THE UNIVERSITY OF DANANG
DANANG UNIVERSITY OF SCIENCE AND TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

GRADUATION PROJECT THESIS

MAJOR: INFORMATION TECHNOLOGY

SPECIALTY: SOFTWARE ENGINEERING

PROJECT TITLE:

BUILDING AN APPLICATION TO SUPPORT FINDING PARKING LOCATIONS

Instructor: **Dr. NGUYEN VAN HIEU**
Student: **LE NGOC HUNG**
Student ID: **102200019**
Class: **20T1**

Da Nang, 06/2024

Le Ngoc Hung **BUILDING AN APPLICATION TO SUPPORT FINDING PARKING LOCATIONS** **2024**

THE UNIVERSITY OF DANANG
DANANG UNIVERSITY OF SCIENCE AND TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

GRADUATION PROJECT THESIS

MAJOR: INFORMATION TECHNOLOGY

SPECIALTY: SOFTWARE ENGINEERING

PROJECT TITLE:

BUILDING AN APPLICATION TO SUPPORT FINDING PARKING LOCATIONS

Instructor: **Dr. NGUYEN VAN HIEU**
Student: **LE NGOC HUNG**
Student ID: **102200019**
Class: **20T1**

Da Nang, 06/2024

SUMMARY

Topic title: Building an application to support finding parking locations

Student name: Le Ngoc Hung

Student ID: 102200019 Class: 20T1

Project summary:

This application is developed with the aim of supporting users finding nearby parking locations. It allows users to easily reserve parking lots and make payments for their desired parking lots. By integrating MOMO payment method, this application aims to make it easier for users to reserve space in a parking lot. The main features of the application are providing real-time information about the status of parking lots, allowing users to easily search for and reserve parking lots, helping users easily view their reservation history, providing directions to the desired parking lot in the easiest way possible. Moreover, parking owners and users can contact each other to exchange information about parking lot information by sending and receiving real-time messages.

DA NANG UNIVERSITY
UNIVERSITY OF SCIENCE AND TECHNOLOGY
FACULTY INFORMATION TECHNOLOGY

SOCIALIST REPUBLIC OF VIETNAM
Independence - Freedom - Happiness

GRADUATION PROJECT COMMENT

I. General information:

1. Student name: Le Ngoc Hung
2. Class: 20T1 Student ID: 102200019
3. Topic title: Building an application to support finding parking locations
4. Instructor: Dr. Nguyen Van Hieu Academic title/ degree: Ph.D.

II. Reviews of graduation project

1. About the urgency, novelty, usability of the topic: (2 points)
   …………………………………………………………………………………
   …………………………………………………………………………………
2. About the results of solving the tasks required by the project: (4 points)
   …………………………………………………………………………………
   …………………………………………………………………………………
3. About the form, structure and layout of the graduation project: (2 points)
   …………………………………………………………………………………
   …………………………………………………………………………………
4. The topic includes scientific value/article/problem solving of the enterprise or school: (1 point)
   …………………………………………………………………………………
   …………………………………………………………………………………
5. Existing shortcomings need to be supplemented or modified:
   …………………………………………………………………………………
   …………………………………………………………………………………

III. Spirit and attitude of the student (1 point):

………………………………………………………………………………..
………………………………………………………………………………..

IV. Evaluation:

1. Evaluation point: …. /10
2. Suggest: ☐ Defense permitted ☐ Edit to defend ☐ Defense not permitted

Da Nang, date……month……. 2024

Instructor

DA NANG UNIVERSITY
UNIVERSITY OF SICIENCE AND TECHNOLOGY
FALCUTY INFORMATION TECHNOLOGY

THE SOCIALIST REPUBLIC OF VIETNAM
Independence - Freedom - Happiness

GRADUATION PROJECT REQUIREMENTS

Student Name: Le Ngoc Hung Student ID: 102200019

Class: 20T1 Faculty: Information Technology Major: Software Engineering

1. *Topic title:*

   Building an application to support finding parking locations
2. *Project topic: ☐has signed intellectual property agreement for final result*
3. *Initial figure and data:*

   None
4. *Content of the explanations and calculations:*

   Introduction: Provide an overview of the topic, objectives, significance, and requirements of the project, methods used, and the structure of the report.

   Theoretical Basis: Present the theoretical foundations applied in the project. System Analysis and Design: Present the analysis and design involved in building the system.

   System analysis and design: Present the analysis documents and design documents in system construction and the workflow of the system.

   Implementation and Evaluation: Describe the implementation, installation, operation of the system, deployment and evaluate the achieved results.

   Conclusion: Summarize the findings, lessons learned during the system development process, limitations, and directions for future development
5. *Drawings, charts (specify the types and sizes of drawings):*

   None
6. *Name of instructor: Dr. Nguyen Van Hieu*
7. *Date of assignment: 10/04/2024.*
8. *Date of completion: 19/06/2024.*

Da Nang, date 19 month 06 year 2024

Head of Division Software Engineering **Instructor**

PREFACE

During my time studying at the University of Science and Technology - University of Da Nang and throughout the process of completing my graduation project, I have received a lot of help and enthusiastic guidance from the esteemed professors and lecturers of the Information Technology faculty. I would like to express my profound gratitude to all the teachers who have devotedly taught and imparted specialized knowledge, as well as inspired my learning spirit. Thanks to their dedicated guidance, I have been able to successfully complete my graduation project.

I would like to extend my deep gratitude to Dr. Nguyen Van Hieu, who has diligently guided and supported me throughout the process of completing my graduation project. He helped me understand the project's objectives, provided valuable advice in choosing the topic, and was always available to meet and discuss and clarify any doubts. His guidance and assistance have played a crucial role in my successful completion of this project.

To have reached this point today, I am very grateful to my family and loved ones for giving me the opportunity, nurturing, encouraging, motivating, and providing the best conditions for me to continue my studies. I also want to thank my friends who have always accompanied and encouraged me throughout my learning journey.

Throughout my study and the completion of my graduation project, I acknowledge that there have been shortcomings. I hope to receive valuable feedback from teachers, lecturers, and peers so that I can improve, enhance the results, and accumulate more useful experience for future endeavors.

Sincerely thanks!

ASSURRANCE

I hereby certify that:

1. The graduation project report titled: Building an application to support finding parking locations is the result of my own research under the direct guidance of Dr. Nguyen Van Hieu.
2. I have independently read, researched, translated documents, and synthesized the knowledge that contributed to this report, ensuring that there is no plagiarism from any source.
3. The theoretical framework presented in the thesis is based on the reference materials, as indicated in the reference section of the report.

If any violation is found, I will take full responsibility.

Student Performed

Le Ngoc Hung

TABLE OF CONTENT

SUMMARY .... **iii**

PREFACE .... **v**

ASSURRANCE .... **vi**

LIST OF TABLES .... **ix**

LIST OF FIGURES .... **x**

LIST OF SYMBOLS, ACRONYMS .... **xii**

INTRODUCTION **1**

1. The purpose of implementing the project **1**

2. The objectives of the project **1**

3. Scope and subjects of research **1**

3.1. Scope 1

3.2. Research Subjects 2

4. Research method **3**

5. The structure of the thesis **3**

CHAPTER 1: THEREOTICAL FOUNDATION **4**

1.1. Technology used **4**

1.1.1. Android with Kotlin 4

1.1.2. Back-end server 5

1.1.2.1. Typescript 5

1.1.2.2. MongoDB Atlas 6

1.2. AWS **7**

1.3. Firebase Cloud Messaging **7**

1.4. RESTful API **8**

1.4.1. Introduction to RESTful API 8

1.4.2. RESTful API workflow 9

1.4.3. Status codes 10

1.5. Google Maps API **11**

1.6. Chapter conclusion 12
CHAPTER 2: SYSTEM ANALYSIS AND DESIGN 13
2.1. Project requirements analysis 13
2.1.1. Main features of the project 13
2.1.2. Target users 13
2.2. System Design Analysis Diagram 14
2.3. Business processes 17
2.4. Use-case specification 20
2.5. Database design 25
2.6. Chapter conclusion 29
CHAPTER 3: BUILDING APPLICATION AND TESTING 30
3.1. Building application 30
3.2. Application testing 55
3.3. Deployment 56
3.4. List of APIs 57
CHAPTER 4: CONCLUSION AND FUTURE ORIENTATION 58
4.1. Achieved results 58
4.1.1. In terms of theoretical 58
4.1.2. In terms of application 58
4.2. Drawbacks 58
4.3. Future development 58
REFERENCES 59

LIST OF TABLES

Table 2.1. Login Use-case specification..21
Table 2.2. Add new parking lot specification 23
Table 2.3. Reserve parking lot specification 24
Table 2.4. Find nearby parking lots specification 24
Table 2.5. Table Accounts...26
Table 2.6. Table Lots..27
Table 2.7. Table Orders 27
Table 2.8. Table Payments 28
Table 2.9. Table Momos...28
Table 2.10. Table Notification...29
Table 2.11. Table Contacts...29
Table 2.12. Table Message...29
Table 3.1. Table Tools and Libraries..30
Table 3.2. Table Application testing 56

LIST OF FIGURES

Figure 1.1. Android development with Kotlin 5
Figure 1.2. Introduction to TypeScript 6
Figure 1.3. Introduction to MongoDB Atlas 7
Figure 1.4. Introduction to Amazon Web Services 7
Figure 1.5. Introduction to Firebase Cloud Messaging 8
Figure 2.1. System Use-case diagram 14
Figure 2.2. Use-case Adding new parking lot 15
Figure 2.3. Use-case Editing personal information 15
Figure 2.4. Use-case Viewing parking lot detail 16
Figure 2.5. Use-case Reserving lot 16
Figure 2.6. Login activity diagram 17
Figure 2.7. Add new parking lot activity diagram 18
Figure 2.8. Reserve parking lot activity diagram 19
Figure 3.1. Register new account page 31
Figure 3.2. Login page 32
Figure 3.3. Homepage 33
Figure 3.4. List of parking lots 34
Figure 3.5. View parking lot information 35
Figure 3.6. View parking lot detail (1) 36
Figure 3.7. View parking lot detail (2) 37
Figure 3.8. Pay for parking lot order (1) 38
Figure 3.9. Pay for parking lot order (2) 39
Figure 3.10. Pay for parking lot order (3) 40
Figure 3.11. Pay for parking lot order (4) 41
Figure 3.12. Pay for parking lot order (5) 42
Figure 3.13. Show route to the parking location 43
Figure 3.14. View parking history (1) 44
Figure 3.15. View parking history (2) 45
Figure 3.16. Recent chats with others 46
Figure 3.17. Detail chat screen 47
Figure 3.18. View notifications 48
Figure 3.19. View and change personal information 49

Figure 3.20. Add new parking lot (1) 50
Figure 3.21. Add new parking lot (2) 51
Figure 3.22. Add new parking lot (3) 52
Figure 3.23. Add new parking lot (4) 53
Figure 3.24. Add new parking lot (5) 54

LIST OF SYMBOLS, ACRONYMS

API	Application Programming Interface
Json	The abbreviation of JavaScript Object Notation, which is a data format adhering to a specific rule that most programming languages today can read. JSON is an open standard for exchanging data on the web.
XML	XML (Extensible Markup Language) is a markup language created by the World Wide Web Consortium (W3C) to define the syntax for encoding documents, enabling humans and machines to read it.

INTRODUCTION

1. The purpose of implementing the project

Nowadays, in large cities such as Hanoi, Ho Chi Minh, Da Nang…, the density of cars per capita is relatively high. The current situation can be observed when moving on streets, where roads are narrow and sidewalks are small, resulting in parking on the road or sidewalk, which only partially meets the small parking needs of people when traveling. Although there are many parking lots, people often waste time searching for parking spaces near their desired locations because they are not aware of where the parking lots are.

Therefore, the development of a parking lot search application is essential to address the difficulties mentioned above. Currently, there are some applications on the market aimed at solving this issue, such as Viettel's MyParking. However, this application still has some limitations, which will be discussed later in this report. Recognizing the essential needs and problems faced by vehicle owners participating in traffic, I have decided to choose the topic "Building an application to support finding parking locations" specifically for Vietnamese users.

2. The objectives of the project

The objective and scope of the project are to design and develop an application that can run on the Android platform, fulfilling basic functions such as posting parking spots, displaying parking locations on a map, searching for and reserving parking spots. This application can generate passive income for individuals who have unused parking spaces at home by allowing them to post these spaces for those in need, as well as help parking lots find consumers easily. Users can both be individuals searching for parking spots and parking lot owners.

3. Scope and subjects of research

3.1. Scope

The scope of this project encompasses the development and implementation of a mobile application for searching and booking parking spots. The key areas covered in this project include:

- User Management: Handling user registration, login, and profile management.

- Parking Spot Search and Booking: Implementing functionalities for users to search for available parking spots, view details, and book them.
- Parking Spot Management: Allowing parking spot owners to add, update, and manage their parking spots.
- Notifications: Implementing a system to send real-time notifications to users about their bookings and other relevant updates.
- Payment Integration: Facilitating secure online payments for parking spot bookings.
- Map Integration: Utilizing Google Maps API for location services and visual representation of parking spots.
- Data Storage and Security: Ensuring secure storage and management of user and parking spot data using Firebase services.

The project aims to provide a comprehensive solution that addresses the needs of both users looking for parking spots and parking spot owners managing their spaces.

3.2. Research Subjects

The research subjects involved in this project include:

- End Users: Individuals who use the application to search for, book, and manage parking spots. These users are primarily vehicle owners who need convenient parking solutions.
  - Profile:
    - Age: 18-60 years
    - Gender: All genders
    - Occupation: Varied, including office workers, shoppers, and residents of urban areas
    - Technology Proficiency: Basic to intermediate level of smartphone usage
- Parking Spot Owners: Individuals or entities that own parking spaces and wish to offer them for booking through the application. These users manage their parking spots and monitor bookings.
  - Profile:
    - Age: 25-60 years
    - Gender: All genders

  - Occupation: Property owners, businesses, or individuals with extra parking space
  - Technology Proficiency: Basic to intermediate level of smartphone and web application usage

4. Research method

The tasks required to complete the project include:

- Brainstorming ideas and detailing all aspects of the project.
- Researching theoretical foundations, analyzing functionalities, and systems.
- Designing the corresponding database.
- Designing the interface, implementing functionalities.
- Testing, debugging, and refining the product to completion.

5. The structure of the thesis

- Chapter 1: Theoretical foundation
- Chapter 2: System analysis and design
- Chapter 3: Building application and testing
- Chapter 4: Conclusion and future development
- References

CHAPTER 1: THEREOTICAL FOUNDATION

1.1. Technology used

1.1.1. Android with Kotlin

Kotlin is a versatile programming language developed by JetBrains in 2011 and publicly introduced in 2016. Kotlin is designed to run on the Java Virtual Machine (JVM) and is an object-oriented programming language, interoperable with Java, and supports functional programming features. Kotlin has been officially supported by Google for Android application development since Android Studio version 3.0.

With many outstanding advantages, Kotlin has become one of the most popular programming languages in the developer community. Below are the advantages of Kotlin:

- Safety: Kotlin is designed to ensure safety during application development. Kotlin supports static type checking, allowing programmers to detect and fix data type-related errors before the application is compiled. Additionally, Kotlin also supports null-safety features to avoid null-related errors.
- Simplicity: Kotlin has simple syntax, easy to read, and understand. This makes source code development and maintenance easier.
- Efficiency: Kotlin allows programmers to write concise and understandable source code, reducing development time and increasing the efficiency of the application development process.
- Compatibility: Kotlin is an interoperable language, allowing programmers to interact with libraries or other source code in different languages such as Java, C++, and Python.
- Extensibility: Kotlin supports extension functions, allowing programmers to extend classes and methods of existing libraries without modifying the original source code.
- Community: Kotlin has a large community of programmers, with many documents and open-source projects shared publicly on the Internet. This helps programmers easily find supporting documents and answers to questions during development.

