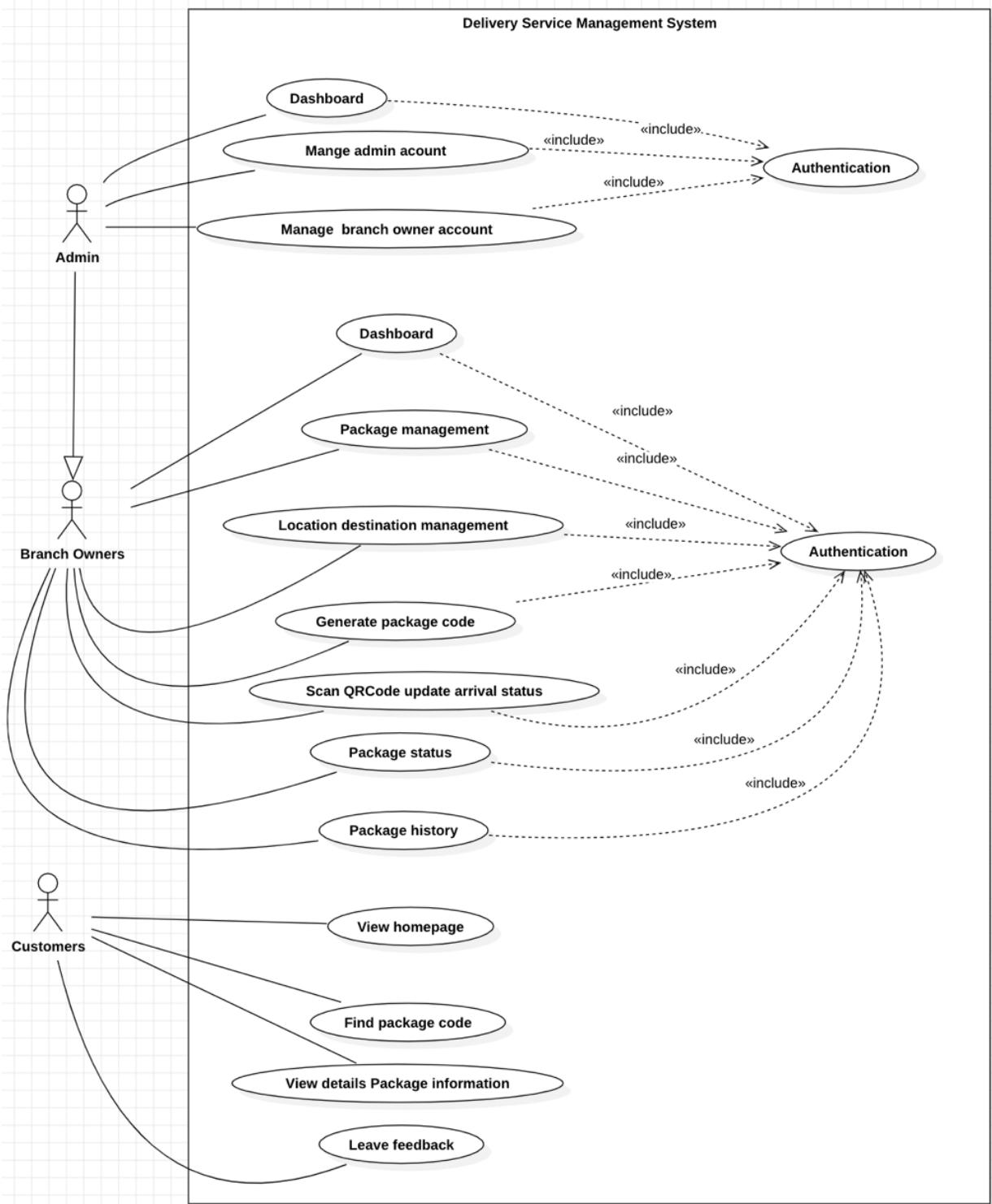


Figure 8: Use case diagram

Specification use case diagram of the functions include:

➤ **Admin:**

- **Authentication:** All types of users must have authentication are enabled to access any feature.
- **Dashboard:** Admin can see a summary data such as the total number of branch owner, incoming, pending and delivered package, total profit of any branch owners which have summarize with useful data include the total of user, pie chart of package status.
- **Manage admin account:** Admin can use the crud operation to create, update and delete admin account of any admins on the system.
- **Manage branch owner account:** Admin also can use the crud operation to create, update and delete branch owner account as well as manage admin account.

➤ **Branch Owner:**

- **Package management:** This use case, branch owner can use the crud operation to create, update and delete of any packages. in this package there are three main statuses include incoming, pending and delivered package, it is used to update arrival status of any package when it delivered to any destination as well as branch owner need to manage package to another destination.
- **Location Destination:** Branch owner can set origin source to destination for delivery any package that customers are request for purchase delivery service to deliver their own package to another branch owner location.
- **Generate QR Code:** this feature, branch owner can generate package code into QR code for scan verify package to avoid delivered incorrect package.
- **Scan QR Code:** If any package is delivered to destination, branch owner can scan qr code to confirm the package are arrival to any branch owner, so these branch owners obtain the package need to scan qr code to verify package, thus system will update arrival status automatically into specific branch owner.
- **Package Status:** After getting the package, branch owner clicks on mark delivered button for package are arrival on their own destination, else not, they can click on the mark not delivered button to confirm the package are not delivered yet.

- **Package History:** This feature, branch owner can view the history of package and download an invoice PDF that details information includes specific date, destination, display sender and receiver phone number and show details about package, branch owner information.

➤ **Customers:**

- **View Homepage:** This use case use for customer, they can view the homepage website.
- **Find Package code:** customer can find their own package by search the find now button to find their own package with details information and attached with QR Code to verify package and can track the package are delivered or not delivered yet.
- **View Details package information:** The system will show about the details information of any package that customer request to send to another branch owner.
- **Leave Feedback:** In the end of system, customers can comment to review about their own feedback to rate service system for delivery service good or not good.

3.3.2. Database Schema

As shown in Figure 11 is database schema that I create after analyzing the main functional of the project, one I can find the given needed and the categories in the different table. Form the conception of association entity and relational model, we can find the relations and the cardinalities between each table. In addition, we can define the conceptual model of data with the attributes in each entity.

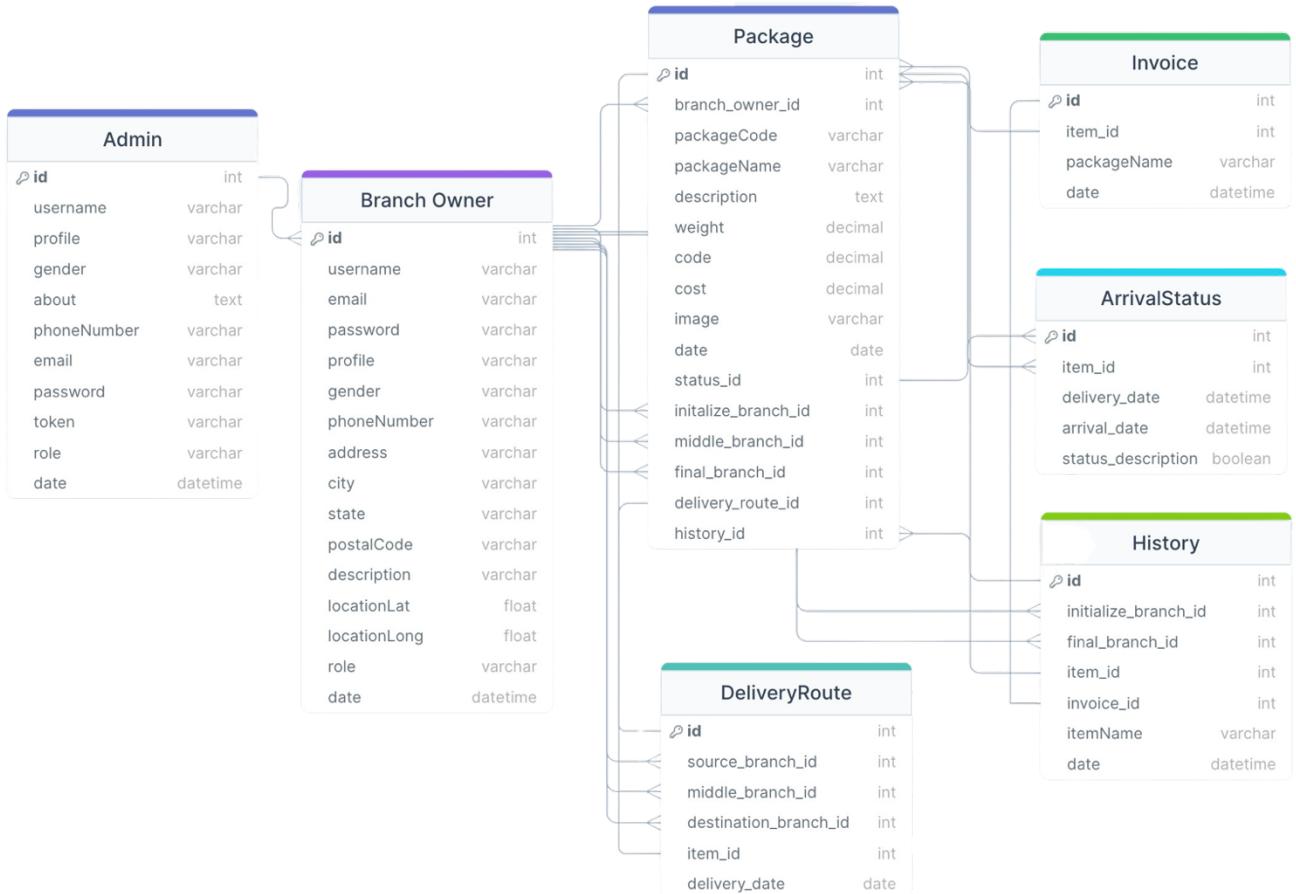


Figure 9: Database Schema

There are 7 tables in the database schema of the system as shown above to represent the overall database schema in our system that makes the system work.

Figure 9 is shown about the overall of database schema that I designed after analyzing the concept and main functionalities of the whole project. Here is the summarize of any table include:

- **Admin:** This table stores information about administrators in the system, including their username, profile, gender, contact details, role, and creation date. It is used for managing system administrators.
- **Branch Owner:** The branch owner table contains details about branch owners, including their username, contact information, address, role, and creation date. This table is used to manage branch owner accounts.

- **Package:** This table manages information related to items/packages for delivery. It includes package details, weight, cost, images, and delivery route information. It is crucial for tracking and managing the delivery of packages.
- **Arrival Status:** Arrival status records details about the arrival of packages, such as delivery and arrival dates, and status descriptions. This information helps in tracking and verifying the status of delivered items.
- **Delivery Route:** The Delivery Route table tracks the routes taken by packages during delivery. It includes source, middle, and destination branches, along with the delivery date. This table is essential for mapping delivery routes.
- **History:** This table records the history of item movements between branches, including the branch from which an item originated, the final branch, and related details. It is used for keeping a record of item movements.
- **Invoice:** The invoice table manages invoice information associated with items. It includes details like package name, creation date, and related item. It is essential for billing and financial tracking.

