



MINISTRY OF EDUCATION AND TRAINING

TRƯỜNG ĐẠI HỌC FPT

FPT UNIVERSITY

Automatic Parking Management and Parking Fee Collection System

Capstone Project Document

GFA22SE51	
Group Members	Khúc Ngọc Thái (SE151055) Tăng Chí Cường (SE150676) Hồ Hữu Phát (SE151312) Thuỷ Võ Anh Hoàng (SE150780) Nguyễn Quang Dũng (SE150721)
Supervisors	Kiều Trọng Khánh Đặng Ngọc Minh Đức
Capstone Project code	FA22SE50

- Ho Chi Minh, 12/2022 -

This page is intentionally left blank

Table of Contents

Acknowledgements	8
Acronyms & Definitions	9
I. Project Introduction	10
1. Overview	10
1.1 Project Information	10
1.2 Project Team	10
2. Product Background	10
3. Existing Systems	11
3.1 Parking System Of Pearl Plaza	11
3.2 Aeon Tân Phú Parking System	11
3.3 Conclusion	11
4. Business Opportunity	11
5. Software Product Vision	11
6. Project Scope & Limitations	11
6.1 Major Features	11
6.2 Limitations & Exclusions	12
II. Project Management Plan	13
1. Overview	13
1.1 Scope & Estimation	13
1.2 Project Objectives	14
1.3 Project Risks	14
2. Management Approach	15
2.1 Project Process	15
2.2 Quality Management	15
2.3 Training Plan	16
3. Project Deliverables	16
4. Responsibility Assignments	17
5. Project Communications	17
6. Configuration Management	18
6.1 Document Management	18

6.2 Source Code Management	18
6.3 Tools & Infrastructures	18
III. Software Requirements Specification	19
1. Product Overview	19
2. User Requirements	20
2.1 Actors	20
2.2 Use Cases	20
2.2.1 Diagram	20
2.2.2 Descriptions	21
3. Functional Requirements	23
3.1 System Functional Overview	23
3.1.1 Screen flow	23
3.1.2 Screen details	25
3.1.3 Screen Authorization	27
3.1.4 Non-Screen Functions	28
3.2 Booking Progress	29
3.2.1 Booking Home	29
3.2.2 Create booking	30
3.2.2 Confirmation page	32
3.2.3 QR Scan	33
3.3 History	34
3.3.1 Booking history	34
3.3.2 Booking ticket	35
3.3.3 Parking history	36
3.3.4 Parking ticket	37
3.3.5 Done history	38
3.3.6 Done ticket	39
3.3.7 Canceled history	40
3.3.8 Canceled ticket	41
3.4 Profile	42
3.4.1 Profile information	42
3.4.2 Top-up	43

3.4.3 Transaction history	44
3.4.3 Feedback	45
3.4.4 Change password	46
3.5 Customer Authorization	48
3.5.1 Sign In	48
3.5.2 Sign Up	50
3.5.3 Forgot password	51
3.5.4 OTP Verification	52
3.6 Authorization	53
3.6.1 Login	53
3.6.2 Reset password	54
3.7 Show admin reports	55
3.8 Get car park list	56
3.9 Get ticket list	57
3.10 Get transaction history	58
3.11 Get employee list	59
3.12 Show owner reports	60
3.13 Show staff's reports	60
3.14 Manage car park	61
4. Non-Functional Requirements	63
4.1 External Interfaces	63
4.2 Quality Attributes	63
4.2.1 Usability	63
4.2.2 Reliability	63
4.2.4 Compatibility	63
4.2.5 Security	63
5. Requirement Appendix	63
5.1 Business Rules	63
5.2 Application Messages List	65
IV. Software Design Description	66
1. System Design	66
1.1 System Architecture	66

1.2 Package Diagram	67
1.2.1 Back-end	67
1.2.2 Front-end	68
1.2.3 Mobile	69
1.2.4 IoT	69
2. Database Design	71
3. Detailed Design	75
3.1 Check-in	75
3.1.1 Class Diagram	75
3.1.2 Sequence Diagram	76
3.2 Check-out	77
3.2.1 Class Diagram	77
3.2.2 Sequence Diagram	78
3.3 Booking	79
3.3.1 Class Diagram	79
3.3.2 Sequence Diagram	80
V. Software Testing Documentation	81
1. Scope of Testing	81
2. Test Strategy	81
2.1 Testing Types & Levels	81
2.2 Supporting Tools	81
3. Test Plan	81
3.1 Human Resources	81
3.2 Test Environment	81
3.3 Test Milestones	81
4. Test Cases & Test Reports	81
VI. Release Package & User Guides	82
1. Deliverable Package	82
2. Installation Guides	82
2.1 System Requirements	82
2.2 Installation Instruction	83
2.2.1 Install camera system	83

2.2.2 Install embedded system	85
2.2.3 Install python packages	85
3. User Manual	87
3.1 Overview	87
3.2 Booking	87
3.3 Check-in	89
3.4 Check-out	90
3.5 Manual check-out	91
3.5 Update car park information	92
3.6 Create a new employee's account	94

Acknowledgements

We would like to express our deepest appreciation to our mentors, Mr. Kiều Trọng Khánh & Mr. Đặng Ngọc Minh Đức, for putting so much time and effort into this project.

Our gratitude to Mr. Lâm Hữu Khanh Phương, Mr. Lê Vũ Trường, Mr. Ngô Đặng Hà An and Mr. Nguyễn Thé Hoàng. Your support and suggestions are greatly appreciated.

Special thanks to Mr. Nguyễn Ngọc Quang, Mr. Nguyễn Tuấn Anh and Mr. Lương Hoàng Nhân for helping us with the project as our demonstrators, as well as providing the necessary tools and location for this project.

Acronyms & Definitions

Acronym	Definitions
Anaconda	Anaconda is a distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, etc.), that aims to simplify package management and deployment.
API	Application Program Interface
APMS	Automatic Parking Management and Parking Fee Collection System
AWS	Amazon Web Services: A third-party service that is used to deploy front-end and back-end applications
BR	Business Rule: Rules and constraints to be implemented into the system
CUDA	CUDA programming was designed for computing with NVIDIA's graphics processing units (GPUs). CUDA enables developers to reduce the time it takes to perform compute-intensive tasks, by allowing workloads to run on GPUs and be distributed across parallelized GPUs.
ERD	Entity Relationship Diagram: describes entities existing in the system and their relationships with each other
EX	Exclusion: Features that will not be included in the projects
FE	Features: Major features of the project
Firebase	A Backend-as-a-Service (BaaS). It provides developers with a variety of tools and services to help them develop apps. It is built on Google's infrastructure.
IoT	Internet of Things: is the concept of connecting any device to the Internet and to other connected devices, allow hardware to interact real time with the back-end using cloud
LI	Limitations of the project
OTP	One Time Password
QR	Quick Response: a type of barcode, contains data that can be scanned to retrieve
SHA256	An algorithm for hashing

I. Project Introduction

1. Overview

1.1 Project Information

- Project name: Automatic Parking Management and Parking Fee Collection System
- Project code: APMS
- Group name: GFA22SE51
- Software type: Web, Mobile App

1.2 Project Team

Full Name	Role	Email	Mobile
Kieu Trong Khanh	Mentor	khanhkt@fe.edu.vn	0908133304
Dang Ngoc Minh Duc	Mentor	ducdnm2@fpt.edu.vn	0989699299
Khuc Ngoc Thai	Leader	thaiknse151055@fpt.edu.vn	0949936732
Nguyen Quang Dung	Member	dungnqse150721@fpt.edu.vn	0377162315
Ho Huu Phat	Member	phathhse151312@fpt.edu.vn	0932781745
Thuy Vo Anh Hoang	Member	hoangtvase150780@fpt.edu.vn	0914108001
Tang Chi Cuong	Member	cuongtcse150676@fpt.edu.vn	0908436393

2. Product Background

Nowadays, as people's overall life quality has increased drastically compared to the past, more and more have chosen cars as their primary transportation. In many car parks, the process of parking a car requires is mostly done manually, which is extremely time-consuming. This process is also wasting so much manpower and financial resources because many functions must be completed by hand. Moreover, since most car parks in Vietnam are inadequate to support the high volume of vehicles that are using the service, drivers are tempted by the idea of booking a lot in advance to save up a huge amount of time finding a vacancy in the car park instead of having to look around to search for an available car park.

3. Existing Systems

3.1 Parking System Of Pearl Plaza

The parking system of Pearl Plaza provides visitors with the ability to take and scan RFID cards automatically, so it is not required to hire security staff at the entrance. However, visitors still have to pay the fee directly at the exit. Visitors can not book the car park in advance before coming. These are the reasons why traffic jams still occasionally happen.

3.2 Aeon Tân Phú Parking System

Aeon Tan Phu has an automated RFID provider at the parking gate, allowing visitors to get in more conveniently. At the exit, you can also pay the fee by using an online wallet. This method can save a lot of traffic time. However, many visitors complain that the car park is always out of service when they show up.

3.3 Conclusion

RFID-based parking systems are no longer an unfamiliar concept. Multinational buildings, malls, and other public areas have been applying these systems thoroughly to support their management. An automatic parking system is what we truly need, especially after social distancing in COVID times. However, the older systems (such as Pearl Plaza, Aeon Tan Phu, etc.) can not afford the best requirements. It's essential to have a system that allows operating contactless. The combination of RFID-based system, QR, and image-processing technologies allows APMS to help customers have a better way to manage their car parking system.

4. Business Opportunity

Many employees have requested a system that would automate the process of parking cars and allow them to see whether the car park still has available slots or not. Such a system like this would save many customers' time and improve overall satisfaction. It will also reduce the resources needed for running and maintaining the car park so that the company can direct those resources to other sections which could make good use of them. Also, this system should allow drivers to book a lot before they come to ensure that at the time they come, a parking space will always be available.

5. Software Product Vision

An automatic payout system for car park owners will be developed within one month from the first release. A plan for the addition of support for other types of vehicles such as motorcycles, bicycles, and other countries' license plates has been considered, which should be implemented 3 months after the initial release. Extending features for human resource management are also being considered, given the fact that car park owners want to have more control over their staff.

6. Project Scope & Limitations

6.1 Major Features

FE-01: As a customer, I want to manage my transactions

FE-02: As a customer, I want to book a parking slot

FE-03: As a system administrator, I want to manage accounts that exist on the system

FE-04: As a system administrator, I want to manage car parks

FE-05: As a staff, I want to manage parking tickets

FE-06: As a staff, I want to configure car park settings

FE-07: As an owner, I want to view my car park statistics

6.2 Limitations & Exclusions

- LI-01: The system only supports the registered users
 - LI-02: The current system currently only for cars and works with Vietnamese citizen license plate
 - LI-03: Only works with clear, visible number plates (not fully covered, heavily modified, etc.)
 - LI-04: The current system demands electricity and an Internet connection in order to be fully functional
-
- EX-01: The current system does not support the individual parking slot management
 - EX-02: The current system does not support human resource management
 - EX-03: The current system does not support vehicle management, nor fake/fraud license plate detection
 - EX-04: The current system does not support data migration, nor integrating with other systems

II. Project Management Plan

1. Overview

1.1 Scope & Estimation

#	WBS Item	Complexity	Est. Effort (man-days)
1	<i>Car check-in / check-out</i>		112
1.1	Recognize the number plate of the car	Complex	14
1.2	Prepare dataset	Medium	7
1.3	Preprocess dataset	Complex	14
1.4	Notify user after checking in or checking out	Medium	7
1.5	Calculate fees	Medium	7
1.6	Movement detection	Complex	14
1.7	Generate QR code	Medium	7
1.8	Automate process	Complex	14
1.9	License plate recognition	Complex	14
1.10	Car recognition	Complex	14
2	<i>Booking</i>		33
2.1	Get number of available car park	Simple	4
2.2	Get the list of car park	Simple	4
2.3	Get car park detail	Simple	4
2.4	Create a booking	Medium	7
2.5	Cancel a booking	Medium	7
2.6	Get booking detail	Medium	7
3	<i>Manage account</i>		71
3.1	Register an account	Medium	7
3.2	Login	Medium	7
3.3	Logout	Medium	7
3.4	Get personal information	Simple	4
3.5	Update password	Simple	4
3.6	Top-up the money	Complex	14
3.7	Get history of tickets	Complex	14

3.8	Get ticket detail	Medium	7
3.9	OTP verification	Medium	7
6	<i>Car park</i>		46
6.1	View car park information	Medium	7
6.2	Update price table	Medium	7
6.4	Update car park's information	Medium	7
6.5	Get parking history of specific license plate	Medium	7
6.6	Create new car park	Simple	4
6.7	Make statistics	Complex	14
7	<i>Install Hardware</i>		61
7.1	Config ESP32	Medium	7
7.2	Connect system to firebase	Medium	7
7.3	Install servo	Medium	7
7.4	Install camera and connect to image-processing module	Complex	20
7.5	Setup in real environment	Complex	20
8	<i>Feedback</i>		15
8.1	Create feedback	Simple	4
8.2	Get feedback list	Medium	7
8.3	Get feedback detail	Simple	4

Total Estimated Effort (man-days) **338**

1.2 Project Objectives

#	Testing Stage	Test Coverage	No. of Defects	% of Defect	Notes
1	System Test	>= 80%	8	4%	Mostly on license plate reading

1.3 Project Risks

#	Risk Description	Impact	Possibility	Response Plans
1	The team don't have enough budget	Medium	10	Get a loan at the bank or find a sponsor
2	A member is sick so the tasks belong to that	High	10	Assign other members help with the tasks until the member return

	member cannot be done			
3	A member is out	High	5	Assign members to the task belong to drop out member and work overtime
4	Deadline for the task is miscalculated	Medium	25	Plan and reschedule tasks to still meet time needed

2. Management Approach

2.1 Project Process

The project process will follow the Scrum framework of the Agile methodology.

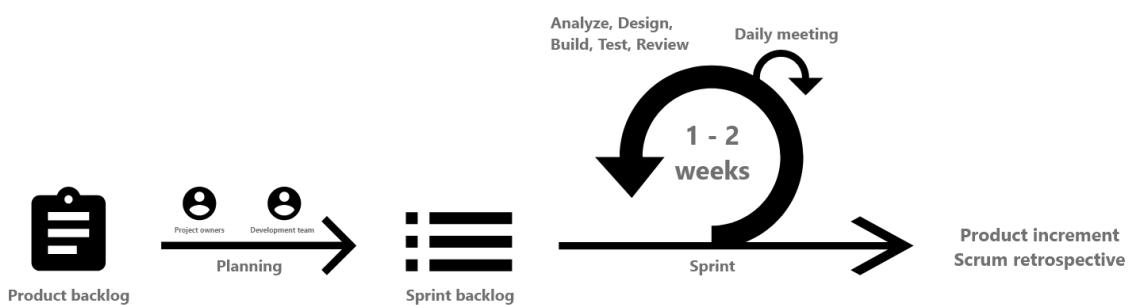


Figure 1. Scrum process diagram (Ref: scrum.org)

Why we picked the Scrum framework: there are many new features that the development team is not familiar with, such as integrating number plates/cars recognizing modules, research on other car parking systems and processes, etc. If we use the waterfall method, it would be very difficult to detect errors until later in the development process, which would be costly and time-consuming to fix. Scrum framework allows us to develop the product incrementally, and divide the project into multiple sprints so that we can review the features more frequently, also we can find errors earlier in the process, making those less tricky to deal with.

2.2 Quality Management

- Daily meeting: a small meeting (approximately 15 mins) will be hosted every working day to resolve any problem each team member is having.
- System testing: every core function should be tested, with multiple edge cases included. 80% test coverage or higher is preferred.
- Reviewing: the development team will have a review session every week to ensure that everything is going in the right direction. For small tasks, each team member will review each other and give suggestions. In case of a big milestone, mentors will be involved.

2.3 Training Plan

Training Area	Participants	When, Duration	Waiver Criteria
.NET, Repository Pattern, Unit of Works Pattern	Tang Chi Cuong, Khuc Ngoc Thai	12/9/2022 - 15/9/2022	Mandatory
JIRA, Bitbucket	All members within the development team	12/9/2022 - 13/9/2022	Mandatory

3. Project Deliverables

Sprint #	Deliverable	Due Date	Notes
1	Getting requirements	10/09/2022	Project Management Plan, Project Introduction Document
2	Diagrams	17/09/2022	Context Diagram, ERD Diagram, Use Case Diagram
	Screen flows	17/09/2022	For Admin, Staff and User
	Prototypes	17/09/2022	Web and Mobile
3	Simple Functions for User, Admin, Staff	24/09/2022	Login, Logout, View and Update Profile, Add Vehicle, Update Vehicle
4	Plate Recognition Module	1/10/2022	
	Top - up Money on Mobile	1/10/2022	
	Manage Staff for Admin	1/10/2022	Create, Update, View, Delete Staff
	Manage User for Staff	1/10/2022	View, Delete User
5	Book parking slot on mobile	8/10/2022	
	Pay the parking fee	8/10/2022	
	Manage Booking	8/10/2022	
	Manage Parking Slot	8/10/2022	
6	First Build	15/10/2022	Have main features but may not have authentication, authorization and validation
7	Second Build	5/11/2022	Complete all functions and Fix Bug
8			
9			
10	Third Build	19/11/2022	Fix Bug and Improve
11			
12	Final Product	26/11/2022	

13	Finalization	3/12/2022	Documents, User guides
14	Practice for Presentation	10/12/2022	Prepare Presentation Slides, Rehearsal, Final Product
15	Final Presentation		Complete Capstone Project

4. Responsibility Assignments

Responsibility	Khuc Ngoc Thai	Ho Huu Phat	Tang Chi Cuong	Thuy Vo Anh Hoang	Nguyen Quang Dung
Project Planning & Tracking	D	R	R	R	R
Prepare Report 1 (Project Introduction)	R	R	D	D	D
Prepare Report 2 (Project Management Plan)	D	D	R	R	R
Prepare Report 3 (Software Requirement Specification)	R	D	D	D	D
Prepare Report 4 (Software Design Document)	R	D	D	D	D
Prepare Report 5 (Software Test Document)	R	D	D	D	D
Prepare Report 6 (Software User Guides)	R	D	D	D	D
Backend related functions	D	R	D	R	R
Frontend related functions	R	R	R	D	D
Mobile related functions	S	D	S	R	R
IoT related functions	R	R	R	D	S
Image Processing related functions	S	D	R	R	R

5. Project Communications

Communication Item	Who/ Target	Purpose	When, Frequency	Type, Tool, Method(s)
Sprint planning	All team members	Confirm tasks for the next sprint	8:30 a.m every Sunday	Google Meet, Jira
Daily Scrum	All team member	Plan for the day, resolve issues	8 a.m every working day	Google Meet, Jira
Sprint Retrospective	All team member	Discuss improvements for next sprint	8 a.m every Sunday	Google Meet, Jira

6. Configuration Management

6.1 Document Management

- Google Drive:
 - + [Link to document](#) (permission required)
 - + Description: All documents and all resources needed (diagrams, images, videos, etc.) will be kept in Google Drive. Each type of document will have its folder, and the format will either be .doc, .docx, or .pdf, to keep consistency and ensure that the display of every document will be the same among different devices.
- Confluence:
 - + [Link to Confluence](#) (permission required)
 - + Description: All links leading to Google Drive will be kept here. A copy of meeting notes will also be saved here for easier access.
- Google Meet:
 - + [Link to Google Meet](#) (permission required)
 - + Description: All team meetings will be held here.

6.2 Source Code Management

- Bitbucket:
 - + [Link to source code](#) (permission required)
 - + Description: Bitbucket is a Git repository management tool created by Atlassian. This can easily be integrated with Jira and Confluence to automate and keep track of every pull request made by team members. CI/CD is fully supported, which makes the deployment process much simpler.

6.3 Tools & Infrastructures

Category	Tools / Infrastructure
Technology	ReactJS 18.2.0 (Front-end), Flutter 3.7.0-4.0.pre.5 (Mobile), .NET 6 (Back-end), Python 3.7.9 (Image Processing)
Database	FireBase, PostgreSQL 13.7
IDEs/Editors	Visual Studio Code 1.71, Visual Studio 2022, Arduino CC 2.0.0
Diagramming	Diagrams.net, Figma
Documentation	MS Word/Excel 2016, Google Docs(Reports/Documents), MS Powerpoint 2016 (Slides)
Version Control	Bitbucket(Source Codes), Google Drive/Confluence(Documents)
Deployment server	Amazon Web Service
Project management	Jira (Schedule, Tasks, Defects)
Prototype	Figma

III. Software Requirements Specification

1. Product Overview

Automatic Parking Management and Parking Fee Collection System, or APMS, is a product aimed at delivering a contactless solution by automating the process of managing public car parking lots (supermarkets, parks, etc.), thus reducing the time and human resources for the parking process. The context diagram below illustrates the external entities and system interfaces for release 1.0.

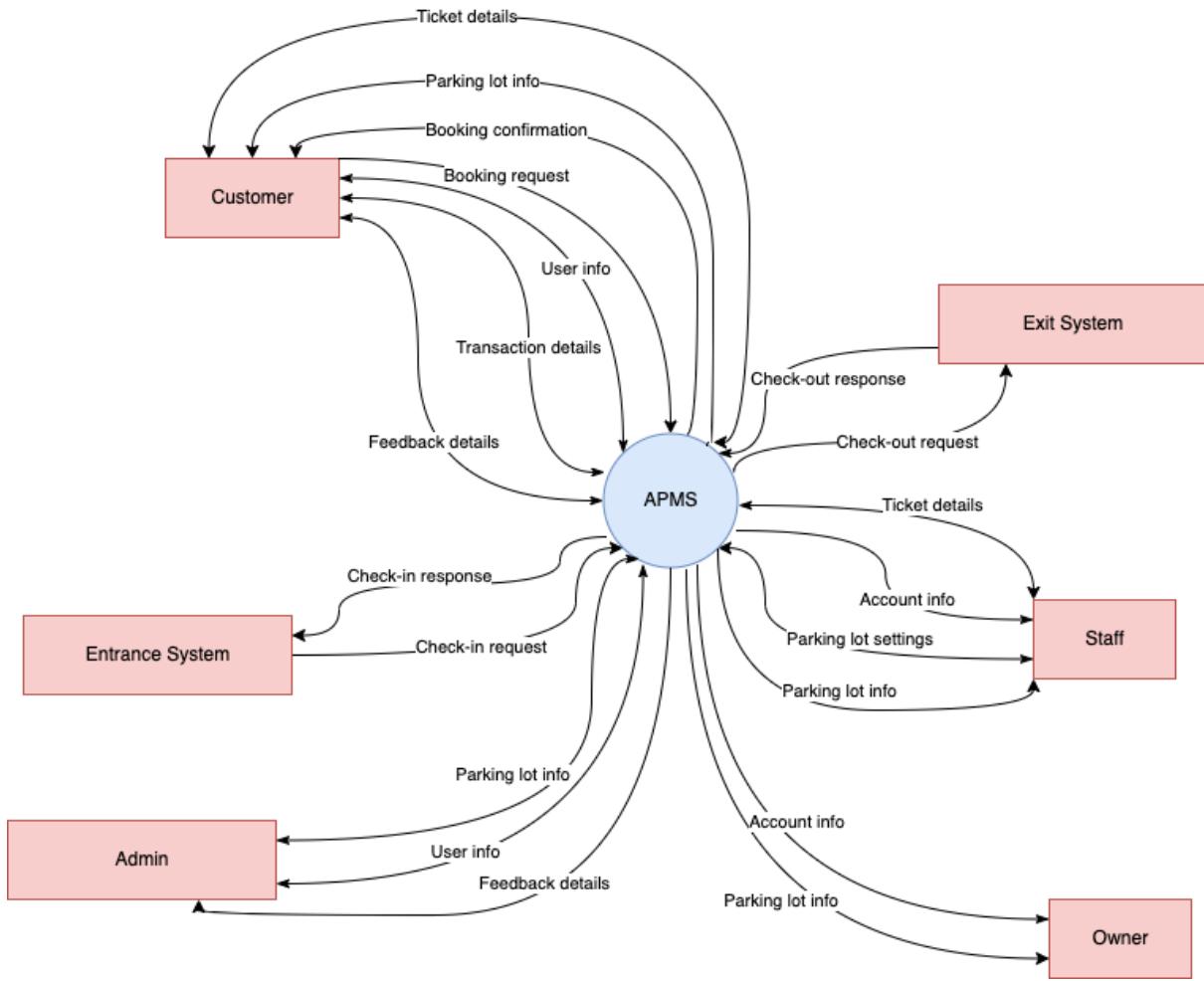


Figure 2. Context diagram of the APMS

2. User Requirements

2.1 Actors

#	Actor	Description
1	Guest	Users who are not the members of the system
2	Authenticated User	Users who are the members of the system
3	Admin	Authenticated users with the admin role, who manage all car parks and accounts in the system
4	Owner (Car Park Owner)	Authenticated users with the owner role, can make statistics their own car park
5	Staff	Authenticated users with the staff role, manage tickets, system configuration and information of their own car park
6	Customer	Authenticated users with the customer role, mostly interact with the system using the mobile application to book a parking slot and check-in at a car park
7	System Handler	Helps handle check-in/check-out processes, and automatically cancels expired tickets

2.2 Use Cases

2.2.1 Diagram

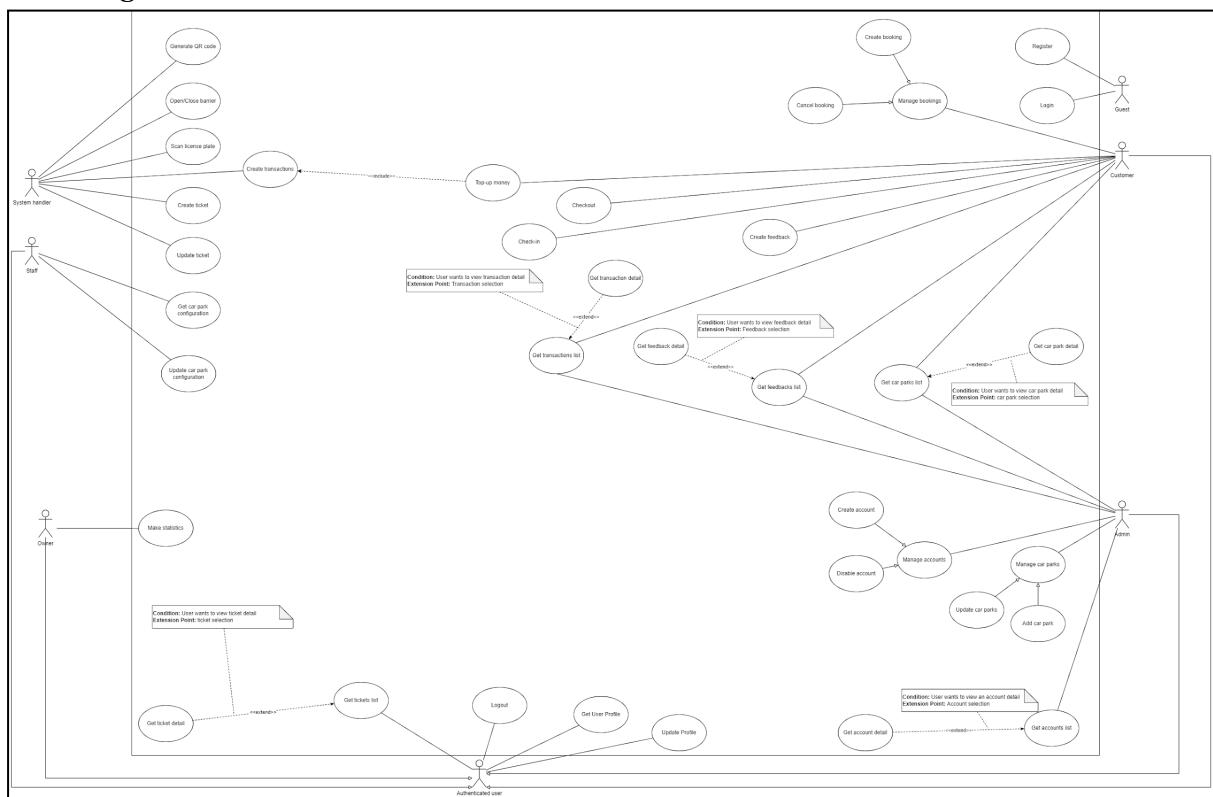


Figure 3. Use case diagram of the APMS

Ref: [Unified Modeling Language, v2.5.1 \(omg.org\)](http://www.omg.org/spec/UML/v2.5.1/)