In summary, Kotlin is a versatile, safe, simple, efficient, compatible, extensible, flexible, mobile application-supporting, modern programming language with a large community of programmers. With all these advantages, I have chosen Kotlin as the main programming language to build the application for this project.

Figure 1.1. Android development with Kotlin

1.1.2. Back-end server

1.1.2.1. Typescript

TypeScript is an open-source project developed by Microsoft. It can be considered as an advanced version of JavaScript by adding optional static typing and object-oriented features that are not present in JavaScript. TypeScript can be used to develop applications for both client-side (Angular2) and server-side (NodeJS).

TypeScript utilizes all features of ECMAScript 2015 (ES6) such as classes and modules. It does not stop there; if ECMAScript 2017 is released, TypeScript will likely upgrade its version to incorporate the latest techniques from ECMAScript.

Below are some advantages of TypeScript:

- Easy development of large projects: By using the latest techniques and object-oriented programming, TypeScript makes it easy to develop large projects.
- Multiple framework choices: Nowadays, JavaScript frameworks are gradually encouraging the use of TypeScript for development, such as AngularJS 2.0 and Ionic 2.0.
- Support for features of the latest JavaScript versions: TypeScript always ensures the full use of the latest JavaScript techniques, such as the current version ECMAScript 2015 (ES6).

- It is an open source, so it's free to use. Moreover, it is still supported by the huge community.

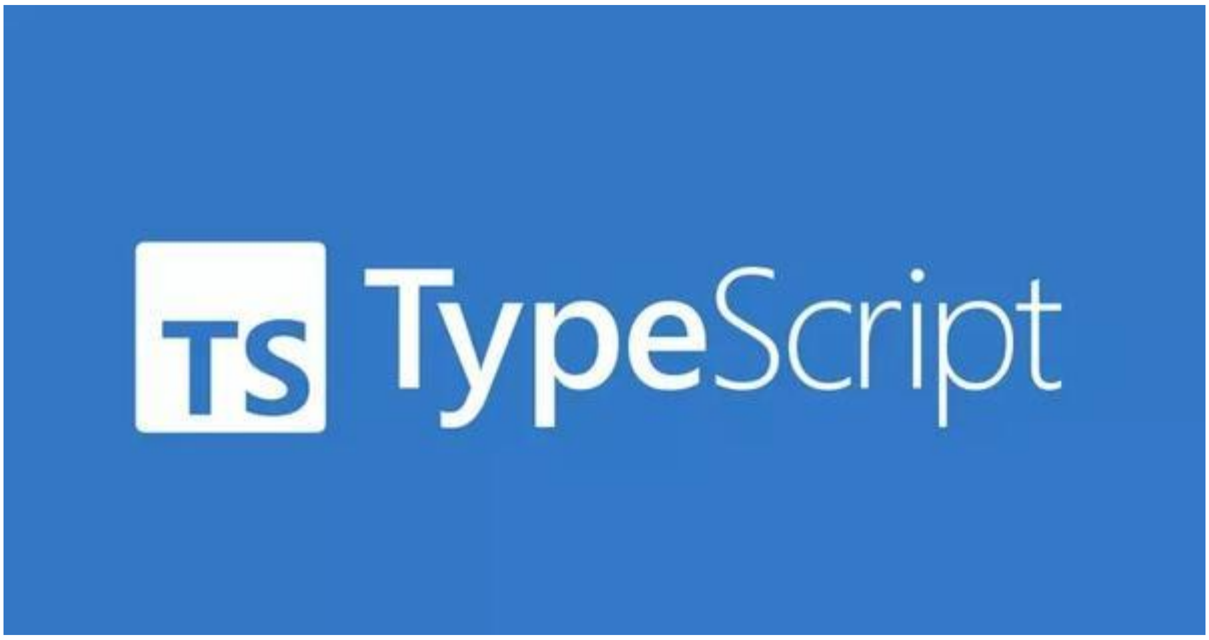


Figure 1.2. Introduction to TypeScript

1.1.2.2. MongoDB Atlas

MongoDB is an open-source NoSQL database and one of the leading NoSQL databases, used by millions of users. MongoDB is written in C++. Additionally, MongoDB is a cross-platform database, operating on the concepts of Collection and Document. It provides high performance, high availability, and easy scalability.

NoSQL is an open-source type of database management system that does not use Transact-SQL for querying information. NoSQL stands for "Not Only SQL" or "None-Relational SQL," as it is often called. This database is developed on JavaScript Framework with JSON data type. (The syntax of JSON is "key: value.") NoSQL emerged as a patch for the shortcomings and limitations of the relational RDBMS data model in terms of speed, features, scalability, and memory cache.

Below are some advantages of MongoDB:

- Less schema: Since the schema is generated to group objects into a cluster for easy management. For example, creating a schema named Students will only include objects related to students in this schema. In MongoDB, however, a single collection can contain multiple different documents. Each document can have different fields, content, and sizes.
- Clear object structure.
- No complex joins.

- Extremely large scalability: Scaling data without worrying about foreign keys, primary keys, constraint checking, etc. MongoDB allows replication and sharding, making scaling easier.

Figure 1.3. Introduction to MongoDB Atlas

1.2. AWS

Amazon Simple Storage Service (Amazon S3) is an object storage service with scalable storage capacity, high availability, security, and industry-leading performance. You can use Amazon S3 to store and retrieve data of any volume, at any time, from anywhere.

S3 is secure because AWS provides:

- Data encryption for stored data, which can occur in two ways: encryption at the client-side and encryption at the server-side.
- Multiple copies are maintained to facilitate data recovery in case of data corruption.
- Versioning, where each edit is stored for use when needed.

You can store any type of data in any format. When discussing storage capacity on S3, the number of objects that can be stored on S3 is unlimited. An object is a basic entity in S3, consisting of data, keys, and metadata.

Figure 1.1. Introduction to Amazon Web Services

1.3. Firebase Cloud Messaging

Firebase Cloud Messaging (FCM) is a cross-platform messaging solution that lets you reliably send messages at no cost.

Using FCM, you can notify a client app that new email or other data is available to sync. You can send notification messages to drive user re-engagement and retention. For use cases such as instant messaging, a message can transfer a payload of up to 4096 bytes to a client app.

With FCM, you can send two types of messages to clients:

- Notification messages, sometimes thought of as "display messages." These are handled by the FCM SDK automatically.
- Data messages, which are handled by the client app.

Notification messages contain a predefined set of user-visible keys. Data messages, by contrast, contain only your user-defined custom key-value pairs. Notification messages can contain an optional data payload. The maximum payload for both message types is 4096 bytes, except when sending messages from the Firebase console, which enforces a 1000-character limit.

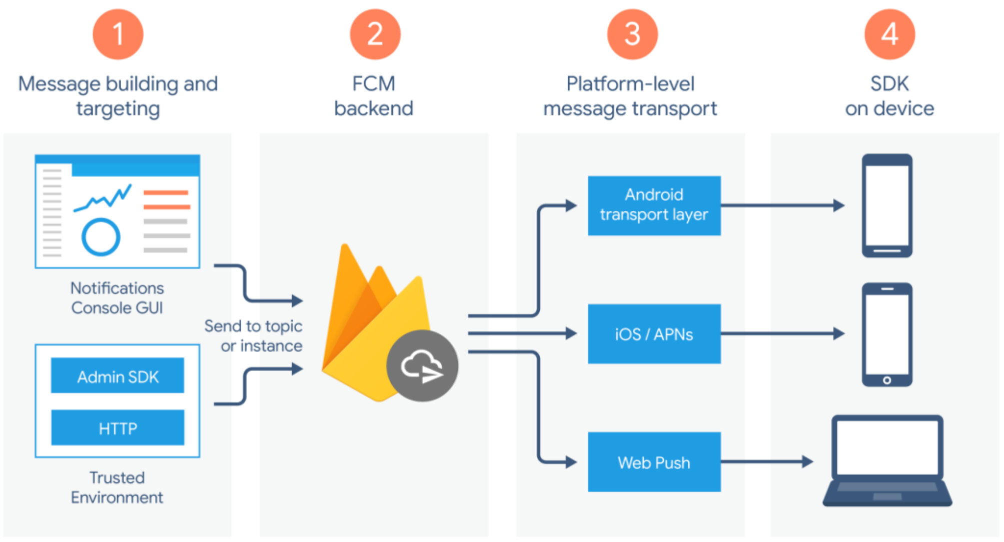

1
Message building and targeting
2
FCM backend
3
Platform-level message transport
4
SDK on device
Notifications Console GUI
Admin SDK
HTTP
Trusted Environment
Send to topic or instance
Android transport layer
iOS / APNs
Web Push

Figure 1.1. Introduction to Firebase Cloud Messaging

1.4. RESTful API

1.4.1. Introduction to RESTful API

A REST API (also called a RESTful API or RESTful web API) is an application programming interface (API) that conforms to the design principles of the representational state transfer (REST) architectural style. REST APIs provide a flexible, lightweight way to integrate applications and to connect components in microservices architectures.

API (Application Programming Interface) is a set of rules and mechanisms through which an application or component interacts with another application or component. APIs can return data that your application needs in common data formats such as JSON or XML.

REST (Representational State Transfer) is a type of data structure transformation, an architectural style for writing APIs. It uses simple HTTP methods to facilitate communication between machines. Therefore, instead of using a URL to handle some user information, REST sends an HTTP request like GET, POST, DELETE, etc. to a URL to process data.

RESTful API is a standard used in designing APIs for web applications to manage resources. RESTful is one of the commonly used API design styles today for different applications (web, mobile, etc.) to communicate with each other.

The most important function of REST is to define how to use HTTP methods (such as GET, POST, PUT, DELETE...) and how to format URLs for web applications to manage resources. RESTful does not dictate the logic of application code and is not limited by the programming language of the application. Any language or framework can be used to design a RESTful API.

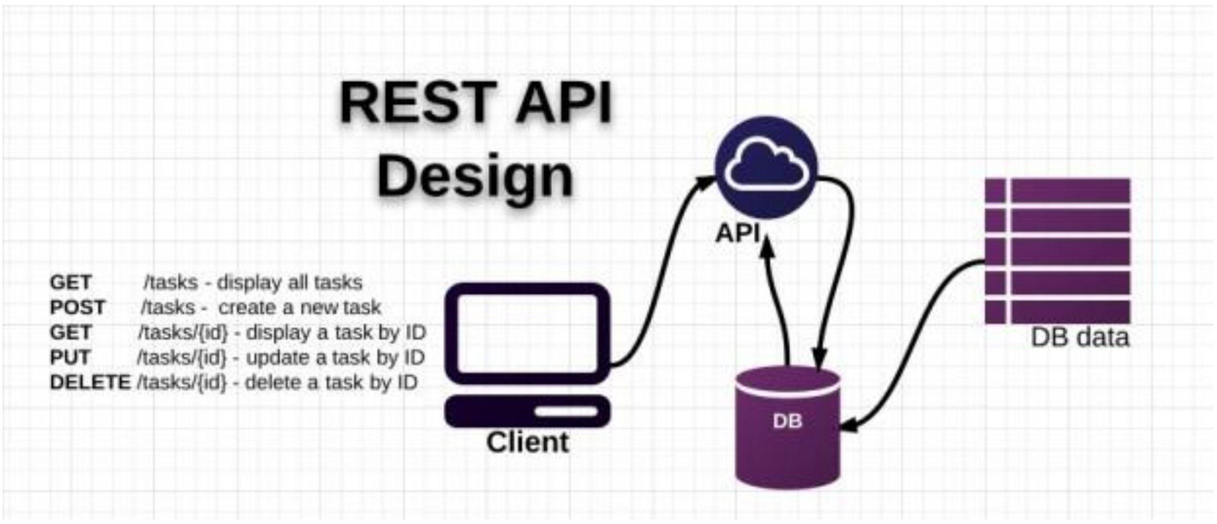

REST API
Design
GET /tasks - display all tasks
POST /tasks - create a new task
GET /tasks/{id} - display a task by ID
PUT /tasks/{id} - update a task by ID
DELETE /tasks/{id} - delete a task by ID
Client
API
DB
DB data

Figure 1.6. Introduction to RESTful API

1.4.2. RESTful API workflow

REST primarily operates based on the HTTP protocol. The basic operations mentioned above will use specific HTTP methods:

- GET (SELECT): Retrieves a Resource or a list of Resources.
- POST (CREATE): Creates a new Resource.

- PUT (UPDATE): Updates information for a Resource.
- DELETE (DELETE): Deletes a Resource.

These methods or operations are commonly referred to as CRUD, corresponding to Create, Read, Update, Delete actions.

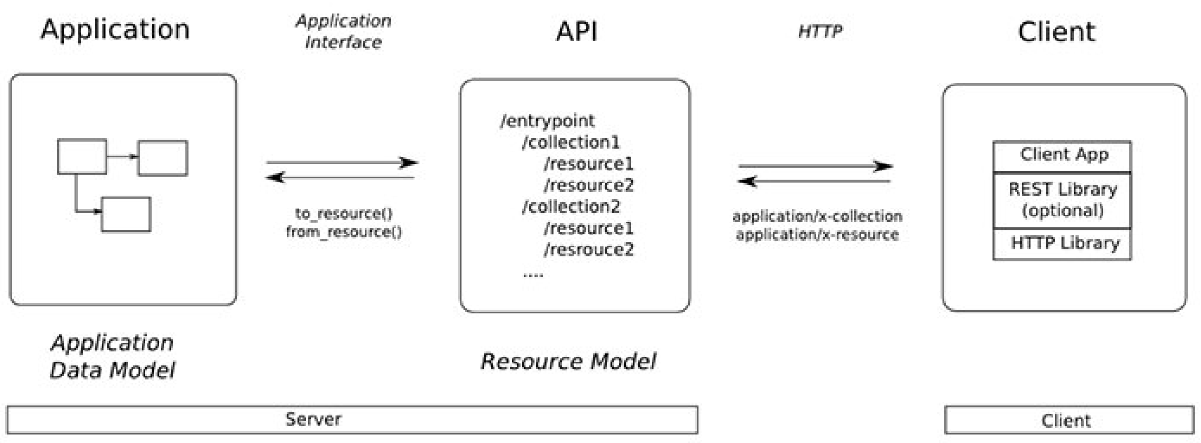


Figure 1.7. Workflow of RESTful API

1.4.3. Status codes

When we request an API, there are several status codes to identify the outcome:

- 200 OK – Successful response for GET, PUT, PATCH, or DELETE methods.
- 201 Created – Returned when a Resource has been successfully created.
- 204 No Content – Returned when a Resource has been successfully deleted.
- 304 Not Modified – Client can use cached data.
- 400 Bad Request – Invalid request.
- 401 Unauthorized – Authentication is required for the request.
- 403 Forbidden – Access is forbidden.
- 404 Not Found – Resource not found at the requested URI.
- 405 Method Not Allowed – Method is not allowed for the current user.
- 410 Gone – Resource is no longer available, old version no longer supported.
- 415 Unsupported Media Type – Resource type not supported.
- 422 Unprocessable Entity – Data could not be validated.
- 429 Too Many Requests – Request rejected due to rate limiting.