3.4. Activity Diagram

3.4.1. Add New Admin Account Diagram

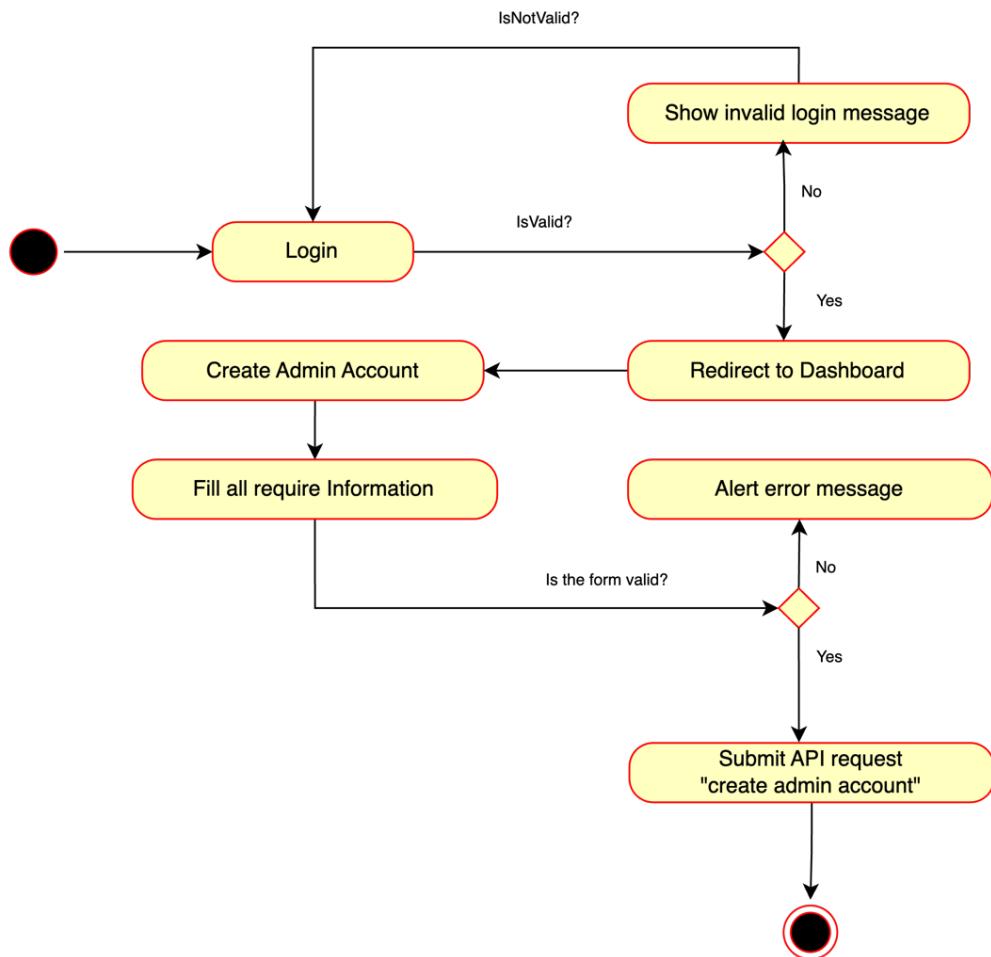


Figure 10: Activity diagram of add new admin account

In Figure 10 is an activity diagram for admin create new admin's account:

- First, initiate the process by logging into the system. This login step is essential to access the administrative functions.
- Second, upon a successful login, the user gains access to the admin dashboard, which provides them with the necessary administrative controls and features.
- Third, within the admin dashboard, the user can locate and select the "Create an Admin Account" option from the sidebar. This specific feature is designed for adding new admin accounts.
- Fourth, the user is prompted to input all the required data and details for the new admin account. This typically includes information such as the admin's username, password, email, and other relevant particulars.

- Finally, after entering all the necessary information, the user completes the process by clicking the "Create an Admin Account" button. This action signifies the user's intent to create the new admin account.

3.4.2. Add New Branch Owner Account Diagram

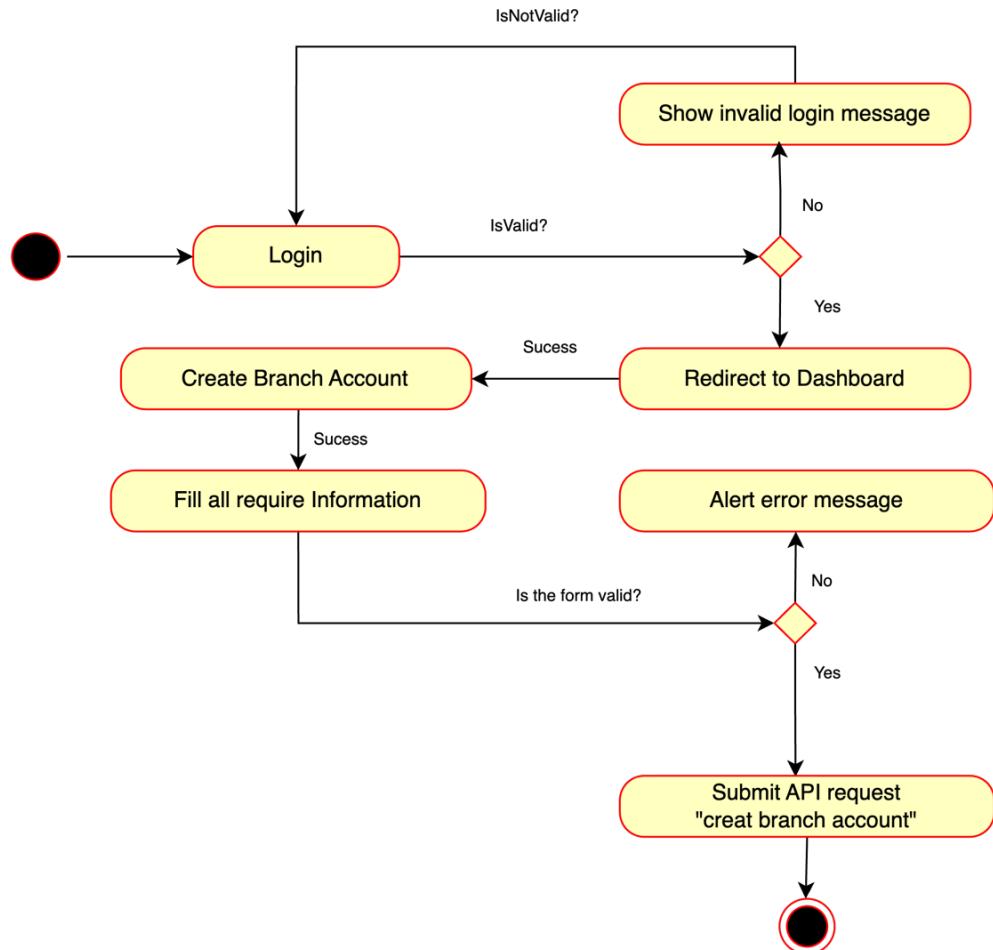


Figure 11: Activity of Add new branch owner account

In Figure 11 is an activity diagram for admin create new branch owner's account, it familiar like create admin's account as below:

- First, log in to the system to access administrative functions.
- Second, after a successful login, proceed to the sidebar and select "Create a Branch Account."
- Third, enter all the required data and details for the branch owner.
- Finally, click the "Create a Branch Account" button to complete the process, successfully adding a new branch owner account. the system will submit API request for "create branch account".

3.4.3. Add new Item for Delivery

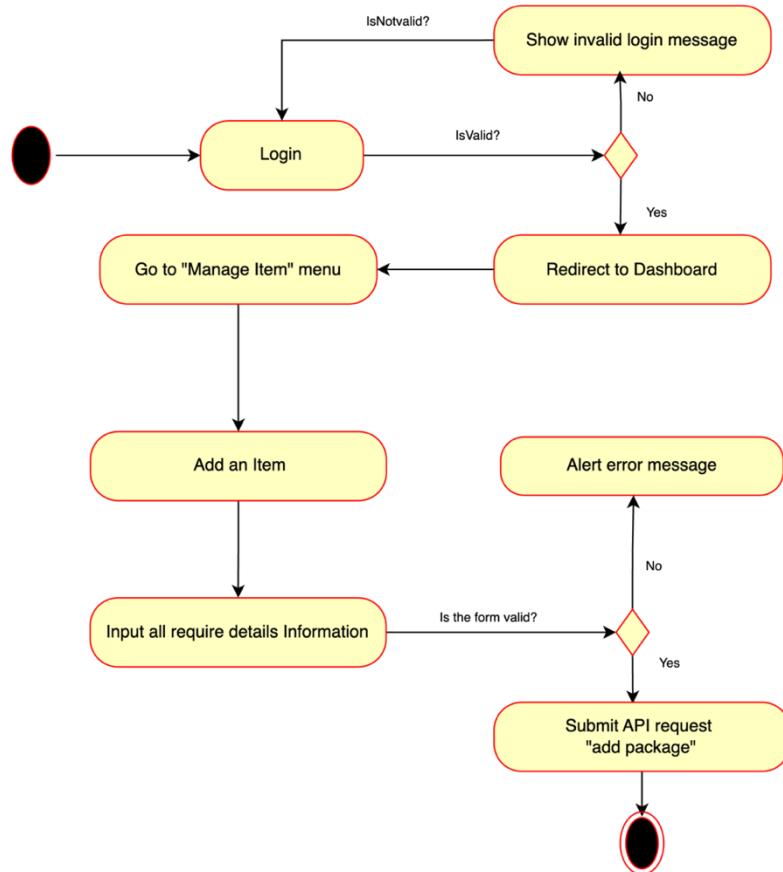


Figure 12: Activity diagram of add package for delivery

In Figure 12 is an activity diagram of add new item for delivery on the system:

- First, the Branch Owner must log into the system for access.
- Second, upon a successful login, the Branch Owner navigates to the "ManageItems" section via the sidebar.
- Third, within this section, they click the "Add an Item" button, triggering a popuppage.
- In this popup page, they are required to fill in all the necessary information about the item.
- An essential step in this process, they need to upload an image file of the item.
- Once all the information is filled out, they can click the "Submit" button to complete the process of adding a new item for delivery.

- Fourth, if the product has been delivered to the set destination, the Branch Owner has the option to click the "Mark as Delivered" button.
- Last, they can choose to select all items if multiple products have been delivered and then click the button, as mentioned earlier, to complete the process of adding new items for delivery.

3.4.4. Scan QR Code for Tracking Package

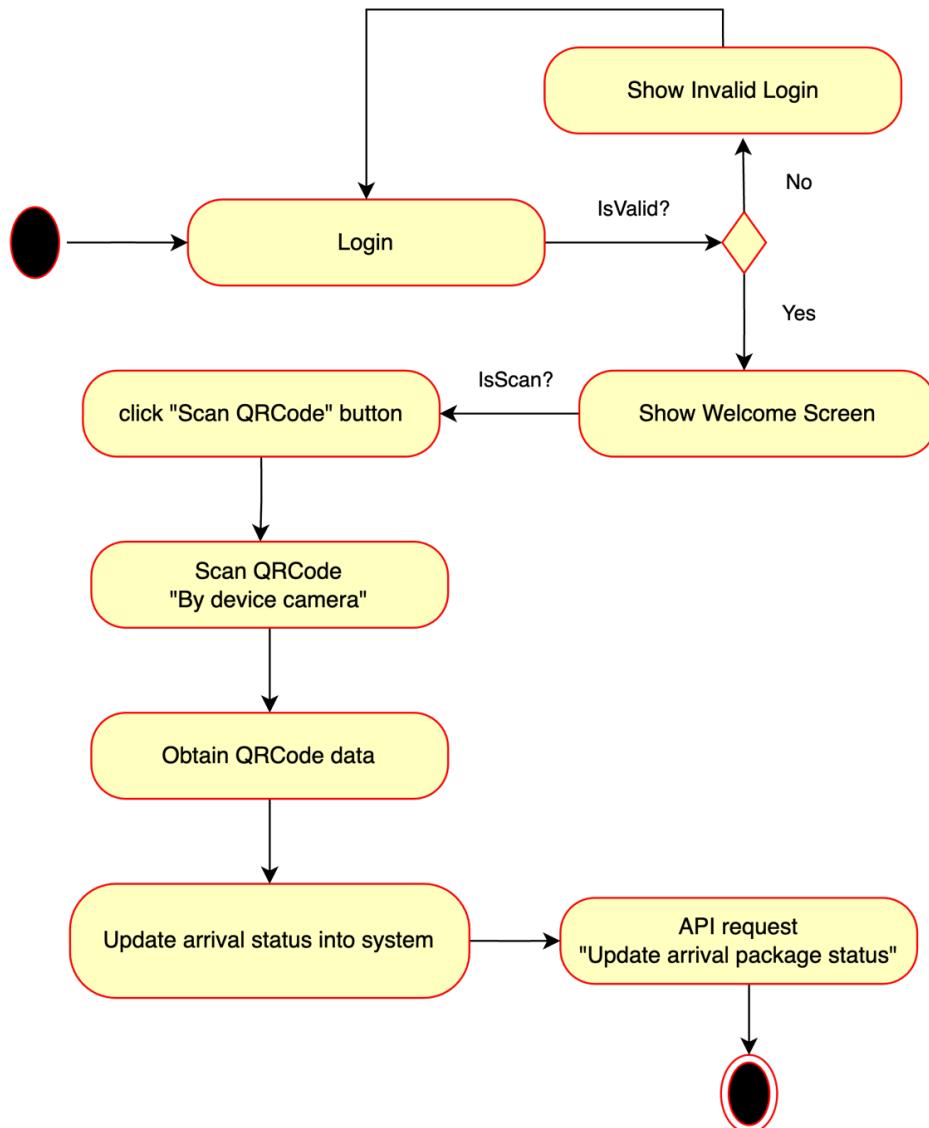


Figure 13: Activity diagram of QR Code scan

In Figure 13 is an activity diagram to shows scan QR Code for Tracking Item:

- First, the Branch Owner needs to access the "Manage Item" section in the sidebar.
- Second, they should select any item to preview its details by clicking the eye-buttonicon, which will pop up a details page.
- Third, on this details page, users can view the item's QR code, which is attached below the item information.
- Fourth, the Branch Owner can use their camera to scan the QR code for verify their own package.
- Last, after scanning the code, they can view the item code and continue the process of using the QR Code Scanner for tracking the item.