Link to diagram: [Use-Case Diagram](#)

2.2.2 Descriptions

ID	Use Case	Actors	Use Case Description
01	Register	Guest	User must register to use the features of the system as a customer
02	Login	Guest	Determine actor's identity, role, and permissions
03	Get car park list	Admin, Customer	User views a list of car parks existing in the system
04	Get car park detail	Authenticated user	User views the details of a car park. Owners and staff can only view the detail of their car parks
05	Get user profile	Authenticated user	User views his own account profile
06	Update user profile	Authenticated user	User updates his own account profile
07	Get feedback list	Admin	Admin views all feedbacks.
08	Get feedback detail	Admin, Customer	User views a selected feedback detail
09	Get ticket list	Authenticated user	Admin can view all tickets. Owners and staff can only view all tickets of their car parks. Customers can only view their tickets
10	Get ticket detail	Authenticated user	User views the detail of the selected ticket
11	Make statistics	Owner	Owner makes statistics of their car park
12	Update car park configuration	Staff	Staff configures their car park settings: reservation fee, parking fee per hour, etc.
13	Top-up money	Customer	Customer can add more money to their account
14	Check-in	Customer	Customer checks in at a car park
15	Check-out	Customer	Customer checks out of a car park
16	Create feedbacks	Customer	Customer creates feedbacks to inform us about system's issues or tell us about their experiences using the system
17	Manage bookings	Customer	Customer creates and cancels bookings
18	Create bookings	Customer	Customer creates booking for a parking slot at a specific car park
19	Cancel bookings	Customer, Staff	Customer cancels their bookings, Staff cancels bookings of their car park
20	Manage accounts	Admin	Admin can manage the accounts in the system

21	Create accounts	Admin	Admin can create accounts with Admin, Staff and Owner role
22	Deactivate accounts	Admin	Admin can deactivate accounts in the system
23	Get account list	Admin	Admin views the list of accounts in the system
24	Get account detail	Admin	Admin views the selected account detail
25	Manage car parks	Admin	Admin manages car parks in the system
26	Add car parks	Admin	Admin can add the car parks in the system
27	Update car parks	Admin	Admin can update the information of the car parks in the system
28	Remove car parks	Admin	Admin can remove car parks from the system
29	Get booking list	Authenticated user	Owner and staff can only view their car park bookings; customer can only view their bookings, and admin can view all bookings
30	Get booking detail	Authenticated user	User views the selected booking detail
31	Logout	Authenticated user	User logs out of the system
32	Create transaction	System handler	System creates a transaction when customer top up their account, book a slot or check-out at a car park
33	Create ticket	System handler	System creates a ticket ticket when customer checks in to a car park
34	Update ticket	System handler	System updates the status and total fee of the ticket when the customer checks out of a car park, or changes status of the ticket to canceled if it is expired
35	Scan license plate	System handler	System scans and recognizes license plate number when customer checks in or checks out at the car park
36	Generate QR code	System handler	System generates a QR code for customer to scan when checking in and checking out
37	Open/Close barrier	System handler	System opens or closes barriers after customer has checked in or checked out successfully

3. Functional Requirements

3.1 System Functional Overview

3.1.1 Screen flow

- Customer screen flow:

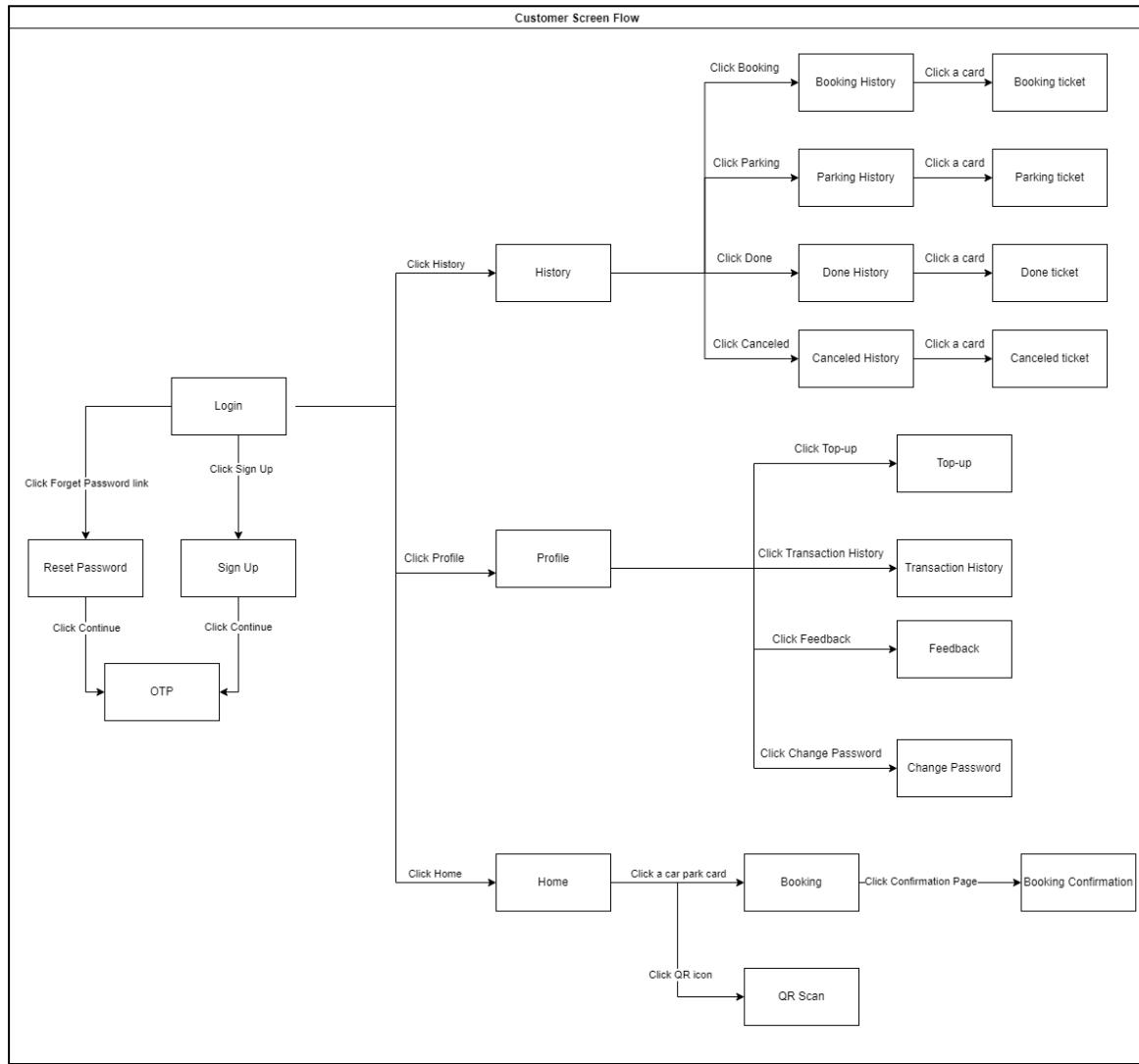


Figure 4. Screen flow diagram for Customer mobile application

- Admin screen flow:

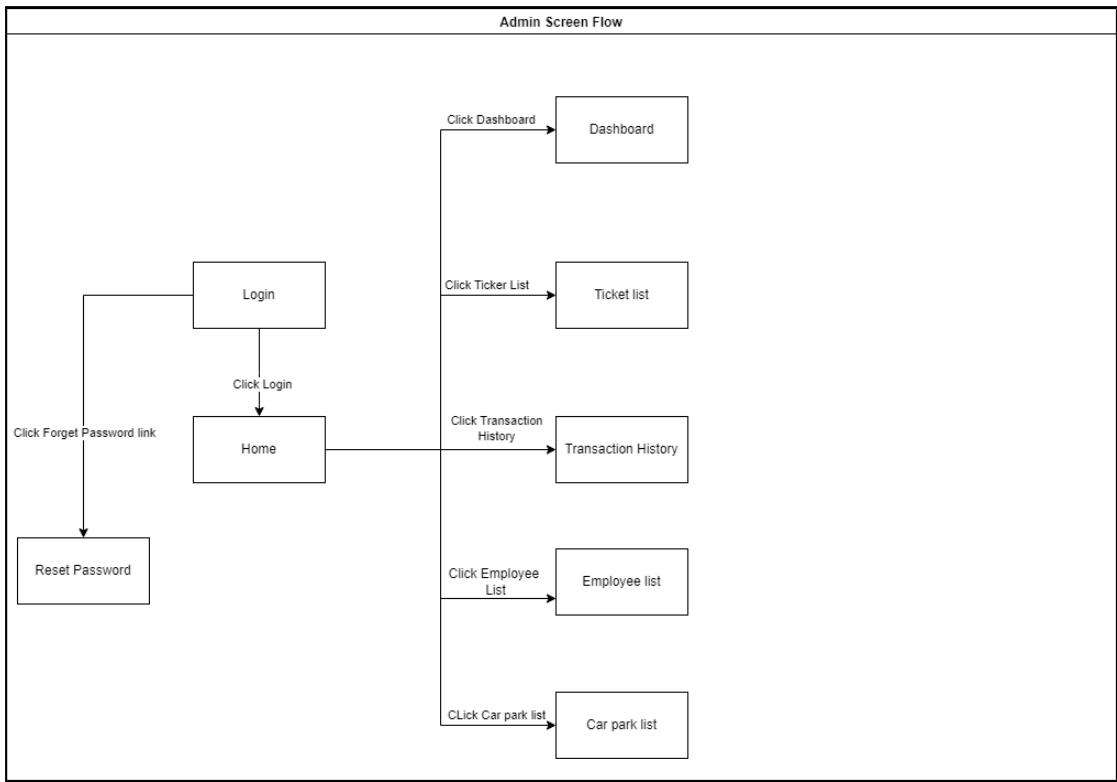


Figure 5. Screen flow diagram for Admin web application

- Owner screen flow:

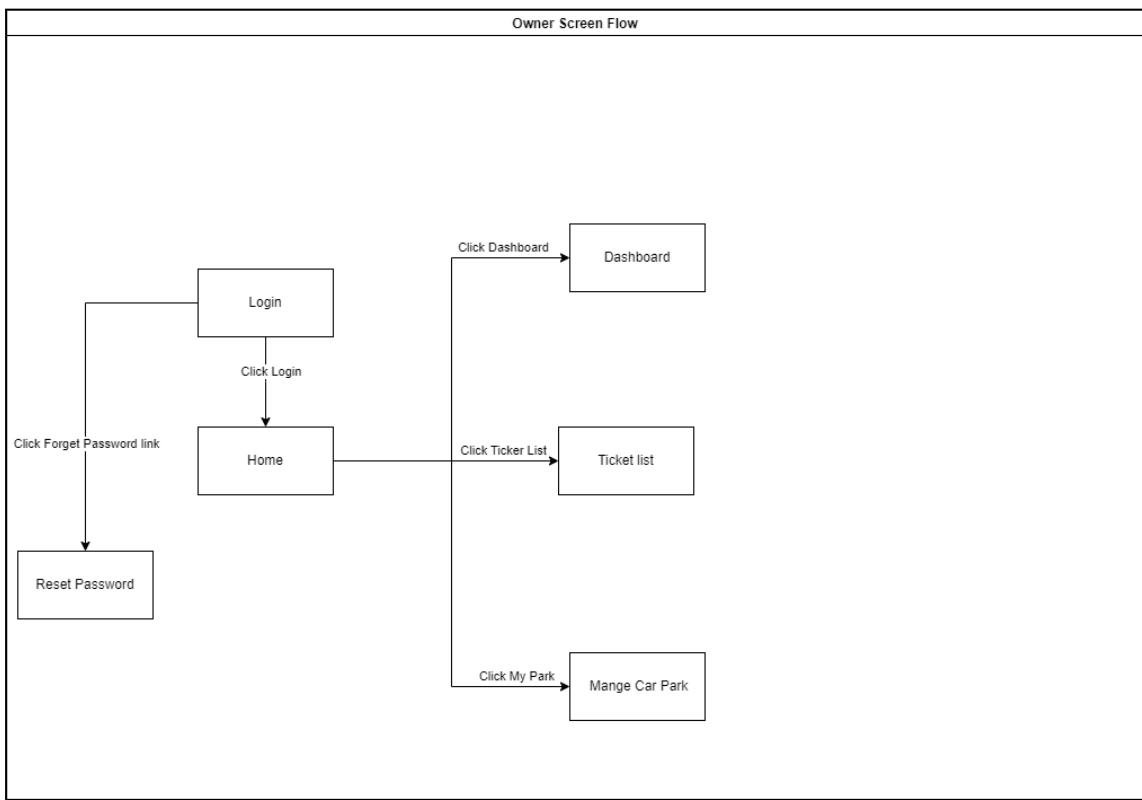


Figure 6. Screen flow diagram for Owner web application

- Staff screen flow:

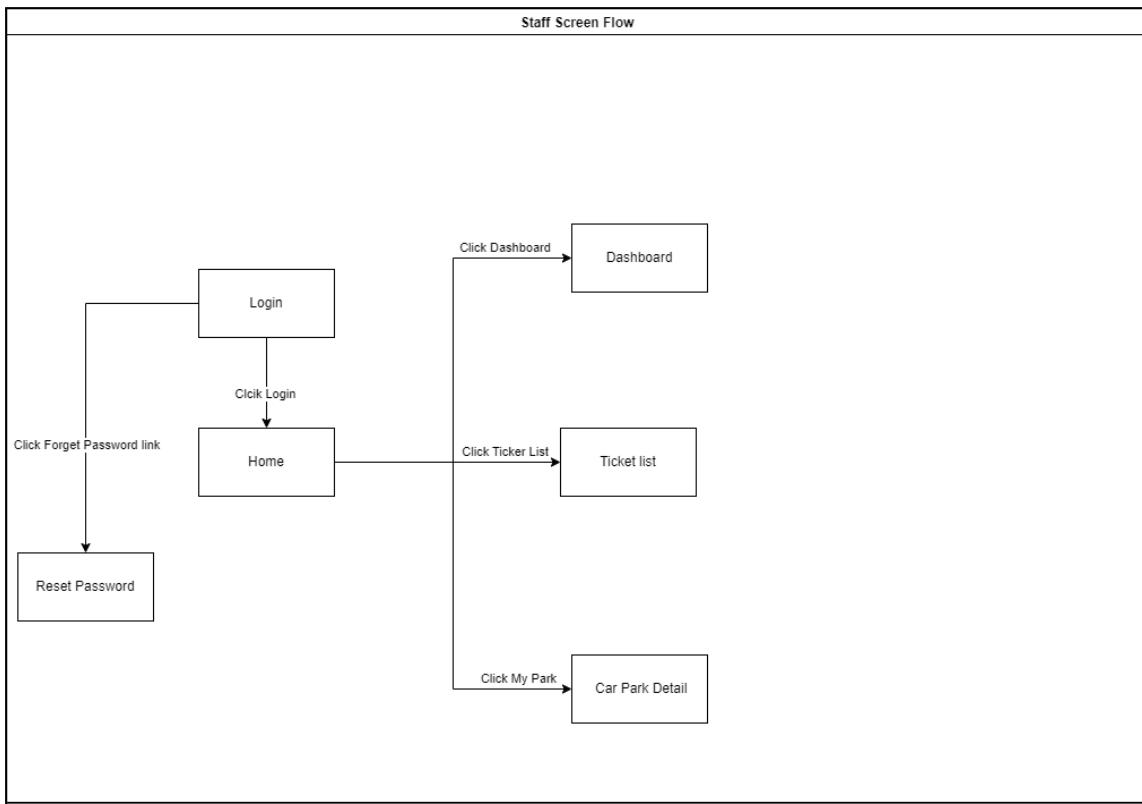


Figure 7. Screen flow diagram for Staff web application

3.1.2 Screen details

- Customer mobile application:

#	Feature	Screen	Description
1	Create account	Sign up	Display a register form for creating a customer
2	Authenticate	Sign in	Allow logging into the system using customer phone number
3	Authenticate	Forget password	Help customer retrieve their password
4	Book a slot	Home	Give user list of car parking order by distance
5	Book a slot	Create booking	Allow user to create a booking at selected car park
6	Book a slot	Booking confirmation	For customer to confirm booking detail before the system create a booking
7	Check in/out	QR Scan	Allow customer to use phone camera to scan QR code and check in or out if available
8	Get booking tickets	Booking history	Display a list of booking tickets currently in the system
9	Get booking tickets	Booking ticket	Display details of a chosen booking ticket and allow actor to cancel the ticket

10	Get parking tickets	Parking history	Display a list of parking tickets currently in the system
11	Get parking tickets	Parking ticket	Display details of a chosen booking ticket
12	Get done tickets	Done history	Display a list of done tickets in the past
13	Get done tickets	Done ticket	Display details of a chosen done ticket
14	Get canceled tickets	Canceled history	Display a list of canceled tickets in the past
15	Get canceled tickets	Canceled ticket	Display details of a chosen canceled ticket
16	Manage account	Profile	Display account's information
17	Manage account	Change password	Allow customer to change old password
18	Top-up	Top-up	Allow customer to pay a fund he/she wants
19	Get transaction history	Transaction history	Display history of all transactions performed by customer
20	Create feedback	Send feedback	Allow customer to send a feedback

- Admin web application:

#	Feature	Screen	Description
1	Authenticate	Sign in	Allow logging into the system using customer phone number
2	Authenticate	Forget password	Help user retrieve their password
3	Get admin reports	Dashboard	Display statistics of total of users, booking, check-in; charts of system overview, employees, booking stats; list of booking details under many kinds of chart
4	Get car parks	Car park list	Display a car park list that allows user to view or edit car park details or add a new car park
5	Get tickets	Ticket list	Display ticket list that allows user to view ticket details
6	Get transaction history	Transaction history	Display a list of all transactions in the past
7	Get employees	Employee list	Display an employee list that allows user to view, edit employee details, deactivate an employee or add a new employee

- Owner web application:

#	Feature	Screen	Description
1	Authenticate	Sign in	Allow logging into the system using customer phone number
2	Authenticate	Forget password	Help user retrieve their password
4	Get tickets	Ticket list	Display ticket list that allows user to view ticket details
5	Manage car park	My park	Allow user to view or update information, fee and price of the car park and change permission of employees

- Staff web application:

#	Feature	Screen	Description
1	Authenticate	Sign in	Allow logging into the system using customer phone number
2	Authenticate	Forget password	Help user retrieve their password
3	Get staff reports	Dashboard	Display booking list of current day, scanned plate and car park overview
4	Get tickets	Ticket list	Display ticket list that allows user to view ticket details
5	Manage car park	My park	Allow user to view or update information, fee and price of the car park if he/she has permission

3.1.3 Screen Authorization

- Mobile application:

Screen	Unauthenticated User	Authenticated User
Sign up	X	
Sign in	X	
Forget password	X	
Home		X
Create booking		X
Booking confirmation		X
QR Scan		X
Booking history		X
Booking ticket		X
Parking history		X
Parking ticket		X
Done history		X
Done ticket		X
Canceled history		X
Canceled ticket		X
Profile		X
Change password		X
Top-up		X
Transaction history		X
Send feedback		X

- Web application:

Screen	Unauthenticated User	Admin	Owner	Staff
Sign in	X			
Forget password	X			
Admin dashboard		X		
Owner dashboard			X	
Staff dashboard				X
Car park list		X		
Ticket list			X	X
Transaction history		X		
Employee list		X		
My park		X		

3.1.4 Non-Screen Functions

#	Feature	System Function	Description
1	Security	Password Hashing	All passwords are hashed with salt using SHA256
2	Booking management	Automatically cancel bookings	Automatically cancel bookings that are expired
3	Image processing	Detect vehicle and its number plate	Recognize plate number from pictures

3.2 Booking Progress

3.2.1 Booking Home

- **Function trigger:** Actor goes to the home screen after logging in or clicks “Home” button in the bottom menu
- **Function description:** Actor can search for nearest car park here
- **Screen Layout:**

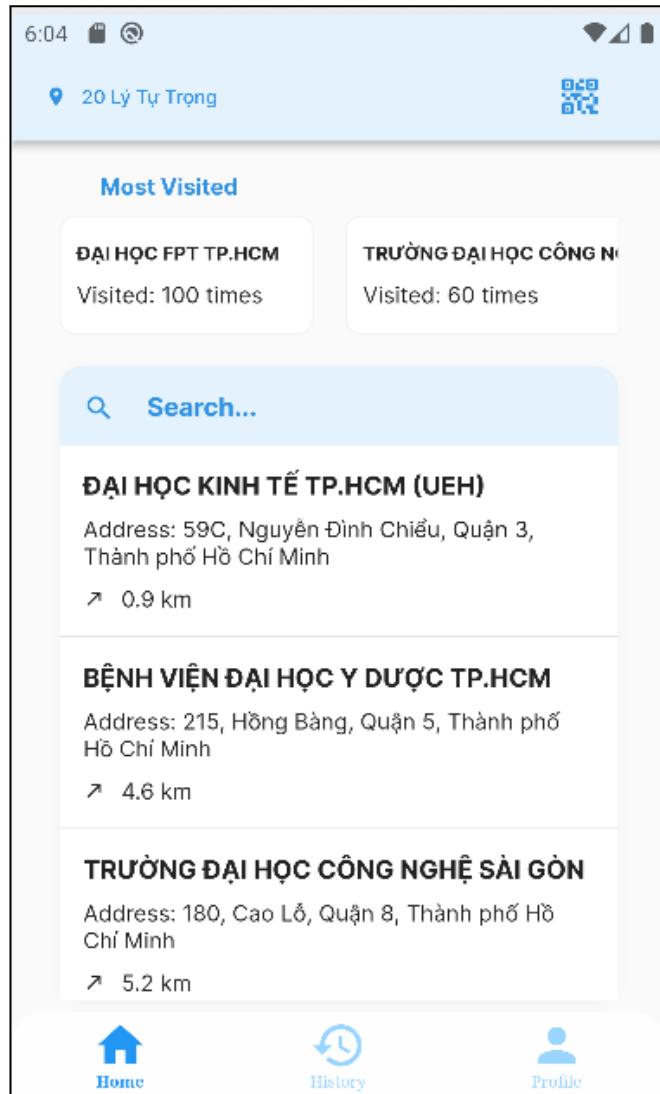


Figure 8. Home screen

- **Function details:**

Fields:

#	Field name	Description	Read only	Mandatory	Control type	Data type
1	Car park search	Fill car park information	No	Yes	Text box	String

Buttons/Links:

#	Function	Description	Validation	Outcome
2	QR Scan	Click to open camera and scan QR	No	Proceed to QR scan screen
3	Home	Click to go to home	No	Switch to home screen
4	History	Click to view ticket history	No	Switch to ticket history screen
5	Profile	Click to view profile information	No	Switch to profile screen

3.2.2 Create booking

- **Function trigger:** Actor clicks on a car park card in the middle of the screen
- **Function description:** Actor can book a ticket
- **Screen Layout:**

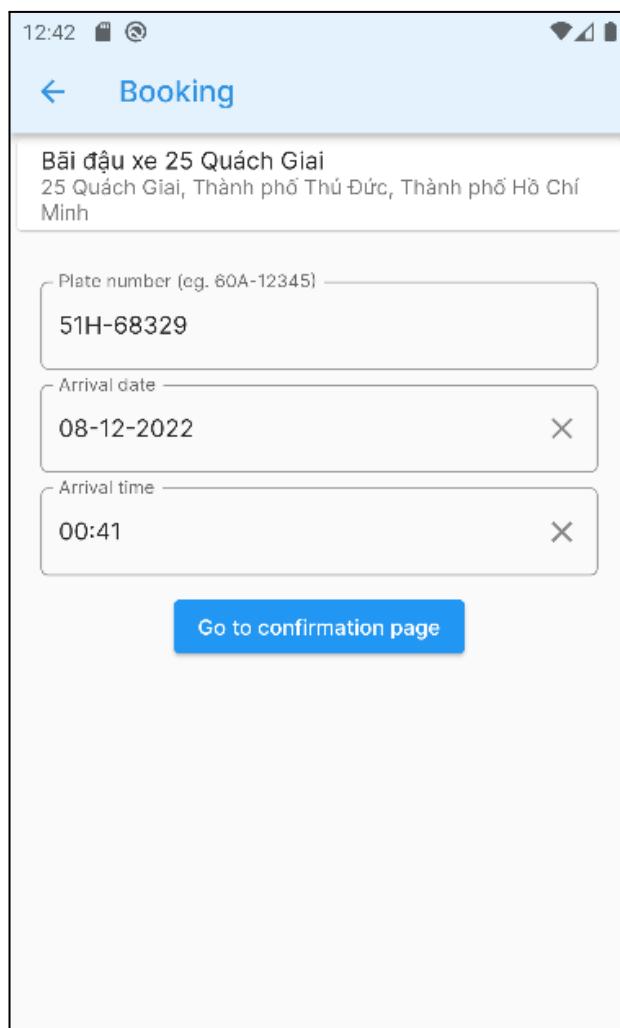


Figure 9. Create Booking screen

- **Function details:**

Fields:

#	Field name	Description	Read only	Mandatory	Control type	Data type
1	Plate number	Fill car's plate number	No	Yes	Text box	String
2	Arrival date	Fill arrival date	No	Yes	Text box/Date	Date
3	Arrival time	Fill arrival time	No	Yes	Text box/Time	Time

Buttons/Links:

#	Function	Description	Validation	Outcome
4	Go to confirmation page	Click to confirm inputted booking details	Yes	Proceed to confirmation page

3.2.2 Confirmation page

- **Function trigger:** Actor clicks on “Go to confirmation page” button
- **Function description:** Actor gets booking confirmation
- **Screen Layout:**

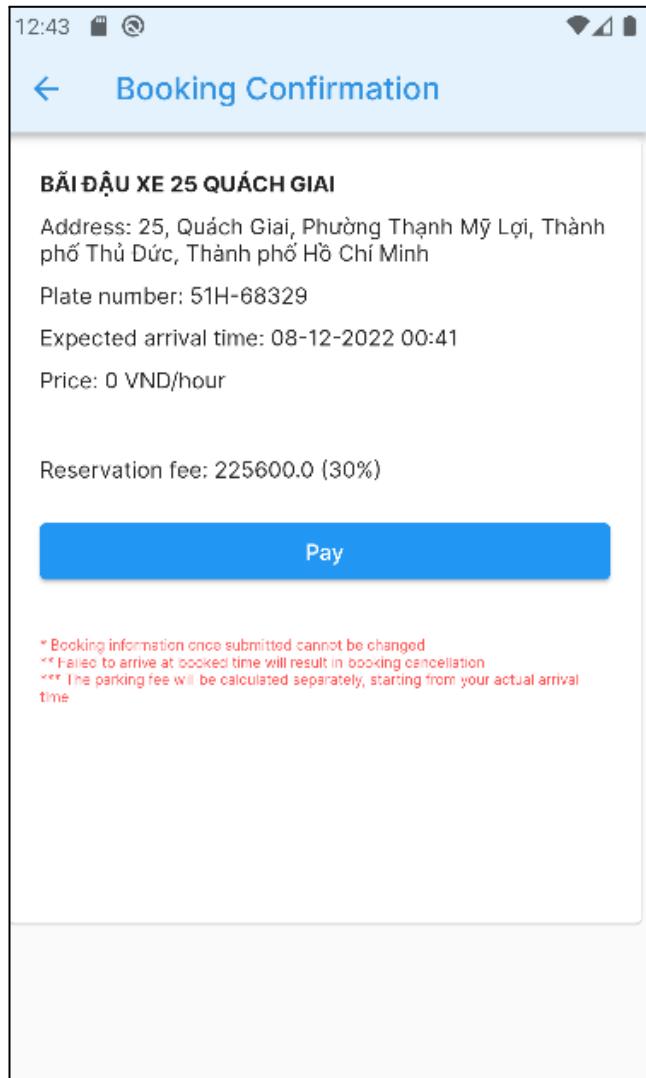


Figure 10. Booking Confirmation screen

- **Function details:**

Buttons/Links:

#	Function	Description	Validation	Outcome
1	Pay	Click to pay reservation fee	Yes	Proceed to pay reservation fee

3.2.3 QR Scan

- **Function trigger:** Actor clicks on “QR Scan” button on the top right of home screen
- **Function description:** Actor scans QR to check in or check out
- **Screen Layout:**