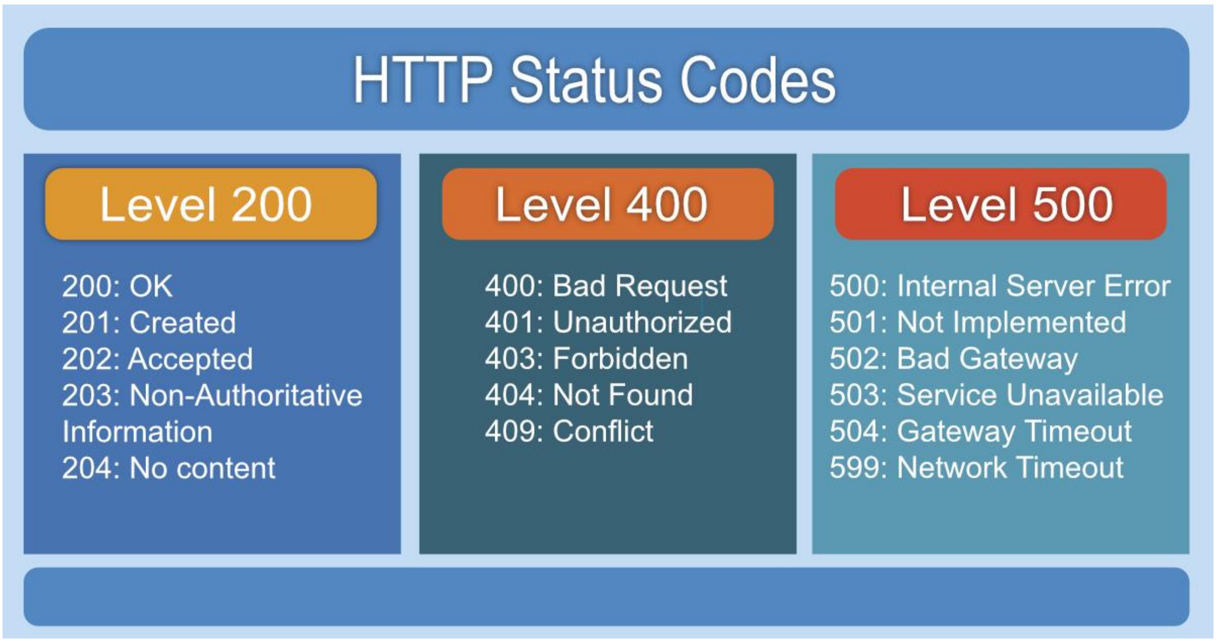


Figure 1.8. HTTP Status Codes

1.5. Google Maps API

The Google Maps API is a set of application programming interfaces provided by Google that allows developers to incorporate Google Maps features and data into their applications. To use the Google Maps API in their application, developers need to register and have an account on the Google Cloud Platform. They will then receive an API key to access the API's features. Additionally, Google provides a variety of documentation and examples to guide developers on how to use the API and integrate it into their applications.

The Google Maps API is a powerful set of tools and services provided by Google that enable developers to integrate various mapping features and geographic data into their applications. Here are some key aspects and functionalities of the Google Maps API:

- **Map Display**: Developers can embed interactive maps directly into their websites or mobile apps. These maps can be customized in terms of style, markers, overlays, and layers to suit specific needs.
- **Geocoding**: The API provides geocoding services, allowing developers to convert addresses (forward geocoding) or coordinates (reverse geocoding) into geographic locations on the map.

- **Directions and Routing**: Developers can incorporate routing and navigation functionalities into their applications, enabling users to get directions between two or more locations, including options for different modes of transportation like driving, walking, or public transit.
- **Places Search**: The API offers powerful place search capabilities, allowing users to search for various types of places (such as restaurants, businesses, landmarks) within a specified area or along a route.

Figure 1.9. Introduction to Google Maps API

1.6. Chapter conclusion

In this chapter, I have provided the theoretical foundation and all the technology used to build the application.

CHAPTER 2: SYSTEM ANALYSIS AND DESIGN

2.1. Project requirements analysis

2.1.1. Main features of the project

Currently, there are existing applications for finding and booking parking spaces on the market, such as My Parking developed by Viettel, which is already in use. After using and referencing this application, I have identified some basic functionalities for my project like My Parking as follows:

- Support for search and booking by time, and quick, convenient online payment.
- Integration with Google Maps, allowing users to easily search for and accurately locate parking spots.
- Availability of multiple parking locations in densely populated areas, especially shopping centers, making it easy for users to find parking spots.

Additionally, I propose to improve my application with several distinctive features:

- Unified application for both users and parking lot owners: Unlike My Parking, which is divided into two separate applications—one for parking lot owners and one for users searching for parking spots—my application will provide both functionalities in a single app, allowing users to switch roles easily.

Support for families with unused space: My application will enable households with unused space to generate income by creating their own parking spots on the map for users to choose from.

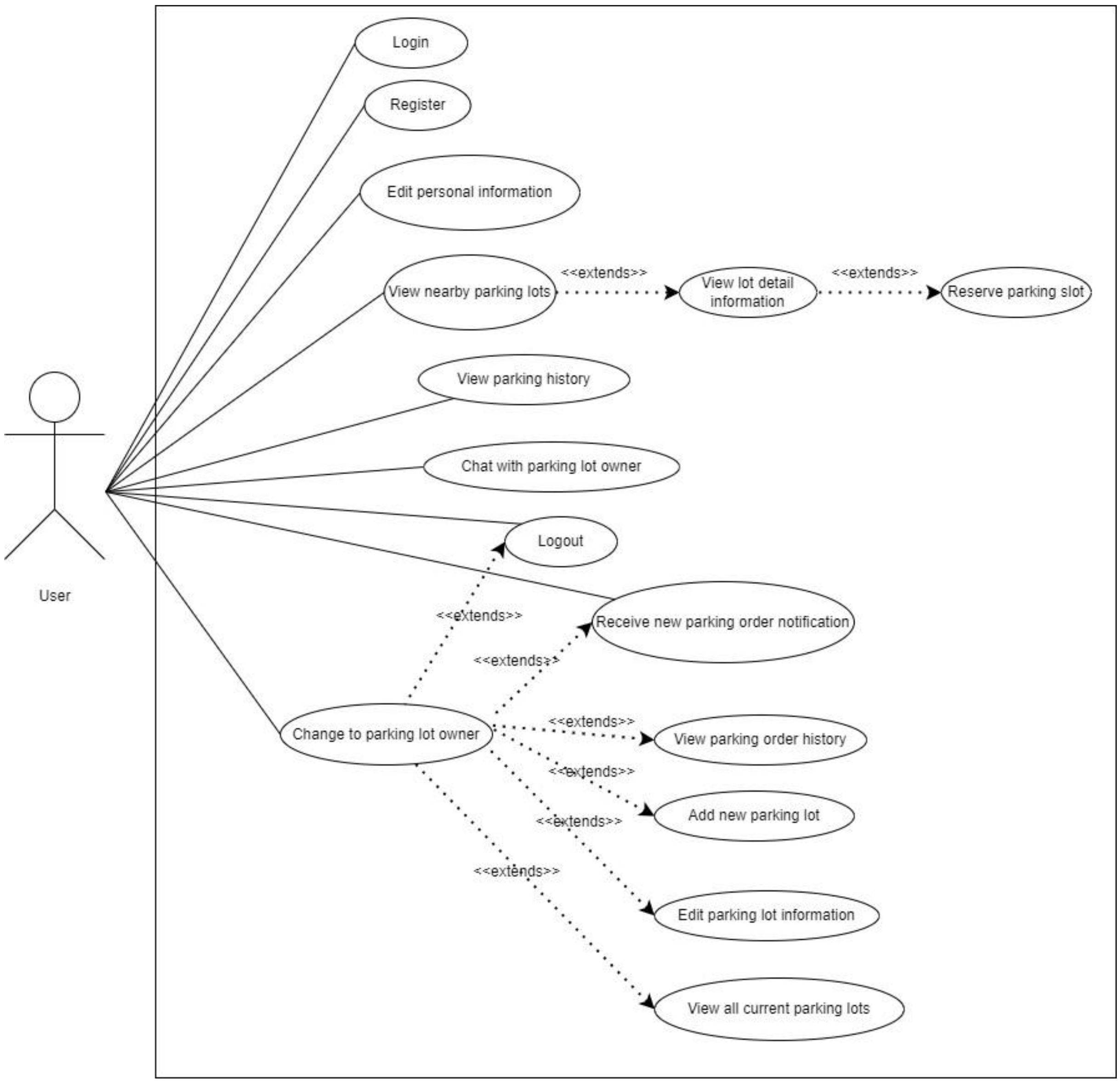
2.1.2. Target users

The application is designed for two primary types of users: general users and parking owners.

- General users: These are the everyday users who interact with the application to find and book parking spaces. They can view detailed information about available parking spaces, including location, availability, and price. Users can book a parking spot and make payments directly through the application.
- Parking locations owners: These are individuals or businesses that own parking spaces and wish to list them on the application. Owners can add new parking spots to the system and edit details of existing spots.

2.2. System Design Analysis Diagram

- System Use-case diagram



User
Login
Register
Edit personal information
View nearby parking lots
<<extends>>
View lot detail information
<<extends>>
Reserve parking slot
View parking history
Chat with parking lot owner
Logout
Receive new parking order notification
Change to parking lot owner
<<extends>>
View parking order history
Add new parking lot
Edit parking lot information
View all current parking lots

Figure 2.1. System Use-case diagram

- Use-case Adding new parking lot

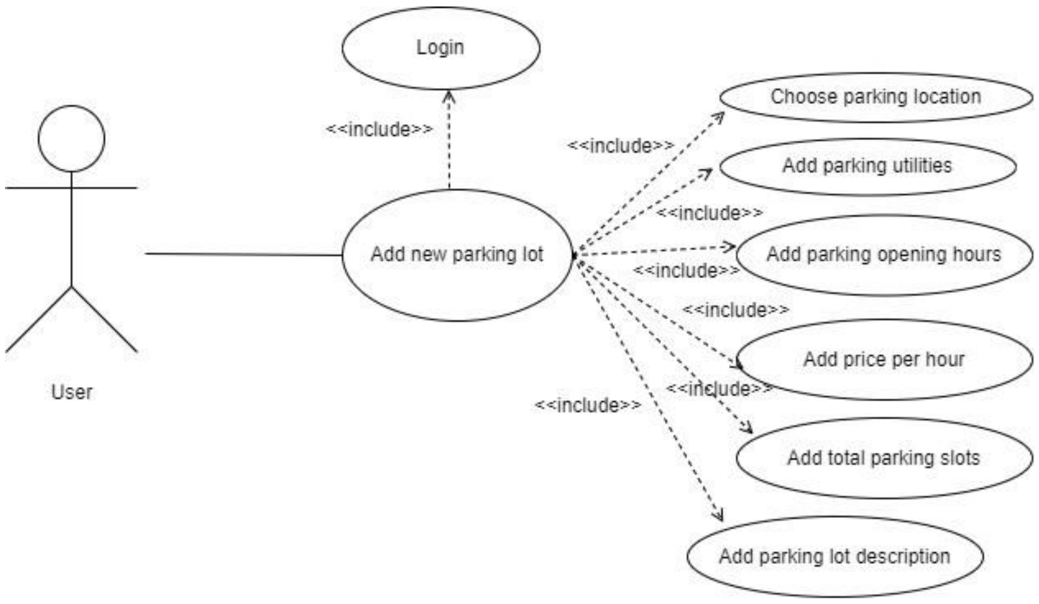


Figure 2.2. Use-case Adding new parking lot

- Use-case Editing personal information

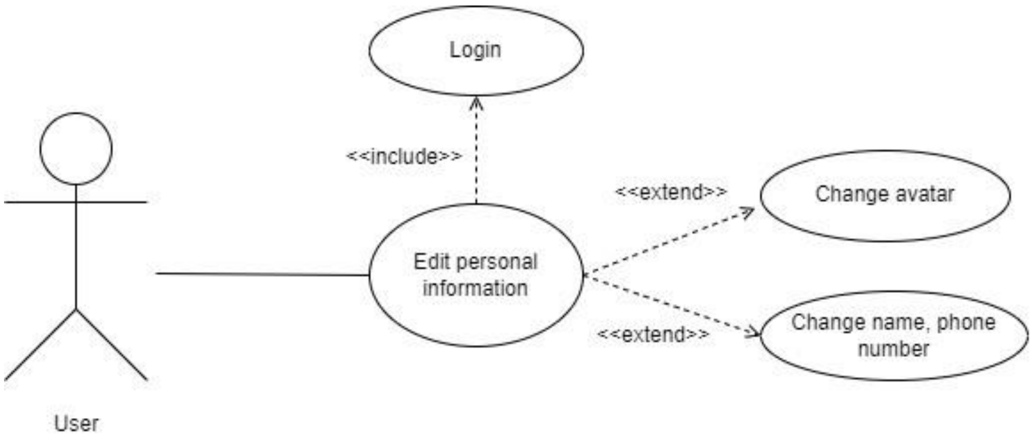


Figure 2.3. Use-case Editing personal information

- Use-case Viewing parking lot detail

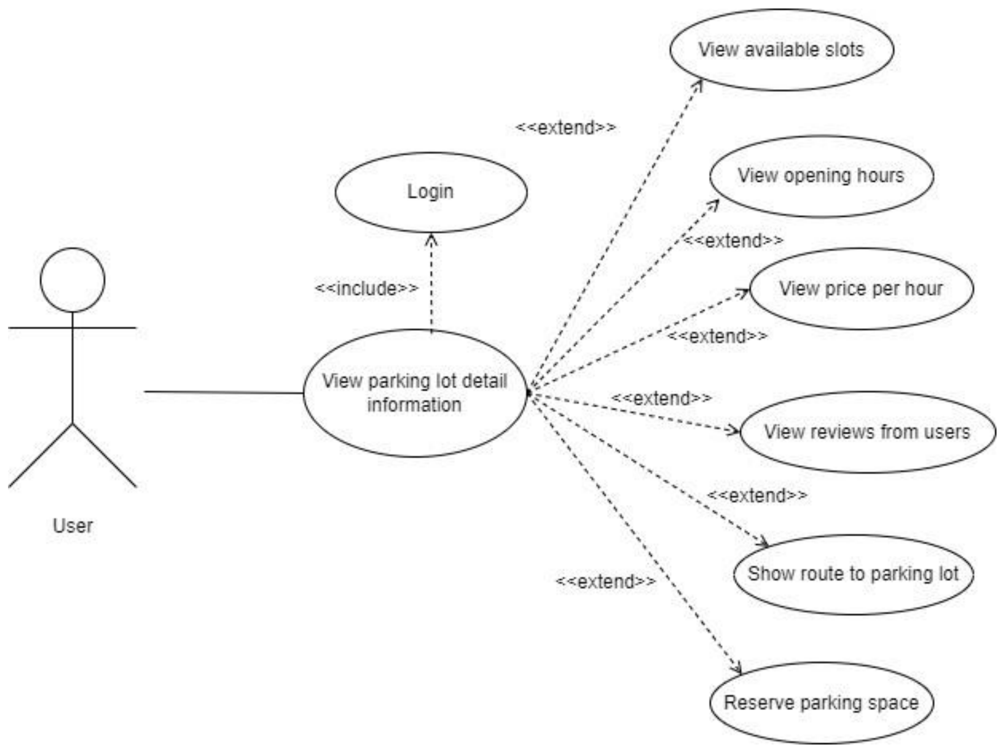


Figure 2.4. Use-case Viewing parking lot detail

- Use-case Reserving lot

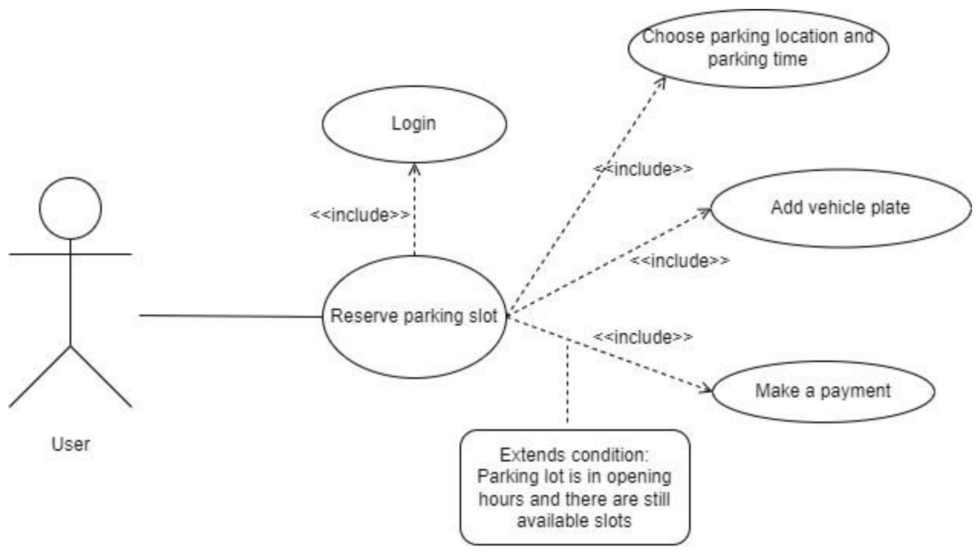

Choose parking location and parking time
Login
<<include>>
Add vehicle plate
<<include>>
<<include>>
Reserve parking slot
<<include>>
Make a payment
User
Extends condition: Parking lot is in opening hours and there are still available slots