3.5. Programming Language

JavaScript [1] is a high-level, prototype-based, object-oriented programming language that is crafted to minimize implementation dependencies. I opted for the JavaScript programming language for my project due to its approachable learning curve, user-friendly object-oriented features, and its widespread adoption across various industries globally. Furthermore, JavaScript offers a blend of speed, security, and dependability.



Figure 14: JavaScript logo

3.6. Frameworks and Technology

For this Project my team choose **MERN Stack [2]** include MongoDB, Express, ReactJS and NodeJS. MERN Stack is a collection of powerful technologies and robust, used to develop scalable master web applications comprising backend, front-end, and database components. It is JavaScript that is used for the faster and easier development of full-stack web

applications. MERN Stack is a technology that is a user-friendly full-stack JavaScript framework for building applications and dynamic websites.



Figure 15: Mern Stack logo

HTML [3], CSS [4], JavaScript [5]: These are important for website development because they are the website that I use to build the website, and most websites can build these three web basics. So, I use **HTML [3]** to build the structure for my website and use **CSS [4]** for designing the UI for my website to look good and have more interest from many users, and the final one that is **JavaScript [5]** I use this programming language for behaviors of the website that can interact with users.



Figure 16: Html,css,javascript logo

SASS [6] is a CSS preprocessor that streamlines style sheet development with features like nesting, variables, and mixins. I chose Sass for its ability to enhance CSS writing and maintenance. My experience with Sass has been positive, as it has improved code organization and readability while reducing redundancy. Compared to traditional CSS, Sass offers greater modularity and reusability, making it a powerful tool for efficient styling in web development.



Figure 17:Scss logo

ReactJS [7] is a popular JavaScript library for building user interfaces due to its component-based architecture, virtual DOM, and strong community support. I chose it for its efficiency and strong community support. My experience has been positive, as it simplifies complex applications and offers a rich ecosystem of tools. Compared to alternatives like Angular, Vue JS provides flexibility, a smaller learning curve, and a lighter development approach.

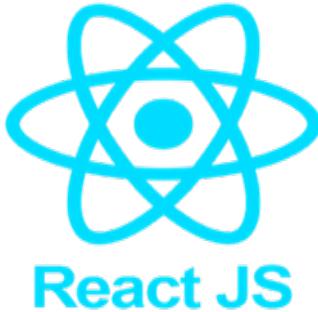


Figure 18: ReactJS logo

React Native [8] is a framework for developing mobile applications with JavaScript and React. I chose it because of its ability to construct cross-platform programs with a single codebase, cutting development time and cost. My experience with React Native has been great, as it enables efficient development and maintenance of both iOS and Android apps. When compared to native app development, React Native speeds up the process by allowing code sharing and providing a large library of pre-built components.

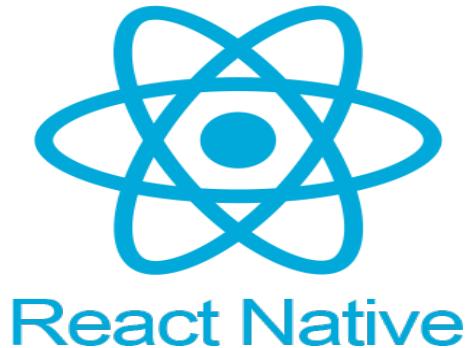


Figure 19: React Native logo

Node JS [9]: is a cross-platform, open-source server environment that can run on Windows, Linux, Unix, macOS, and more. Node.js is a back-end JavaScript runtime environment, runs on the V8 JavaScript Engine, and executes JavaScript code outside a web browser. Node.js lets developers use JavaScript to write command line tools and for server-side scripting. The ability to run JavaScript code on the server is often used to generate dynamic web page content before the page is sent to the user's web browser. I choose Node JS because it allows developer to use JavaScript on both the front-end and back-end. This enables a unified development language, as developer can write code in a single language throughout the entire stack. The challenge of this technology is managing callbacks, promises, and async/await patterns correctly requires careful handling to avoid issues like callback hell or race conditions.



Figure 20: NodeJS logo

Express JS [10]: is a lightweight and adaptable Node.js web application framework that simplifies the development of web apps and APIs. It was chosen for its lightweight nature, solid performance, and wide middleware support, which simplifies server-side development. I choose Express JS because It provides a straightforward and efficient way to create server applications. In comparison to other similar technologies, such as Hapi or Koa, Express.js is widely adopted and offers a large community and ecosystem, making it a dependable choice for web development in Node.js.



Figure 21: ExpressJS logo

MongoDB [11] is a NoSQL database management system that employs a document-oriented dataparadigm. It was chosen for its flexibility, scalability, and simplicity of working with semi-structured and unstructured data, making it well-suited for current online applications. My experience using MongoDB has been positive. It makes data storage and retrieval easier, especially when working with quickly changing data models. In comparison to traditional relational databases such as MySQL, MongoDB excels at horizontal scaling and provides greater agility when reacting to changing application requirements. It's an excellent solution for projects that require dynamic and adaptable data structures.



Figure 22: MongoDB

Docker [12] is an open platform for app development, shipping and running. Docker allows me to build, test, and deploy applications quickly. Docker packages software into standardized units called containers that have everything the software needs to run including libraries, system tools, code, and runtime. I used Docker for building the image and running the container of the code.



Figure 23: Docker logo

3.7. Version Control System

In the programming field such as web development, vision control is type of practice that tracks and provides control over changes to source code. I chose **GitLab** [13] for project version controls. For my project, version control is very importance because it helps me to manage my project. I can store my source code every time that I complete any functionality. Easy to get it back when writing something wrong in my project.



Figure 24:Gitlab logo

SourceTree [14] is a graphical user, which is a distributed version management system. It provides a visual interface to manage and track changes in source code, as well as collaborate with others on software development projects. I chose SourceTree because of its simple interface and powerful Git integration. It simplifies difficult version control proceduresand allows you to easily visualize branching, merging, and commit history.



Figure 25: SourceTree logo

3.8. Choice of Tools

Visual Studio Code [15] is a standalone source code editor that runs on Windows, macOS, and Linus. The top pick for Website development, with more extensions support. I used it because it is working fast for my computer, and it can create and compile the project similar other tools.

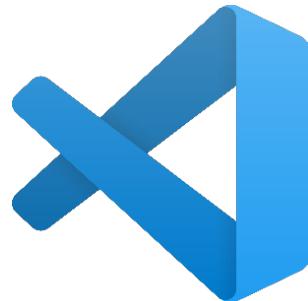


Figure 26: Visual studio code logo

StarUML [16] is a software engineering tool for system modeling using the Unified Modeling Language, as well as Systems Modeling Language, and classical modeling notations. I use this tool to construct use case diagram and activity flow chart diagram of my project.



Figure 27: StarUML logo

draw.io [17] is free online diagram software for making flowcharts, process diagrams, org charts, UML, ER and network diagrams, I use this tool because it is free software application for making flowchart and another diagrams.



draw.io

Figure 28:Draw.io

Postman [18] is an API platform for developers to design, build, test and iterate their APIs. It is an HTTP client that tests HTTP requests, utilizing a graphical user interface, through which we obtain different types of responses that need to be subsequently validated. I used postman for testing project API after I build it complete in each functionality before, I integrate it with front-end. When we have swagger configuration in our project, we just import the link of our API description, it generates all the requests for us to test easily.



Figure 29: Postman logo

Trello [19] is a collaborative work management app designed to track team projects, highlight tasks underway, show who they are assigned to, and detail progress towards completion. My team always use this tool for manage and assign represent each task to do and has a due time of any completed tasks.



Figure 30: Trello logo

Figma [20] is a cloud-based design tool that is similar to Sketch in functionality and features, but with big differences that make Figma better for team collaboration with additional offline features enabled by desktop applications for macOS and Windows. I use this tool for designUI for my whole project before considering to developing user interface for the whole system.



Figure 31: Figma logo

CHAPTER IV: PROJECT IMPLEMENTATION

In this section, the details about the implementation of the project during the internship. I explain how to set up tools and technology, project implementation, and installment.

4.1. Project setup

In this section, I will explain how to set up Environment for Frontend, Backend also Mobile App. To set up the required environment and tools for your project, follow these steps:

- Install **Node JS** version 18.18.0
- Install the **IDE**. In this case, I chose Visual Studio Code for the code editor.
- Install the **Postman** for testing API.
- Install the **Expo GO** for open-running React Native mobile applications.

4.1.1. Backend project setup

After preparing the necessary environment, I initialized the project using the command **npm init -y**. Next, I installed the required libraries with **npm install library-name**. Even after the project is up and running, we can add more libraries using the same command. After setup, you can start running the server by using **npm server.js** or **nodemon server.js**.

4.1.2. Frontend Project setup

To create a React JS project, you need to follow a few steps below:

- Prerequisites: Ensure you have Node.js and npm installed.
- Create App: In the terminal, **run npx create-react-app project-name**.
- Navigate: Use **cd project-name** to enter the project directory.
- Start Server: Run **npm start** to view the app in a browser.
- Project Structure: Main files include public/index.html (main HTML), src/App.js (main React component), and src/index.js (JS entry point).
- Production Build: Use **npm run build** for an optimized build.
- Development: Customize App.js, add components, and install libraries **with npm install**.

4.1.3. Mobile Project setup

Here is the following step need to set up react native for mobile app:

- Step1: installing Expo-CLI by using the command line:
- **npm install -g expo cli** or you might need to install necessary other dependencies by using the command line: **npm install**
- Step2: Creating the project name by using the command line:
- **npx create-expo-app [your-project-folder-name]**
- Step3: Install the application by using the command Line: **npx expo start**

4.2. Project Structure

The application structure of ReactJS is basically the structure of folder, sub-folders and files included in a project. When we create the project, we will get an overview of the application structure as shown in Figure 32.