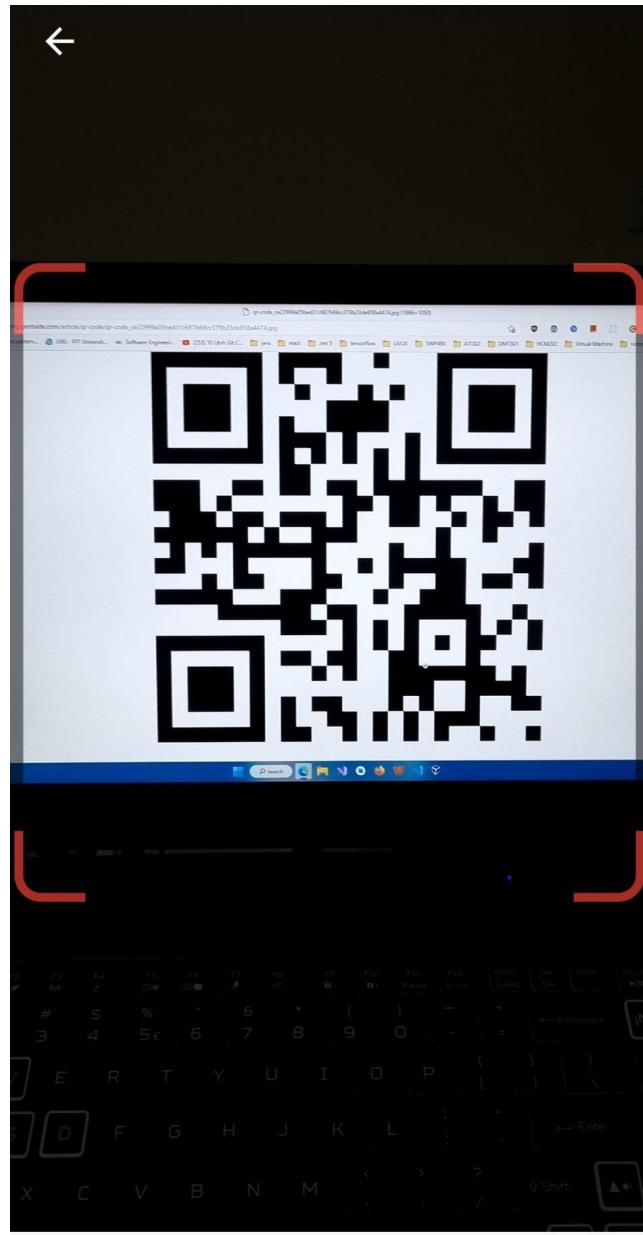


Figure 11. QR Scan screen

- **Function details:**

Buttons/Links:

#	Function	Description	Validation	Outcome
1	Pay	Click to pay reservation fee	Yes	Proceed to pay reservation fee

3.3 History

3.3.1 Booking history

- **Function trigger:** Click “Booking” in the top menu
- **Function description:** Actor can view booking history
- **Screen Layout:**

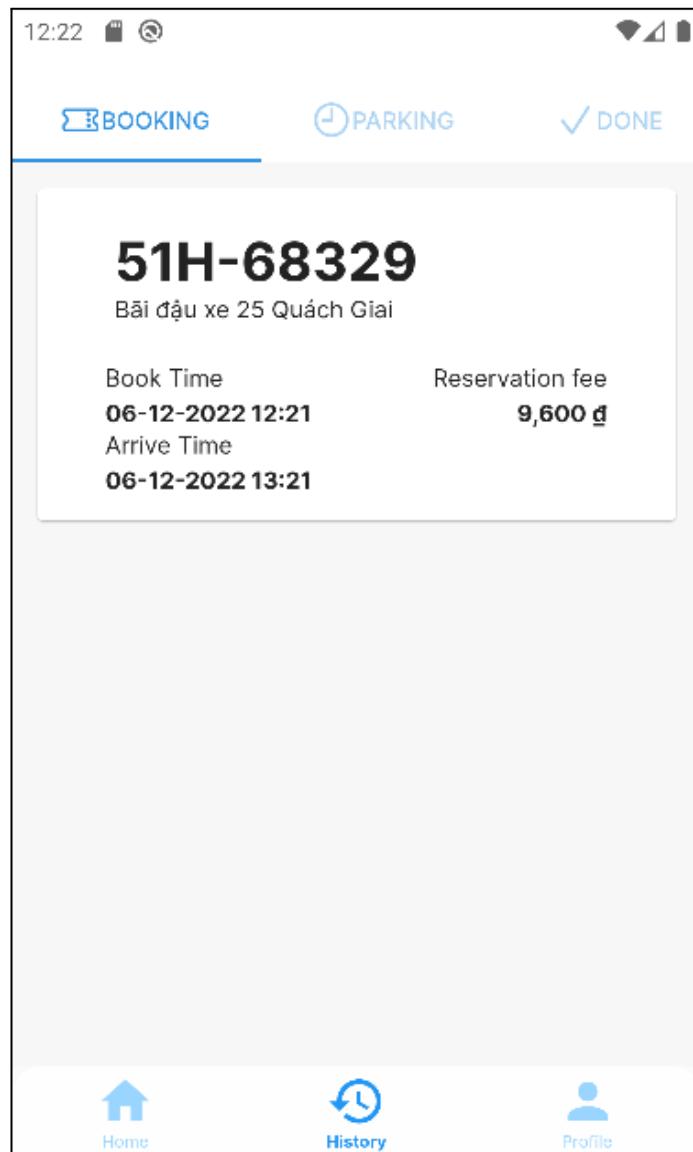


Figure 12. Booking History screen

- **Function details:**

Buttons/Links:

#	Function	Description	Validation	Outcome
1	Booking ticket card	Click on a card to view the booking ticket detail	No	Proceed to booking ticket

3.3.2 Booking ticket

- **Function trigger:** Click on a card to view booking ticket
- **Function description:** Actor can view booking ticket detail or cancel booking
- **Screen Layout:**

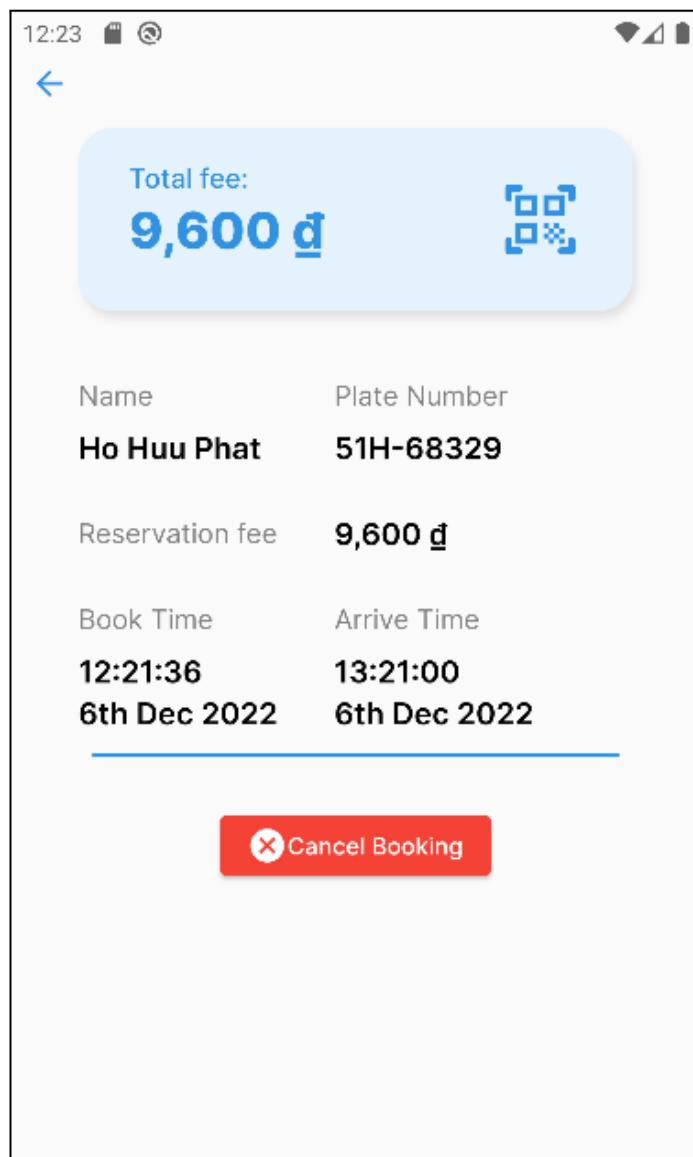


Figure 13. Booking Ticket screen

- **Function details:**

Buttons/Links:

#	Function	Description	Validation	Outcome
1	Cancel Booking	Click to cancel booking	No	Proceed to QR scan screen

3.3.3 Parking history

- **Function trigger:** Click “Parking” in the top menu
- **Function description:** Actor can view parking ticket detail
- **Screen Layout:**

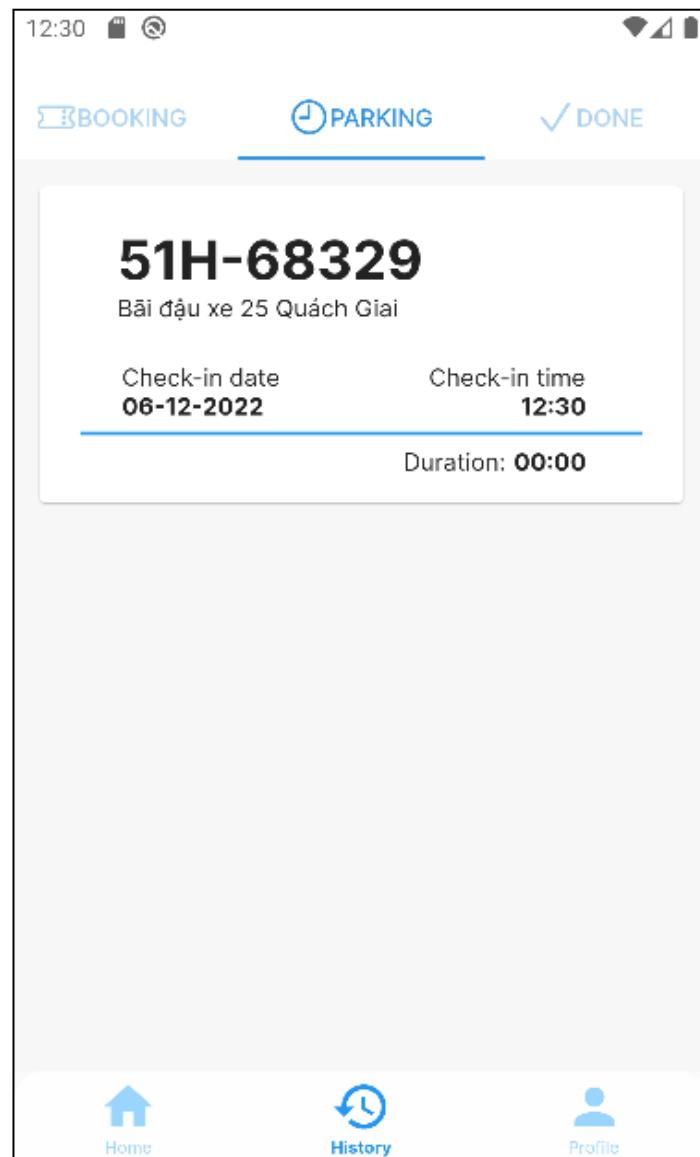


Figure 14. Parking History screen

- **Function details:**

Buttons/Links:

#	Function	Description	Validation	Outcome
1	Parking ticket card	Click on a card to view the parking ticket detail	No	Proceed to parking ticket

3.3.4 Parking ticket

- **Function trigger:** Click on a card to view parking ticket
- **Function description:** Actor can view parking ticket history
- **Screen Layout:**

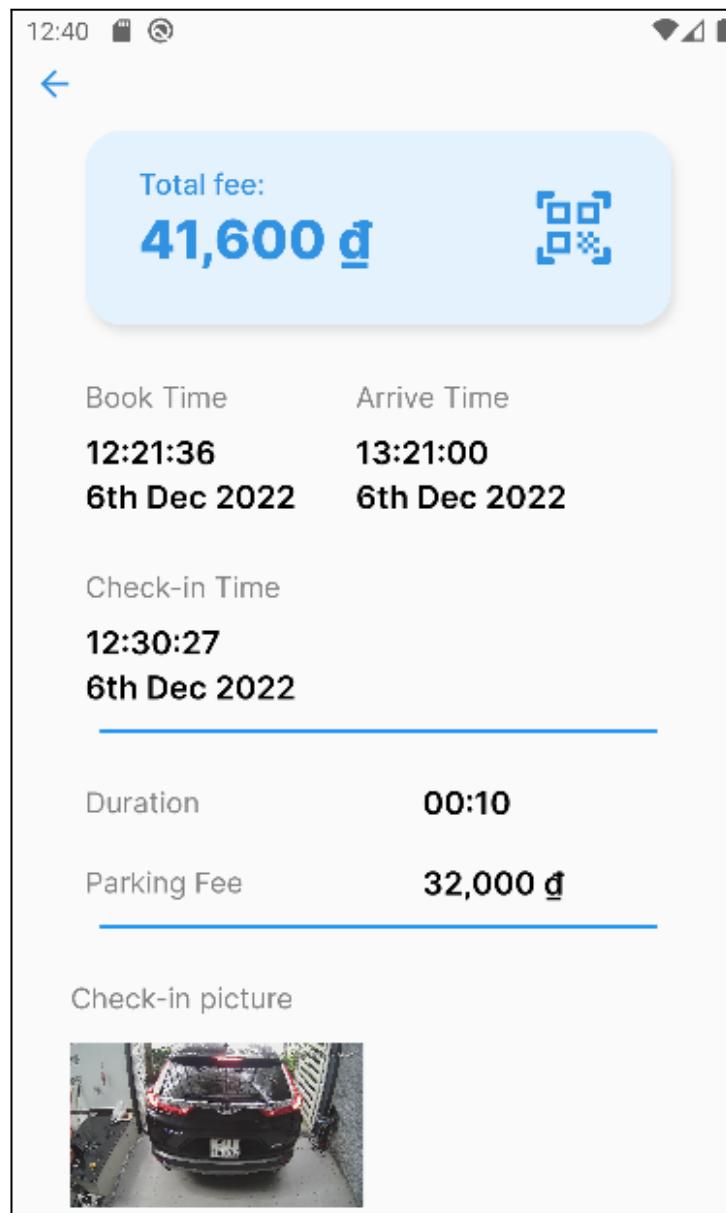


Figure 15. Parking Ticket screen

3.3.5 Done history

- **Function trigger:** Click “Done” in the top menu
- **Function description:** Actor can view done history
- **Screen Layout:**

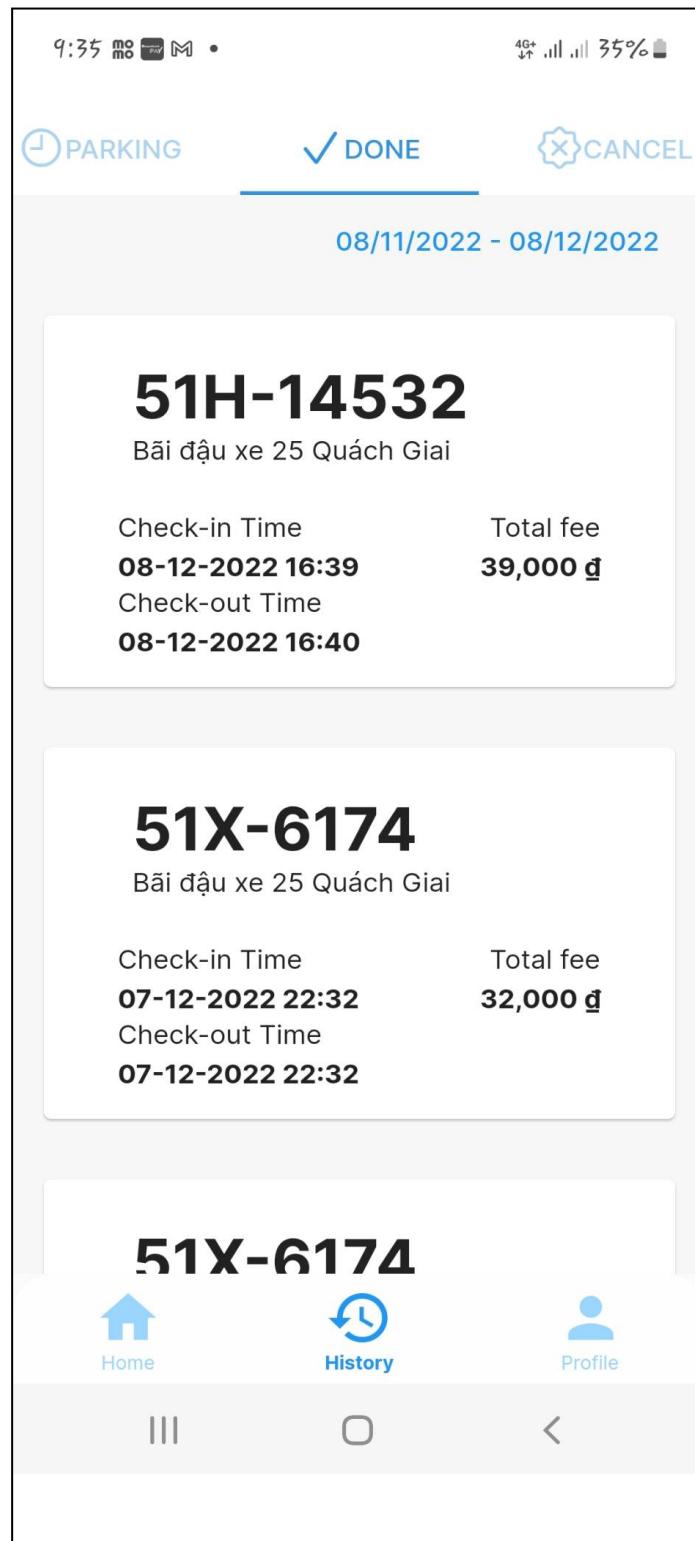


Figure 16. Done History screen

- **Function details:**

Buttons/Links:

#	Function	Description	Validation	Outcome
1	Done ticket card	Click on a card to view the done ticket detail	No	Proceed to done ticket

3.3.6 Done ticket

- **Function trigger:** Click on a card to view done ticket
- **Function description:** Actor can view done ticket detail
- **Screen Layout:**

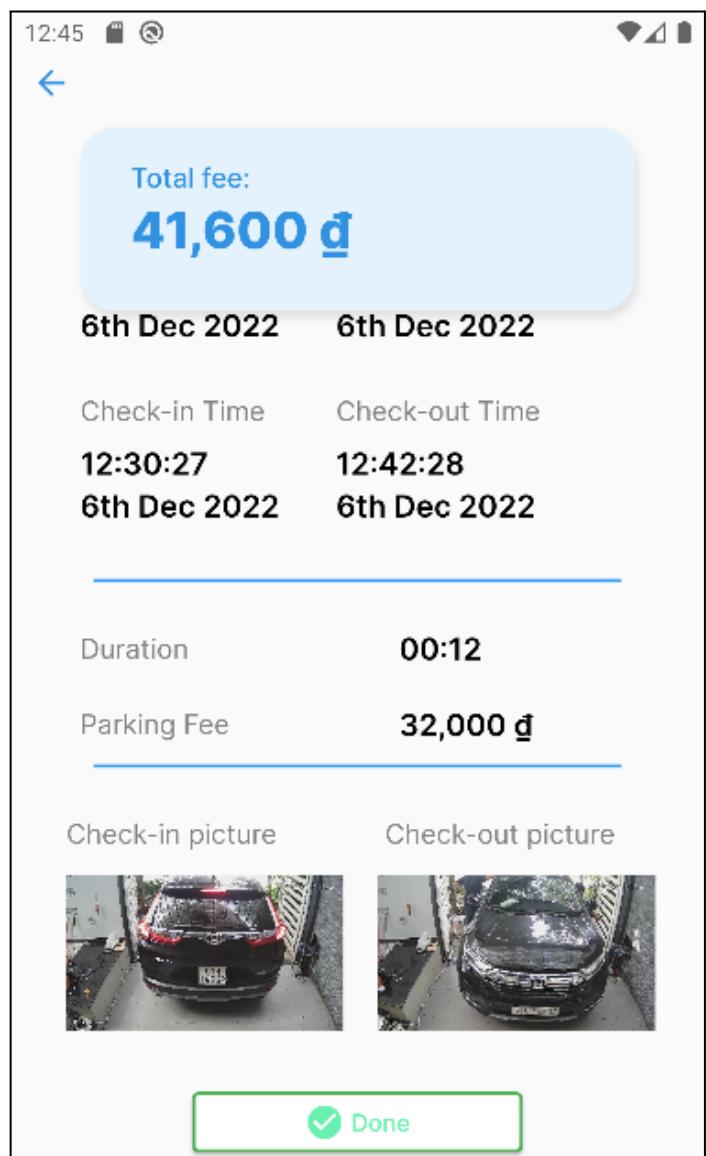


Figure 17. Done Ticket screen

3.3.7 Canceled history

- **Function trigger:** Click “Canceled” in the top menu
- **Function description:** Actor can view history of booking that has been canceled
- **Screen Layout:**

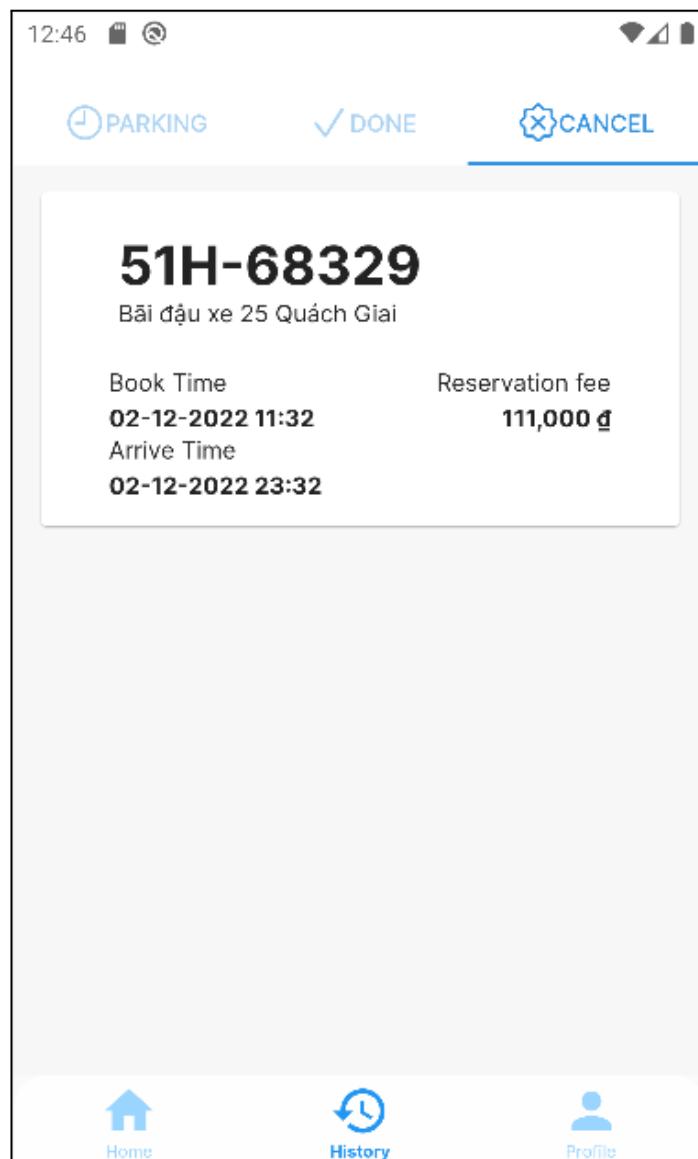


Figure 18. Canceled History screen

- **Function details:**

Buttons/Links:

#	Function	Description	Validation	Outcome
1	Canceled ticket card	Click on a card to view the canceled ticket detail	No	Proceed to canceled ticket

3.3.8 Canceled ticket

- **Function trigger:** Click on a card to view canceled ticket
- **Function description:** Actor can view canceled ne ticket detail
- **Screen Layout:**

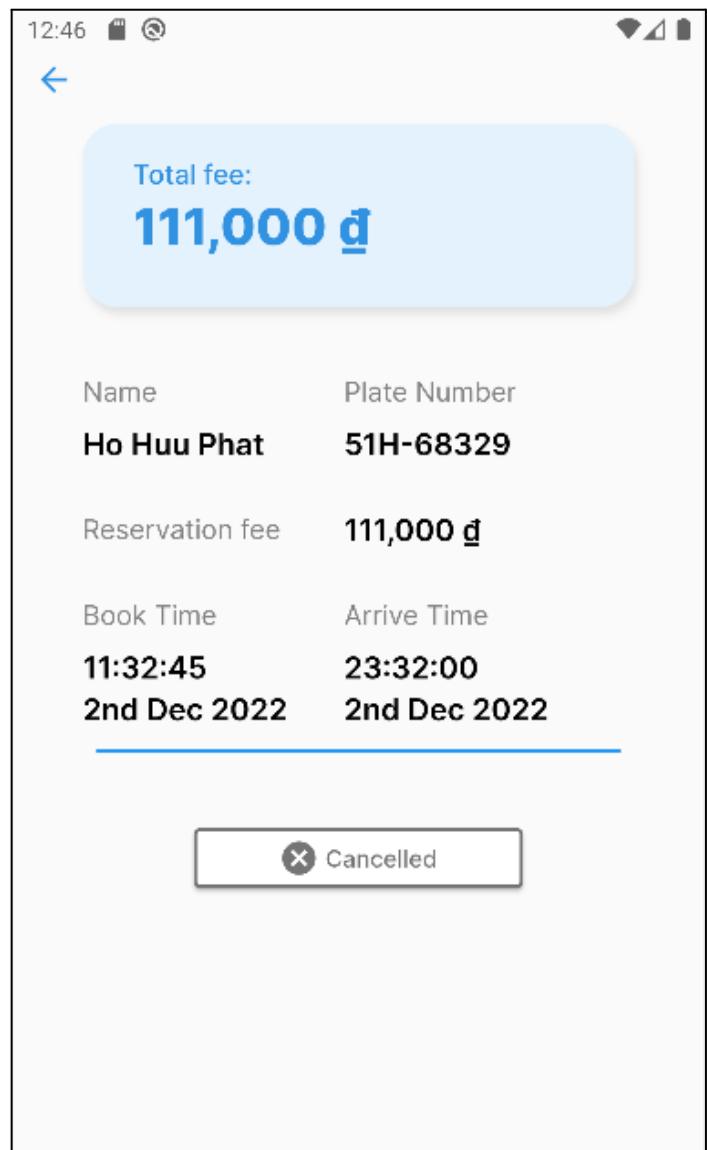


Figure 19. Canceled Ticket screen

3.4 Profile

3.4.1 Profile information

- **Function trigger:** Actor clicks “Profile” button in the bottom menu
- **Function description:** Actor can view or update account balance, view transaction history, send feedback, change password and log out
- **Screen Layout:**

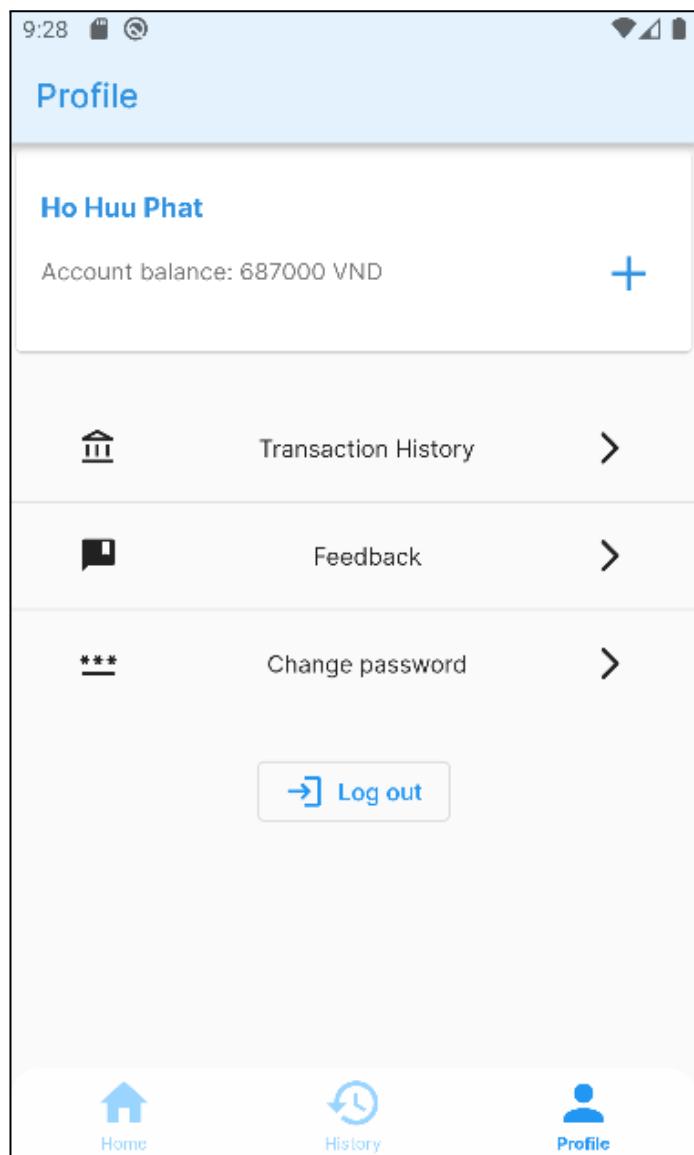


Figure 20. Profile screen

- **Function details:**

Buttons/Links:

#	Function	Description	Validation	Outcome
1	Topup	Click to add funds	No	Proceed to top-up screen
2	Transaction history	Click to view recent transaction	No	Proceed to transaction history screen

4	Feedback	Click to send a feedback	No	Proceed to feedback screen
5	Change password	Click to change current password	No	Proceed to change password screen
6	Log out	Click to log out	No	Log current account out

3.4.2 Top-up

- **Function trigger:** Actor clicks “Top-up” button on the right side of account balance
- **Function description:** Actor can choose a fund to add
- **Screen Layout:**

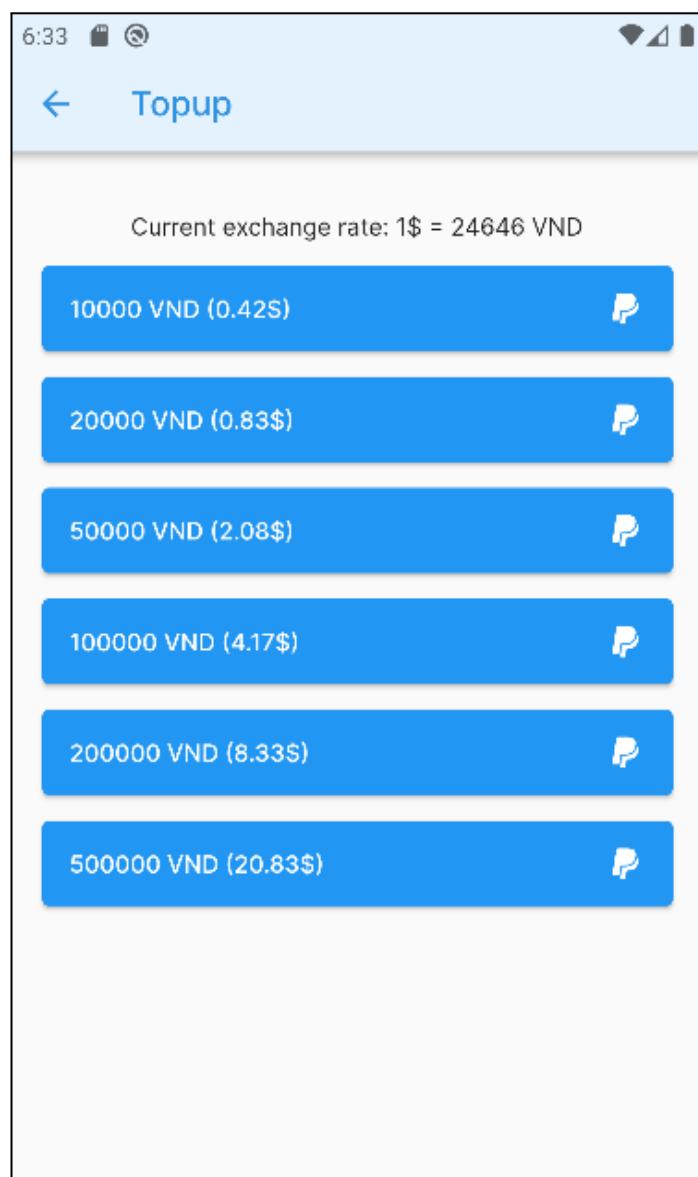


Figure 21. Top-up screen

- **Function details:**

Buttons/Links:

#	Function	Description	Validation	Outcome
1	Fund	Click to choose a fund you want to add	No	Proceed to Paypal

3.4.3 Transaction history

- **Function trigger:** Actor clicks “Transaction” button in the middle of profile
- **Function description:** Actor can view transaction history
- **Screen Layout:**

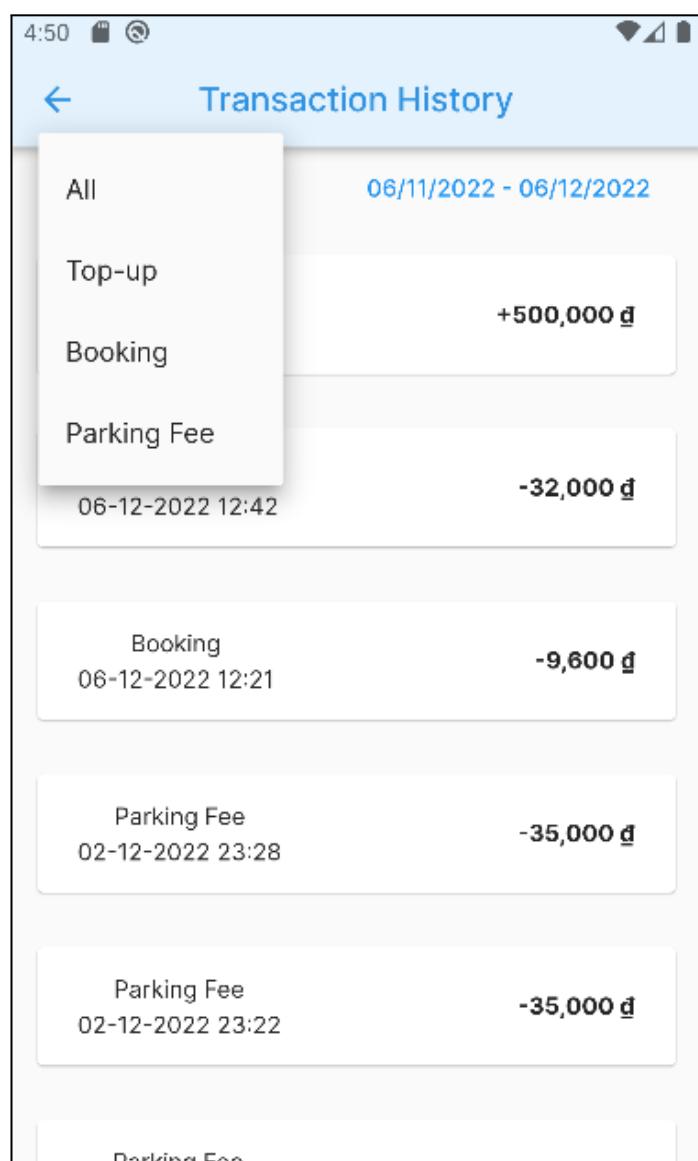


Figure 22. Transaction History screen

- **Function details:**

Fields:

#	Field name	Description	Read only	Mandatory	Control type	Data type
1	Transaction type	Filter transaction list	No	No	Dropdown List	List

3.4.3 Feedback

- **Function trigger:** Actor clicks “Feedback” button in the middle of profile
- **Function description:** Actor can send a feedback
- **Screen Layout:**

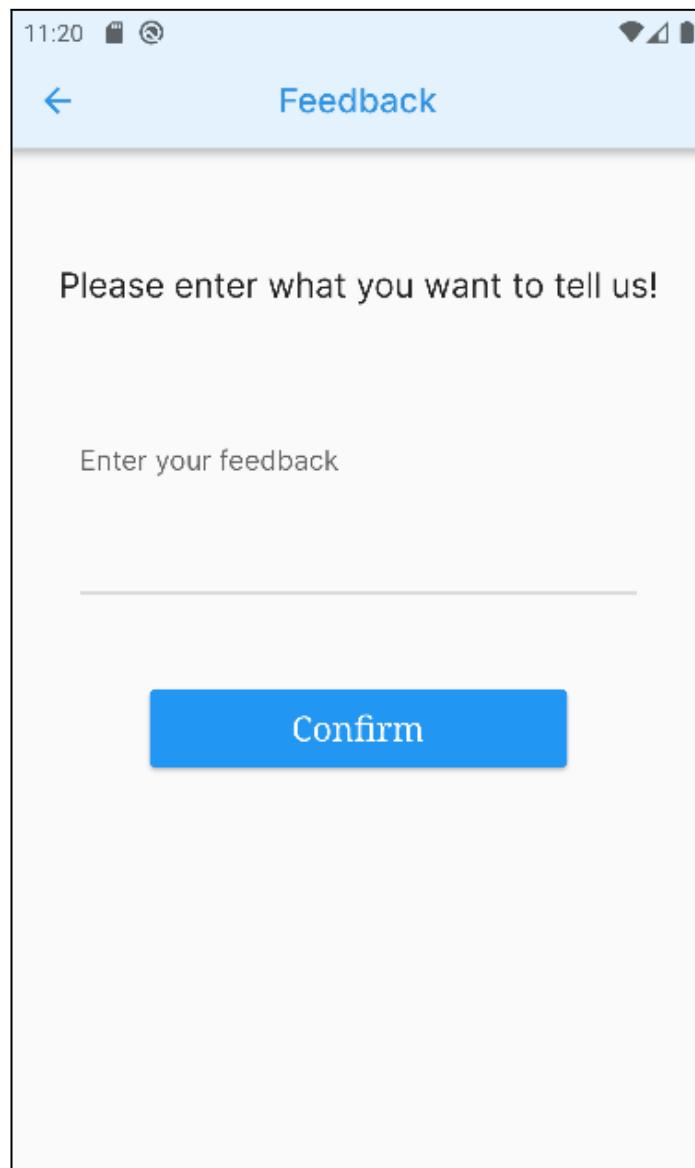


Figure 23. Feedback screen

- **Function details:**

Fields:

#	Field name	Description	Read only	Mandatory	Control type	Data type
1	Enter feedback	Fill feedback detail	No	Yes	Text box	String

Buttons/Links:

#	Function	Description	Validation	Outcome
2	Confirm	Click to confirm sending feedback	Yes	Send inputted feedback

3.4.4 Change password

- **Function trigger:** Actor clicks “Change Password” button in the middle of profile
- **Function description:** Actor can change current password
- **Screen Layout:**

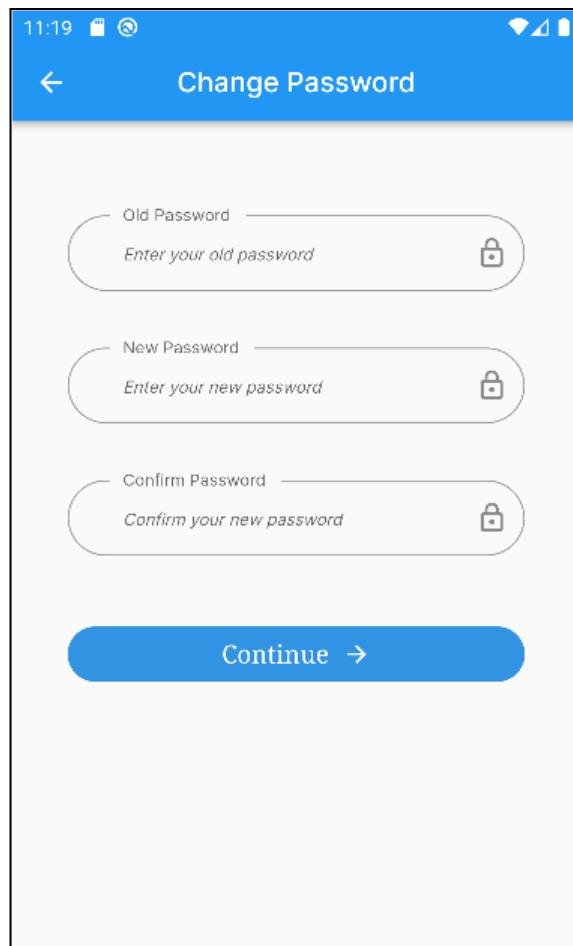


Figure 24. Change Password screen

- **Function details:**

Fields:

#	Field name	Description	Read only	Mandatory	Control type	Data type
1	Enter old password	Fill old password	No	Yes	Password	String
2	Enter new password	Fill new password	No	Yes	Password	String
3	Confirm new password	Re-enter new password	No	Yes	Password	String

Buttons/Links:

#	Function	Description	Validation	Outcome
4	Continue	Click to keep changing password	Yes	Back to profile screen

3.5 Customer Authorization

3.5.1 Sign In

- **Function trigger:** Actor goes to Sign In screen
- **Function description:** Actor fill out a sign-in form and submit to sign in account
- **Screen layout:**

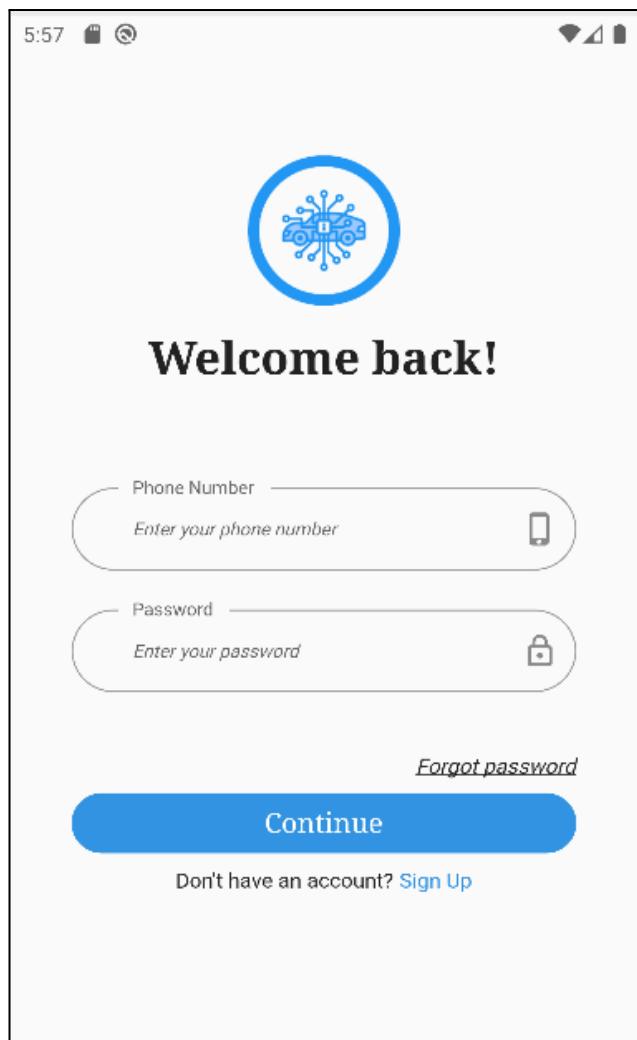


Figure 25. Sign In screen

- **Function details:**

Fields:

#	Field name	Description	Read only	Mandatory	Control type	Data type
1	Phone number	Fill Phone number	No	Yes	Text box	Number
2	Password	Fill Password	No	Yes	Password	String

Buttons/Links:

#	Function	Description	Validation	Outcome
3	Continue	Click to continue signing in	Yes	Proceed to homepage
4	Forgot password	Click to reset password	No	Proceed to reset password view
5	Sign Up	Click to sign up a new customer account	No	Proceed to sign-up screen

- The Actor will successfully log in to the system when the phone number is available in the system and the password is matched.
- When Actor unsuccessfully logs in to the system, error messages will be displayed under the fields.

3.5.2 Sign Up

- **Function trigger:** The actor clicks on the “Sign Up” button under the “Login” button
- **Function description:** The actor fills out a sign-in form and submits to sign up a new account with inputted details
- **Screen Layout:**

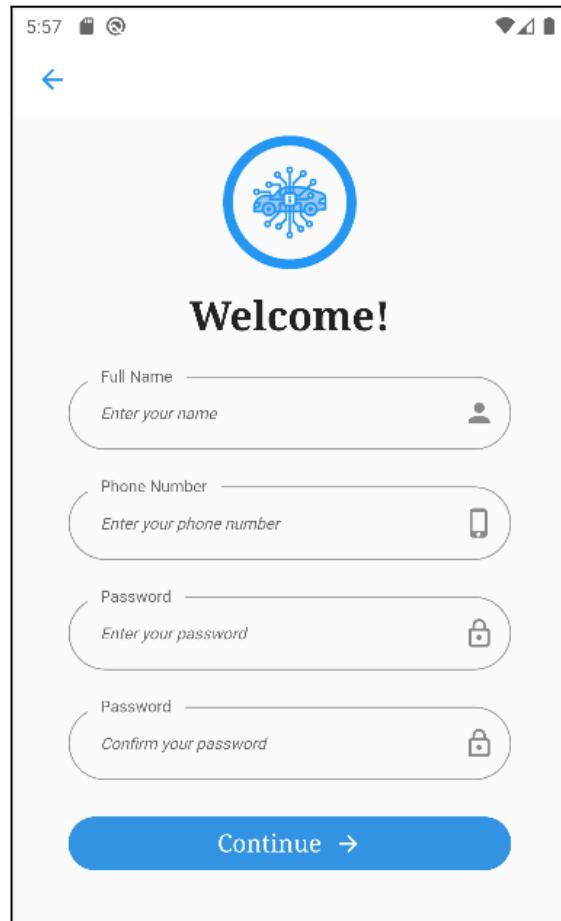


Figure 26. Sign Up screen

- **Function details:**

Fields:

#	Field name	Description	Read-only	Mandatory	Control type	Data type
1	Full name	Fill full name	No	Yes	Text box	String
2	Phone number	Fill phone number	No	Yes	Text box	Number
3	Password	Fill password	No	Yes	Password	String
4	Confirm password	Re-enter above password	No	Yes	Password	String

Buttons/Links:

#	Function	Description	Validation	Outcome
5	Continue	Click to sign up account	Yes	Proceed to Sign In screen

3.5.3 *Forgot password*

- **Function trigger:** Actor clicks on the “Forgot Password” link
- **Function description:** Actor enters phone number and receive OTP message
- **Screen Layout:**

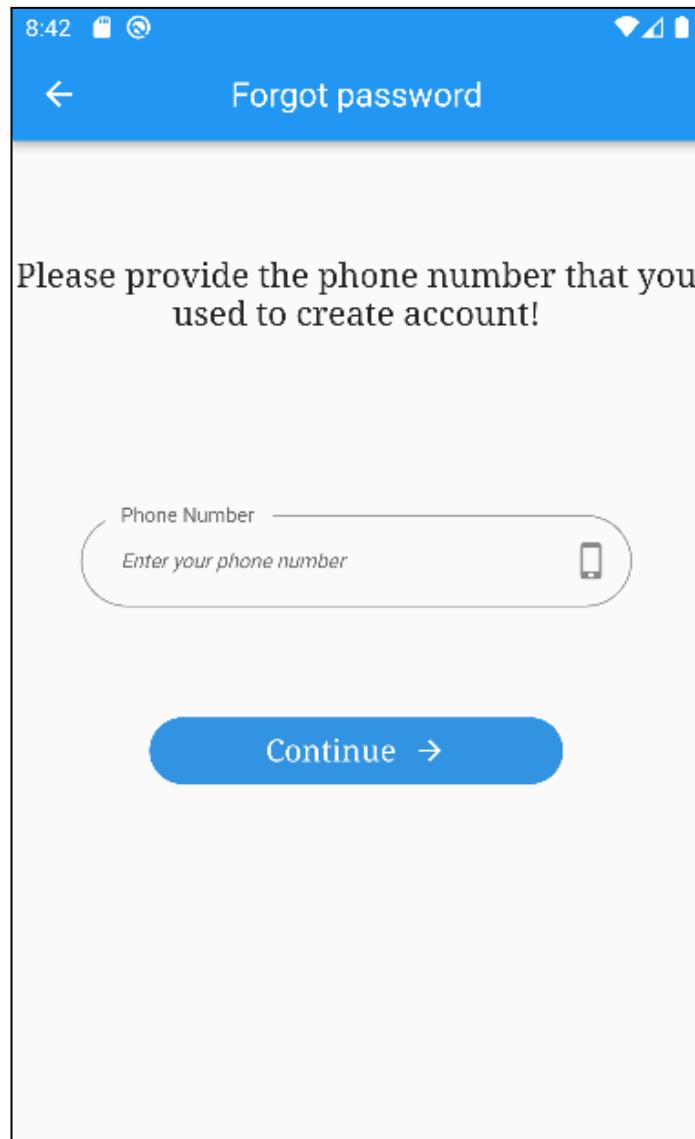


Figure 27. Forget Password screen

- **Function details:**

Fields:

#	Field name	Description	Read-only	Mandatory	Control type	Data type
1	Phone number	Fill phone number	No	Yes	Text box	Number

Buttons/Links:

#	Function	Description	Validation	Outcome
2	Continue	Click to confirm phone number	Yes	Proceed to OTP confirmation screen

3.5.4 OTP Verification

- **Function trigger:** Actor clicks on the “Forgot password” button above the “Continue” button
- **Function description:** System will send an OTP message to the Actor’s phone number for resetting password or signing up
- **Screen Layout:**

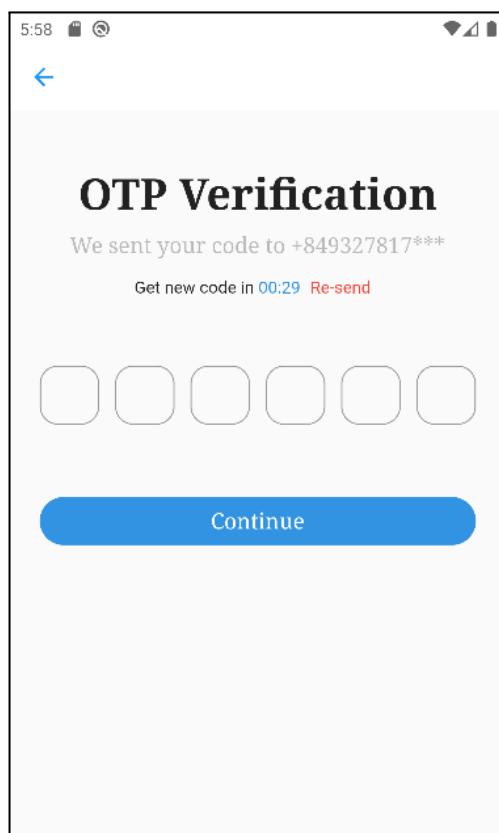


Figure 28. OTP Confirmation screen

- **Function details:**

- The system will check if the phone number exists in the system or not.
- If the phone number exists, the system will send an email containing a reset password OTP message to the inputted phone number.
- If the phone number does not exist, an error message will be displayed under the incorrect field.
- After clicking the reset password link, the Actor is required to input a new password for their account.

3.6 Authorization

3.6.1 Login

- **Function trigger:** Actor tries to access the <http://apms.ga> website
- **Function description:** Actor fills out a login form and submits to login account
- **Screen layout:**

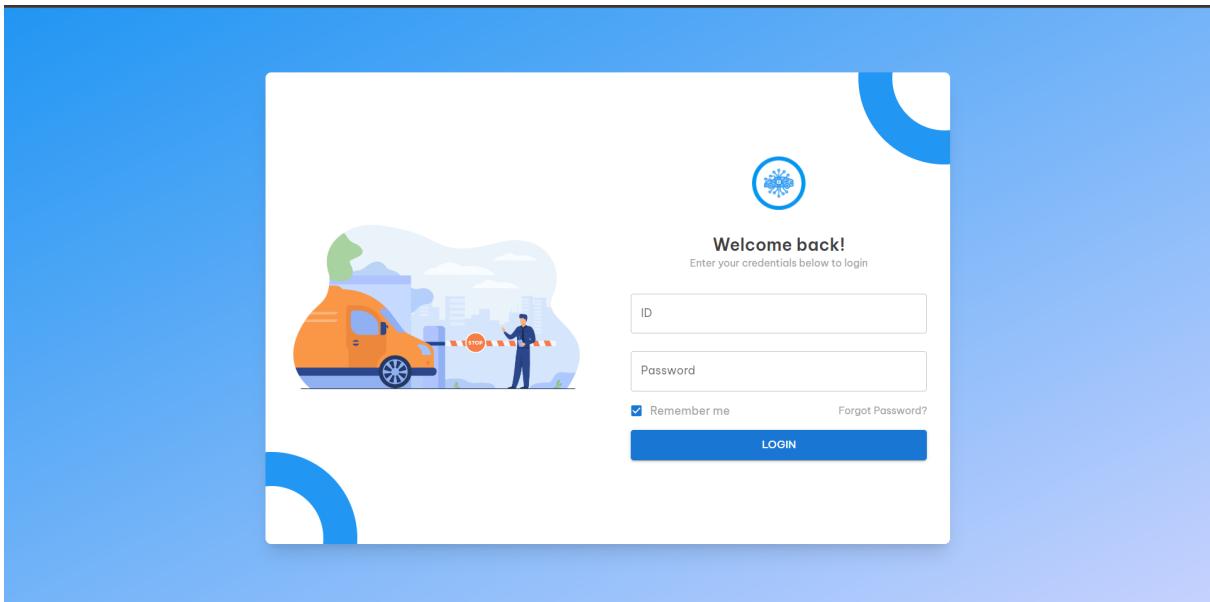


Figure 29. Login screen

- **Function details:**

Fields:

#	Field name	Description	Read only	Mandatory	Control type	Data type
1	ID	Fill account ID	No	Yes	Text box	String
2	Password	Fill Password	No	Yes	Password	String

Buttons/Links:

#	Function	Description	Validation	Outcome
3	Login	Click to login account	Yes	Proceed to homepage
4	Forgot password	Click to reset password	No	Proceed to reset password view

- The Actor will successfully log in to the system when their phone number and password are correct.
- When Actor unsuccessfully logs in to the system, error messages will be displayed under the fields.

3.6.2 Reset password

- **Function trigger:** Actor clicks on the “Forgot password” button above the “Login” button
- **Function description:** System will send a Reset Password OTP message to the Actor’s phone number so that the actor can reset his password
- **Screen Layout:**

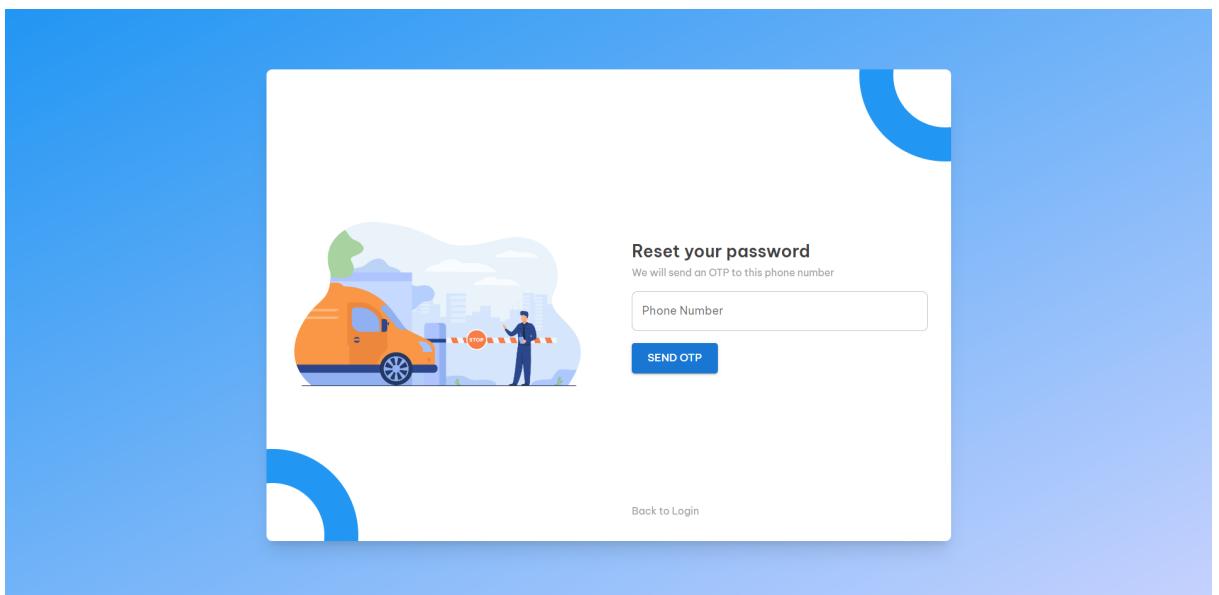


Figure 30. Reset Password screen

- **Function details:**
 - The system will check if the phone number exists in the system or not.
 - If the phone number exists, the system will send an email containing a reset password OTP message to the inputted phone number.