Figure 2.5. Use-case Reserving lot

2.3. Business processes

- Login process

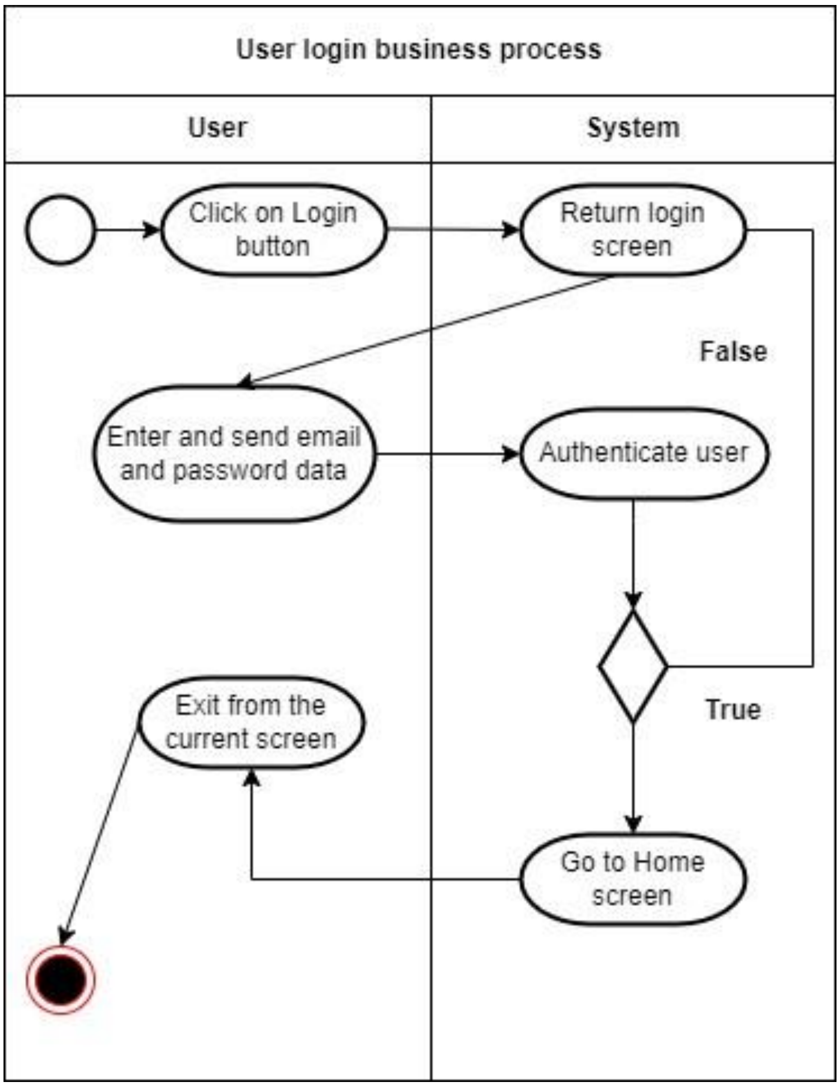

User login business process
User
System
Click on Login button
Return login screen
False
Enter and send email and password data
Authenticate user
Exit from the current screen
True
Go to Home screen

Figure 2.1. Login activity diagram

- Add new parking lot process

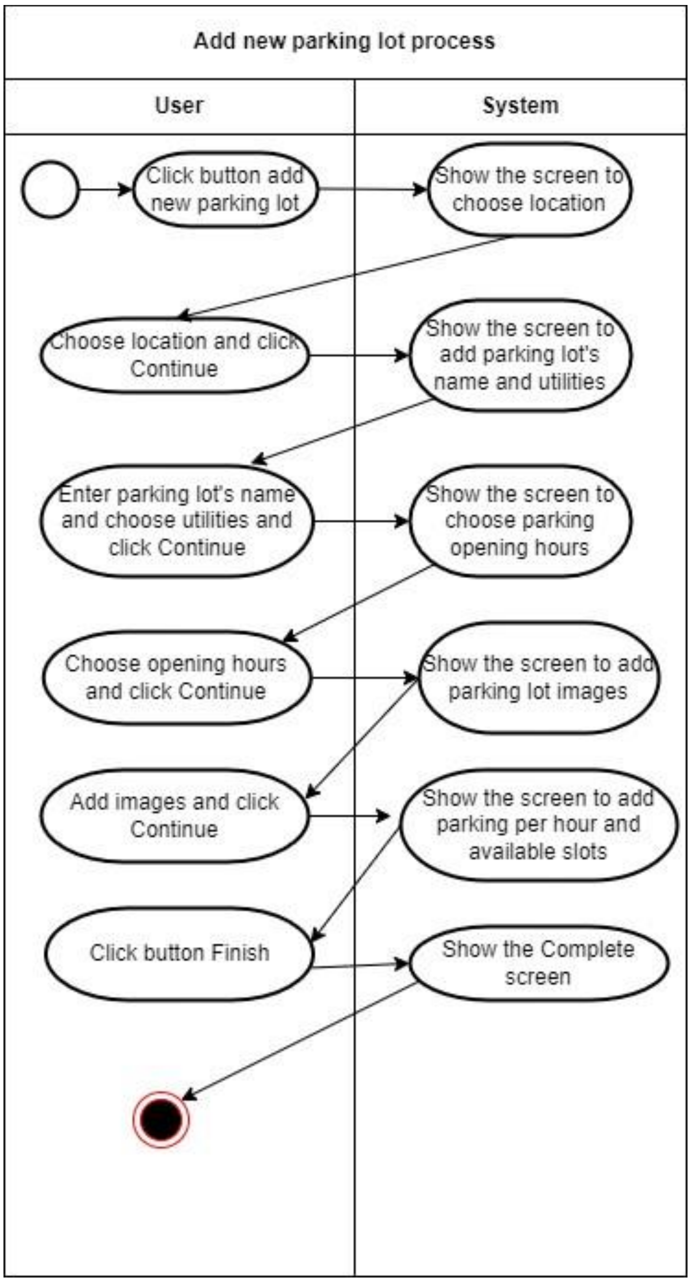

Add new parking lot process
User
System
Click button add new parking lot
Show the screen to choose location
Choose location and click Continue
Show the screen to add parking lot's name and utilities
Enter parking lot's name and choose utilities and click Continue
Show the screen to choose parking opening hours
Choose opening hours and click Continue
Show the screen to add parking lot images
Add images and click Continue
Show the screen to add parking per hour and available slots
Click button Finish
Show the Complete screen

Figure 2.2. Add new parking lot activity diagram

- Reserve parking slot process

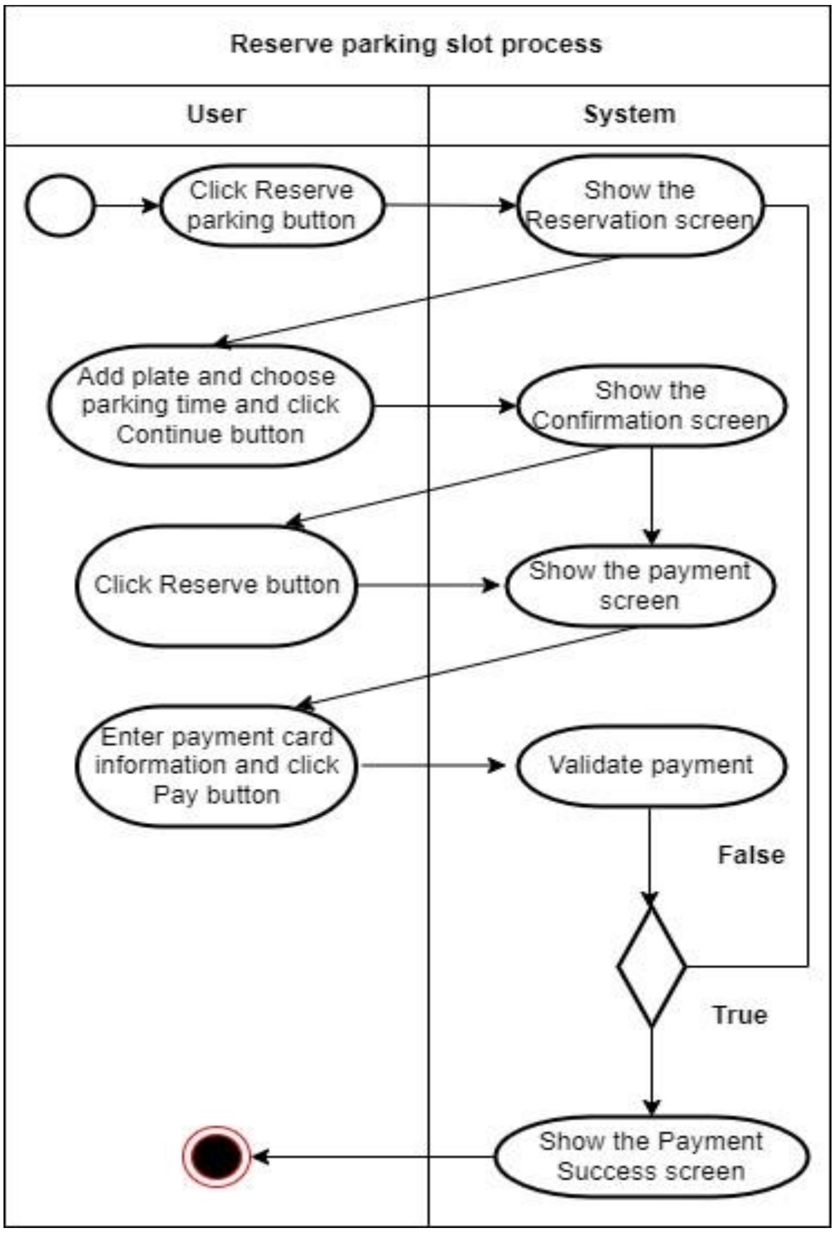

Reserve parking slot process
User
System
Click Reserve parking button
Show the Reservation screen
Add plate and choose parking time and click Continue button
Show the Confirmation screen
Click Reserve button
Show the payment screen
Enter payment card information and click Pay button
Validate payment
False
True
Show the Payment Success screen

Figure 2.3. Reserve parking lot activity diagram

- Find nearby parking lots process

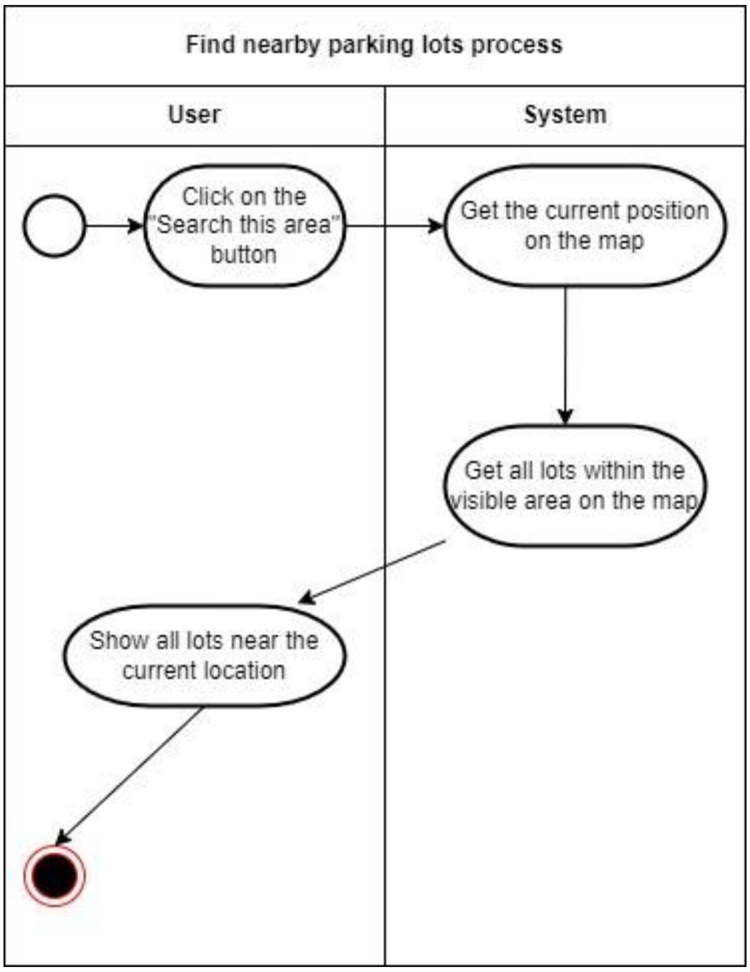

Find nearby parking lots process
User
System
Click on the "Search this area" button
Get the current position on the map
Get all lots within the visible area on the map
Show all lots near the current location

Figure 2.9. Find nearby parking lots activity diagram

2.4. Use-case specification

- Login use case specification

Use case name	Login		
Actor	User		
Pre-condition	User already have account		
Purpose	Allow logging in and using application with specific functions.		
Triggered event	Click on the Login button after entering email and password.		
Main path	Order of action	Performer	Action
	1.	User	Click the Login button
	2.	System	Navigate to the Login screen

	3.	User	Enter email and password
	4.	User	Click the login button
	5.	System	Validate email and password
	6.	System	Navigate to the Home screen
Alternative path	5A.	System	Show error message: email and password must not be empty.
	5B.	System	Show error message: email is not registered yet.
	5C.	System	Show error message: wrong password
	5D.	System	Show error message: incorrect email format
Post condition	Login success and move to the home screen		

Table 2.1. Login Use-case specification

- Add new parking lot specification

Use case name	Add new parking lot		
Actor	User		
Pre-condition	User is registered as a parking lot owner		
Purpose	Allow user to add new parking lot.		
Triggered event	User clicks the “Add new parking lot” button.		
Main path	Order of action	Performer	Action
	1	User	Click the “Add new parking lot” button
	2	System	Navigate to the add parking lot screen.
	3	User	Choose the current location or the desirable location.
	4	User	Click the “Continue” button

	5	System	Navigate to the “Add name and utilities” screen.
	6	User	Enter parking lot name and choose utilities for the parking lot and click “Continue” button.
	7	System	Navigate to the “Add time and images” screen.
	8	User	Add parking hours and images for the parking lot and click “Continue” button.
	9	System	Navigate to the “Add price and available slots” screen.
	10	User	Enter price per hour and available slots and click “Continue” button.
	11	System	Show the preview of the parking lot.
	12	User	Click the “Confirm” button to finish adding new parking lot.
	13	System	Navigate to the “Add parking lot complete” screen.
Alternative path	7A	System	Show error message: parking lot name and utilities must not be empty.
	9A	System	Show error message: images must be selected.