➤ NodeJS structure as shown in Figure 32:

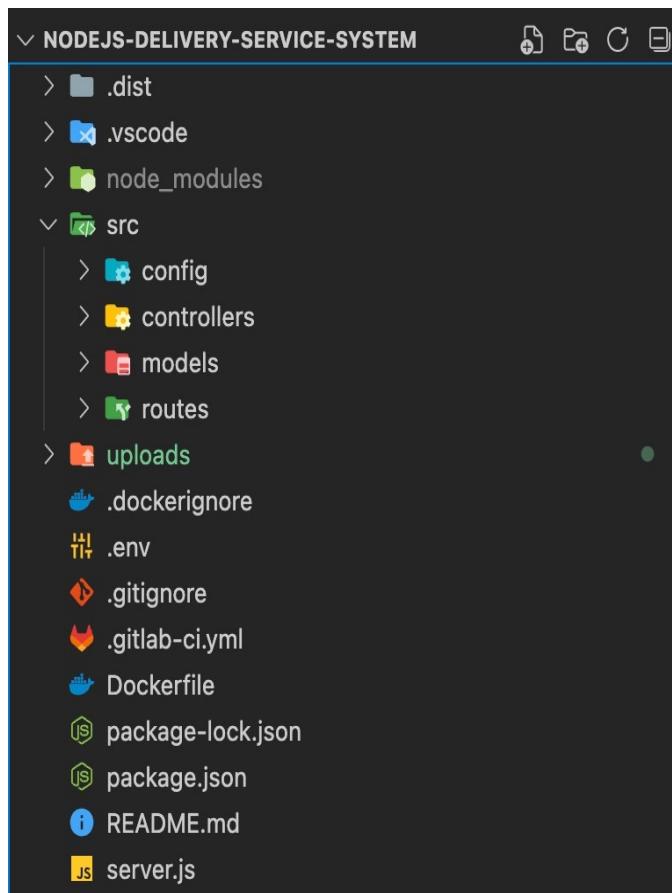


Figure 32: NodeJS 's project structure

- **node_modules:** It stores all the third-party libraries and other packages that project depends on. These packages are specified in **the package.json** file and are installed into the **node_modules** folder.
- **config:** This directory contains configuration files for the application. It is used to store config database connection.
- **controllers:** This directory Contains logic to handle requests and send responses.
- **models:** Defines data models, often representing database schema tables.
- **routes:** The router directory includes the application's route definition. It defines how distinct HTTP requests are routed to the appropriate controller function. It is used to submit a request to a controller and receive a response, as well as to respond to an HTTP request.
- **Uploads:** This directory is used to store files that are uploaded by users through the application. Depending on the nature of your application, these files could be images, documents, audio files, videos, or any other type of file that users might need to upload.
- **File server.js:** handle server running such as port, database connection of the overall application.

➤ ReactJS project structure as shown in Figure 33:

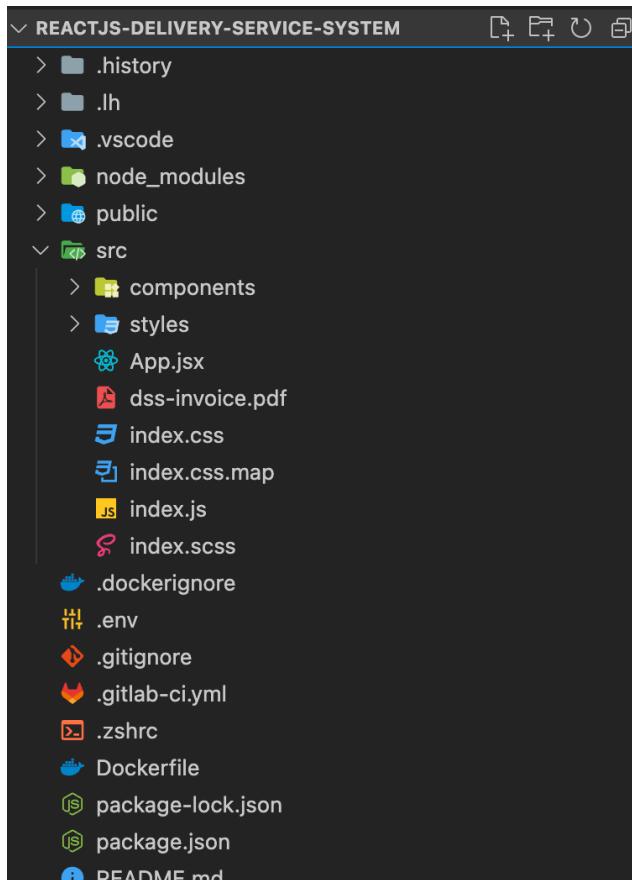


Figure 33: ReactJS 's project structure

- **node_modules:** It stores all the third-party libraries and packages that the project depend on. These packages are specified in the **package.json** file and are installed in the `node_modules` folder.
- **public:** it has the main file called `index.html` is the file that display result into the webpage and within it has the **manifest.json** file is store the **PWAs** (Progressive Web Apps).
- **components:** Contains reusable React components. Components define the visual and interactive parts of your app.
- **styles:** The folder directory contains all CSS files for the React application, with structure and contents varying based on the application's complexity and styling approach.
- **index.js:** is the file connected with the `index.html` in the public folder, this file used to render only one component of the project website.
- **App.jsx:** is used to render in the web browser and it always integrates connect with **index.js** file ready to be displayed.
- **File dotenv:** store some content used for connecting with backend project like **BACKENDAPI URL**.

➤ React Native structure as shown in Figure 34:

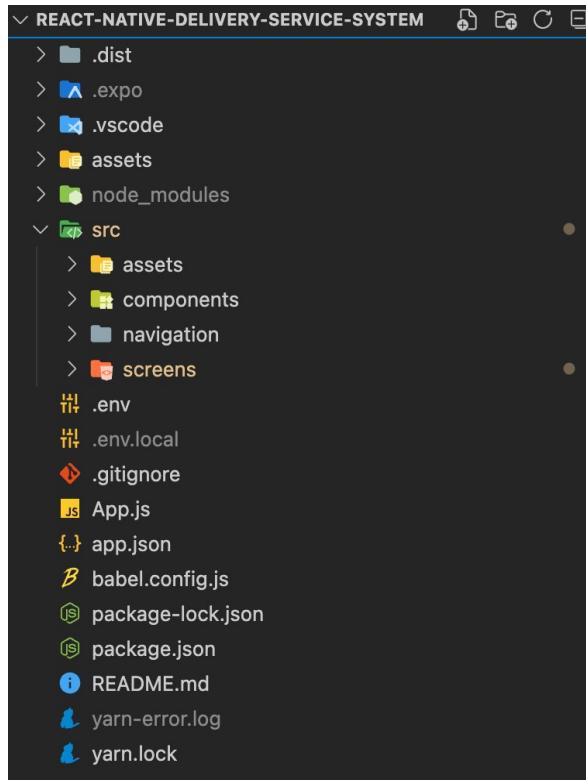


Figure 34: React Native 's project strture

- **node_modules:** contain all libraries and packages of the project after installing another libraries or dependencies it will be stored in **node_modules** folder.
- **assets:** the directory is used to store images and fonts that are required in the application that mostly call to use into the components folder.
- **components:** This folder is used to contain all reusable UI elements and used to wrap the application components and determine their overall layout.
- **navigation:** The navigation folder directory is used to store and manage the app's routing, combining various navigators. Includes stack is used for individual navigators for all different app sections and main navigator combines all stack navigators for app routing.
- **screens:** This directory is used to contain different pages or screens of your app like HomeScreen, Login Screen, etc.
- **File App.js:** This is often the main entry point of your app. It's where you configure the root component and initialize your app.

4.3.Project Implementation flowchart

4.3.1.Generate token flowchart

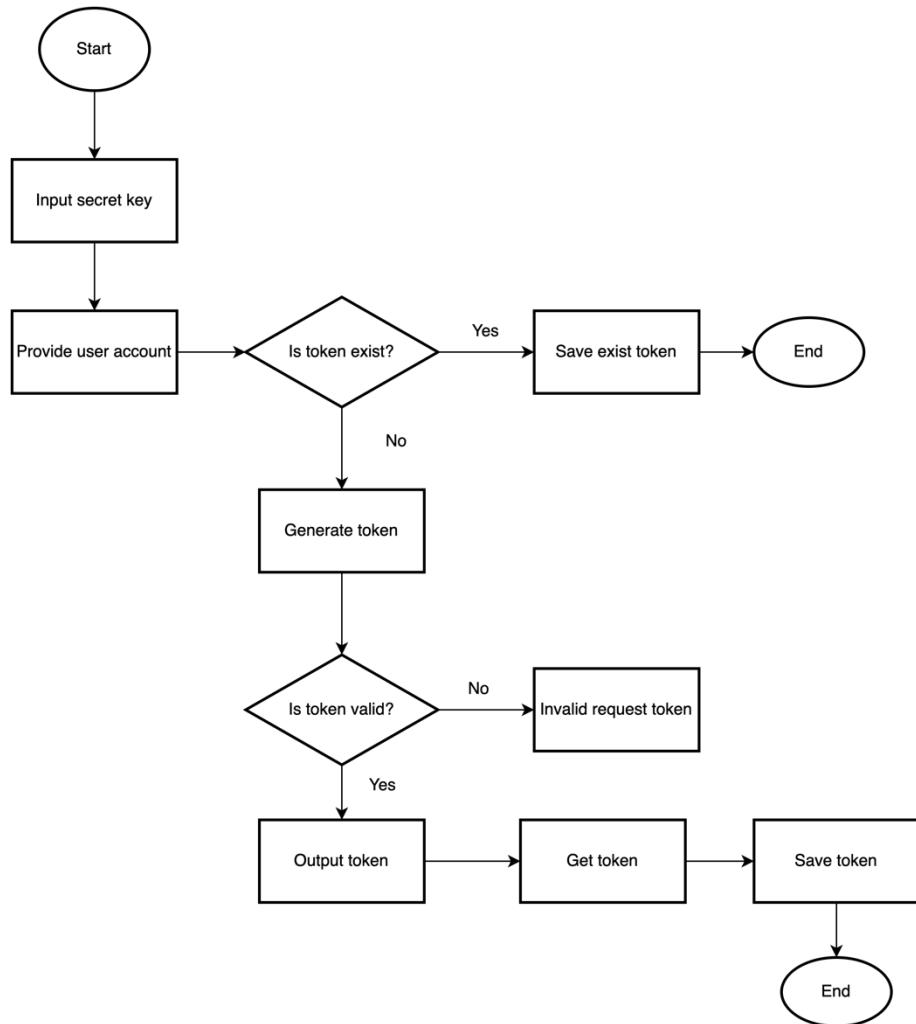


Figure 35: Generate token flowchart

```
const GenerateToken = async (req, res, next) => {
  try {
    const { userAccount, userType } = req.body;

    if (!userAccount || !userType) {
      return res.status(400).json({ error: 'User account and user type are required.' });
    }

    const userModel = getUserModel(userType);
    const user = await userModel.findOne({ userAccount });
    if (!user) {
      return res.status(404).json({ error: 'User account not found.' });
    }

    const secretKey = getSecretKey(userType);
    const existingToken = req.headers.authorization?.split(" ")[1];
    const validatedToken = existingToken ? validateAndReturnExistingToken(existingToken, secretKey) : null;

    if (validatedToken) {
      return res.json({ token: validatedToken });
    }
    // If existing token not valid or not provided, generate a new token
    const newToken = await generateNewToken(user, secretKey, userType);
    res.json({ token: newToken });
  } catch (error) {
    console.error('Error in generate token:', error);
    res.status(500).json({ error: 'Internal server error.' });
  }
};
```

Figure 36: Generate token

In Figure 35 and Figure 36 is shown about to begin with Generate token process flowchart and logical implement code as the following steps:

- First, start with input secret key of any user account implement with backend (NodeJS) projects choose a token library like **jsonwebtoken** for Node.js, which simplifies token generation and verification.
- Second, define a secret key for secure signing. then, create a user object with relevant data, like username and password. each of user get the different token as set roles of user account.
- Third, use the library to generate the token, include options like an expiration time, which determines how long the token is valid or invalid.
- Fourth, the token is used for authentication and authorization purposes. clients send this token in requests for verification. in case for condition validate token use for input secret key and get valid of user account if those account are acceptable to generate token.
- Fifth, after generate token, user can get token to access their own account but need to verify token first before begin authentication flow otherwise, sometime generate token is an error during generated for validate provide with incorrect user account.