- If the phone number does not exist, an error message will be displayed under the incorrect field.
- After clicking the reset password link, the Actor is required to input a new password for their account.

3.7 Show admin reports

- **Function trigger:** Click on “Dashboard” on the left menu
- **Function description:** System displays total of users, booking, check-in; charts of system overview, employees, booking stats; list of booking details
- **Screen Layout:**

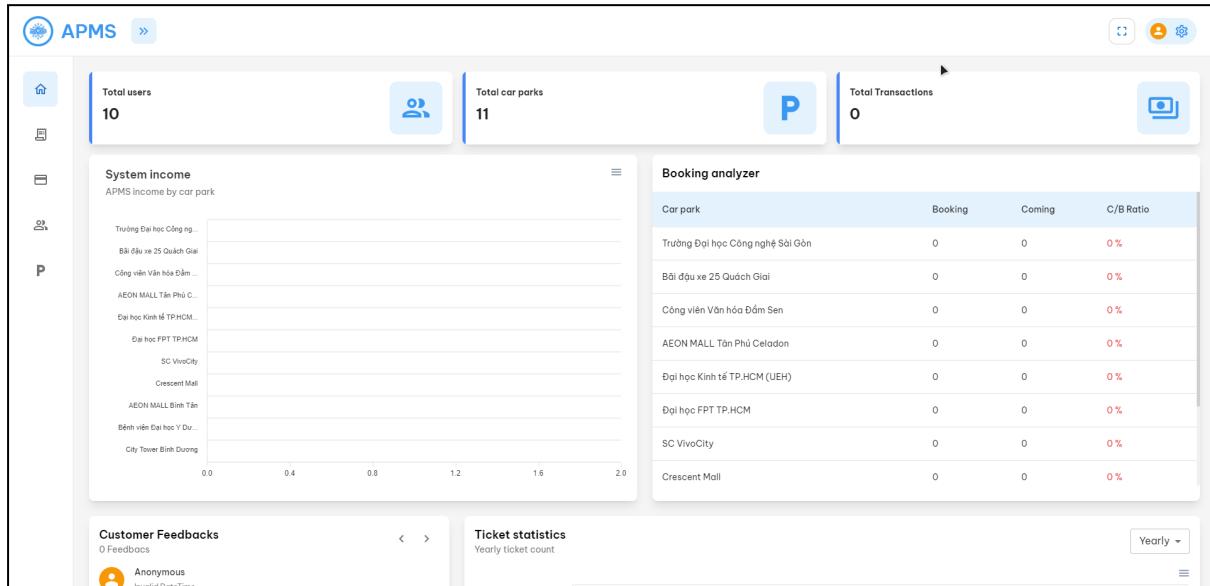


Figure 31. Admin Dashboard screen

3.8 Get car park list

- Function trigger:** Click on “Car parks” on the left menu
- Function description:** System displays a list of car parks currently in the system
- Screen Layout:**

#	Function	Description	Validation	Outcome
1	Add car park	Click to go to add car park form	No	Pop up a form to add a car park
2	Edit car park	Click to edit the car park	No	Display a form to edit the selected car park

Figure 32. Car Park List screen

- Function details:**

Buttons/Links:

#	Function	Description	Validation	Outcome
1	Add car park	Click to go to add car park form	No	Pop up a form to add a car park
2	Edit car park	Click to edit the car park	No	Display a form to edit the selected car park

3.9 Get ticket list

- **Function trigger:** Click on “Ticker List” on left menu
- **Function description:** System displays list of tickets currently in the system
- **Screen Layout:**

The screenshot shows the APMS Billing history screen. At the top, there are navigation icons and a search bar labeled "Search...". Below the header, there is a section titled "Billing history" with a subtitle "Parking history". Underneath this, there are two sets of date selection fields: "Start Date" and "End Date", each with a calendar icon. To the right of these fields is a search bar with the placeholder "Search plate number or phone number...". The main area is a table listing five parking tickets. The columns in the table are: #, Client, Car Park, Plate Number, Book Time, Arrive Time, Check In, Check Out, Amount, and Status. Each row contains details such as the client's name, car park location, plate number (with a VN flag), booking and arrival times, check-in and check-out status, amount paid, and the current status (Pending, Canceled, or Pending). The table has a light blue header and white rows.

Figure 33. Ticket List screen

- **Function details:**

Fields:

#	Field name	Description	Read only	Mandatory	Control type	Data type
1	Start Date	Choose start date to filter	No	No	Dropdown list /Text box	Date
2	End Date	Choose end date to filter	No	No	Dropdown list /Text box	Date
3	Search plate number or phone number	Fill plate number or phone number to search	No	No	Text box	String

Buttons/Links:

#	Function	Description	Validation	Outcome
4	View ticket details	Click to go to get ticketing details	No	Pop-up ticket details

3.10 Get transaction history

- **Function trigger:** Click on “Transaction history” on left menu
- **Function description:** System displays list of tickets currently in the system
- **Screen Layout:**

#	ID	Client	Time	Amount	Status
1	33bc3e45-ba30-4061-b3f9-83944a8dba63	G Luong Hoang Nhan 0123456789	November 21, 2022	90.110 ₫	success
2	32bc3e45-ba30-4061-b3f9-83944a8dba63	F Ho Huu phat 0123456789	November 22, 2022	176.595 ₫	success
3	31bc3e45-ba30-4061-b3f9-83944a8dba63	E Tran Khai Minh Khoi 0513456789	November 26, 2022	24.325 ₫	success
4	30bc3e45-ba30-4061-b3f9-83944a8dba63	D Khuc Ngoc Thai 0413456789	November 20, 2022	117.874 ₫	success
5	29bc3e45-ba30-4061-b3f9-83944a8dba63	C Tang Chi Cuong 0313456789	November 21, 2022	135.162 ₫	success

Figure 34. Transaction History screen

- **Function details:**

Fields:

#	Field name	Description	Read only	Mandatory	Control type	Data type
1	Transaction Date	Choose date to filter	No	No	Dropdown list /Text box	Date
2	Search by date, phone number or transaction id	Fill date or phone number or transaction id to search	No	No	Text box	String

Buttons/Links:

#	Function	Description	Validation	Outcome
4	View ticket details	Click to go to get ticketing details	No	Pop-up ticket details

3.11 Get employee list

- Function trigger:** Click on “Employees” on left menu
- Function description:** System displays list of employees currently in the system
- Screen Layout:**

The screenshot shows the APMS Employee List screen. At the top, there are navigation icons and a search bar labeled "Search something...". Below the header, it says "5 Employees" and "1 new hire". There are three category buttons: "1 admins", "2 owners", and "2 staffs". A blue button "+ ADD EMPLOYEE" is at the top right. The main area is a table with columns: #, ID ↑, Phone, Car Park, Role, and Status. Each row contains an edit icon (pencil) and a delete icon (X). The data in the table is as follows:

#	ID ↑	Phone	Car Park	Role	Status
1	ADM000001	111111111		Admin	Active
2	OWN000001	0902879231	Bãi đậu xe 25 Quách Giai	Owner	Active
3	OWN196734	0903750650	City Tower Bình Dương	Owner	Active
4	STF000001	0377162315	Bãi đậu xe 25 Quách Giai	Staff	Active
5	STF048757	0908750650	City Tower Bình Dương	Staff	Active

At the bottom, there are buttons for "Rows per page: All" and "1-5 of 5".

Figure 35. Employee List screen

- Function details:**

Fields:

#	Field name	Description	Read only	Mandatory	Control type	Data type
1	Search	Search for employee's id or phone number	No	No	Dropdown list /Text box	String
2	Filter	Choose a status filter	No	No	Dropdown list	

Buttons/Links:

#	Function	Description	Validation	Outcome
3	Edit employee	Click to edit employee information	No	Pop-up employee information
4	Deactivate employee	Click to deactivate an employee	Yes	Notify deactivating confirmation
5	Add employee	Click to add a new employee	No	Pop-up employee form

3.12 Show owner reports

- **Function trigger:** Click on “Dashboard” on left menu
- **Function description:** System displays monthly tickets and revenue; system overview; chart of revenue; transaction details
- **Screen Layout:**

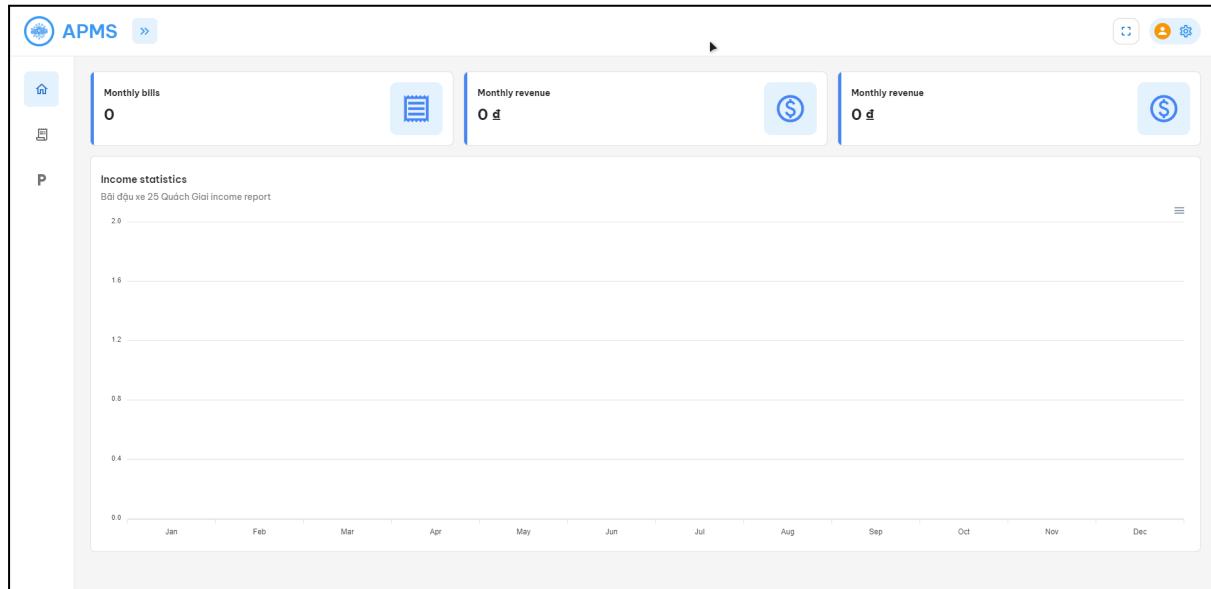
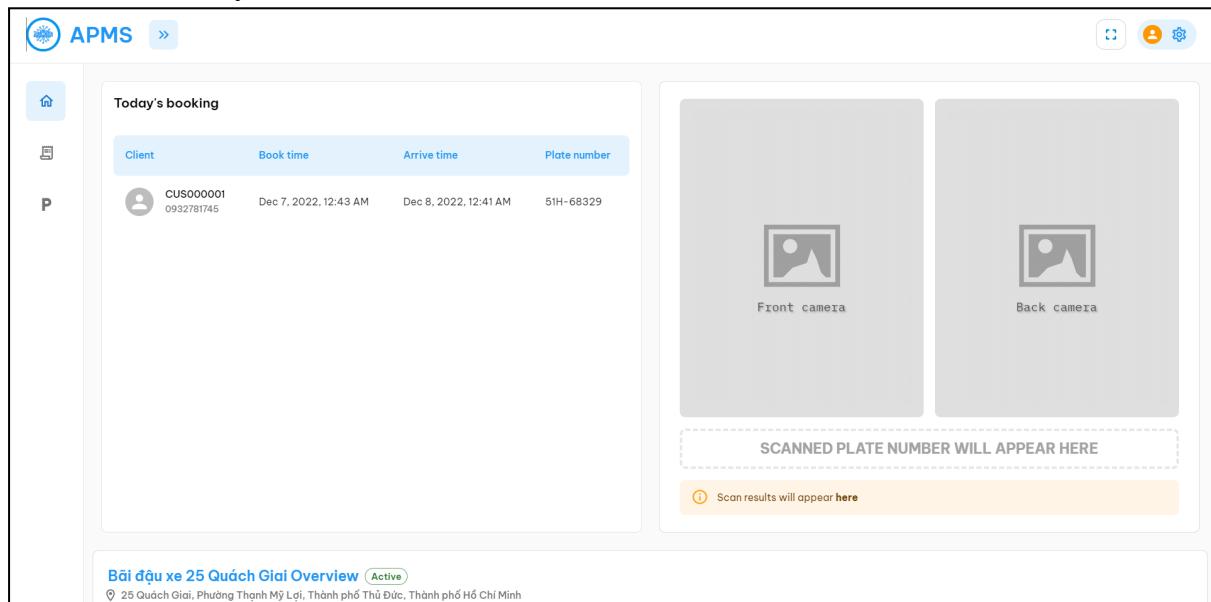


Figure 36. Owner Dashboard screen

3.13 Show staff's reports

- **Function trigger:** Click on “Dashboard” on left menu
- **Function description:** System displays booking list of current day, scanned plate and car park overview
- **Screen Layout:**



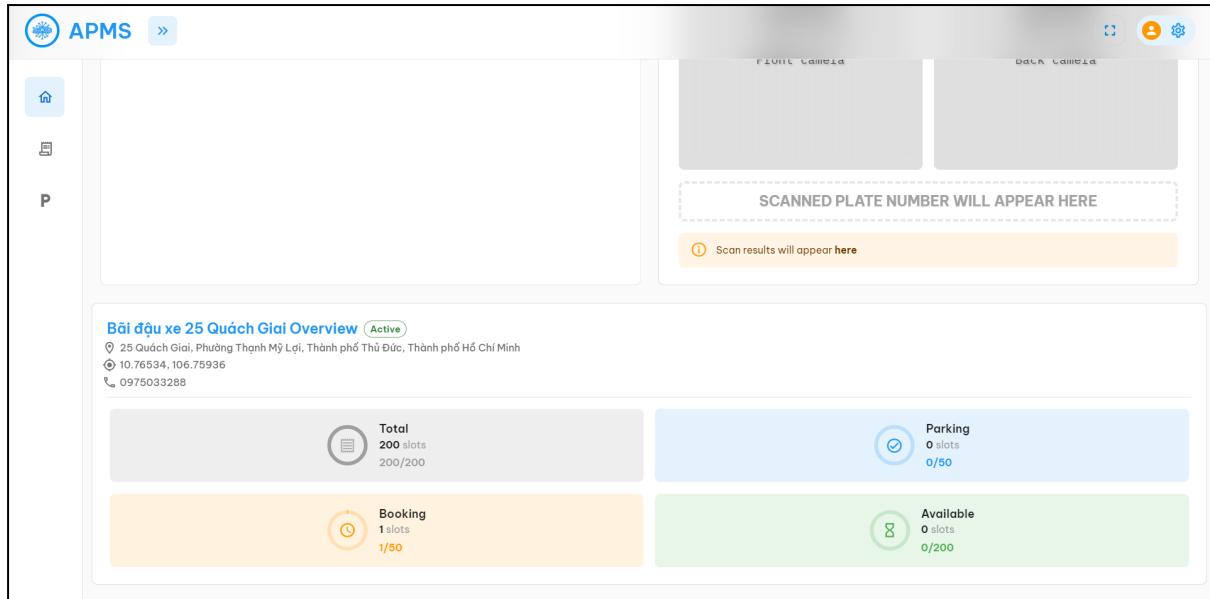


Figure 37-38. Staff Dashboard screen

3.14 Manage car park

- Function trigger:** Click on “My park” on left menu
- Function description:** System displays list of employees of the car park. There are 2 forms to view and update fee and car park’s information
- Screen Layout:**

The screenshot shows the APMS My Park screen. It includes sections for managing staff and car park information:

- Staff Management:** Shows two staff profiles with ID, phone, and role.
- Car park's information:** Details for 'Bãi đậu xe 25 Quách Giai' including address, province, district, ward, phone, capacity, latitude, and longitude.
- Fee & Pricing:** Settings for reservation percentage (30%), start and end times, and fees (30000, 32000).
- Buttons:** UPDATE CAR PARK, UPDATE PRICE TABLE, and + ADD TIME FRAME.

Figure 39. My Park screen

- **Function details:**

Fields:

#	Field name	Description	Read only	Mandatory	Control type	Data type
1	Reservation percentage	Fill Reservation percentage	No	Yes	Text box	Number
2	Start time	Fill or choose start time	No	Yes	Clock/Text box	String
3	End time	Fill or choose end time	No	Yes	Clock/Text box	String
4	Fee	Enter fee for the time frame	No	Yes	Text box	Number
5	Name	Enter car park's name	No	Yes	Text box	String
6	Address number	Enter car park's address number	No	Yes	Text box	String
7	Street	Enter car park's street	No	Yes	Text box	String
8	Province	Choose a province	No	Yes	Dropdown/Text box	String
9	District	Choose a district	No	Yes	Dropdown/Text box	String
10	Ward	Choose a ward	No	Yes	Dropdown/Text box	String
11	Phone	Enter car park's phone number	No	Yes	Text box	Number
12	Capacity	Enter car park's slot capacity	No	Yes	Text box	Number
13	Latitude	Enter Latitude	No	Yes	Text box	Number
14	Longitude	Enter Longitude	No	Yes	Text box	Number

Buttons/Links:

#	Function	Description	Validation	Outcome
15	Remove time frame	Click to remove time frame	No	Remove an existed time frame
16	Update	Click to update the car park's time for the time frame	Yes	Update the time frame based on Actor input
17	Add time frame	Click to create a new time frame	Yes	Create a new line of form for new time frame
18	Update car park	Click to update the car park's information	Yes	Update the car park based on Actor input

4. Non-Functional Requirements

4.1 External Interfaces

- File Storage Service: The system can be integrated with Firebase Storage.
- The system can use HTTP to transfer data.

4.2 Quality Attributes

4.2.1 Usability

- The web application for staff and car park owners should require no more than one day of training to be used
- The mobile application for customers should be used without training

4.2.2 Reliability

- The license plate recognition system should be 92% or higher under normal conditions (the number plate must comply with the Vietnamese law and completely visible, under good lighting condition)

4.2.4 Compatibility

- The mobile application should be compatible with Android 8 and higher
- The web applications should work well with Chrome Browser version 65 and higher, Firefox 52 and higher, Microsoft Edge version 91 or higher

4.2.5 Security

- All input data are validated before being saved to the database
- All passwords will be hashed using a combination of SHA256 and salt
- The system continuously checks for authentication and authorization before executing any features.

5. Requirement Appendix

5.1 Business Rules

ID	Rule Definition
BR-01	Only car with a Vietnamese license plate can be registered
BR-02	Only staff can modify check-in/check-out requests in some special conditions

BR-03	Any issues happen in the parking process that are not related to our system will not be considered as our responsibility
BR-04	Reservation fee is not refundable unless the ticket is canceled by staff
BR-05	The booking will be automatically canceled after 30 minutes of the registered arrival time if no check-in request for the ticket is recorded, and considered as a customer cancellation request
BR-06	If the customer actual arrival time exceeds the registered arrival time by more than 5 minutes and less than 30 minutes, an overdue fee will be added to the total parking fee
BR-07	Overdue fee is the parking fee of one hour
BR-08	Customers' mobile device must be connected to the internet in order to use the system
BR-09	Mobile device must support QR code scanning
BR-10	Parking time is rounded up to hour
BR-11	When the parking fee of a car park is changed, the parking fee of all tickets of that car park created before the change will not be affected
BR-12	When the available slots count equals 0, only the customer who booked a parking slot in advance can check-in
BR-13	The reservation fee must be paid at the time that the customer books a parking slot
BR-14	Customers can only book a parking slot within 24 hours prior to the check-in time
BR-15	One license plate can only be checked in at one car park at a time
BR-16	The parking fee is calculated based on the actual check-in time and check-out time
BR-17	Reservation fee is calculated by the percentage of parking fee calculated from the time of the booking to the scheduled check-in time, set by the owner of the car park
BR-18	Parking fee can only be paid using application's electronic wallet
BR-19	Customers cannot book a slot if their account balance is not enough for the reservation fee
BR-20	Customers cannot check-out if their account balance is not enough for the parking fee
BR-21	The system requires Internet connection and electricity to function normally
BR-22	The revenue earned by the car park (minus the system operation fee) will be paid to the owner monthly (the 1st day of every month)
BR-23	After a top-up transaction is made, a refund is not eligible under any circumstances
BR-24	Only USD is acceptable for a top-up transaction. The conversion rate from VND to USD will be fetched from Forex (https://www.forex.com)
BR-25	The system does not support transferring money from one account to another

5.2 Application Messages List

#	Message code	Message Type	Context	Content
1	MSG01	In red, under the text box	Wrong confirmation password	<i>Confirmation password not match</i>
2	MSG02	In red, under the text box	The * field is required.	<i>Input-required fields are empty</i>
3	MSG03	In red, above submit button	Wrong OTP code	<i>Confirmation code is incorrect</i>
4	MSG04	Toast message	Cancel a booking successfully	<i>Cancel booking successfully</i>
5	MSG05	Toast message	Create new password successfully	<i>Create a new password successfully. Please log in with newly created password</i>
6	MSG06	In red, under the text box	Validate phone number	<i>Phone number needs to be from 9 to 10 numbers</i>
7	MSG07	In red, under the text box	Validate password	<i>Password needs to be from 6 to 10 characters</i>
8	MSG08	In red, under the text box	Validate confirm password	<i>Confirm password does not match</i>
9	MSG09	In red, above submit button	No account found to reset password	<i>No account found with given phone number</i>
10	MSG10	Toast message	Update password successfully	<i>Your password has been changed</i>
11	MSG11	In red, above submit button	Invalid login credential	<i>Incorrect phone number or password!</i>
12	MSG12	In red, under the text box	Validate name	<i>Name needs to be at least 3 characters</i>
13	MSG13	Toast message	Send feedback successfully	<i>Thank you for your contribution</i>
14	MSG14	In red, under the text box	Validate feedback's description	<i>Please enter your opinion in the field above!</i>
15	MSG15	Popup	Confirm plate number for check in	<i>Is this the license plate you want to check-in: {plate number}</i>

16	MSG16	Popup	Confirm plate number for check out	<i>Is this the license plate you want to check-out:{plate number}</i>
17	MSG17	Toast message	Check-in failed	<i>Check-in Failed</i>
18	MSG18	Toast message	Check-out failed	<i>Check-out Failed</i>

IV. Software Design Description

1. System Design

1.1 System Architecture

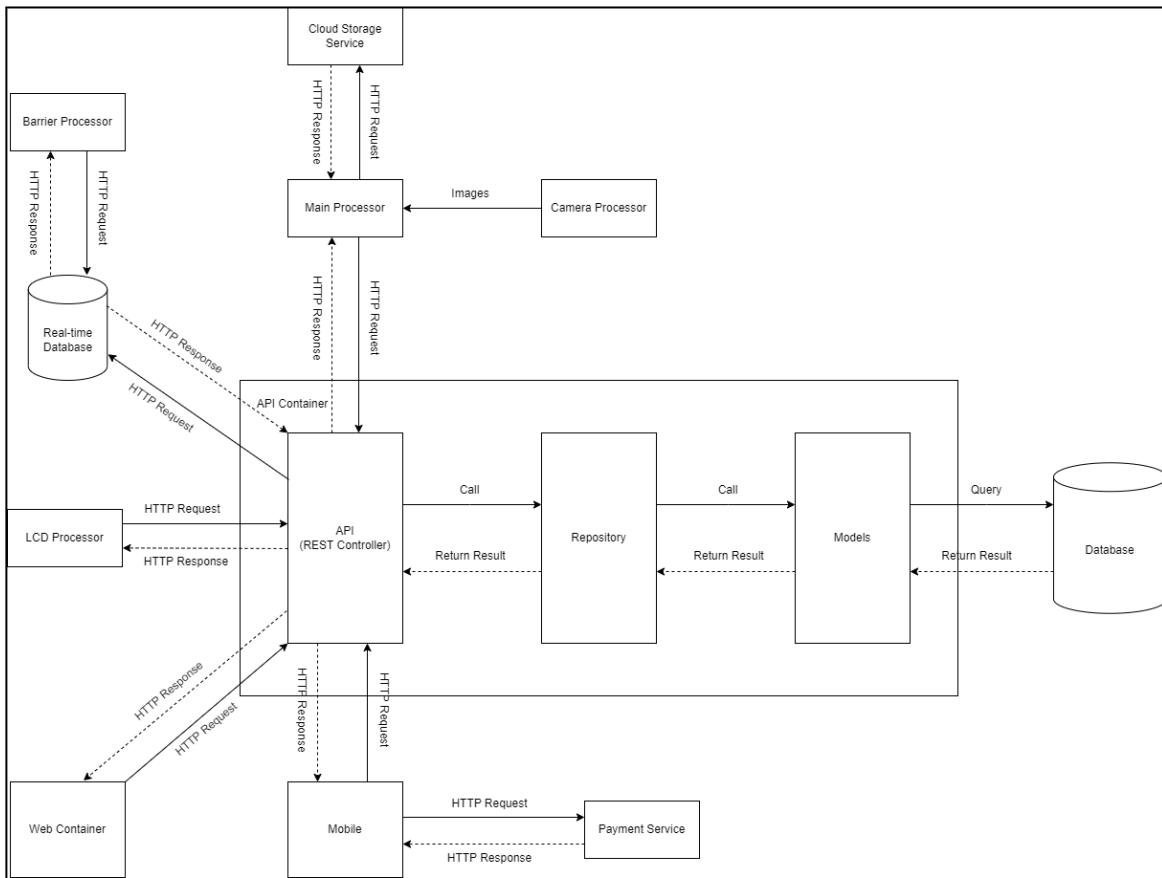


Figure 40. System architecture overview diagram

1.2 Package Diagram

1.2.1 Back-end

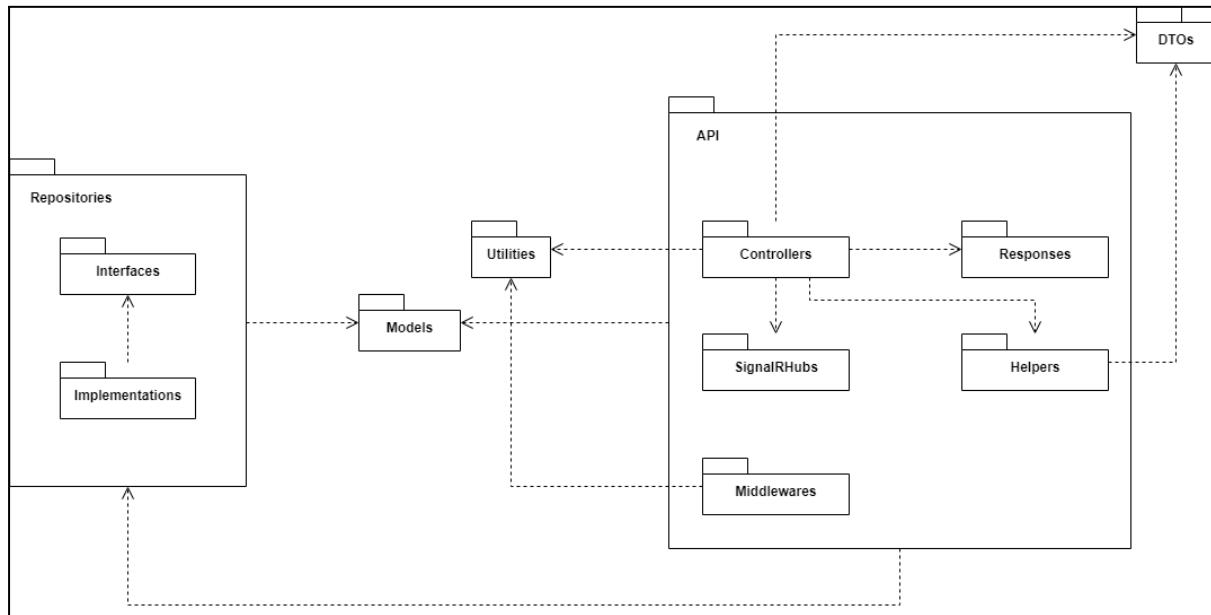


Figure 41. Package diagram (Backend)

Ref: [Unified Modeling Language, v2.5.1 \(omg.org\)](http://www.omg.org/spec/UML/2.5.1/)