	11A	System	Show error message: price and available slots must not be empty.
Post condition	Add new parking lot successful and move to the home screen		

Table 2.2. Add new parking lot specification

- Reserve parking lot specification

Use case name	Reserve parking lot		
Actor	User		
Pre-condition	User already have an account		
Purpose	Allow user to reserve a place at a parking lot.		
Triggered event	User clicks the “Order lot” button.		
Main path	Order of action	Performer	Action
	1	User	Click the “Order lot” button
	2	System	Navigate to the view parking lot detail screen.
	3	User	Enter the parking duration, vehicle plate and click on the “Pay now” button.
	4	System	Navigate to the Order confirmation screen.
	5	User	Click the “Pay now” button.
	6	System	Navigate to the Payment with MOMO screen.
	7	User	Enter the card number, valid time, card holder name and click the “Confirm payment” button.
	8	System	Navigate to the Payment successful screen.

Alternative path	3A	System	Show error message: Vehicle plate must not be empty.
	8A	System	Show error message: Card information incorrect.
Post condition	Reserve parking lot space successful and move to the Show direction to parking location.		

Table 2.3. Reserve parking lot specification

- Find nearby parking lots specification

Use case name	Find nearby parking lot		
Actor	User		
Pre-condition	User already have an account		
Purpose	Allow user to search for nearby parking lots		
Triggered event	User clicks the “Search this area” button.		
Main path	Order of action	Performer	Action
	1	User	Click the “Search this area” button.
	2	System	Get the current location of the user.
	3	System	Find all the parking lots that is within the visible area of the maps and show it on the map screen.
	4	User	View all the parking lots in the current area.
Alternative path	3A	System	If there’s no parking lots nearby, show the maps with no parking lots.
Post condition	Show all the parking lots in the current area.		

Table 2.4. Find nearby parking lots specification

2.5. Database design

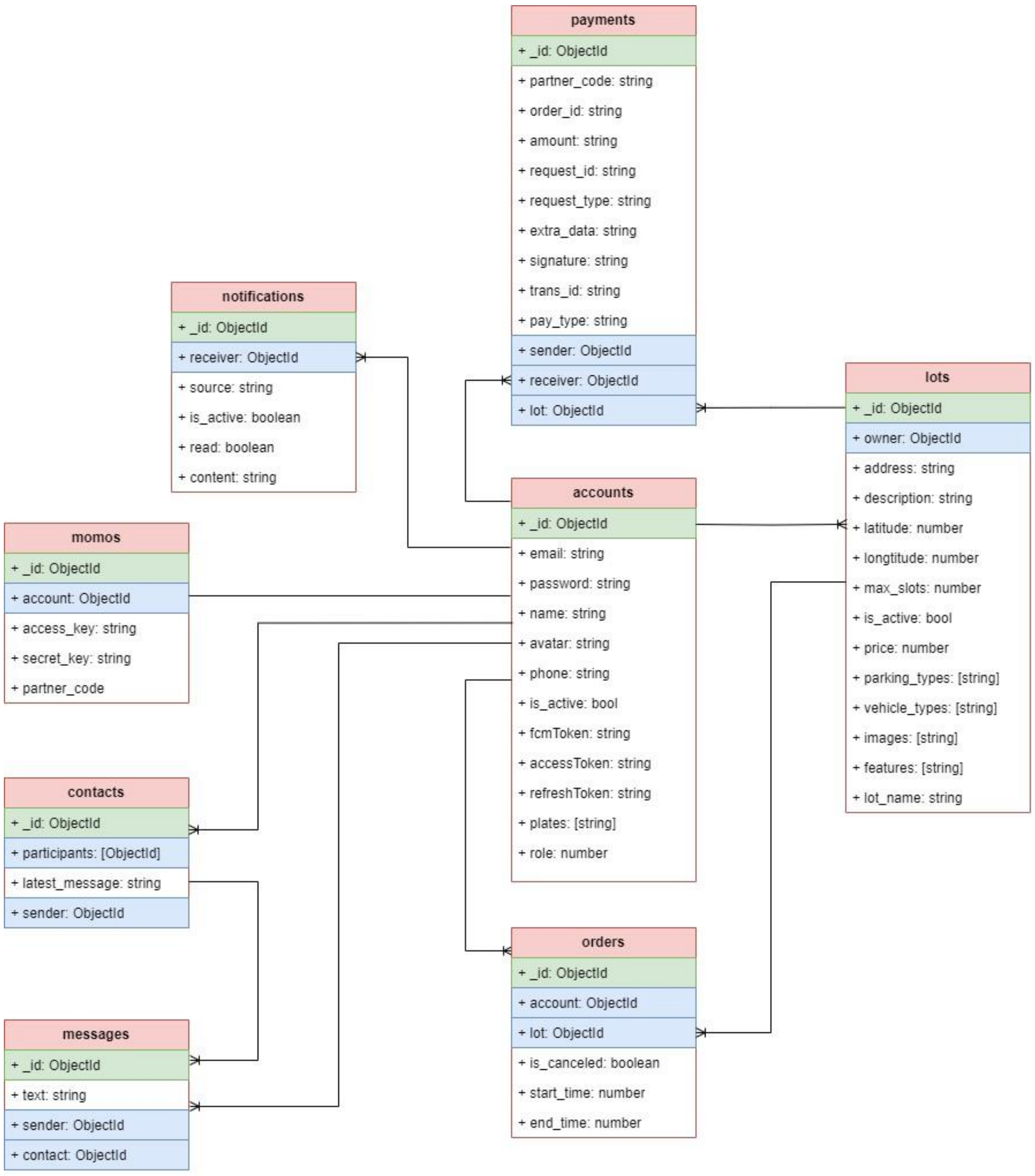


Figure 2.9. Database design

- Table Accounts

Field	Data type	Constraint	Explanation
_id	ObjectId	Primary key	Mongo Id
email	String		Account email

name	String		Account name
password	String		Account password
phone	String		Phone number
avatar	String		Account avatar
is_active	Boolean		Delete flag + true: if account has been deleted + false: if account has not been deleted
fcmToken	String		Token to send push notification
role	Enum		Role in the application (User/Owner)
refreshToken	String		Can be used to refresh access token
plates	[String]		List of plates of vehicles of user

Table 2.5. Table Accounts

- Table Lots

Field	Data type	Constraint	Explanation
_id	ObjectId	Primary key	Mongo Id
owner	String		Owner of the lot
address	String		Lot address
description	String		Lot description
latitude	String		Lot latitude
longitude	String		Lot longitude
max_slots	Boolean		Total available slots of the lot
is_active	String		Delete flag + true: if account has been deleted + false: if account has not been deleted
price	Enum		Rental price per hour

parking_types	String		Type of the parking lot
vehicle_types	[String]		List of plates of vehicles of user
images	[String]		List of images of the lot
features	[String]		List of features of the lot
lot_name	String		Lot name
start_time	String		Start parking time of the lot
end_time	String		End parking time of the lot
order_count	Number		Total orders of the lot
review_count	Number		Total reviews of the lot
review_avg	Number		Average review of the lot

Table 2.6. Table Lots

- Table Orders

Field	Data type	Constraint	Explanation
_id	ObjectId	Primary key	Mongo Id
account	ObjectId		Order of an account
lot	ObjectId		Order at a specific lot
is_canceled	Boolean		Cancel flag: + true: if order has been canceled + false: if order has not been canceled
plate	String		Plate of the vehicle
total_price	Number		Total price of the order
start_time	String		Start order parking time
end_time	String		End order parking time

Table 2.7. Table Orders

- Table Payments

Field	Data type	Constraint	Explanation
_id	ObjectId	Primary key	Mongo Id
partner_code	String		Partner code
order_id	String		Order id of payment
amount	String		Amount of payment
request_id	String		Request id
request_type	String		Request type
extra_data	String		Extra data
signature	String		Signature of payment
trans_id	String		Transfer id
pay_type	String		Pay type
sender	ObjectId		Sender of payment
receiver	ObjectId		Receiver of payment
lot	ObjectId		Lot

Table 2.8. Table Payments

- Table Momos

Field	Data type	Constraint	Explanation
_id	ObjectId	Primary key	Mongo Id
account	ObjectId		Current account
access_key	String		Access key to momo
secret_key	String		Secret key to momo
partner_code	String		Partner code

Table 2.9. Table Momos

- Table Notifications

Field	Data type	Constraint	Explanation
_id	ObjectId	Primary key	Mongo Id
account	ObjectId		Current account
access_key	String		Access key to momo
secret_key	String		Secret key to momo

partner_code	String		Partner code

Table 2.10. Table Notification

- Table Contacts

Field	Data type	Constraint	Explanation
_id	ObjectId	Primary key	Mongo Id
participants	[ObjectId]		List of participants
latest_message	String		Latest message of a contact
sender	ObjectId		Sender id
created_at	String		Time created contact
updated_at	String		Time updated contact

Table 2.11. Table Contacts

- Table Message

Field	Data type	Constraint	Explanation
_id	ObjectId	Primary key	Mongo Id
text	String		Text message
sender	ObjectId		Sender of the message
contact	ObjectId		Contact of the message
created_at	String		Time created message
updated_at	String		Time updated message

Table 2.12. Table Message

2.6. Chapter conclusion

This chapter provides an overview of the technical and business requirements that the system must meet. It covers the process of designing and building the system, presents several basic diagrams that illustrate the main functions of the system, database design and explanation of all tables in the database.

CHAPTER 3: BUILDING APPLICATION AND TESTING

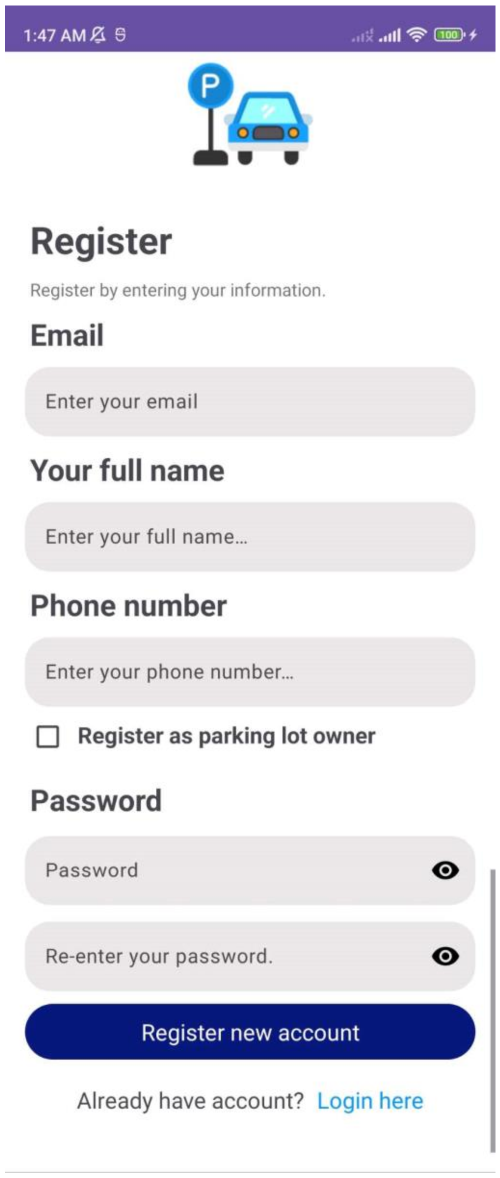
3.1. Building application

- Tools and libraries

Tools/Libraries	Purpose	URL
Android Studio	IDE	https://developer.android.com/studio
Kotlin for Android	Programming Language to build android application	https://kotlinlang.org/
Android XML	Building UI	https://developer.android.com/codelabs/basic-android-kotlin-training-xml-layouts?hl=vi#0
Firebase	Authenticating and storing data	https://firebase.google.com/docs/android/learn-more
Retrofit	Make http request to get data	https://square.github.io/retrofit/
Google Maps	Showing location and route	https://developers.google.com/maps/documentation/android-sdk/start

Table 3.1. Table Tools and Libraries

- Illustration of Key Functions


1:47 AM

Register

Register by entering your information.

Email

Enter your email

Your full name

Enter your full name...

Phone number

Enter your phone number...

☐ Register as parking lot owner

Password

Password

Re-enter your password.

Register new account

Already have account? Login here

Figure 3.1. Register new account page

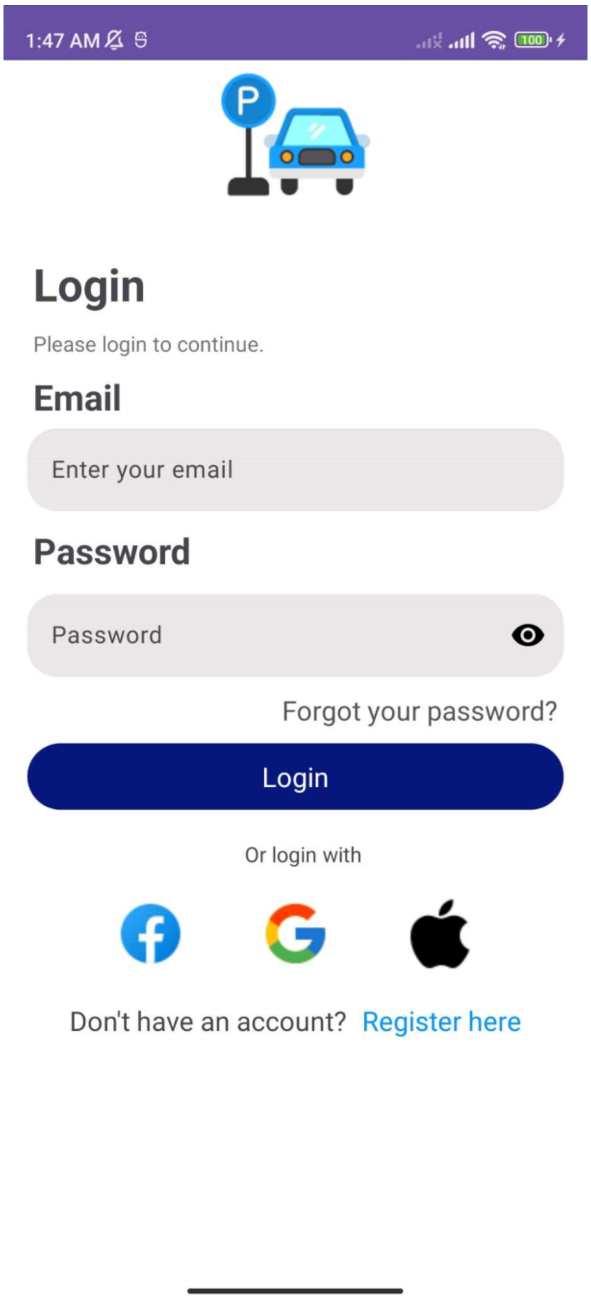
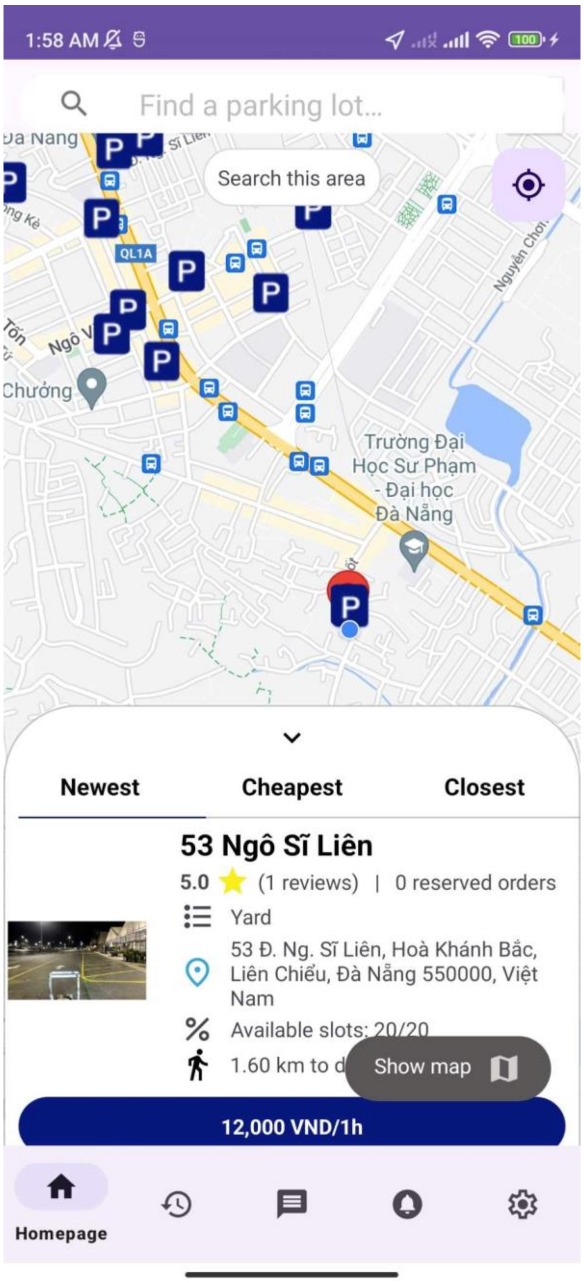