4.3.2.Verify token flowchart

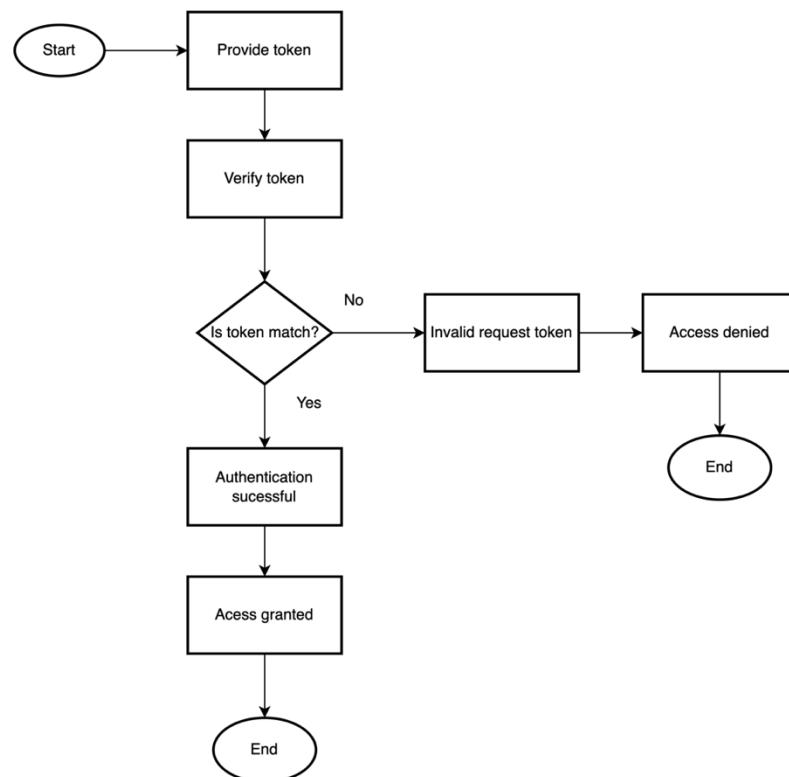


Figure 37: Flowchart of verify token

```

const Admin = require("../models/Admin.js");
const jwt = require('jsonwebtoken');

const adminAuth = async (req, res, next) => {
    try {
        const token = await req.headers.authorization.split(" ")[1];
        const decodedToken = await jwt.verify(token, process.env.ADMIN_SECRET_KEY);
        const user = await decodedToken;
        req.user = user;
        next();
    } catch (error) {
        return res.json({
            message: "Invalid Request!",
            error
        })
    }
}

module.exports = { adminAuth}

```

Figure 38: Implement verify token

In Figure 37 and Figure 38 shown about the process of verify token flowchart process are include:

- First, the process begins start with authentication flow by provide token.
- Second, user need to enter verification token, which is typically provide as part of the authentication process.
- Third, the backend of the system will check if the token entered by the user matches the expected token or generate for user at this moment.
- Fourth, if the entered token matched with expected token for condition “yes” it will continue to the next step, indicating an authentication successful then redirect to access granted before complete process.
- Fifth, if the entered token does not match with the expected token, it indicates an invalid request token, with the token not matching the authentication process considered a failure confirm the user’s account is unable to login on the system, it also displays access denied ending the process.

4.3.3. QR Code Scanner flow chart

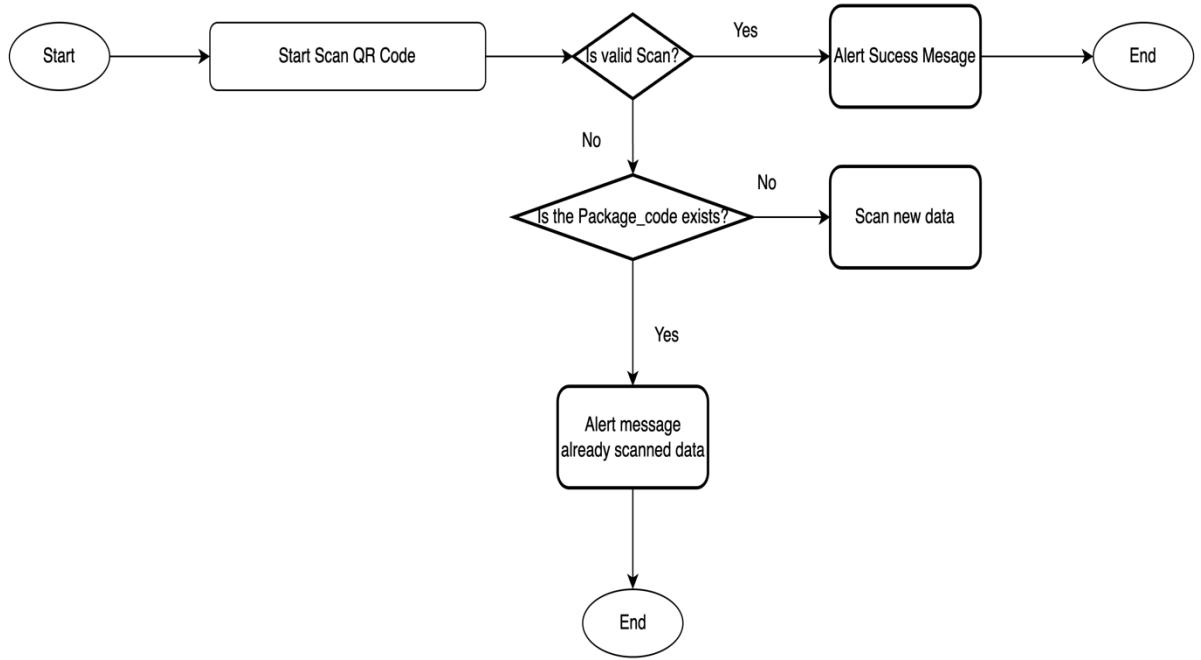


Figure 39: Flowchart of QRCode scanner

```

const Scanner = ({ onScan }) => {
  const [hasPermission, setHasPermission] = useState(null);
  const [scanned, setScanned] = useState(false);
  const scannerRef = useRef(null);

  useEffect(() => {
    (async () => {
      const { status } = await BarcodeScanner.requestPermissionsAsync();
      setHasPermission(status === 'granted');
    })();
  }, []);

  useEffect(() => {
    // Reset the scanned state when the component is unmounted
    return () => {
      setScanned(false);
    };
  }, []);

  const handleBarcodeScanned = ({ type, data }) => {
    setScanned(true);
    onScan(data); // Pass the scanned data to the parent component
  };

  return (
    <BarcodeScanner
      ref={scannerRef}
      onBarcodeScanned={scanned ? undefined : handleBarcodeScanned}
      style={{ flex: 1 }}
      barCodeTypes={[BarcodeScanner.Constants.BarCodeType.qr]}
      focusDepth={0} // Change the focus depth (you can experiment with different values)
      focusStyle={{
        borderColor: 'blue', // Change the border color when focused
        borderRadius: 16, // Change the border radius when focused
        borderWidth: 3,
      }}
    />
  );
}

You, 3 months ago • #002-create-qrcode-app
  
```

Figure 40: QRcode scanner implement code

In Figure 39 shown about the package was verify by using QR code:

- First, the process begins when the user initiates the QR code scanning.
- Second, the user scans a QRCode in mobile application.
- Third, the system mobile will check if the scan is valid or not. If condition Yes, navigate processed to the next step, meanwhile condition No, it requires user to scan new data ending process.
- Fourth, If the scan is valid, the system alerts the user that the scan was successful. the process ends after a successful scan alert.
- Fifth, the system checks if the package code from the QRCode already exists in the database. for Yes, if the package code already exists, the system alerts the user that the data has already been scanned, meanwhile No, prompt the user to scan new data.
- Last, after entering the code, they click the "Find Now" button to check whether the item has been delivered or is still pending delivery.

In Figure 40 shown about how to handle scan the QR Code using the third-party library implement logical code in backend server as the following steps:

- First, import the third-party barcode scanning library and its dependencies.
- Second, Begin the QR code scanning process and initialize scanner use by **BarcodeScanner** component is rendered and wait scanning input.
- Third, check if the QRCode has been scanned, for condition no continue waiting for QRCode scanning. Otherwise, if condition yes process to the next step for scan new QRCode data. Set a state variable scanned to true to indicate that scanning has occurred.
- Finally, handle scanned data by using **handleBarcodeScanned** function with the scanned data. next step, pass data to parent component by the **onScan** function is called, passing the scanned data to the parent component for further processing. use the **onScan** callback function to pass the scanned data to the parent component complete the scanning process.

CHAPTER V: CONCLUSION

In conclusion, during my 2-month and one-week internship as a web developer at the Institute of Technology of Cambodia (ITC), I had the privilege of contributing to the successful completion of the delivery service system project. From this project I have used new technologies and familiar with many useful tools for development including web Development and mobile applications.

5.1. Completed and Uncompleted Tasks

After finishing the internship program at the ITC, the achievement result of complete and uncomplete tasks, the table provides an overview result of the completed and uncomplete tasks foreach role (admin, branch owner, and user) in the Delivery Service System. I will brief the summary of the table below:

5.1.1. Admin Roles Table

In the Admin role table, I have completed 10 features without any uncompleted tasks. All the requirements have been fully developed. The table below presents a summary of all the tasks thatI completed for the admin role.

| Roles | Features | Status | Responsible by |
|-------|---------------------------------|--------|------------------------|
| Admin | Admin Dashboard | ✓ | ROTHA Dapravith |
| | Authentication | ✓ | ROTHA Dapravith |
| | Branch Owner Account Management | ✓ | SOK Pagnavath |
| | Change Password | ✓ | ROTHA Dapravith |
| | Change Personal Information | ✓ | SOK Pagnavath |
| | List of Admin | ✓ | ROTHA Dapravith |
| | Logout | ✓ | ROTHA Dapravith |
| | Monitoring all Branches | ✓ | ROTHA Dapravith |
| | Update Profile | ✓ | SOK Pagnavath |

Table 3:Admin role summary task

5.1.2. Branch Owner Role Table

There are many features of the Branch Owner role table. All requirements have been fulfilled completely. The table below summarizes all the tasks that I completed for the branch owner role. Hence, the summary task for the branch owner role in the table below:

| Roles | Features | Status | Responsible by |
|---------------|------------------------------------|--------|-----------------|
| Branch Owners | Login Autehtication | ✓ | ROTHA Dapravith |
| | Dashboard | ✓ | SOK Pagnavath |
| | Package management | ✓ | SOK Pagnavath |
| | Manage Delivered Package | ✓ | SOK Pagnavath |
| | View all Pending Package | ✓ | SOK Pagnavath |
| | View all Incoming Package | ✓ | SOK Pagnavath |
| | View Delivered Package | ✓ | SOK Pagnavath |
| | Set Source Destination | ✓ | SOK Pagnavath |
| | Generate Package Code into QR Code | ✓ | ROTHA Dapravith |
| | Scan QR Code | ✓ | ROTHA Dapravith |
| | Mark Delivered Package | ✓ | SOK Pagnavath |
| | Package History | ✓ | SOK Pagnavath |

Table 4: Branch-owner role summary task

5.1.3. Customer Role Table

The customer role summary table still lacks two features that have been finished. In general, all requirements for customers are met and fulfilled. Consequently, the task summarization is as follows:

| Roles | Features | Status | Responsible by |
|-----------|-------------------------------|--------|-----------------|
| Customers | Find Package Code | ✓ | ROTHA Dapravith |
| | View Details Package | ✓ | ROTHA Dapravith |
| | View Tracking Branch Location | ✓ | SOK Pagnavath |
| | Leave Feedback | ✓ | ROTHA Dapravith |
| | Advance Search Filter | ✗ | SOK Pagnavath |
| | Display Location | ✗ | SOK Pagnavath |

Table 5: Customer role summary task

5.1.4. Backend Task Summary Table

Table 6 shows a summary of the backend task fetch API that is used to make sure the web application works properly. It includes Modules for the API Products, which can retrieve pending, incoming, and delivered items. The summary table for the backend tasks is displayed below:

| Modules | Features | Status | Responsible by |
|--------------|---------------------------------|--------|------------------------|
| API Auth | Register Admin's Account | ✓ | ROTHA Dapravith |
| | Login Admin's Account | ✓ | ROTHA Dapravith |
| | Register Branch Owner's Account | ✓ | SOK Pagnavath |
| | Login Branch Owner's Account | ✓ | SOK Pagnavath |
| | Get all list of Admin | ✓ | ROTHA Dapravith |
| | Get all list of Branch Owner | ✓ | ROTHA Dapravith |
| | Update Profile User | ✓ | ROTHA Dapravith |
| | Change Password | ✓ | ROTHA Dapravith |
| | Encrypt and Decrypt Token | ✓ | ROTHA Dapravith |
| | CRUD user of Admin | ✓ | ROTHA Dapravith |
| API Package | CRUD user of Branch Owner | ✓ | SOK Pagnavath |
| | CRUD Package | ✓ | SOK Pagnavath |
| | Get all list of Package | ✓ | SOK Pagnavath |
| | Get Package by Images | ✓ | SOK Pagnavath |
| | CRUD Package history | ✓ | SOK Pagnavath |
| | Get all pending Package | ✗ | SOK Pagnavath |
| | Get all incoming Package | ✗ | SOK Pagnavath |
| | Get all delivered Package | ✗ | SOK Pagnavath |
| | Mark not Delivered | ✓ | SOK Pagnavath |
| | Mark Delivered | ✓ | SOK Pagnavath |
| API Feedback | Create customer's feedback | ✓ | ROTHA Dapravith |
| | Get all customer's feedback | ✓ | ROTHA Dapravith |

Table 6:Back-end summary tasks

5.2. Strong Point

After I work and develop on this project, I have found some strong points on this system website include:

- Comprehensible and attractive User Interface.
- Enable to View Data visualization and graph on Dashboard.
- Authentication and Validation process are work effectively.
- Select Filter option by any Destination.
- QR Code Scanner for Verify the Package Code when package have been delivered to destination branch location.
- Customers can track their deliveries in real-time, gaining transparency and trust in the delivery process.
- Enhanced Customer Experience by leave the feedback to the service system.