Package Descriptions

No	Package	Description
01	Models	A project that contains entity models that match database tables and a DbContext class to access the database
02	Repositories	A project that contains classes and interfaces that manipulate data in the database
03	Interfaces	Contained inside the Repositories project, contains interfaces for data manipulation classes
04	Implementations	Contained inside the Repositories project, contains classes that implement the interfaces contained in the Interfaces packages
05	Utilities	Contained classes that perform password hashing and other application constants
06	DTOs	Contained contracts that are mapped from the entity classes to be returned in the controllers

07	API	Contains controllers, middlewares, services that help send data to clients and process data from clients
08	Controllers	Contained in the API project. Contains classes and controllers that receive data from clients, manipulate and return data to the clients
09	SignalRHubs	Contains SignalR Hubs to execute real-time functionalities
10	Middlewares	Contains custom middlewares of the API project
11	Responses	Contains classes that represent the results returned from the API to the clients
12	Helpers	Contains classes that provide some utilities for the API project

1.2.2 Front-end

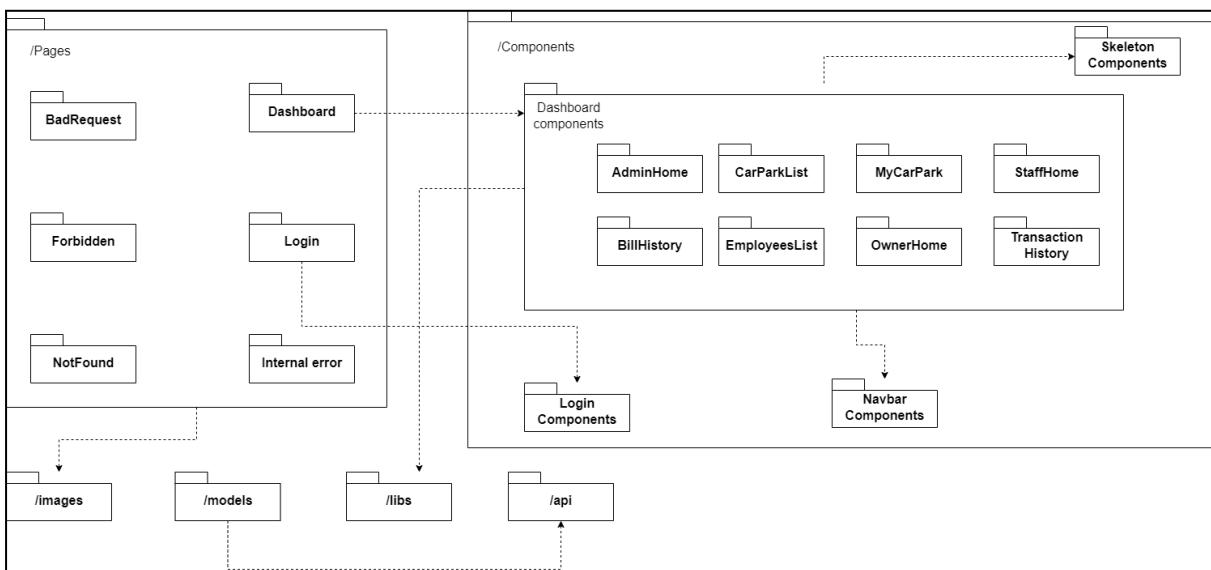


Figure 42. Package diagram (Frontend)

Package Descriptions

No	Package	Description
01	Pages	Root folder for all pages of the website
02	Dashboard	Folder contains all dashboard tabs and serves as a skeleton for the website
03	Login	Folder contains login page to authenticate and authorize users
04	Components	Root folder contains all components which are used throughout the website

05	Images	Contains placeholder and images for the website
06	Models	Contains models for redux global state management
07	Libs	Utilities and tools for formatting data
08	Api	Settings and constants for calling back-end API

1.2.3 Mobile

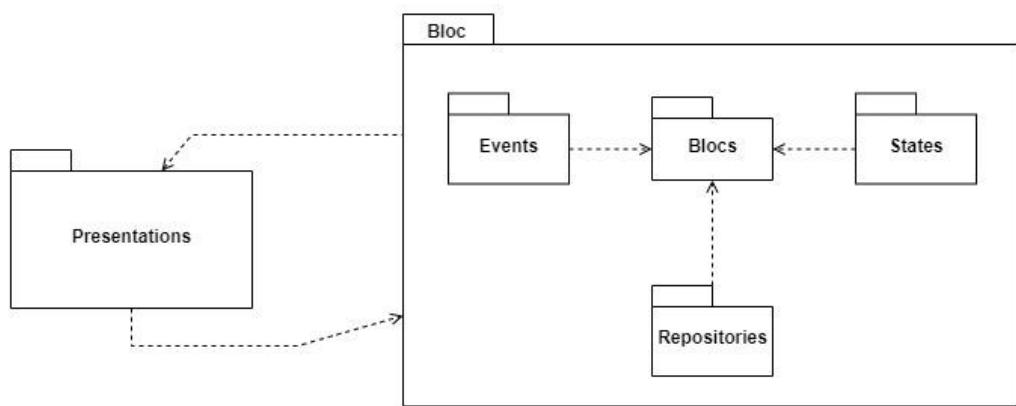


Figure 43. Package diagram (Mobile)

No	Package	Description
01	Presentations	Contains all screen for user to interact and represent data from database
02	States	Represent all states that objects can have in the application
03	Events	Actions can occur in the application
04	Repositories	Intermediary for reading, writing database
05	Blocs	Connect 3 packages : states, event, repositories; let the application know which event will return which state and when to operate actions with the database.

1.2.4 IoT

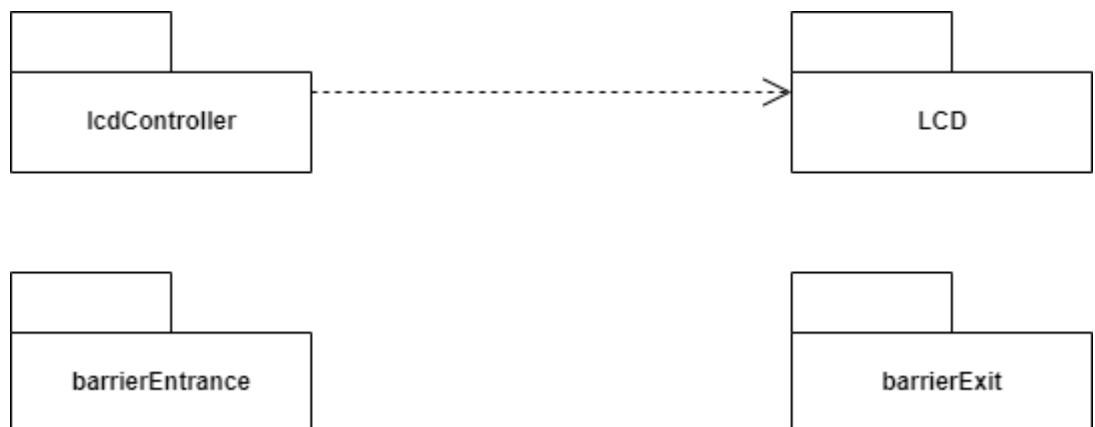


Figure 44. Package diagram (IoT)

Package Descriptions

No	Package	Description
01	LCD	Contains code to display data on the LCD screen
02	lcdController	Contains code to fetch data from API and display the data onto the LCD screen
03	barrierEntrance	Contains code that controls the barrier at the entrance
04	barrierExit	Contains code that controls the barrier at the exit

2. Database Design

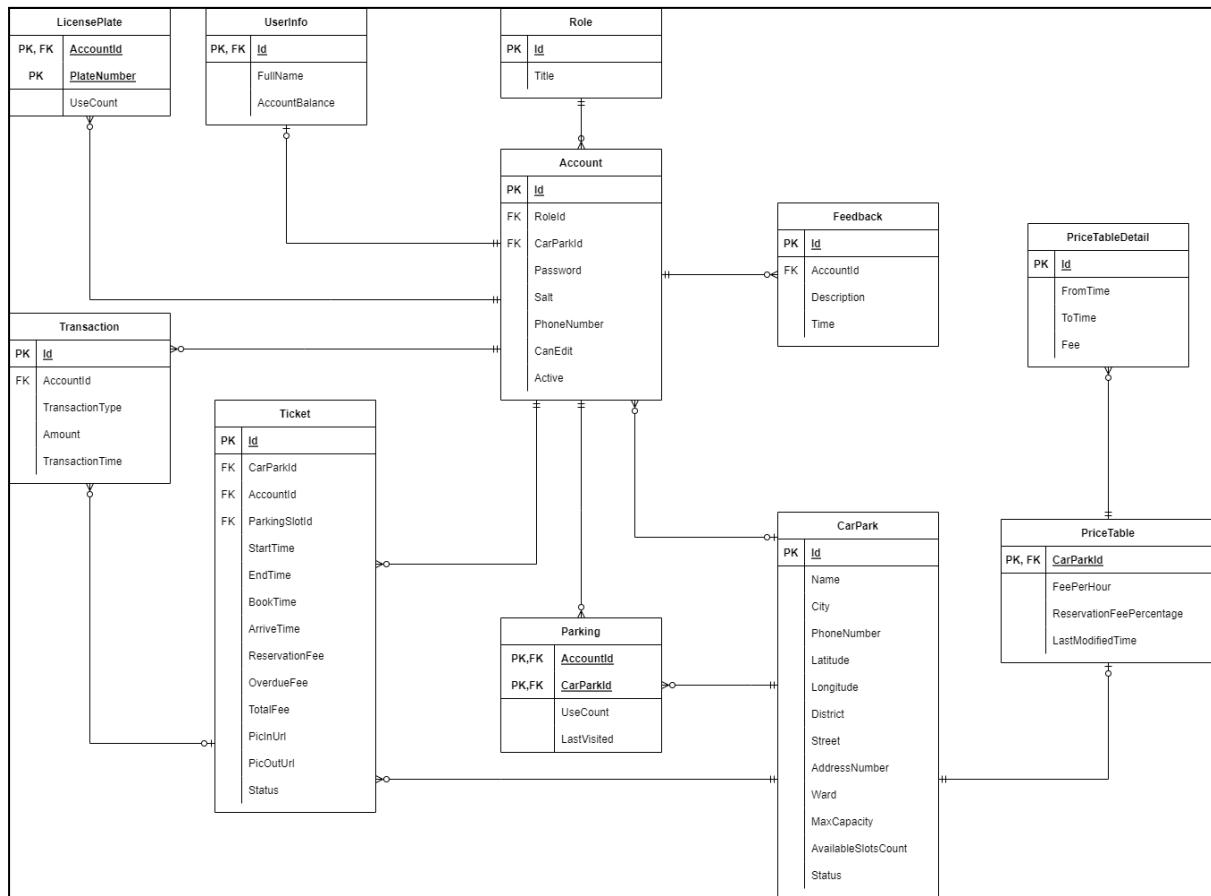


Figure 45. Database diagram

Link to diagram: [Database Diagram](#)

Table Descriptions

No	Table	Description
01	CarPark	<p>Includes all information of a car park</p> <ul style="list-style-type: none"> - Fields: <ul style="list-style-type: none"> + Id: ID of the car park, generated by the system + Name + Address number + Street + Ward + District + City + Latitude: latitude of the car park on the map, used to calculate the distance between customer's current destination to the car park + Longitude: longitude of the car park on the map, used to calculate the distance between customer's current destination to the car park + PhoneNumber + MaxCapacity: Maximum slots count of the car park

		<ul style="list-style-type: none"> + AvailableSlotsCount: current empty slots of the car park + Status: status of the car park, active or removed - Primary keys: Id - Foreign keys: None
02	PriceTable	<p>Each car park has a price table to calculate reservation fee and parking fee</p> <ul style="list-style-type: none"> - Fields: <ul style="list-style-type: none"> + CarParkId: ID of the car park of the price table + ReservationFeePercentage: percentage of the parking to calculate the reservation fee of each tickets + FeePerHour: parking fee per hour + LastModified: the last time the price table is modified - Primary key: CarParkId - Foreign key: CarParkId
03	PriceTableDetail	<p>If owner of the car park does not want to set a fixed parking fee, they can create price table details to calculate parking fee by a time interval</p> <ul style="list-style-type: none"> - Fields: <ul style="list-style-type: none"> + CarParkId: ID of the car park of the price table + FromTime: starting time of the time interval + ToTime: ending time of the time interval + Fee: parking fee in this time interval - Primary key: CarParkId - Foreign key: CarParkId
04	Ticket	<p>Includes all information of a parking ticket</p> <ul style="list-style-type: none"> - Fields: <ul style="list-style-type: none"> + Id: ID of the ticket + CarParkId: ID of the car park the ticket belongs to + AccountId: ID of the customer account the ticket belongs to + PlateNumber: plate number of the vehicle used to book or check in + BookTime: the time customer books a slot + ArriveTime: scheduled time of arrival set by the customer + StartTime: actual check-in time + EndTime: check-out time + PicInUrl: link to the image of the vehicle, captured when customer checks in + PicOutUrl: link to the image of the vehicle, captured when customer checks out - ReservationFee: fee applied when customer books a slot - TotalFee: total parking fee, calculated when customer checks out - PriceTable: price table of the ticket's car park at the time of creation - OverdueFee: fee charged when the actual check in time is later than the scheduled arrival time - CancellerAccountId: ID of the account who cancels the ticket - Status: status of the ticket: pending, checked-in, checked-out or canceled

		<ul style="list-style-type: none"> - Primary key: Id - Foreign keys: CarParkId, AccountId
05	Account	<p>Includes all information of a user of the account</p> <ul style="list-style-type: none"> - Fields: <ul style="list-style-type: none"> + Id: ID of the account, is generated automatically based on role, is used to authentication by admin, staff and owner + Password: password of the account, is used for authentication + Salt: generated by the system, used to enhance security of the system, is unique for each account + PhoneNumber: phone number of the user, unique for each account, is used for authentication by customer + RoleId: ID of the role of the account + CanEdit: represents whether the account is allowed to edit and create tickets of their own car park + CanEditCarPark: represents whether the account is allowed to edit car park, staff and owner can only edit their own car park + CarParkId: ID of the car park the account is working at, applied for staff and owner + Active: represents whether the account is allowed to access the system or not - Primary key: Id - Foreign key: CarParkId, RoleId
06	UserInfo	<p>Includes full name and account balance of the account who has the customer role</p> <ul style="list-style-type: none"> - Fields: <ul style="list-style-type: none"> + Id: ID of the account + FullName: full name of the customer + AccountBalance: account balance of the account, is used to pay for parking and reservation fee - Primary key: Id - Foreign key: Id
07	LicensePlate	<p>Represents license plate that an account has used to book or check in at a car park</p> <ul style="list-style-type: none"> - Fields: <ul style="list-style-type: none"> - AccountId - PlateNumber: number of the license plate account has used - UsageCount: the number of times the account has used the plate number - Primary keys: PlateNumber, AccountId - Foreign key: AccountId
08	Parking	<p>Represents the number of times the account has parked at a car park</p> <ul style="list-style-type: none"> - Fields: <ul style="list-style-type: none"> + AccountId + CarParkId + UseCount: the number of times the account has parked at the car park - Primary keys: CarParkId, AccountId

		<ul style="list-style-type: none"> - Foreign keys: CarParkId, AccountId
09	Transaction	<p>Includes all information of a transaction, created whenever the user tops up their balance, books a slot or checks out at a car park</p> <ul style="list-style-type: none"> - Fields: <ul style="list-style-type: none"> - Id: ID of the transaction - TransactionTime: the time the transaction is made - AccountId: ID of the account who makes the transaction - TransactionType: type of the transaction: Top-up, booking, parking fee - TicketId: ID of the ticket the transaction is paying for - Primary key: Id - Foreign keys: AccountId, TicketId
10	Feedback	<p>Represents the feedback the users create about the system</p> <ul style="list-style-type: none"> - Fields: <ul style="list-style-type: none"> + Id: ID of the feedback + Description: what the feedback is about + Time: the time the feedback is created + AccountId: ID of the account who makes the feedback - Primary key: Id - Foreign key: AccountId
11	Role	<p>Represents the role of each account. Used for authorization, determine the account can access which functions</p> <ul style="list-style-type: none"> - Fields: <ul style="list-style-type: none"> + Id: ID of the role + Title: title of the role - Primary key: Id

3. Detailed Design

3.1 Check-in

3.1.1 Class Diagram

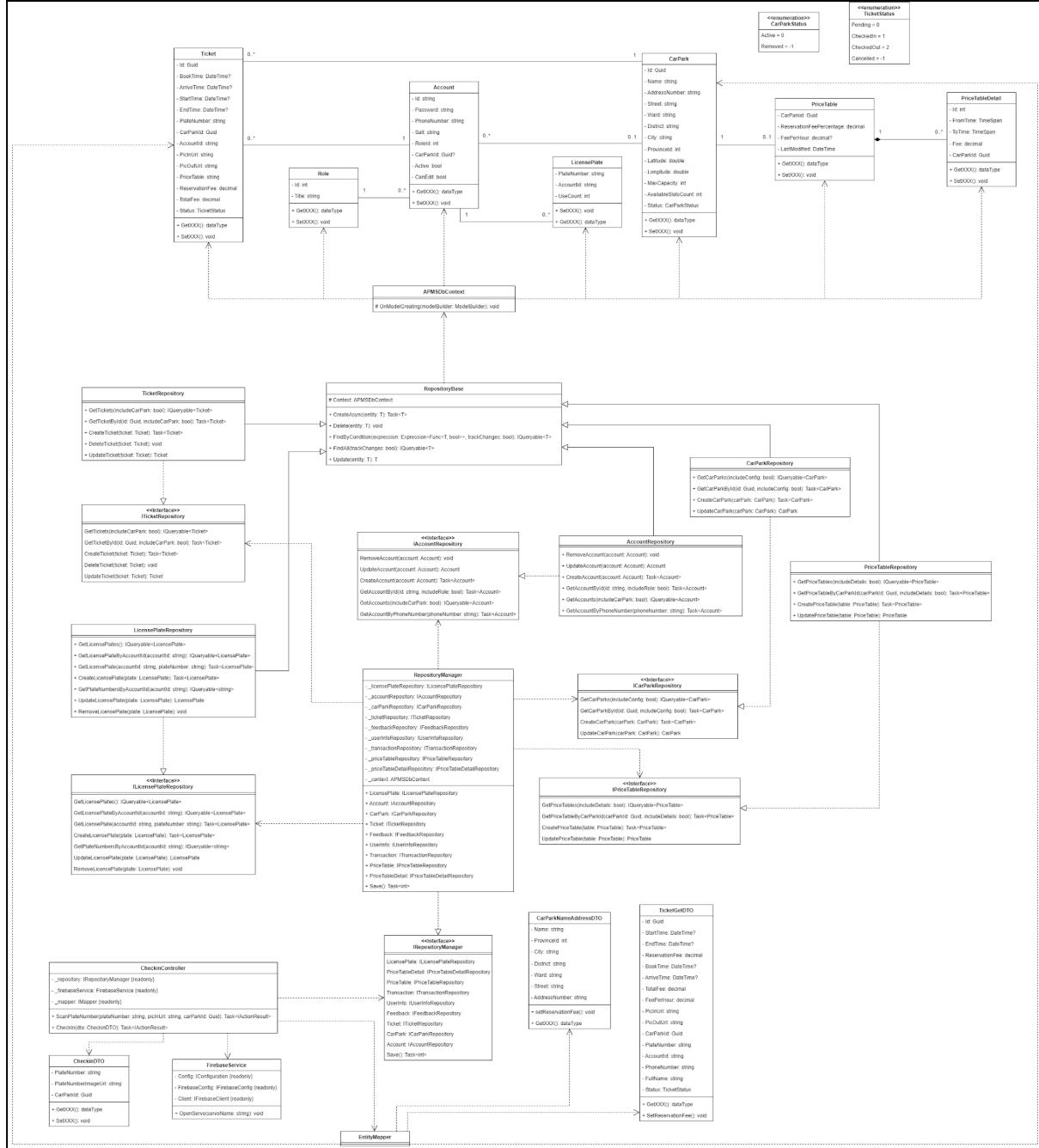


Figure 46. Class diagram

Ref: [Unified Modeling Language, v2.5.1 \(omg.org\)](http://www.omg.org/spec/UML/2.5.1/)

Link to diagram: [Check-in Class Diagram](#)

3.1.2 Sequence Diagram

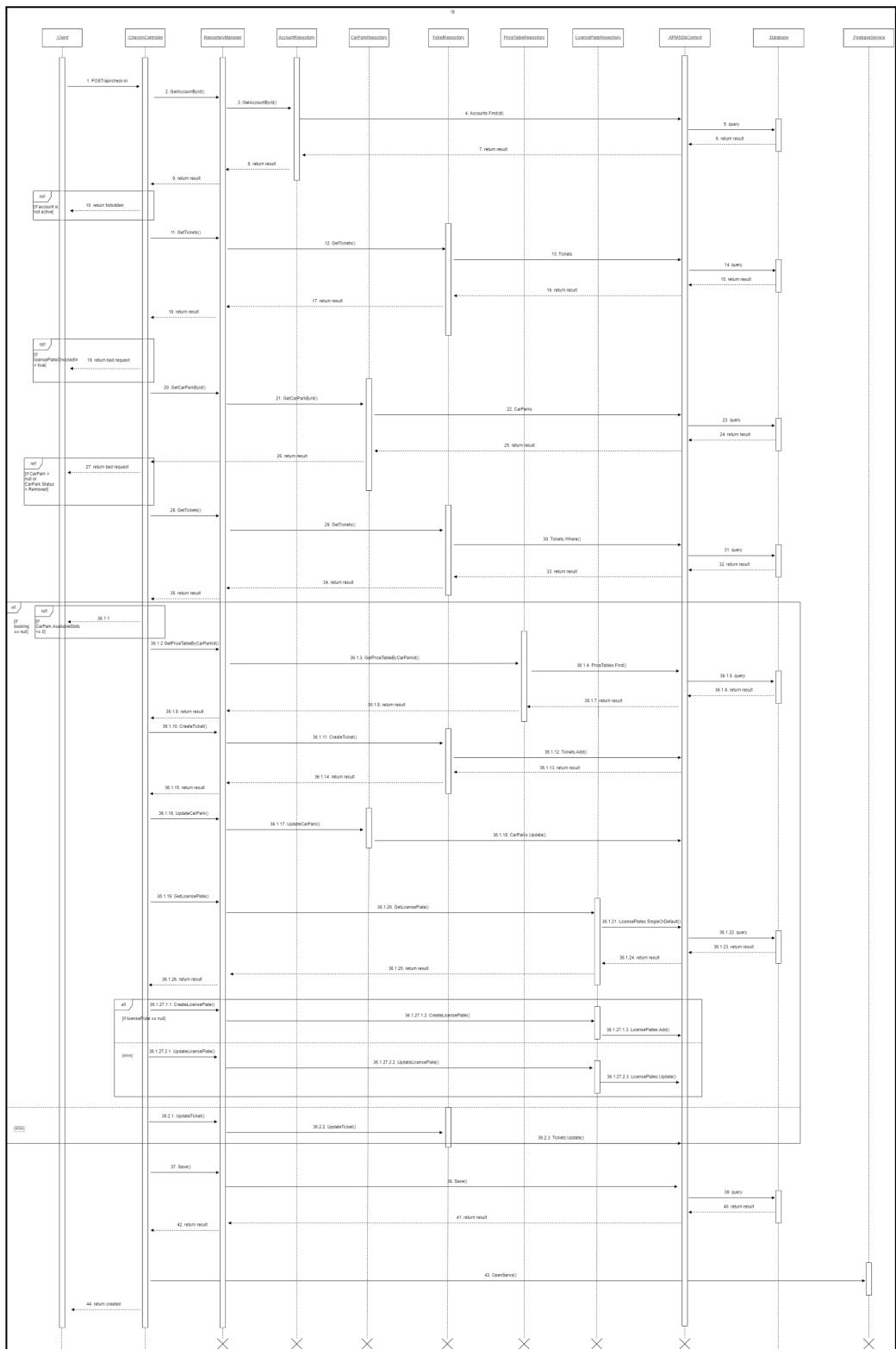


Figure 47. Check-in sequence diagram

Ref: [Unified Modeling Language, v2.5.1 \(omg.org\)](http://www.omg.org)

Link to diagram: [Check-in Sequence Diagram](#)

3.2 Check-out

3.2.1 Class Diagram

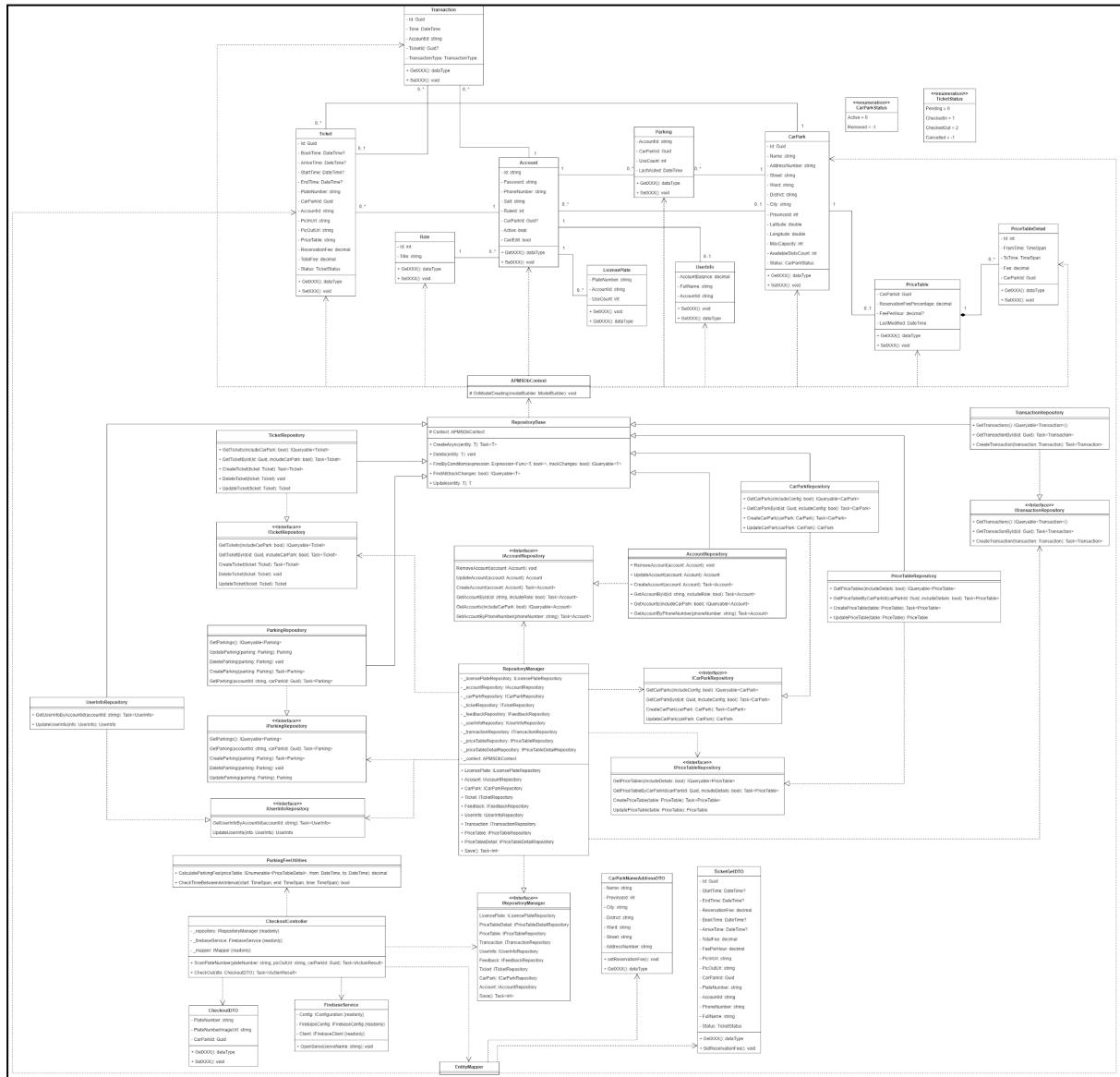


Figure 48. Check-out class diagram

Ref: [Unified Modeling Language, v2.5.1 \(omg.org\)](#)

Link to diagram: [Check-out Class Diagram](#)