Figure 3.2. Login page


1:58 AM

Find a parking lot...

Search this area

Newest Cheapest Closest

53 Ngô Sĩ Liên

5.0 (1 reviews) | 0 reserved orders

Yard

53 Đ. Ng. Sĩ Liên, Hoà Khánh Bắc, Liên Chiểu, Đà Nẵng 550000, Việt Nam

Available slots: 20/20

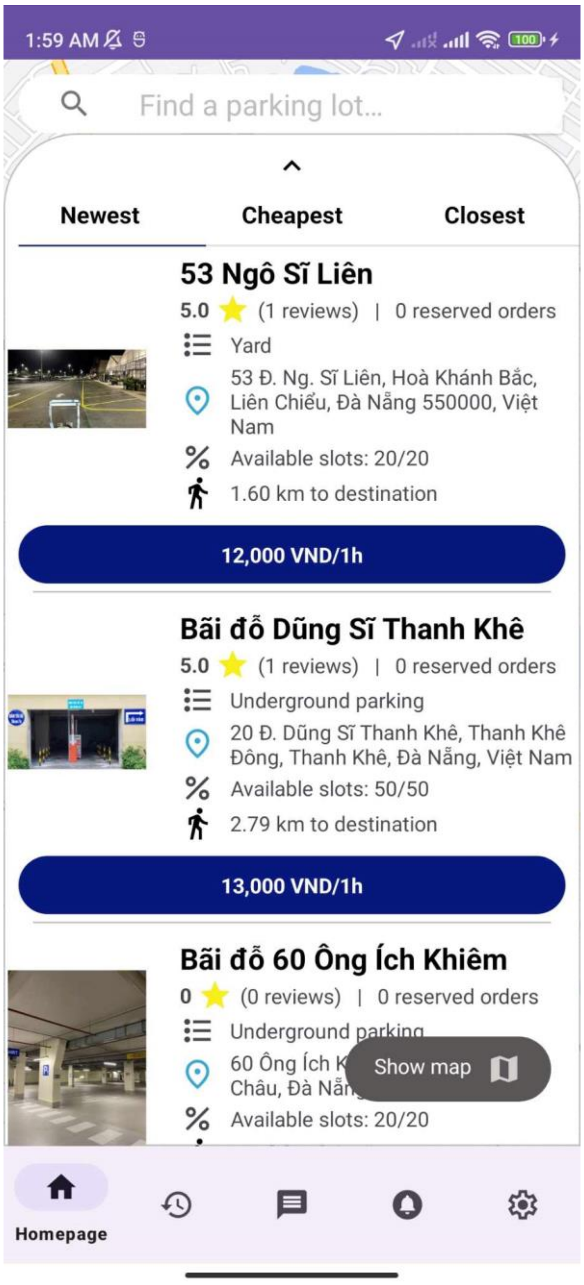
1.60 km to d

Show map

12,000 VND/1h

Homepage

Figure 3.3. Homepage

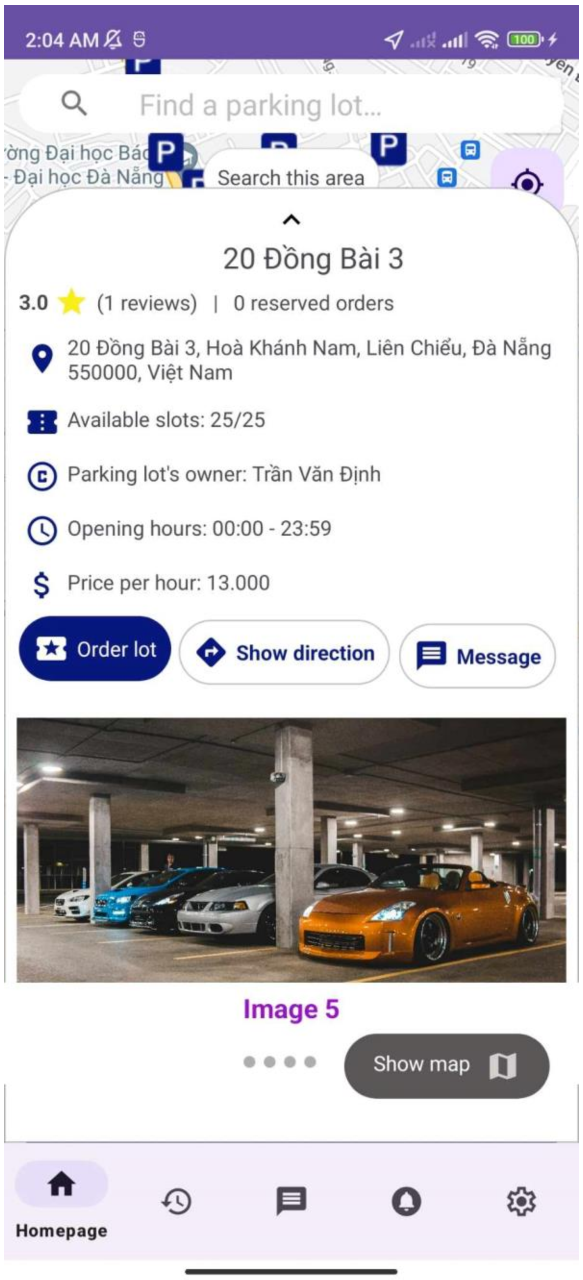


53 Ngô Sĩ Liên
5.0 (1 reviews) | 0 reserved orders
Yard
53 Đ. Ng. Sĩ Liên, Hoà Khánh Bắc, Liên Chiểu, Đà Nẵng 550000, Việt Nam
Available slots: 20/20
1.60 km to destination
12,000 VND/1h

Bãi đỗ Dũng Sĩ Thanh Khê
5.0 (1 reviews) | 0 reserved orders
Underground parking
20 Đ. Dũng Sĩ Thanh Khê, Thanh Khê Đông, Thanh Khê, Đà Nẵng, Việt Nam
Available slots: 50/50
2.79 km to destination
13,000 VND/1h

Bãi đỗ 60 Ông Ích Khiêm
0 (0 reviews) | 0 reserved orders
Underground parking
60 Ông Ích K
Châu, Đà Nẵ
Available slots: 20/20

Figure 3.4. List of parking lots


2:04 AM

Find a parking lot...

Search this area

20 Đồng Bài 3

3.0 (1 reviews) | 0 reserved orders

20 Đồng Bài 3, Hoà Khánh Nam, Liên Chiểu, Đà Nẵng 550000, Việt Nam

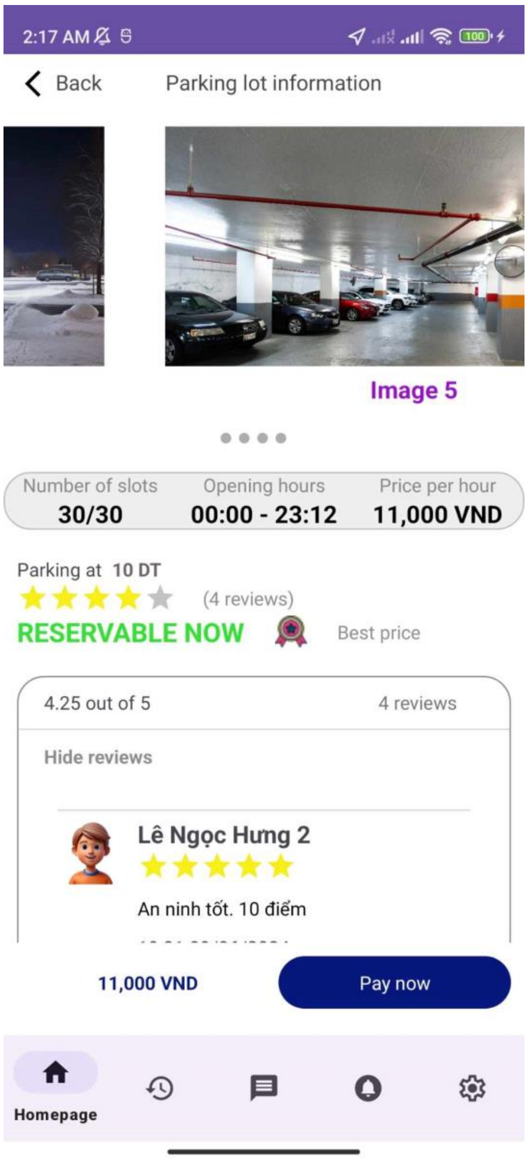
Available slots: 25/25

Parking lot's owner: Trần Văn Định

Opening hours: 00:00 - 23:59

Price per hour: 13.000

Figure 3.5. View parking lot information


2:17 AM

Back Parking lot information

Image 5

Number of slots 30/30

Opening hours 00:00 - 23:12

Price per hour 11,000 VND

Parking at 10 DT

(4 reviews)

RESERVABLE NOW

Best price

4.25 out of 5 4 reviews

Hide reviews

Lê Ngọc Hưng 2

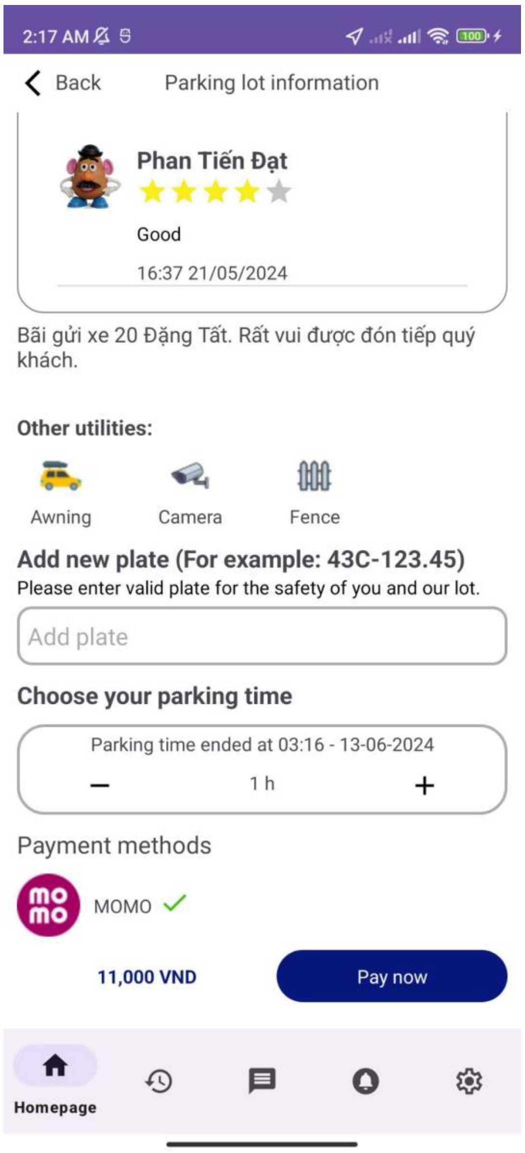
An ninh tốt. 10 điểm

11,000 VND

Pay now

Homepage

Figure 3.6. View parking lot detail (1)


2:17 AM

Back Parking lot information

Phan Tiến Đạt

★★★★☆

Good

16:37 21/05/2024

Bãi gửi xe 20 Đặng Tất. Rất vui được đón tiếp quý khách.

Other utilities:

Awning Camera Fence

Add new plate (For example: 43C-123.45)

Please enter valid plate for the safety of you and our lot.

Add plate

Choose your parking time

Parking time ended at 03:16 - 13-06-2024

— 1 h +

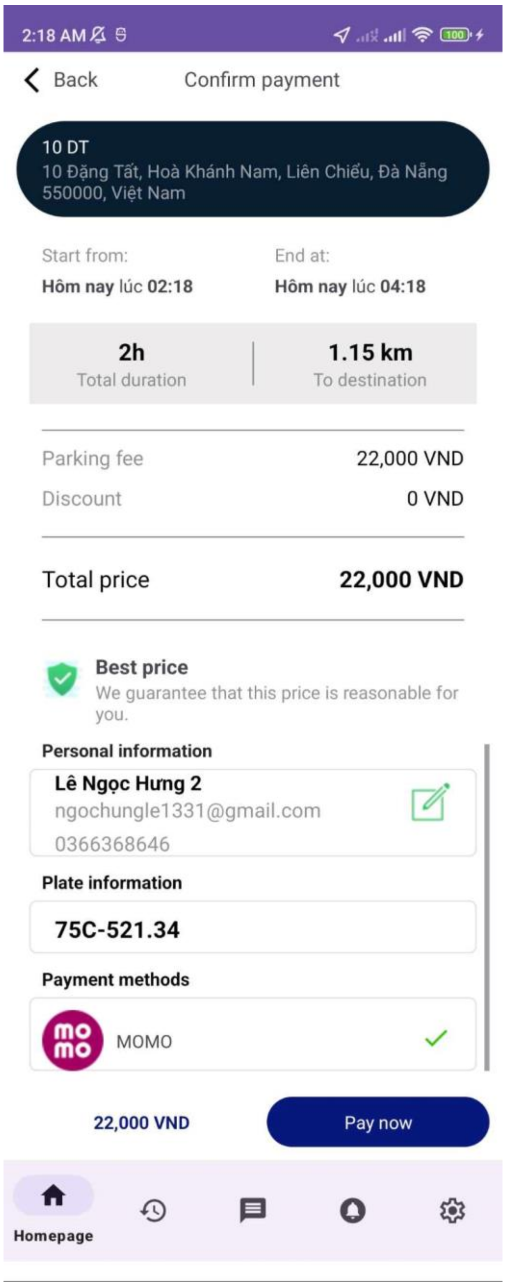
Payment methods

MOMO ✓

11,000 VND Pay now

Homepage

Figure 3.7. View parking lot detail (2)


2:18 AM

Back Confirm payment

10 DT
10 Đặng Tất, Hoà Khánh Nam, Liên Chiểu, Đà Nẵng 550000, Việt Nam

Start from: Hôm nay lúc 02:18
End at: Hôm nay lúc 04:18

2h Total duration
1.15 km To destination

Parking fee	22,000 VND
Discount	0 VND
Total price	22,000 VND

Best price
We guarantee that this price is reasonable for you.