5.3. Weak Point

The system has developed with several strong points, however there are still some areas that need development and improvement:

- Manually create user's account one by one.
- Hard to input location longitudes and location latitudes for branch owner.
- Slow performance when have big data of user and items.
- Frequently update delivered process of stage.
- Security concerns with admin and branch owner data.

5.4. Difficulties

During my internship, I encountered several challenges, primarily stemming from my lack of prior work experience. I had to quickly adapt and learn about new technologies, as I had never been exposed to them before. Despite these hurdles, I believe this internship was a valuable learning opportunity. It allowed me to expand my knowledge, and importantly, I was able to complete the project on time.

5.5. Experiences

During my internship, which lasted for two months and one week, I had the opportunity to gain valuable experiences and improve my skills in various areas such as:

- Understanding the working process and work environment.
- Project Management and Time Management.
- Good Communication with teammates.
- Understand about Software Development Life Cycle and Scrum Master.
- Explored to learn new technology.
- Problem Solving and find the clue to solve the problem.
- Working environment.
- Self-learning, work, and research.

5.6. Perspectives

If I had available time to progress on this project, I want to fill out the lack of any features for improve this project. There are functionalities that I should do in this project:

- Update location tracking to store all branch owners.
- Improve system high performance and high security.
- Improve an interactive user interface and user experience.
- Update QRCode Scanner in mobile app more attractive version.
- Enhanced the system to run more smoothly.
- Complete with undone tasks.

5.7. Summary

The internship provided valuable insights into the real tasks of project work, including teamwork, project management, communication, and troubleshooting. The internship was guided by **Mr. Hok Tin**, an academic supervisor in the department of information and communication engineering. The students learned about new technology and tools for developing web and mobile applications, gaining familiarity with both hard and soft skills. The internship also taught them about the working process and environment, allowing them to apply their knowledge in their future careers. The internship emphasized the importance of teamwork, communication, and problem-solving in project work.

REFERENCES

- [1] *JavaScript Programming Language*. (n.d). Retrieved October 18, 2023, from <https://www.javascript.com/>
- [2] *Mern Stack: A Complete Guide*. (n.d). Retrieved October 18, 2023, from <https://www.mongodb.com/languages/mern-stack-tutorial>
- [3] *Introduction to HTML*. (n.d). Retrieved October 18, 2023, from https://www.w3schools.com/html/html_intro.asp
- [4] *CSS Introduction*. (n.d). Retrieved October 18, 2023, from https://www.w3schools.com/css/css_intro.asp
- [5] *JavaScript.com*. (n.d). Retrieved October 18, 2023, from <https://www.w3schools.com/js/default.asp>
- [6] *Sass*. (n.d). Retrieved October 18, 2023, from <https://sass-lang.com/>
- [7] *React JS*. (n.d). Retrieved October 18, 2023, from <https://legacy.reactjs.org/docs/getting-started.html>
- [8] *ReactNative*. (n.d). Retrieved October 18, 2023, from <https://reactnative.dev/docs/environment-setup>
- [9] *NodeJS*. (n.d). Retrieved October 18, 2023, from <https://nodejs.org/en>
- [10] *Express JS*. (n.d). Retrieved October 18, 2023, from <https://expressjs.com/>
- [11] *MongoDB*. (n.d). Retrieved October 18, 2023, from <https://learn.mongodb.com/learning-paths/introduction-to-mongodb>
- [12] *Docker*. (n.d). Retrieved October 18, 2023, from <https://www.docker.com/>
- [13] *GitLab*. (n.d). Retrieved October 18, 2023, from <https://about.gitlab.com/>
- [14] *SourceTree*. (n.d). Retrieved October 18, 2023, from <https://www.sourcetreeapp.com/>
- [15] *Visual Studio Code*. (n.d). Retrieved October 18, 2023, from <https://code.visualstudio.com/>
- [16] *StarUML*. (n.d). Retrieved October 18, 2023, from <https://staruml.io/>
- [17] *Draw.io* (n.d). Retrieved October 18, 2023, from <https://drawio-app.com/>
- [18] *Postman*. (n.d). Retrieved October 18, 2023, from <https://www.postman.com/>
- [19] *Trello*. (n.d). Retrieved October 18, 2023, from <https://trello.com>
- [20] *Figma* (n.d). Retrieved October 18, 2023, from [https://www.figma.com/](https://www.figma.com)