3.2.2 Sequence Diagram

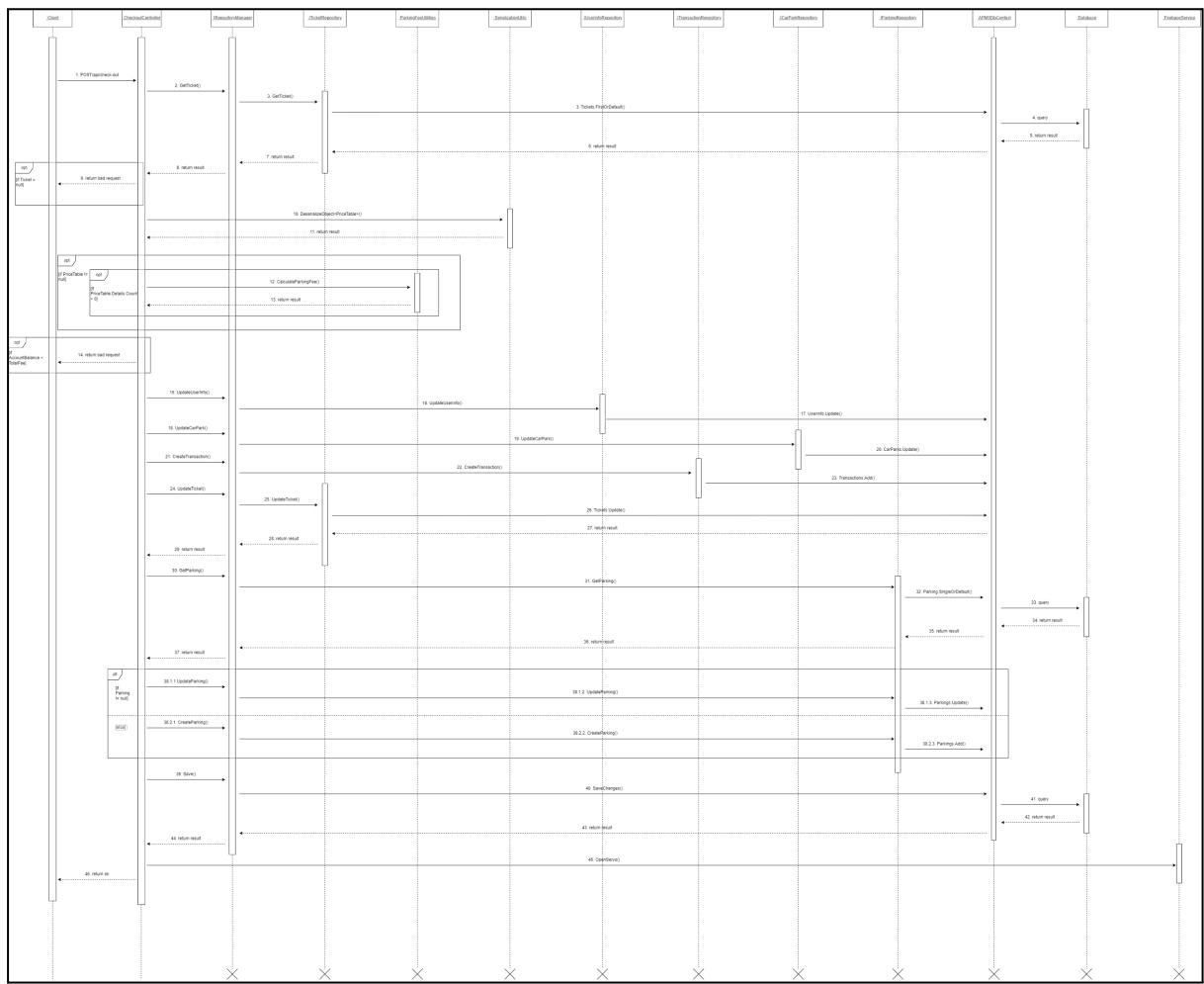


Figure 49. Check-out sequence diagram

Ref: [Unified Modeling Language, v2.5.1 \(omg.org\)](#)

Link to diagram: [Check-out Sequence Diagram](#)

3.3 Booking

3.3.1 Class Diagram

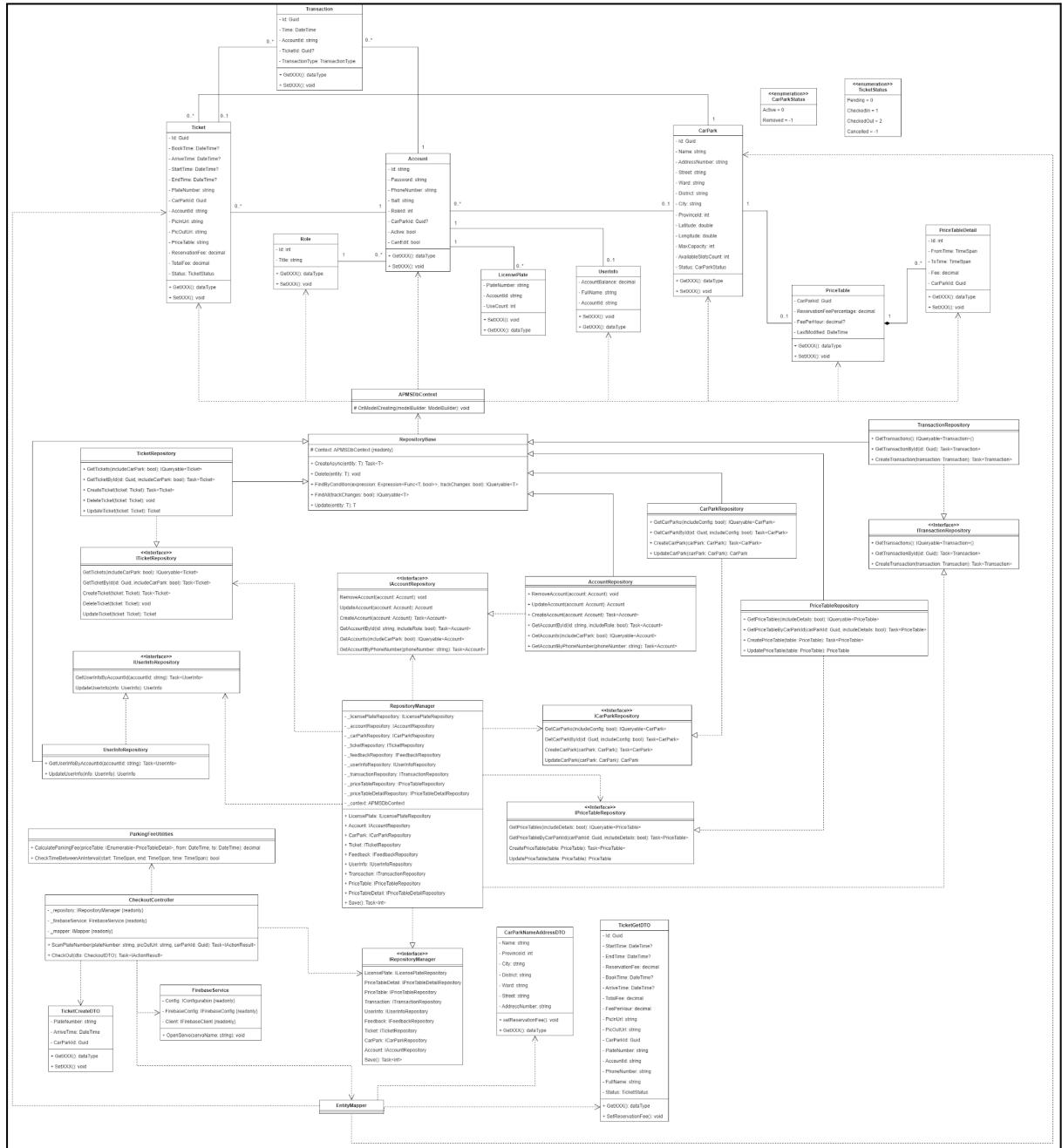


Figure 50. Booking class diagram

Ref: [Unified Modeling Language, v2.5.1 \(omg.org\)](http://www.omg.org/spec/UML/2.5.1/)

Link to diagram: [Booking Class Diagram](#)

3.3.2 Sequence Diagram

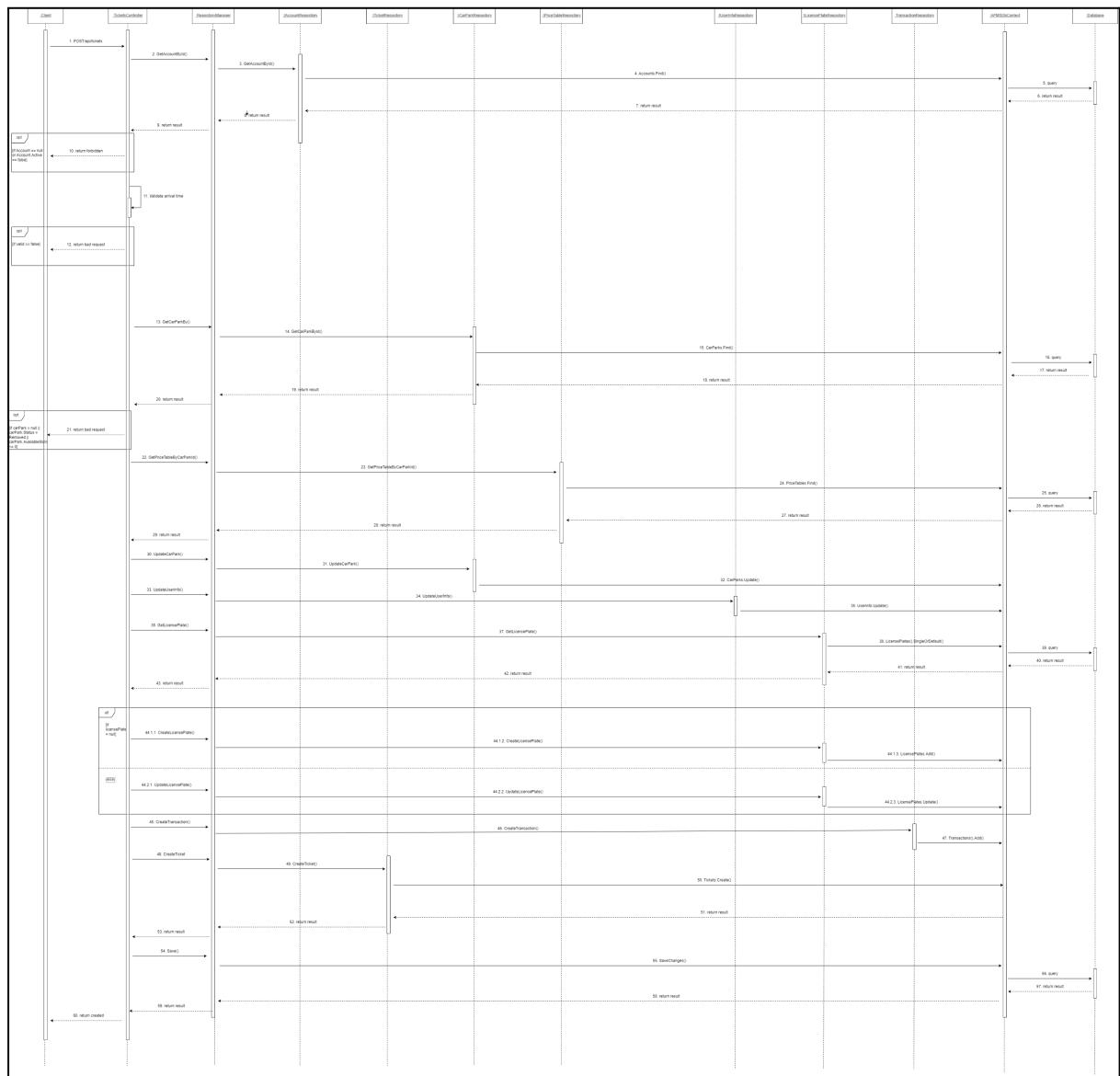


Figure 51. Booking sequence diagram

Ref: [Unified Modeling Language, v2.5.1 \(omg.org\)](http://www.omg.org)

Link to diagram: [Booking Sequence Diagram](#)

V. Software Testing Documentation

1. Scope of Testing

The target will mainly focus on the functions related to the web and mobile application's features and nonfunctional requirements, including but not limited to:

- The functions listed in the functional requirements above (section 3 part III)
- Non-functional requirements listed in the non-functional requirements above (section 4 part III)
- Image processing and IoT related features

2. Test Strategy

2.1 Testing Types & Levels

- Test types: Functional testing
- Test levels: System testing

2.2 Supporting Tools

- Mobile application: Samsung Galaxy S21 FE (8GB/128GB, Android 12)
- Web application: Laptop Asus TUF FX505DT (16GB/512GB, Windows 11)
- IoT and image processing features:
 - + 2 IP KBVision KX-A2111N2 Megapixel cameras
 - + 2 Servo SG90 (barrier)
 - + 1 LG display monitor
 - + 1 LCD
 - + 1 Arduino Uno R3
 - + 2 ESP32
 - + Laptop Asus TUF FX505DT (16GB/512GB, Windows 11)

3. Test Plan

3.1 Human Resources

- Khúc Ngọc Thái: plan & construct test cases, verify results
- Tăng Chí Cường: execute tests and submit reports
- Hồ Hữu Phát: execute tests and submit reports
- Thuỷ Võ Anh Hoàng: execute tests and submit reports
- Nguyễn Quang Dũng: execute tests and submit report

3.2 Test Environment

- Mobile application: will be tested on a Samsung Galaxy S21 FE, running Android 12
- Web application: will be tested on a Laptop Asus TUF FX505DT, running Windows 11

3.3 Test Milestones

Able to construct a test report document before 10th of December, 2022

4. Test Cases & Test Reports

[APMS_TestReport](#)

VI. Release Package & User Guides

1. Deliverable Package

No.	Item	Sub-item	Type	Version
Code package				
1	Back-end	apms_backend.rar	zip	N/A
2	Front-end	apms_frontend.rar	zip	N/A
3	Mobile	apms_mobile.rar	zip	N/A
4	Image processing	dataset.rar	zip	N/A
5	Image processing	apms_imp.rar	zip	N/A
6	Embedded	embedded_system.rar	zip	N/A
Database scripts				
7	Database	script.sql	sql	N/A
Document				
8	Final report	Automatic-Parking-Management-and-Parking-Fee-Collection-System.pdf	pdf	N/A
9	Configuration	Configurations.docx	docx	N/A
Test				
10	System test	APMS_TestReport.xlsx	xlsx	N/A
11	Validate AI model	Detect_Results.xlsx	xlsx	N/A
12	Validate AI model	validate_model.py	py	N/A

2. Installation Guides

2.1 System Requirements

- Python version 3.7.xx or below (some packages aren't compatible with the latest python version at the moment we install - current python version is 3.11.1)
- Windows operating system (either Windows 10 or Windows 11)
- Recommend using a computer with a strong GPU (Graphics Processing Unit) to have best experiences

2.2 Installation Instruction

2.2.1 Install camera system

- Connect an IP camera to your internet router through the ethernet cable. Connect your computer to the wifi address of that router.

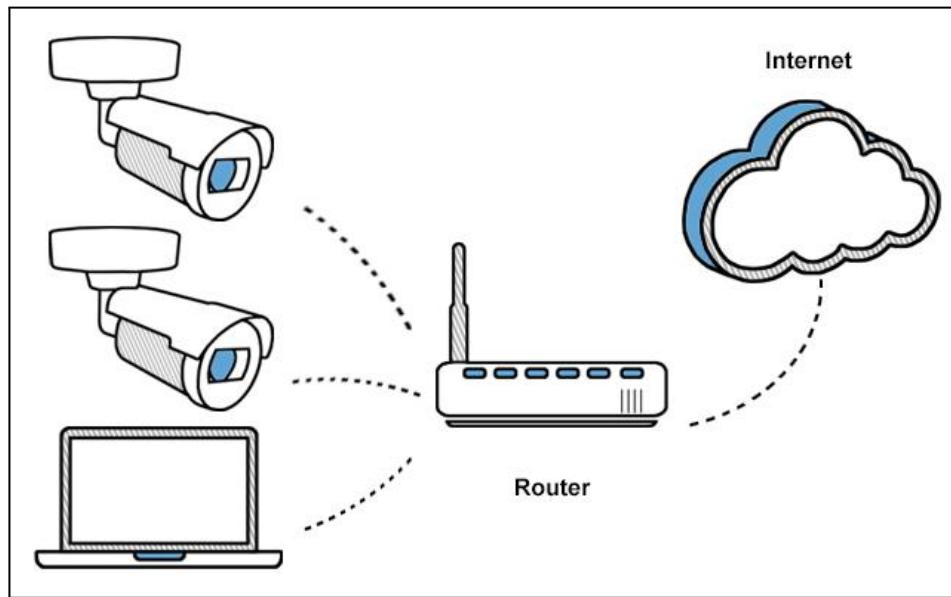


Figure 52. Camera setup

- Download KB Config Tool through this [link](#) to configure your camera via your computer.
- Open KB Config Tool, you can get the camera's IP address here.

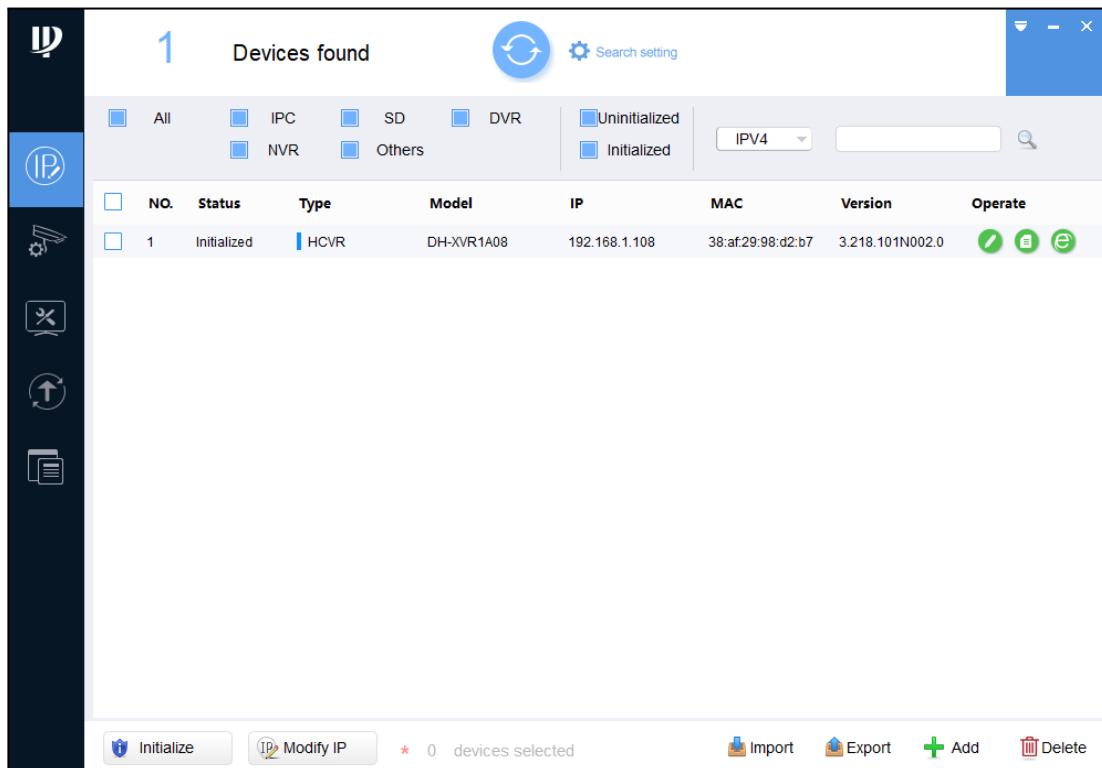


Figure 53. KB Config Tool

- Access that IP address (It's <http://192.168.1.108> for the example above).

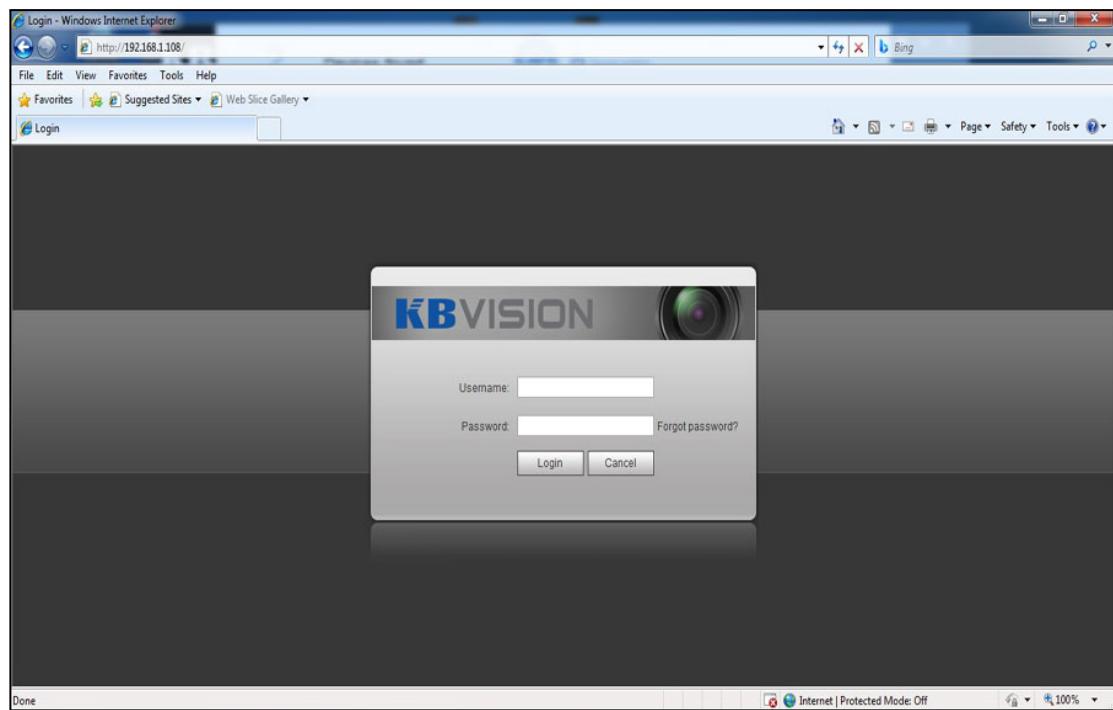


Figure 54. Camera Login Page

- Login with username: admin; password: admin123.

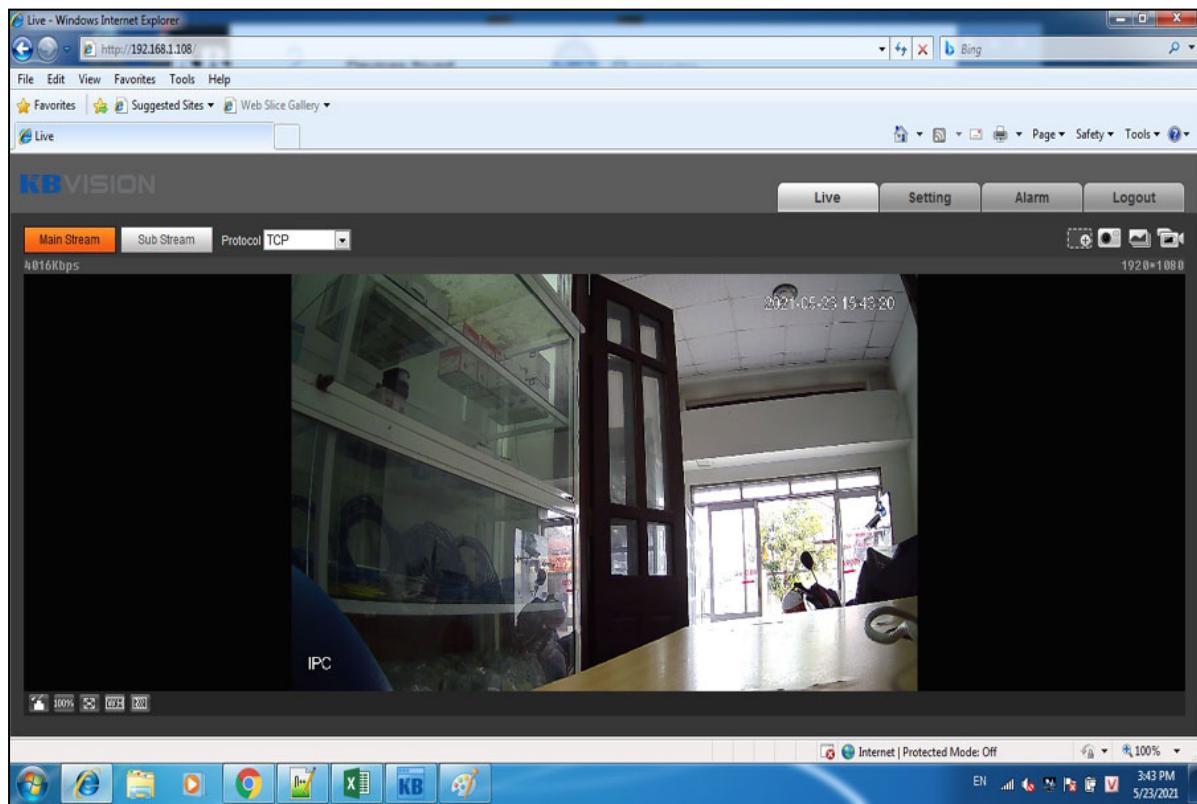


Figure 55. Camera Home Page

2.2.2 Install embedded system

- Install barrier system following the below diagram, then the system will automatically connect to WIFI:

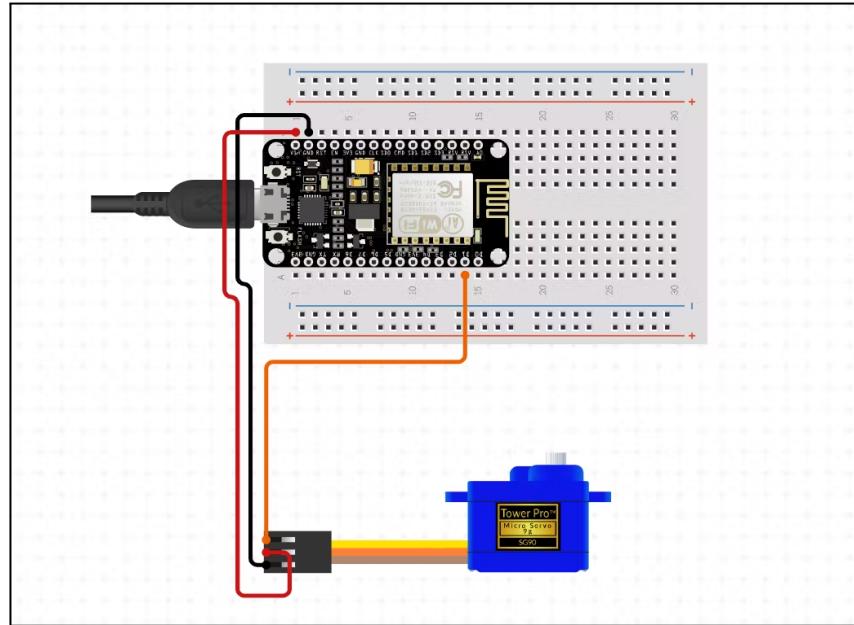


Figure 56. Barrier system

- Install the LCD system following the below diagram and plug it into the USB port to control it:

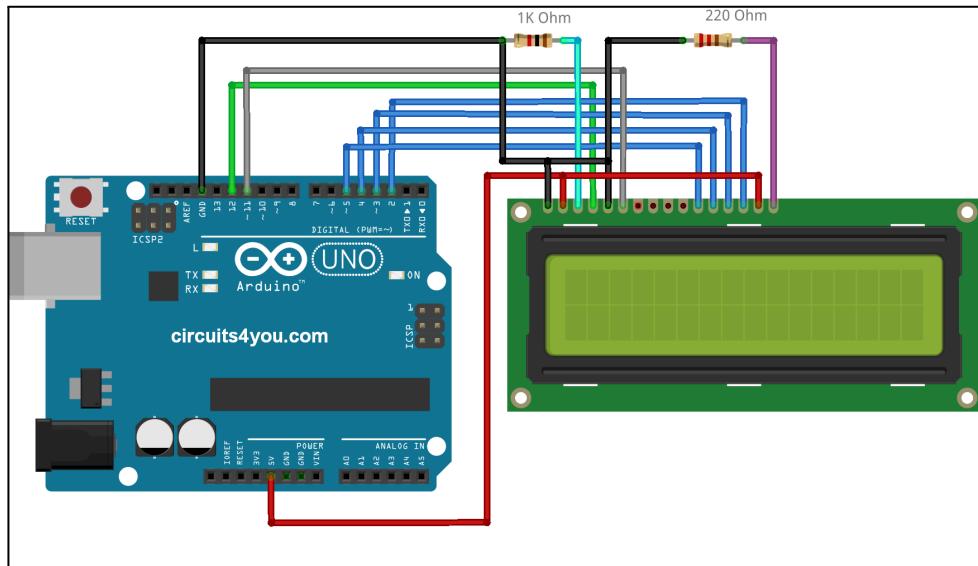


Figure 57. LCD System

2.2.3 Install python packages

- You need to have python pre-install to install packages (If you don't have python installed, you can follow the instructions in this [link](#)).
- Packages need to be installed:
 - + Torch
 - + Paddleocr

- + Dotenv
- + OpenCv
- + Threading
- + PySimpleGUI
- + PIL
- + Qrcode
- + firebase_admin
- + pyrebase

- Open Command Prompt and use this command to install packages:
 - + pip install <package-name>
- Replace <package-name> with the name of the package from above.