Personal information
Lê Ngọc Hưng 2
ngochungle1331@gmail.com
0366368646

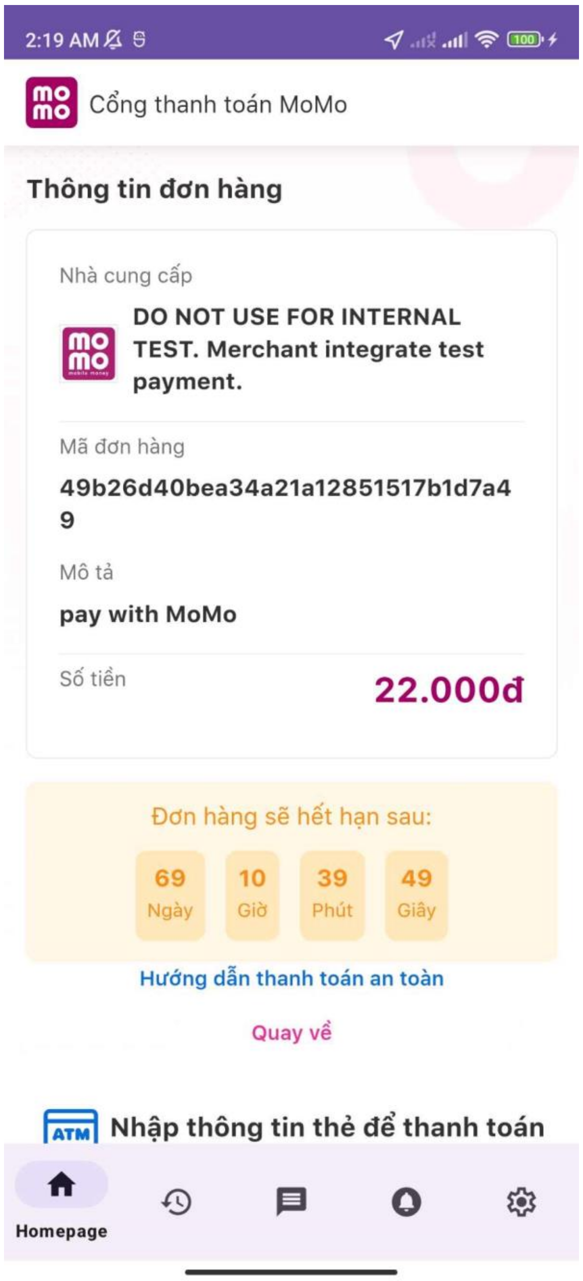
Plate information
75C-521.34

Payment methods
MOMO ✓

22,000 VND
Pay now

Homepage

Figure 3.8. Pay for parking lot order (1)


2:19 AM

Cổng thanh toán MoMo

Thông tin đơn hàng

Nhà cung cấp

DO NOT USE FOR INTERNAL TEST. Merchant integrate test payment.

Mã đơn hàng

49b26d40bea34a21a12851517b1d7a4
9

Mô tả

pay with MoMo

Số tiền 22.000đ

Đơn hàng sẽ hết hạn sau:

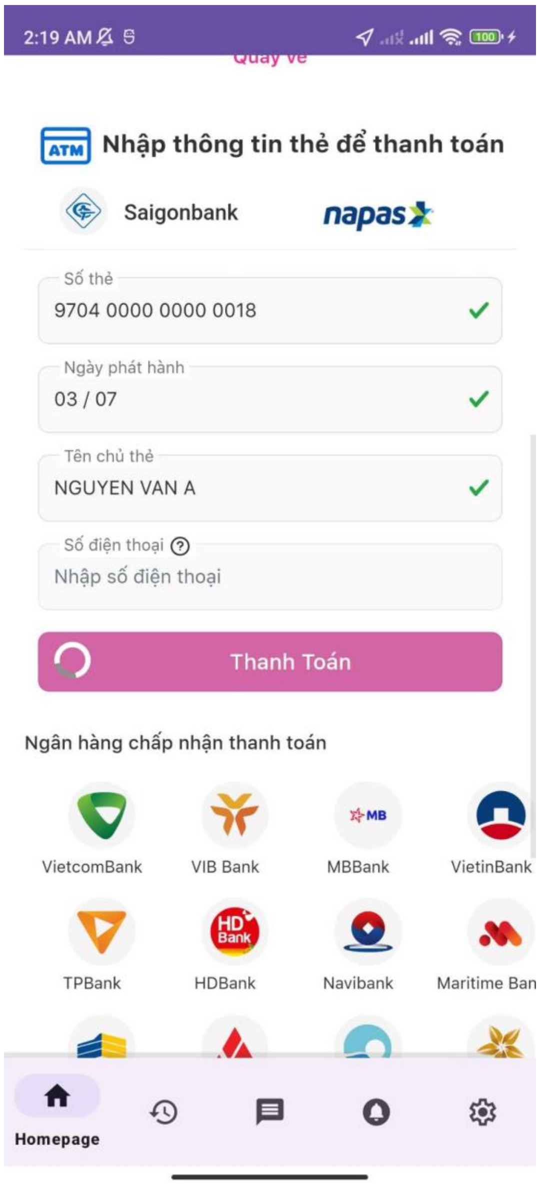
69 Ngày 10 Giờ 39 Phút 49 Giây

Hướng dẫn thanh toán an toàn

Quay về

Nhập thông tin thẻ để thanh toán

Figure 3.9. Pay for parking lot order (2)


2:19 AM

Quay về

Nhập thông tin thẻ để thanh toán

Saigonbank napas

Số thẻ: 9704 0000 0000 0018

Ngày phát hành: 03 / 07

Tên chủ thẻ: NGUYEN VAN A

Số điện thoại: Nhập số điện thoại

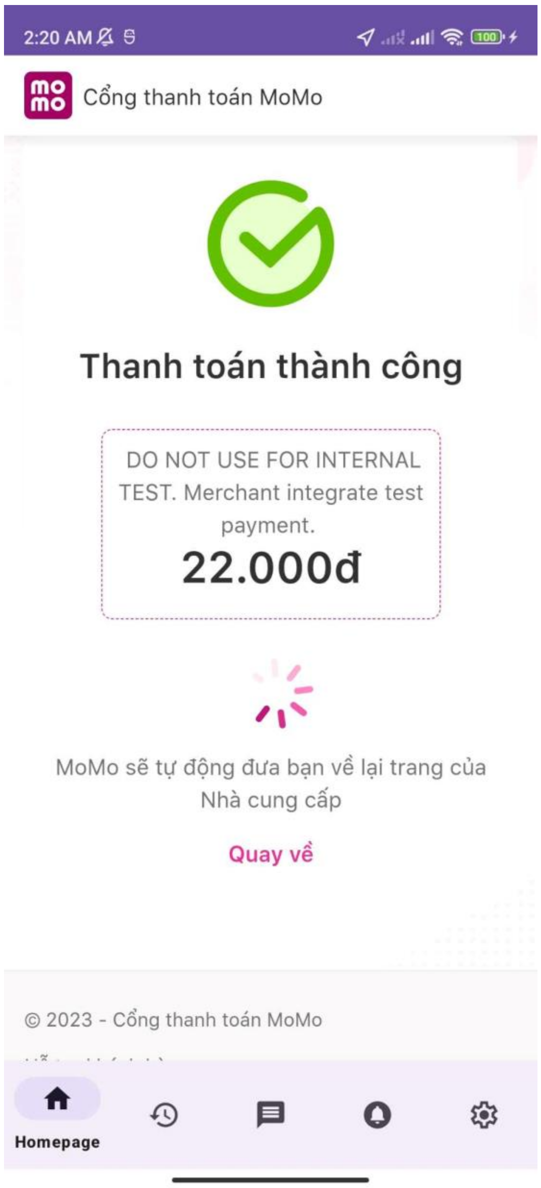
Thanh Toán

Ngân hàng chấp nhận thanh toán

VietcomBank, VIB Bank, MBBank, VietinBank, TPBank, HDBank, Navibank, Maritime Ban

Homepage

Figure 3.10. Pay for parking lot order (3)


2:20 AM

Cổng thanh toán MoMo

Thanh toán thành công

DO NOT USE FOR INTERNAL TEST. Merchant integrate test payment.

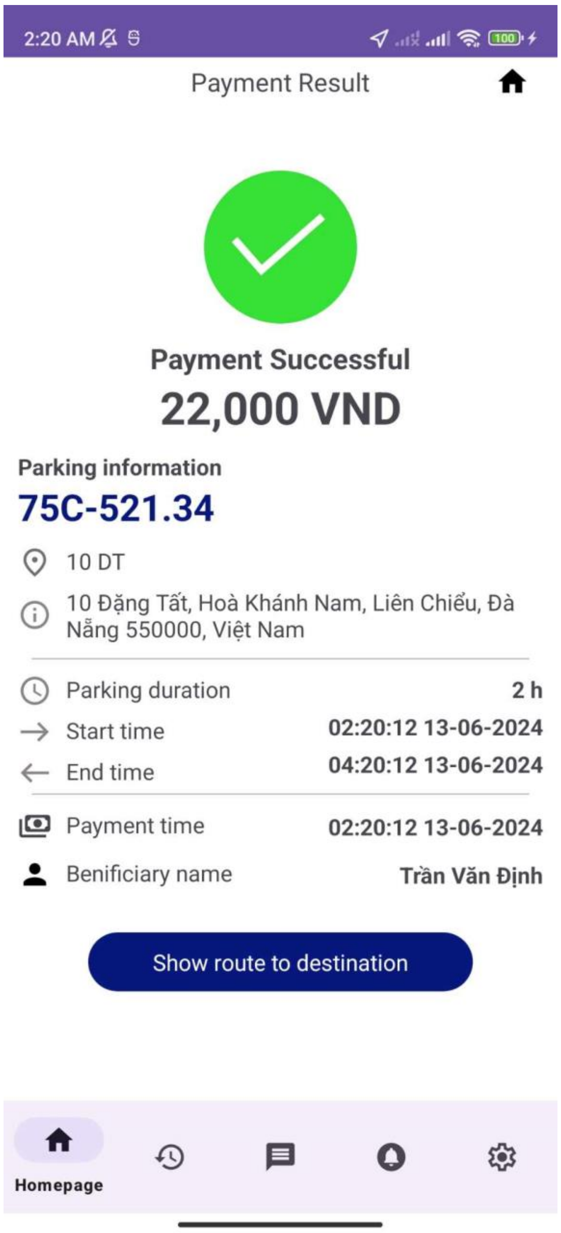
22.000đ

MoMo sẽ tự động đưa bạn về lại trang của Nhà cung cấp

Quay về

© 2023 - Cổng thanh toán MoMo

Figure 3.11. Pay for parking lot order (4)



Payment Result

Payment Successful

22,000 VND

Parking information

75C-521.34

10 DT

10 Đặng Tất, Hoà Khánh Nam, Liên Chiểu, Đà Nẵng 550000, Việt Nam

Parking duration	2 h
Start time	02:20:12 13-06-2024
End time	04:20:12 13-06-2024
Payment time	02:20:12 13-06-2024
Benificiary name	Trần Văn Định

Show route to destination

Figure 3.12. Pay for parking lot order (5)

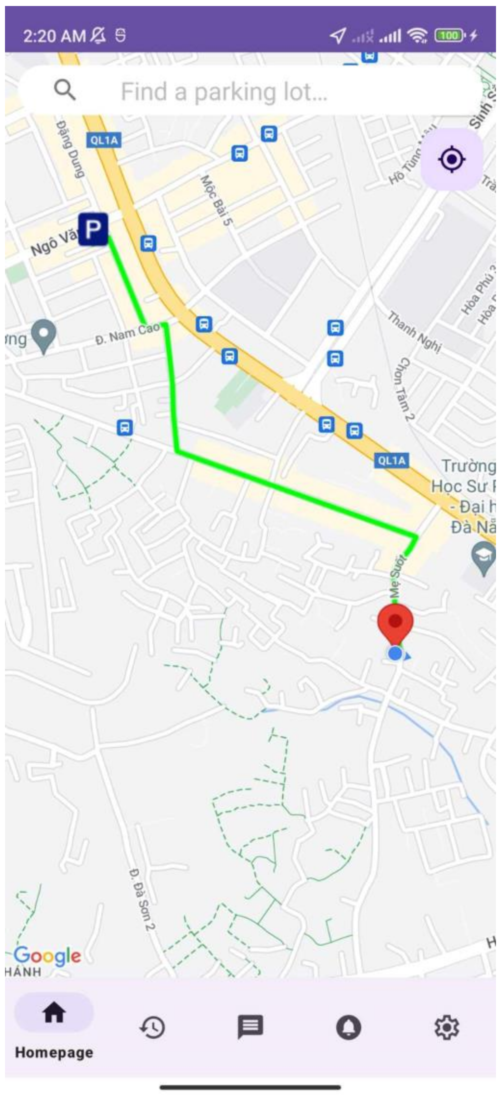
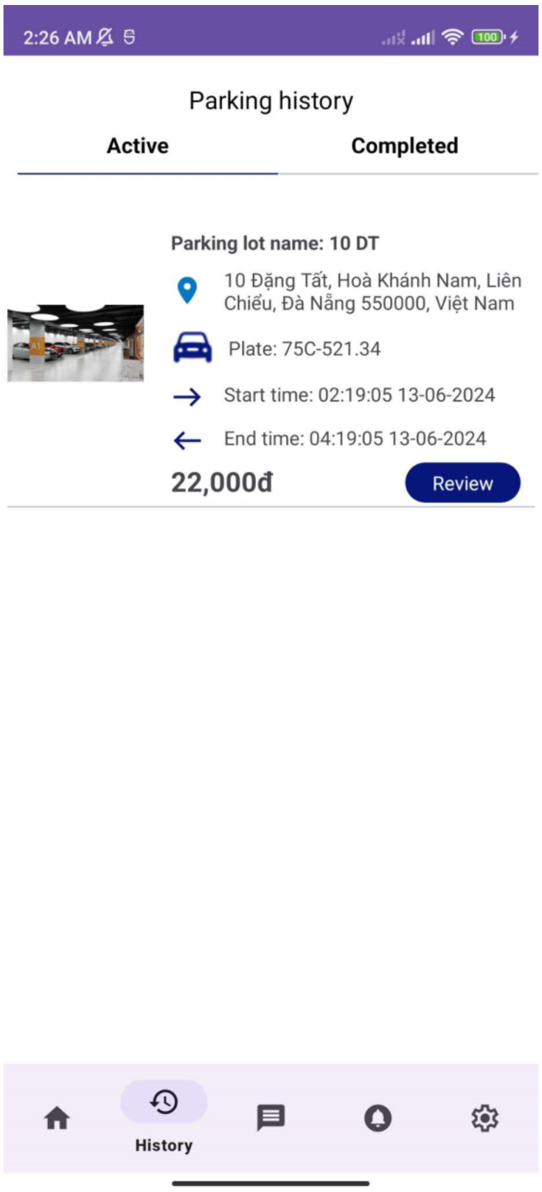