APPENDIX

APPENDIX A: Screenshot of Delivery Service System

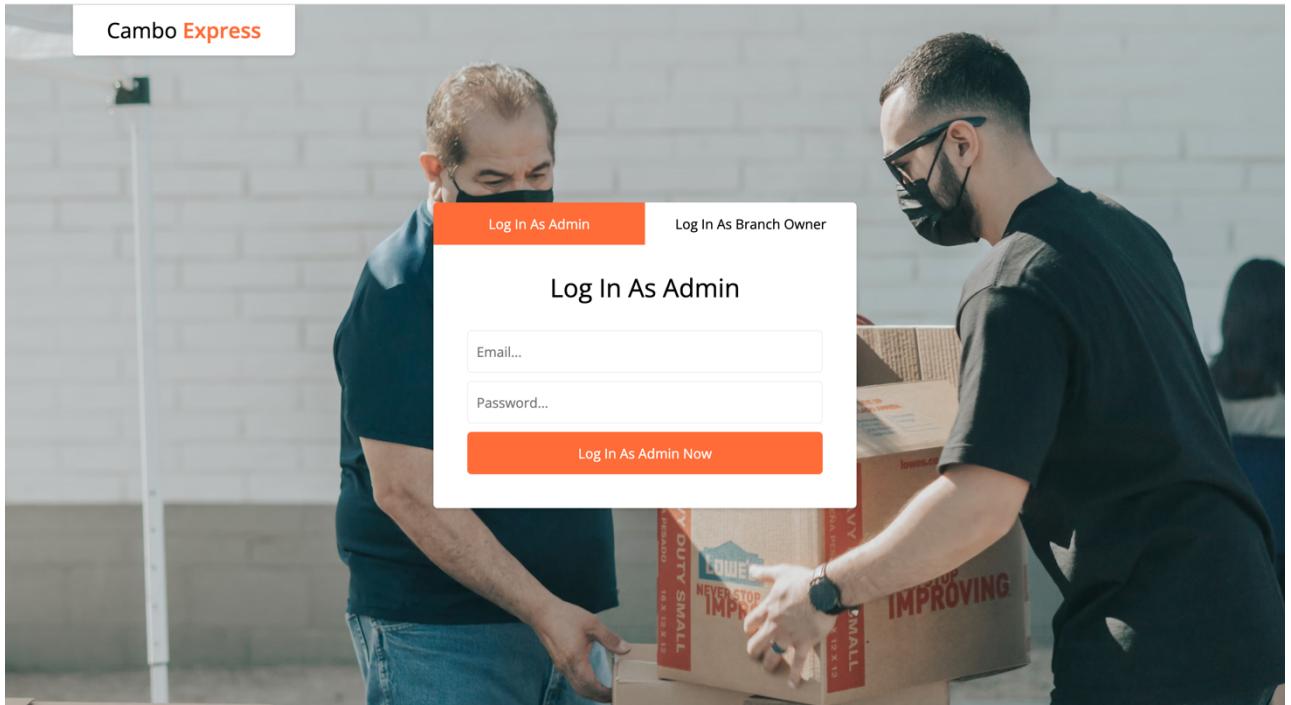


Figure 41: Admin's login page

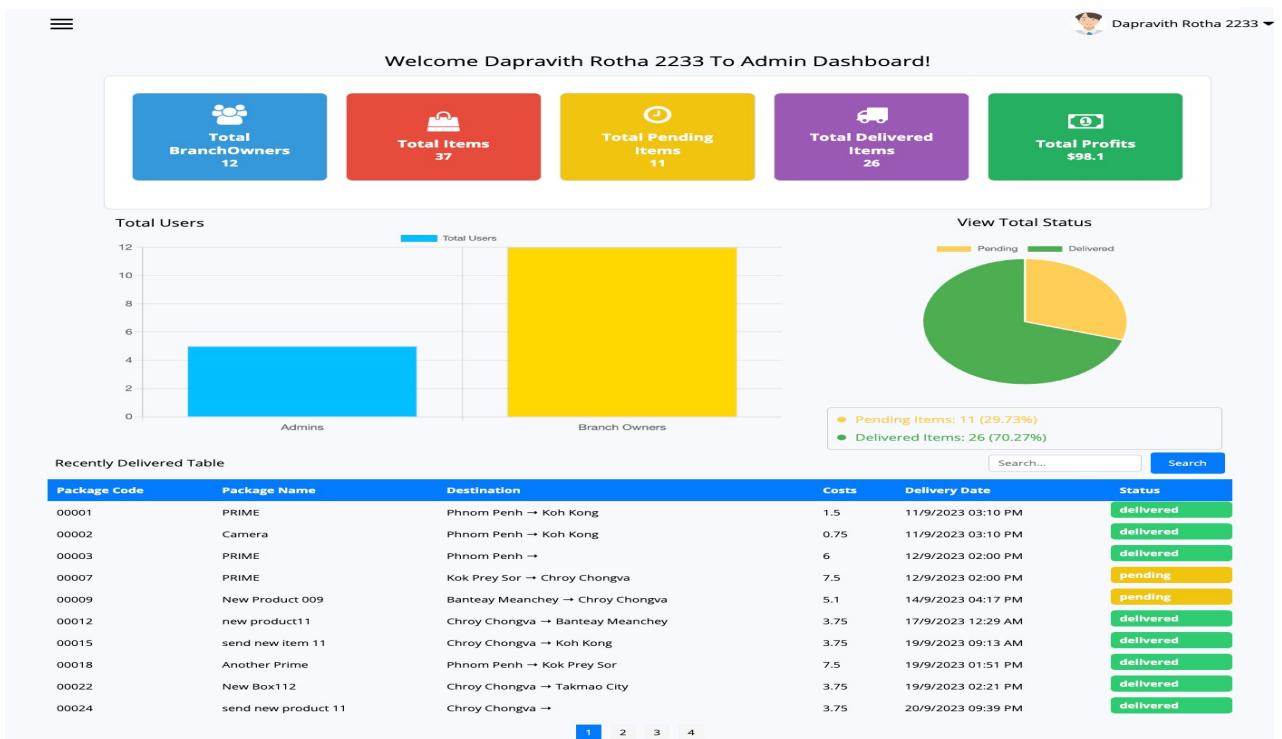


Figure 42: Admin dashboard

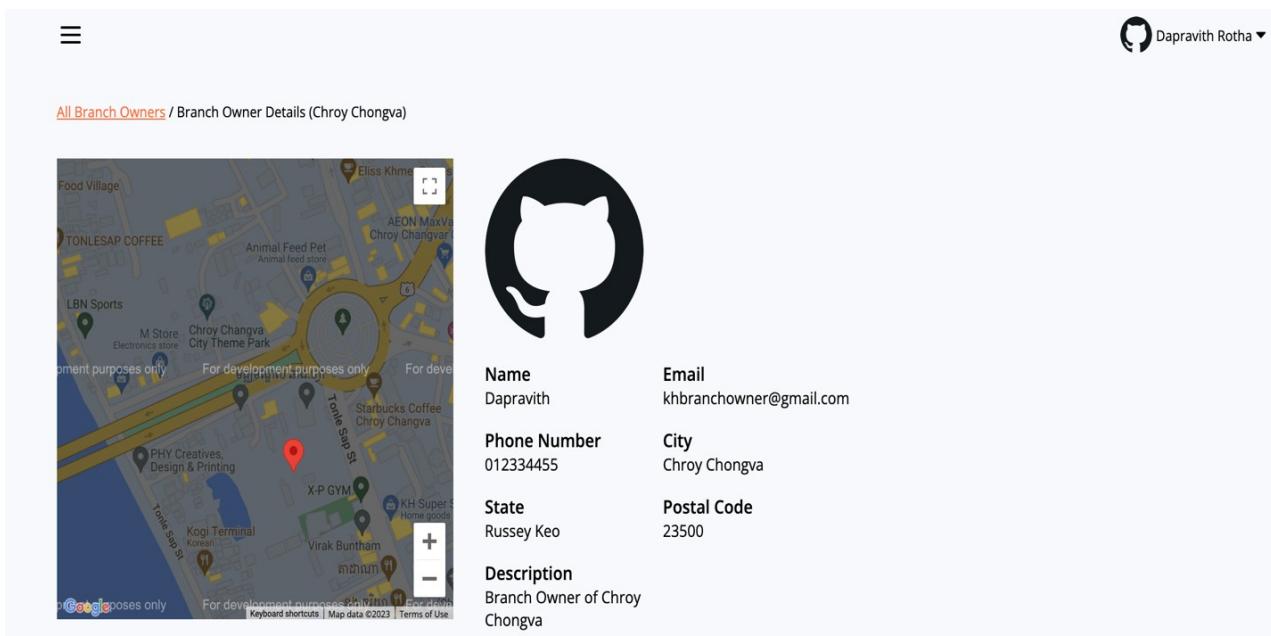


Figure 43: Details information of branch owner account

| Name | Contact Person | Location |
|--------------------|--------------------|------------|
| Chroy Chongva | Dapaprith | zalando |
| Aeon 2 Sen Sok | Chan Thida | Open here |
| Phnom Penh | Pagnavath Sok | ökologisch |
| Siem Reap II | Pagnavath Dev | free to be |
| Kompong Som | Channy Sereyvuthy | PLAY ME |
| Koh Kong | Pagnavath Koh Kong | Open here |
| Kok Prey Sor | John Doe | Open here |
| Banteay Meanche... | Olica Theary | Open here |
| Takmao City | Sok Somnang | Open here |
| Kompong Speu | Nit | Open here |
| Norea Island | Dara Rith | Open here |
| Koh Kaev | AlgoExpert | Open here |

Figure 44: List of branch owner account

≡

Dapravith Rotha 2233 ▾

Create Branch Account

| | |
|---|--|
| Full Name John Doe | Gender <input checked="" type="radio"/> Male <input type="radio"/> Female |
| Phone Number 089 673 945 | Email Address johndoe@gmail.com Please fill out this field. |
| Password Password | Repeat Password |
| Address Address | City Phnom Penh |
| State Toul Kork | Postal Code 12000 |
| Location Latitude 10.2390482 | Location Longitude 12.1239873 |
| Description Describe about branch owner... | Profile Image <input type="button" value="Browse"/> |

Figure 45: Create branch owner account

≡

Dapravith Rotha 2233 ▾

Create Admin Account

| | |
|---|--|
| Full Name John Doe | Gender <input checked="" type="radio"/> Male <input type="radio"/> Female |
| Phone Number 089 673 945 | Email Address johndoe@gmail.com |
| Password Password | Repeat Password |
| About Describe about branch owner... | Profile Image <input type="button" value="Browse"/> |

Figure 46: Create admin account

The screenshot shows a list of five admin accounts. The second account, 'dapravithrotha@gmail.com', is highlighted with a green background and labeled 'You'. Each account entry includes a small profile icon, the email address, and a red trash can icon.

| Rank | Email Address |
|------|-------------------------------------|
| 1 | pagnavathsok@gmail.com |
| 2 | dapravithrotha@gmail.com You |
| 3 | vithdev@gmail.com |
| 4 | admin123@gmail.com |
| 5 | nit@gmail.com |

Figure 47: List of admin account

The screenshot displays the profile of 'Mr. Daprvith Rotha'. It features a large GitHub logo placeholder, a blue 'Edit Profile' button, and a 'Change Password' button. To the right, there are four sections: Name (Mr. Daprvith Rotha), Phone Number (098779398), Email (dapravithrotha@gmail.com), and About (I'm the Admin's Owner of the delivery service system. New admin).

Figure 48: Admin's profile

The screenshot shows the same profile page after changes were made. The 'Name' field now reads 'Mr. Daprvith Rotha 2233'. A green banner at the top right says 'All Changes Saved!'. The other fields remain the same: Phone Number (098779398), Email (dapravithrotha@gmail.com), and About (I'm the Admin's Owner of the delivery service system. New admin). The 'Edit Profile' and 'Change Password' buttons are also present.