****Note:** If your computer has a decent GPU and you want to use it to improve the application's speed then follow these steps:

- Follow instructions from this [link](#) to install Anaconda
- Download and install CUDA from this [link](#)
- Go to PyTorch website: [PyTorch](#) and choose the version compatible with Anaconda version

- Example:

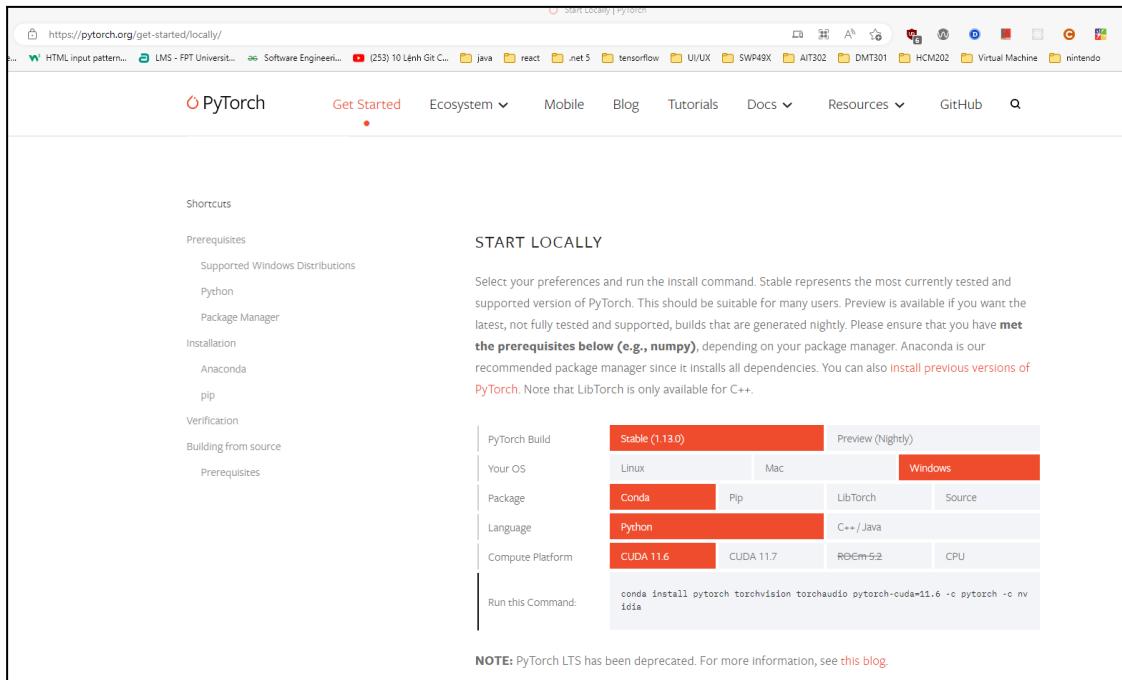


Figure 58. PyTorch Homepage

- After choosing the version, open Command Prompt, type in the command from the website, and then run to install the PyTorch package with CUDA support.
 - + Example command from the figure above

```
conda install pytorch torchvision torchaudio pytorch-cuda=11.6 -c pytorch -c nvidia
```

3. User Manual

3.1 Overview

Mobile Application: The application allows user - customer to use their account, use the money they had deposited in the account to find a suitable car park, then book a parking slot, check-in, check-out, and manage their ticket history.

Web Application: The application allows users - employees of the APMS system to interact with it and perform various actions based on their role in the system

3.2 Booking

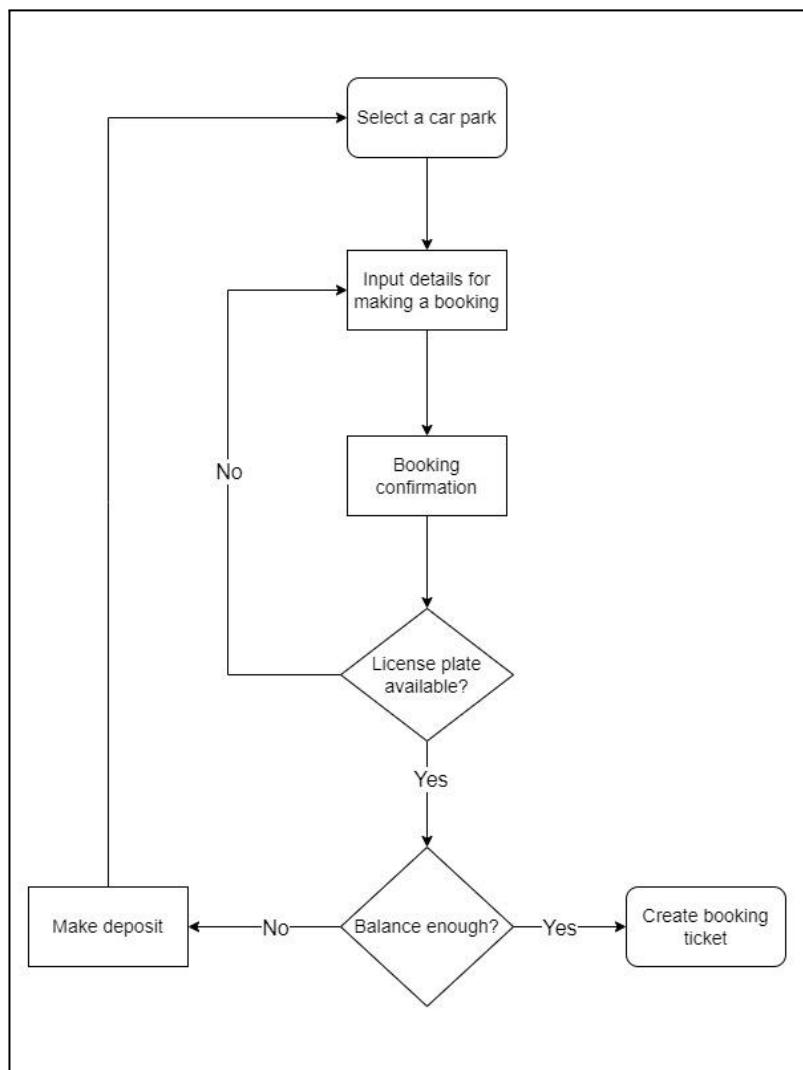


Figure 59. Booking workflow

Description:

This workflow describes the process of booking a parking slot at the desired car park for users.

Instructions:

1. After the user uses their account to login then the user will be redirected to the home page screen (*Figure 8.*), in this screen they see a list of available car parks. Then they can choose one car park from the list and start booking.
2. After a car park is chosen, the user will be redirected to the booking screen (*Figure 9.*). In this screen, the user will be required to enter information to create a booking: license plate number, book date, and book time. When all information is filled in, the user presses on “Go to confirmation page” to confirm their booking.
3. On the booking confirmation screen (*Figure 10*), the user can see all the information about booking a ticket before our system creates one. When the user is satisfied with all the details of the booking ticket then the user can press “ Pay ”, our system will do final verifications before creating a ticket.
4. First, our system will check for the availability of the license plate. Is there any vehicle with this license plate in any car park? Does this license plate number appear on any booking tickets? If the license plate has none of the above criteria then it is available to create a booking. If one of the criteria appears then our system will respond with an error and require the user to change the booking detail.
5. Second, our system will check the user account’s balance. If the balance is enough to make a booking then our system will subtract the booking fee from the user's balance and create a booking ticket.
 - 5.1. If the user's balance is not enough then our system will require the user to deposit by using Paypal and then start over from step 1.

3.3 Check-in

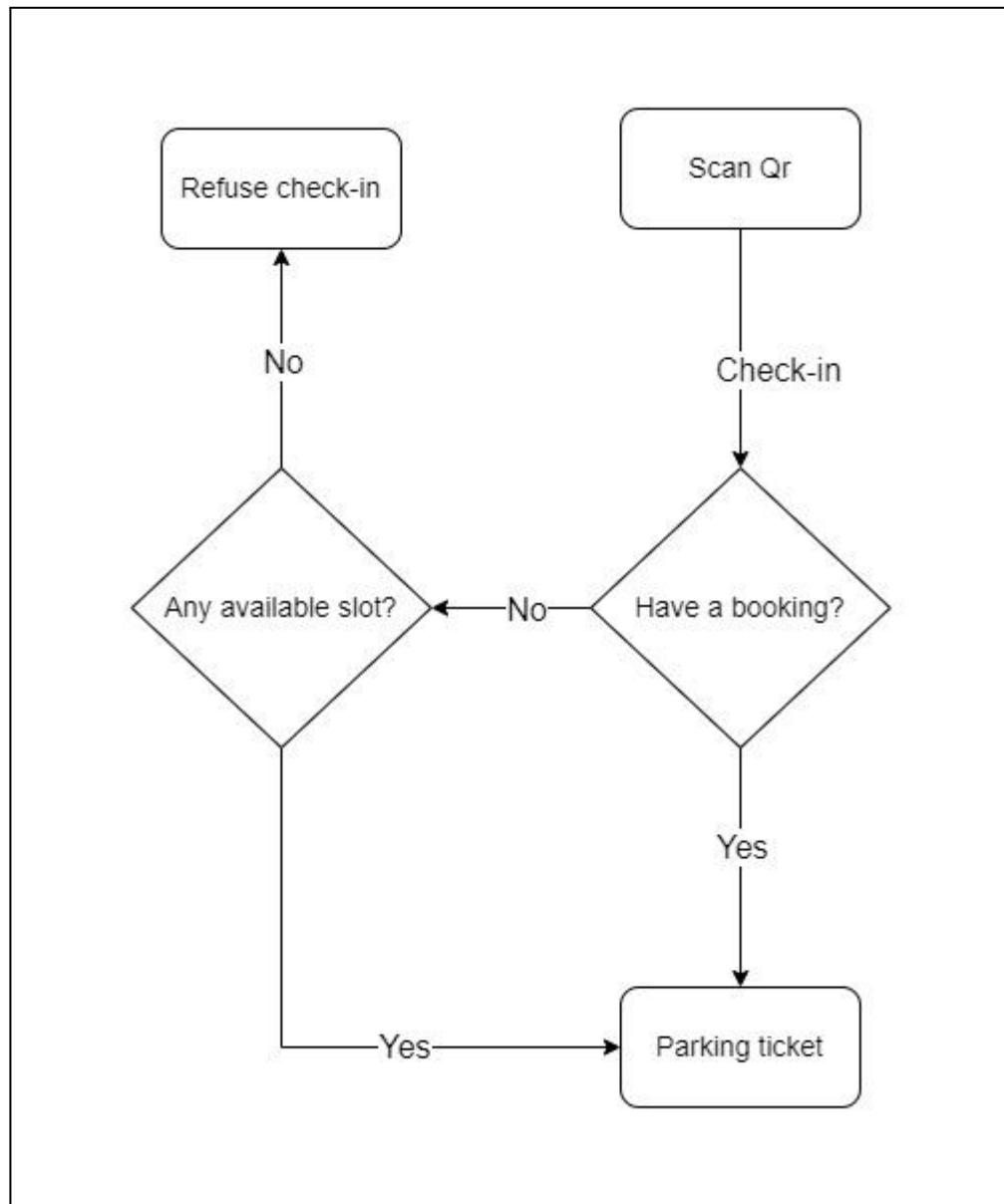


Figure 60. Check-in workflow

Description:

This workflow describes the process of checking in when users want to go into the car park

Instructions:

1. The user can go to the scan QR screen (*Figure 11.*) when they are in the homepage screen (*Figure 8.*) or ticket detail screen (*Figure 13.*) - when the user has booked a slot
2. When the user is checking in, after the mobile application receives data from the QR code then it will prompt a pop-up for the user to double-check if the system is reading the right license plate.
3. After the user confirms that it is the right license plate number, our system will check if the user has booked any slot or slot.

4. If the user has booked a slot, our system will change the booked ticket to a parking ticket then the user can continue to go into the car park and park their car.
 - 4.1. If the user has not booked any slot, our system will check for available slots at the current car park.
 - 4.1.1. If there's no available slot left then the user can't go in and they have to find a different place to park.
 - 4.2. If the car park still has available slots then our system will create a new parking ticket for the user then the user can go in and park their car.

3.4 Check-out

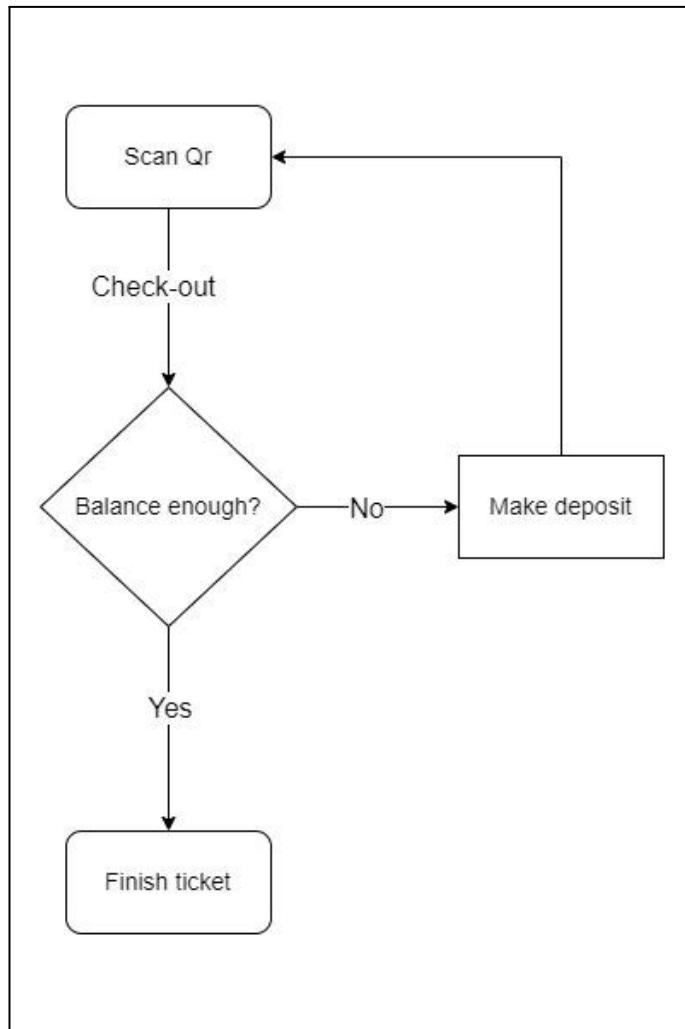


Figure 61. Check-out workflow

Description:

This workflow describes the process of checking out when users want to leave the car park

Instructions:

1. The user can go to the scan QR screen (*Figure 11.*) when they are in the homepage screen (*Figure 8.*) or ticket detail screen (*Figure 15.*)

2. When the user is checking in, after the mobile application receives data from the QR code then it will prompt a pop-up for the user to double-check if the system is reading the right license plate.
3. After the user confirms that it is the correct license plate number, our system will check their account's balance to see if they can afford the parking fee
 - 3.1. If their balance is not enough then the user has to make a deposit and then start over from step 1.
4. If the user has enough money then our system will finish the parking ticket, subtract the parking fee from the account's balance, and the user can leave the car park.

3.5 Manual check-out

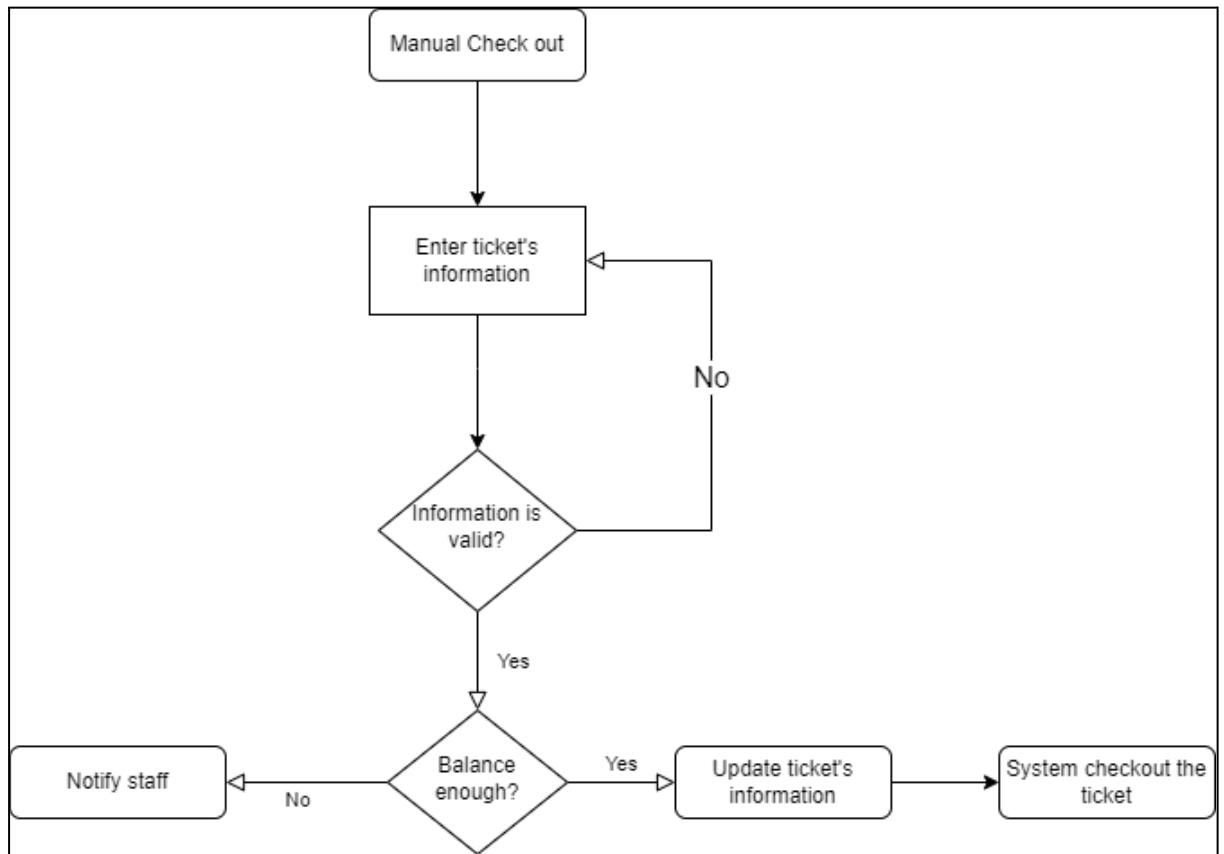


Figure 62. Manual check-out workflow

Description:

This workflow describes the process of staff manually editing and checking out a ticket because the system was unable to perform this action automatically

Instructions:

1. When the system fails to automatically check out the ticket when a customer scans the QR code, it will immediately prompt the staff with the pictures taken by the cameras and the falsely scanned plate number.
2. Staff can then carefully examine the pictures to figure out the correct plate number of the vehicle.

3. The staff will then enter the correct plate number into the text box and send it to the system along with the car park id.
 - 3.1. If the system can not find any ticket with the provided plate number, it will throw an error for the staff and they will have to re-enter the plate number again.
4. If the system finds a ticket with the status of “checked-in” and has matched plate number, it will then check for the customer’s account balance, if the balance is enough to pay for the ticket’s fee, the system will then perform a check out operation for the ticket and user can leave the car park.
 - 4.1. If the customer’s account balance is not sufficient, the system will notify the staff for further action.

3.5 Update car park information

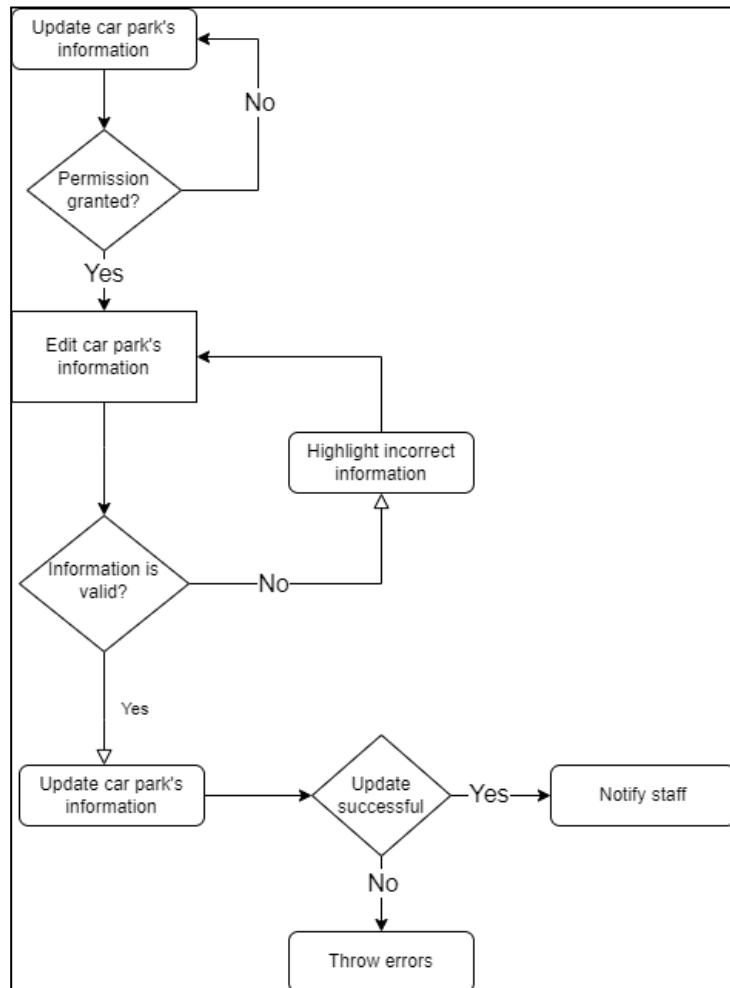


Figure 63. Update car park information workflow

Description:

This workflow describes how an employee can change the car park’s information in case the existing information is incorrect or need to be updated to match the current properties of the car park

Instructions:

1. From the dashboard, the employee can access the update page through clicking the Car Park List (for admin) or My Park (for owner and staff) from the sidebar.
2. User can then edit the information about the car park.
 - 2.1. If the user is a staff, they must have permission from the owner of the car park to perform the mentioned action.
3. After finishing editing the car park's information, the user can submit the information for validation.
4. If all information is valid, the system will then send a request to update the car park to the server.
 - 4.1. If there is any validation error that occurs, the system will notify the user about the invalid details and it is required to update that property until it is valid to update the car park's information
5. The server will then perform the update action and notify the user about the result
 - 5.1. If the update is successful, the system will notify with a toast message and the car park's details will be fetched again to reflect the changes.
 - 5.2. If the update is unsuccessful, the system will notify the user with a toast message containing the reason why it can not perform that action.

3.6 Create a new employee's account

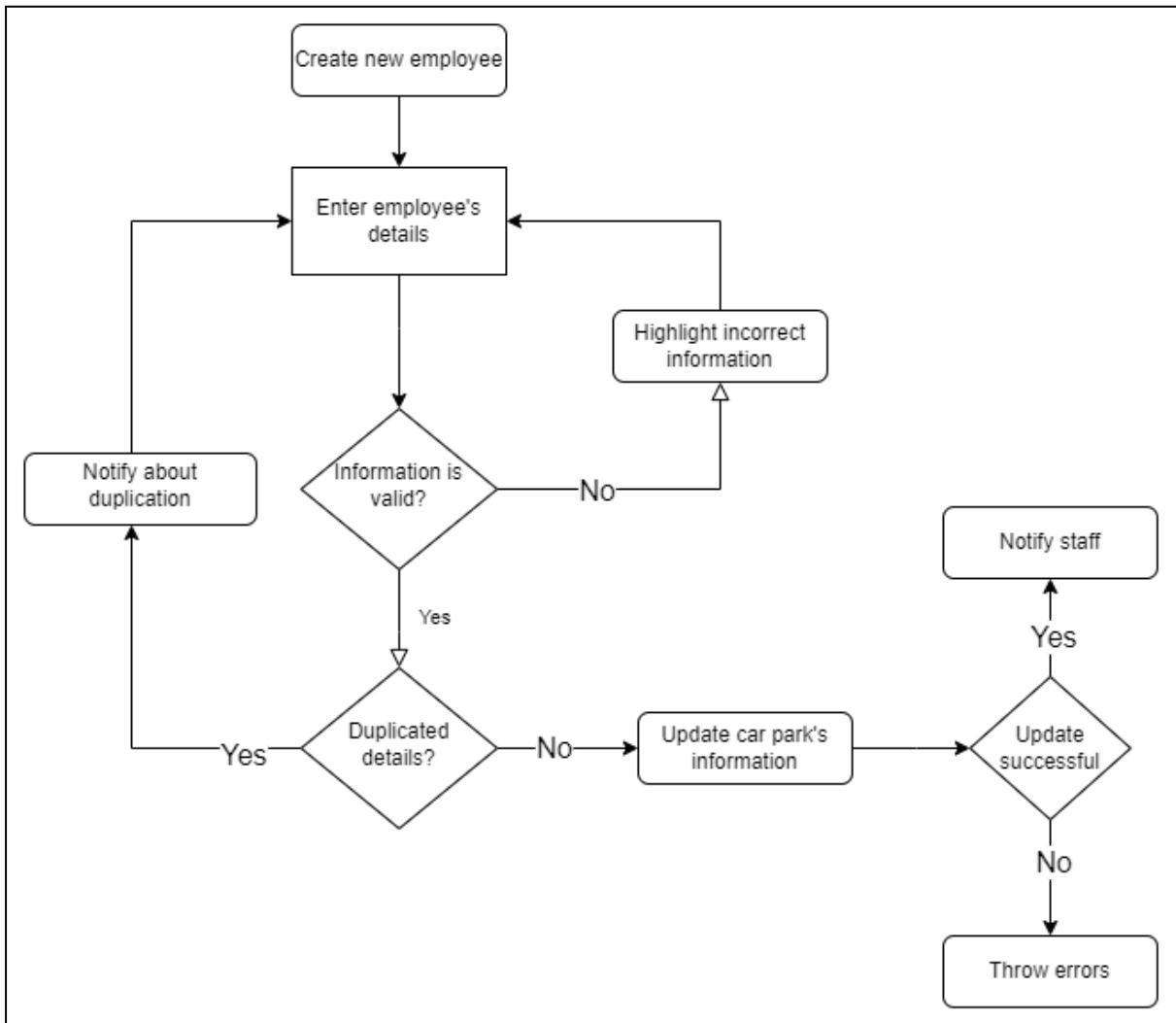


Figure 64. Create a new employee's account workflow

Description:

This workflow describes the process of creating a new employee account (of role Admin, Owner, or Staff)

Instructions:

1. From the dashboard, the admin can navigate to the employee list through the sidebar panel, afterward, clicking the “Add Employee” button will prompt the admin with the create form.
2. Admin will then fill in the new employee’s detail and submit them for validation
3. If all information is valid, the system will make a request to the server for creating a new employee
 - 3.1. If there is any invalidation, the system will highlight the invalid details and it is required to correct them to continue the process.
4. The server will then check in the database for any duplication with provided information from the user.

- 4.1. If there is any duplication (which means that the information the user supplied has been used) the server will throw an error back to the user and the process will have to roll back to step 2.
5. After all validation has been completed, the system will then attempt to create a new account and insert it to the database.
 - 5.1. If the creation is successful, the system will notify with a toast message and the employee list will be fetched again to reflect the changes.
 - 5.2. If the creation is unsuccessful, the system will notify the user with a toast message containing the reason why it can not perform that action.