Figure 49: Edit admin information

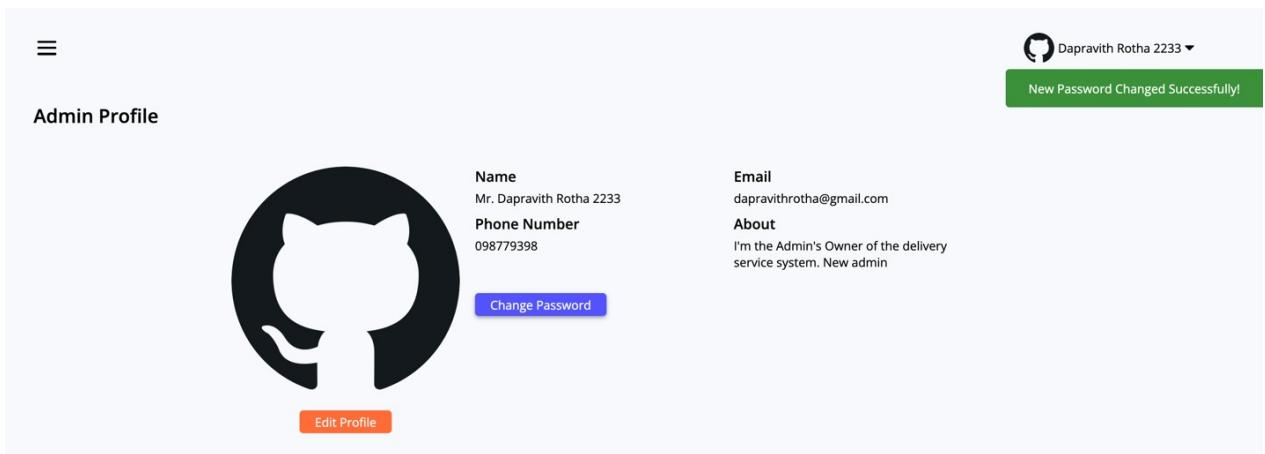


Figure 50: Change password admin

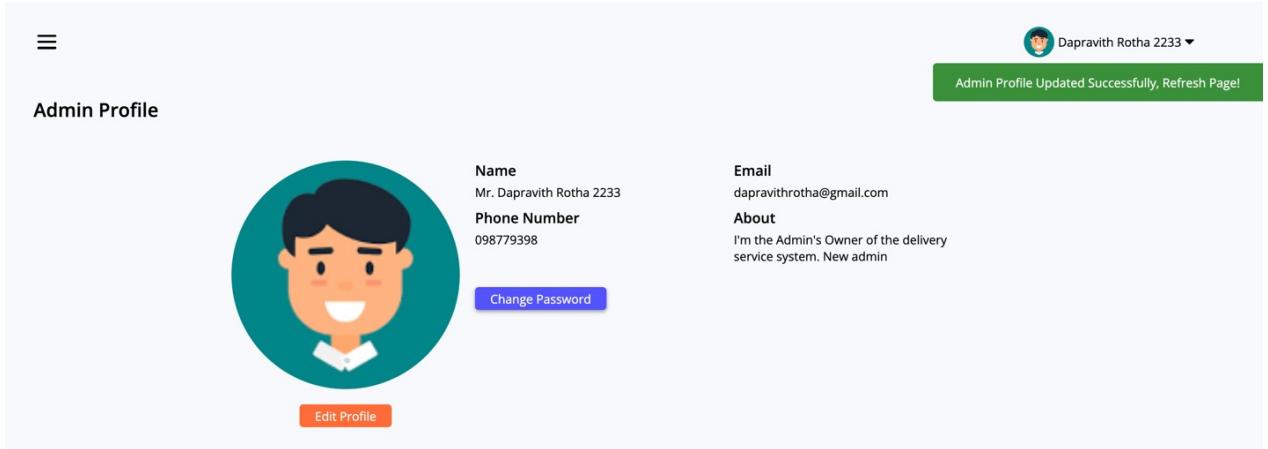


Figure 51: Change admin's profile

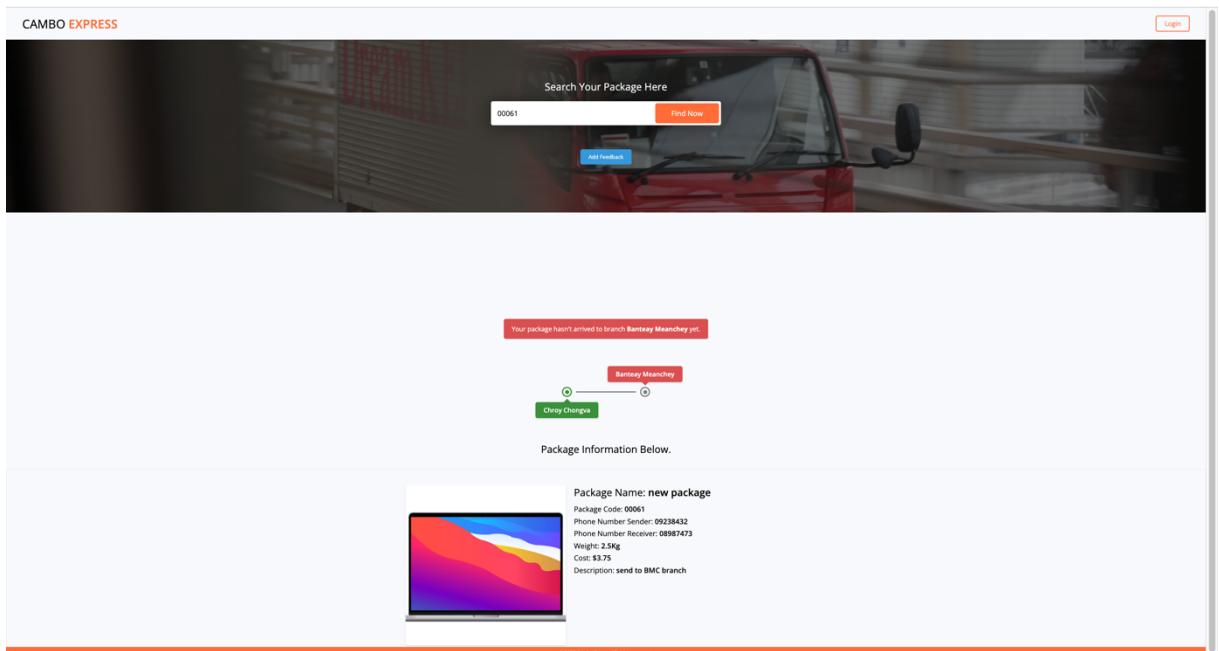


Figure 52: Homepage

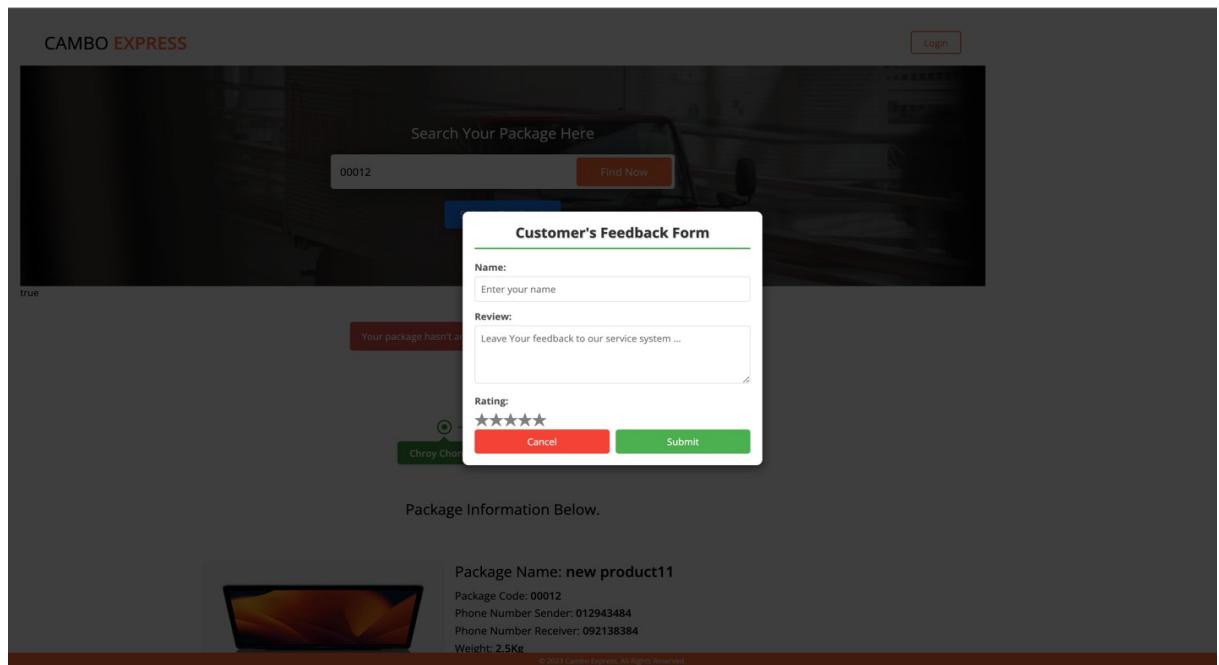


Figure 53: Add new feedbacks

APPENDIX B: QR Code Scanner App

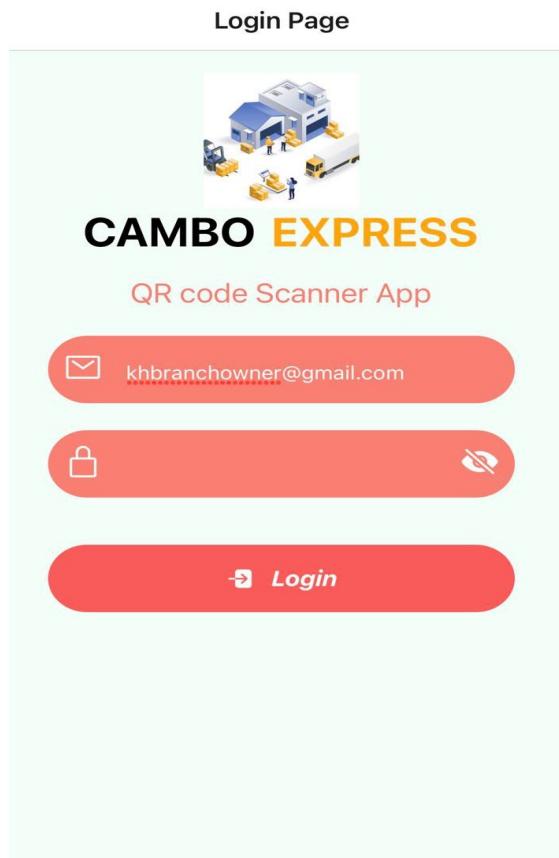


Figure 54: Login screen

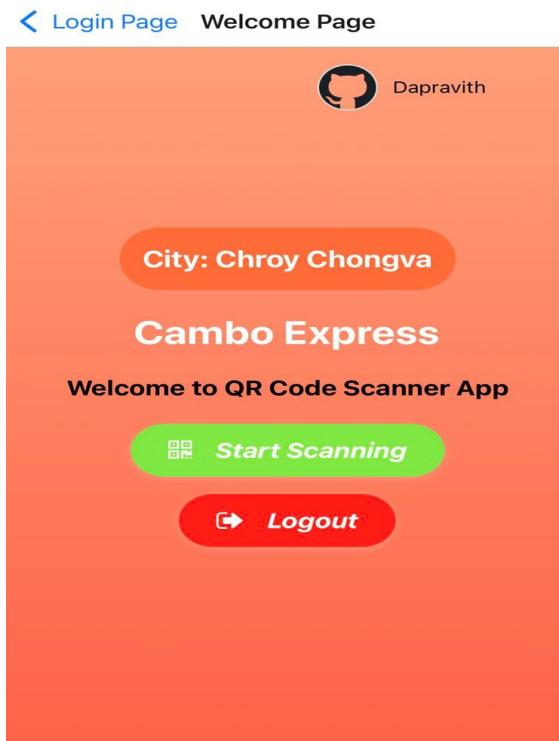
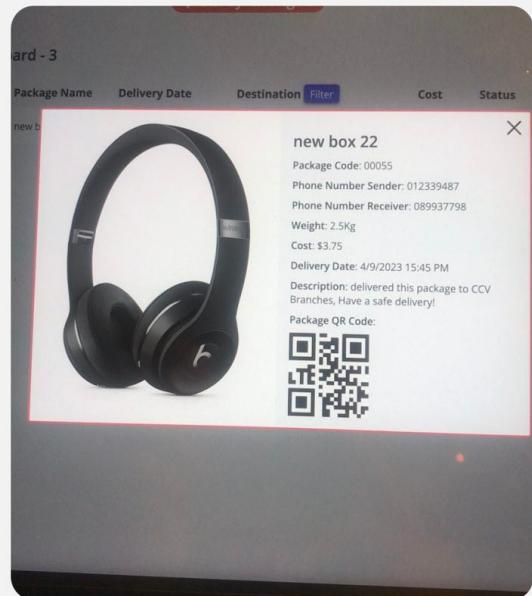


Figure 55: Welcome screen

< QR Scanner



Item has arrived to branch Chroy Chongva:

Item Code: 00055

Figure 56: Scanner screen

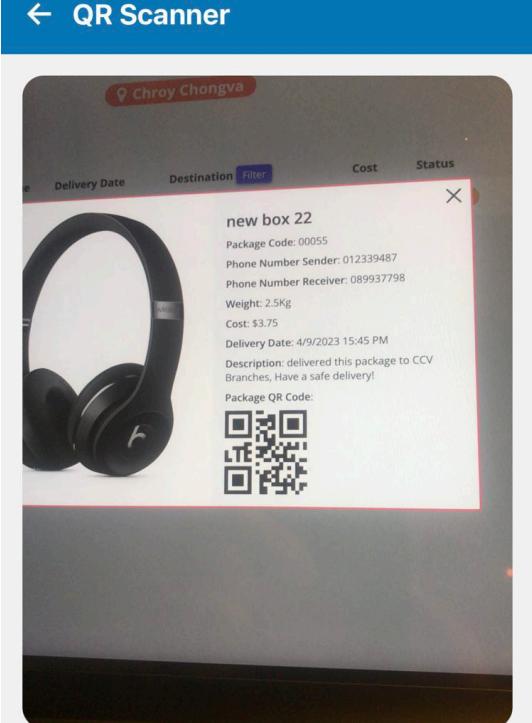


Figure 57: Result of screen

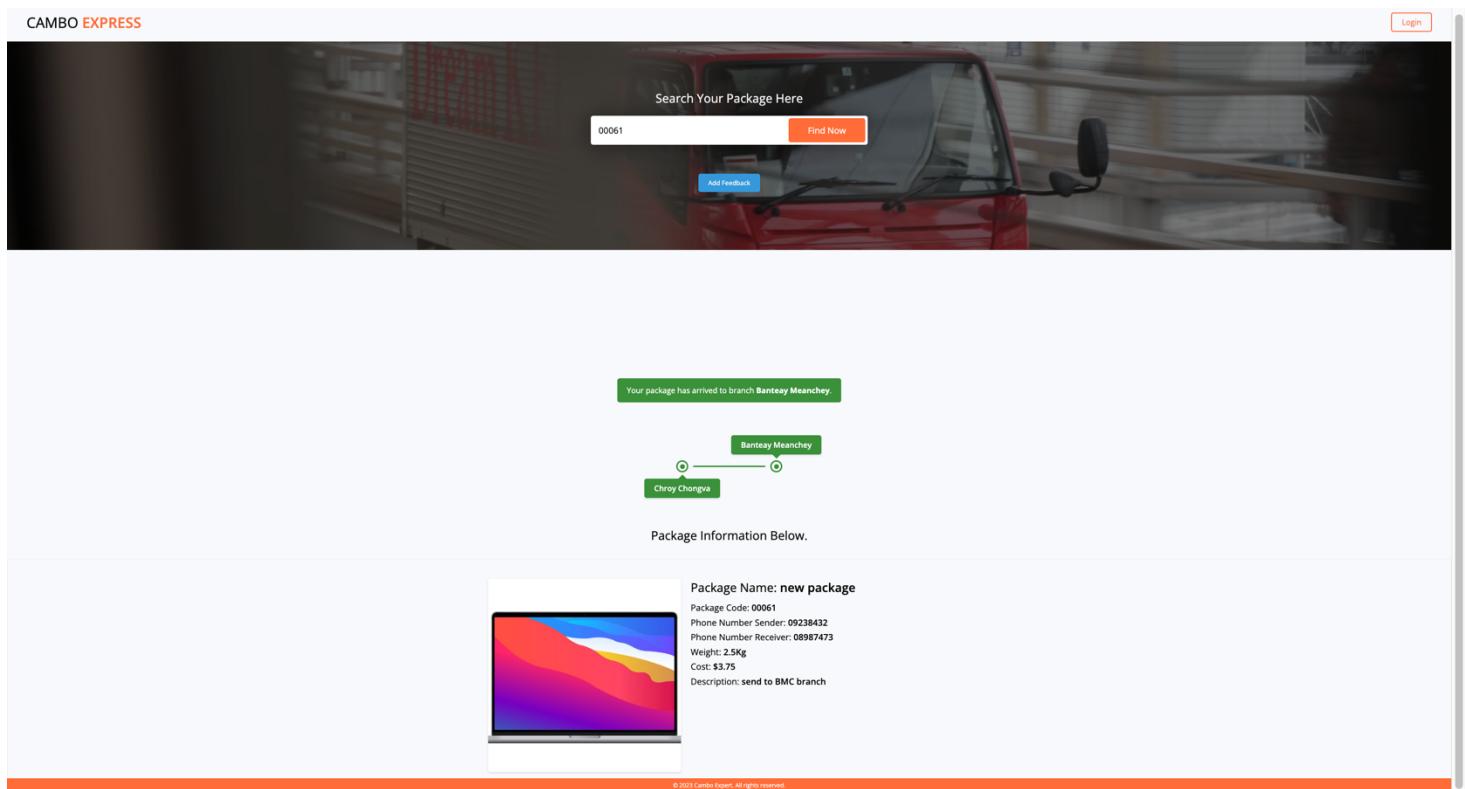


Figure 58: View delivered package