Figure 3.13. Show route to the parking location


2:26 AM

Parking history

Active | Completed

Parking lot name: 10 DT

10 Đặng Tất, Hoà Khánh Nam, Liên Chiểu, Đà Nẵng 550000, Việt Nam

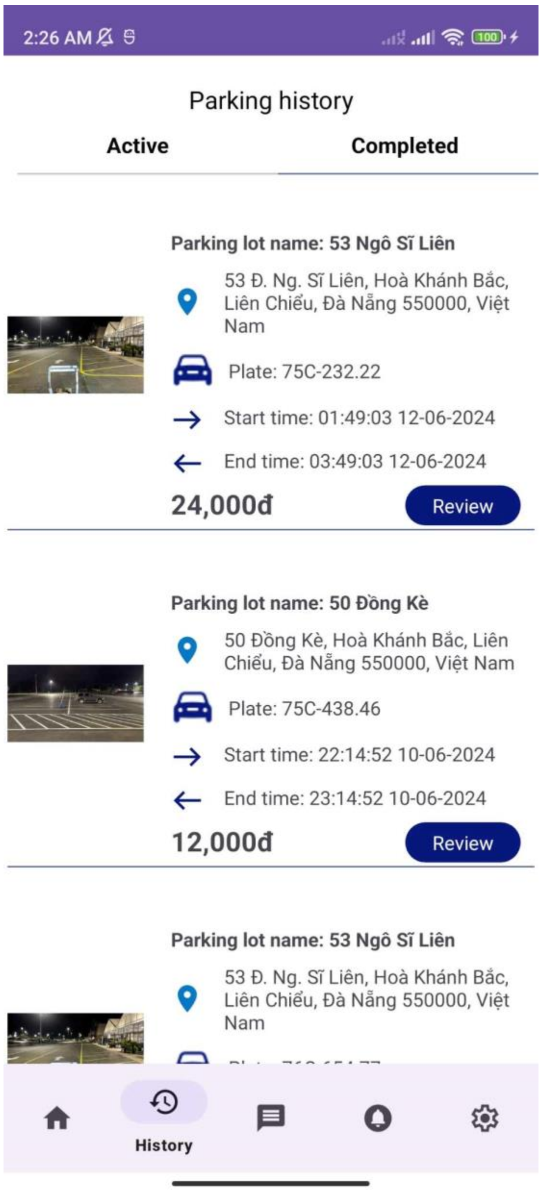
Plate: 75C-521.34

Start time: 02:19:05 13-06-2024

End time: 04:19:05 13-06-2024

22,000đ

Figure 3.14. View parking history (1)


2:26 AM

Parking history

Active Completed

Parking lot name: 53 Ngô Sĩ Liên

53 Đ. Ng. Sĩ Liên, Hoà Khánh Bắc, Liên Chiểu, Đà Nẵng 550000, Việt Nam

Plate: 75C-232.22

Start time: 01:49:03 12-06-2024

End time: 03:49:03 12-06-2024

24,000đ Review

Parking lot name: 50 Đồng Kè

50 Đồng Kè, Hoà Khánh Bắc, Liên Chiểu, Đà Nẵng 550000, Việt Nam

Plate: 75C-438.46

Start time: 22:14:52 10-06-2024

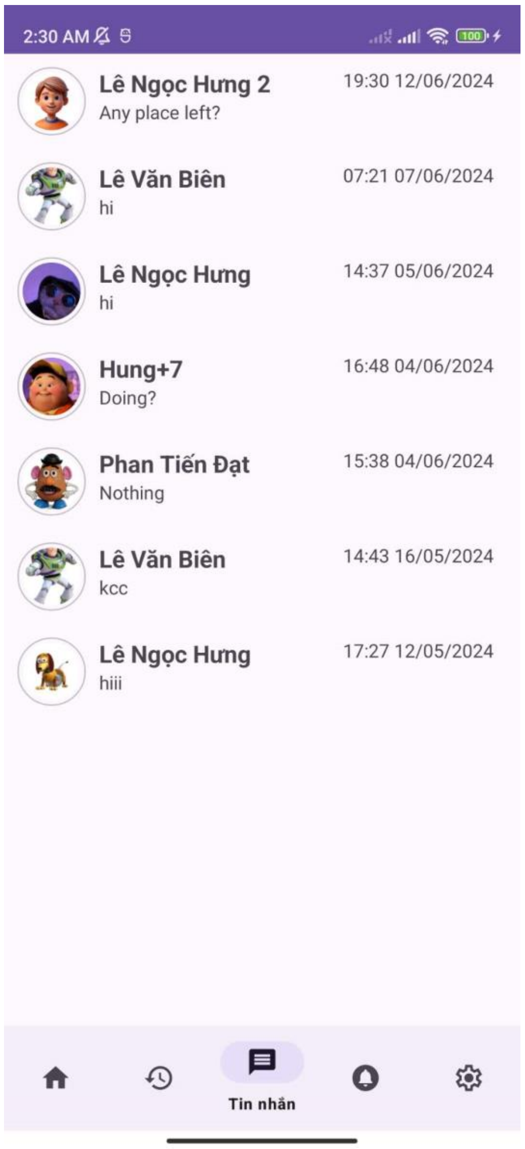
End time: 23:14:52 10-06-2024

12,000đ Review

Parking lot name: 53 Ngô Sĩ Liên

53 Đ. Ng. Sĩ Liên, Hoà Khánh Bắc, Liên Chiểu, Đà Nẵng 550000, Việt Nam

Figure 3.15. View parking history (2)


2:30 AM

Lê Ngọc Hưng 2 — Any place left? — 19:30 12/06/2024

Lê Văn Biên — hi — 07:21 07/06/2024

Lê Ngọc Hưng — hi — 14:37 05/06/2024

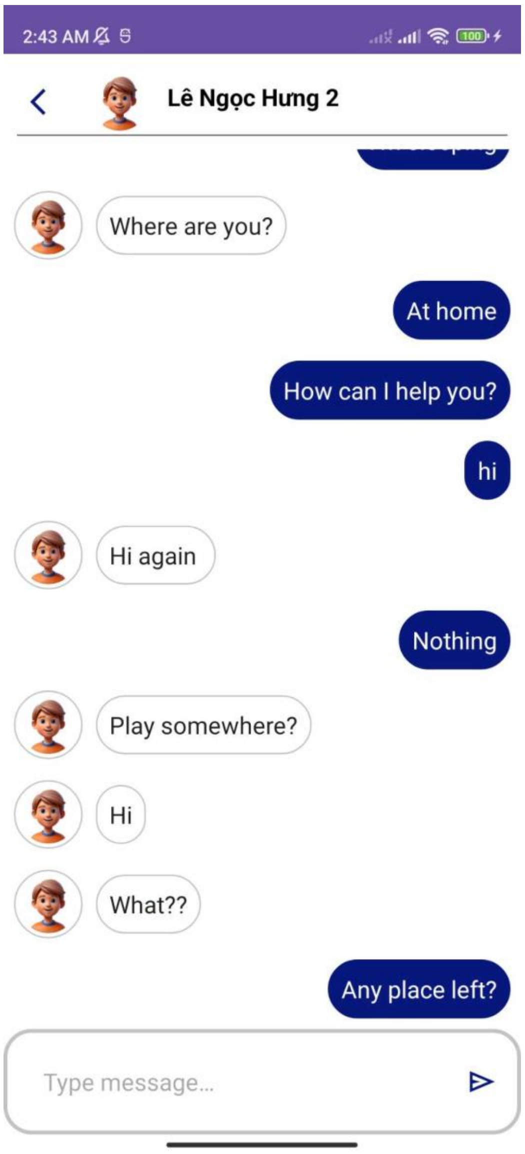
Hung+7 — Doing? — 16:48 04/06/2024

Phan Tiến Đạt — Nothing — 15:38 04/06/2024

Lê Văn Biên — kcc — 14:43 16/05/2024

Lê Ngọc Hưng — hiii — 17:27 12/05/2024

Figure 3.16. Recent chats with others


2:43 AM

Lê Ngọc Hưng 2

Where are you?

At home

How can I help you?

hi

Hi again

Nothing

Play somewhere?

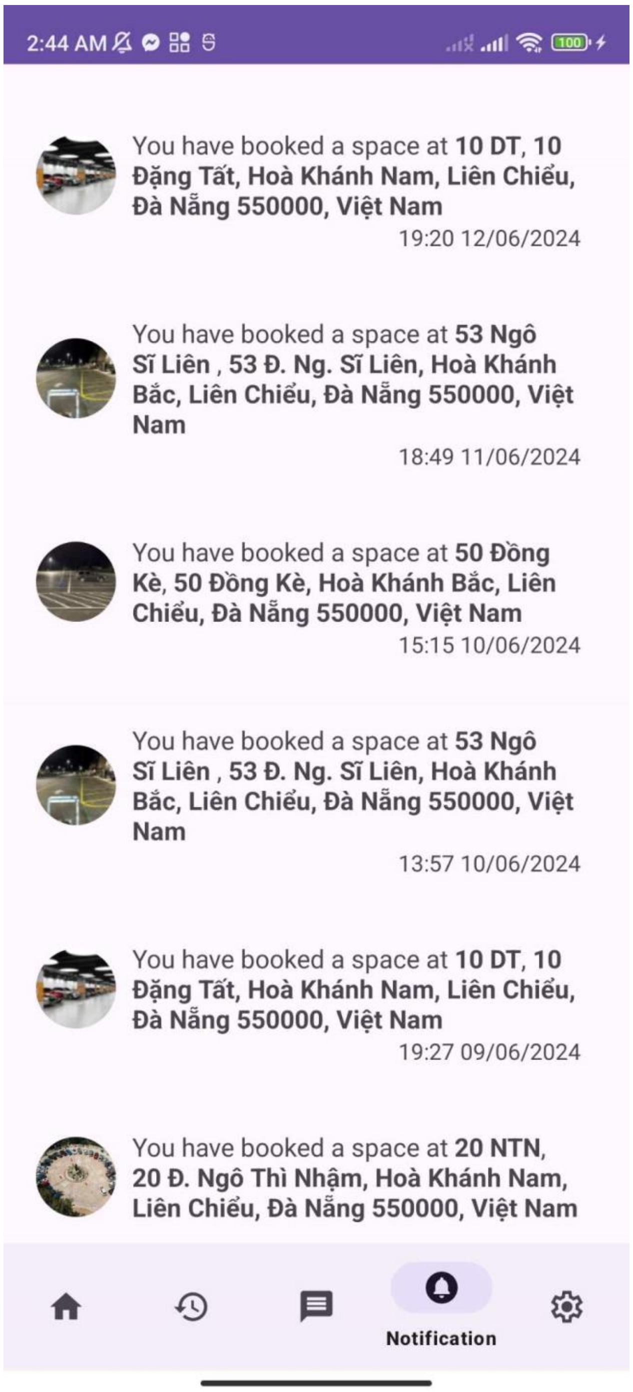
Hi

What??

Any place left?

Type message...

Figure 3.17. Detail chat screen



2:44 AM

You have booked a space at **10 DT, 10 Đặng Tất, Hoà Khánh Nam, Liên Chiểu, Đà Nẵng 550000, Việt Nam**
19:20 12/06/2024

You have booked a space at **53 Ngô Sĩ Liên , 53 Đ. Ng. Sĩ Liên, Hoà Khánh Bắc, Liên Chiểu, Đà Nẵng 550000, Việt Nam**
18:49 11/06/2024

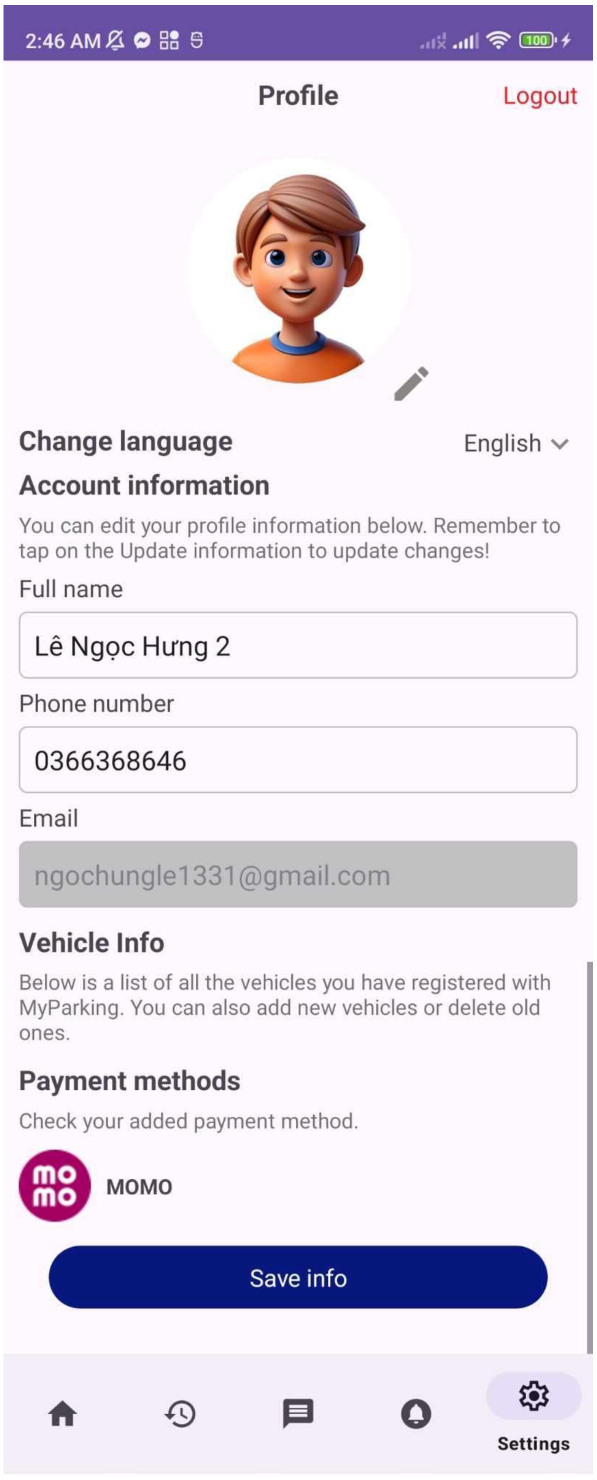
You have booked a space at **50 Đồng Kè, 50 Đồng Kè, Hoà Khánh Bắc, Liên Chiểu, Đà Nẵng 550000, Việt Nam**
15:15 10/06/2024

You have booked a space at **53 Ngô Sĩ Liên , 53 Đ. Ng. Sĩ Liên, Hoà Khánh Bắc, Liên Chiểu, Đà Nẵng 550000, Việt Nam**
13:57 10/06/2024

You have booked a space at **10 DT, 10 Đặng Tất, Hoà Khánh Nam, Liên Chiểu, Đà Nẵng 550000, Việt Nam**
19:27 09/06/2024

You have booked a space at **20 NTN, 20 Đ. Ngô Thì Nhậm, Hoà Khánh Nam, Liên Chiểu, Đà Nẵng 550000, Việt Nam**

Figure 3.18. View notifications


Profile

Logout

Change language

English

Account information

You can edit your profile information below. Remember to tap on the Update information to update changes!

Full name

Lê Ngọc Hưng 2

Phone number

0366368646

Email

ngochungle1331@gmail.com

Vehicle Info

Below is a list of all the vehicles you have registered with MyParking. You can also add new vehicles or delete old ones.

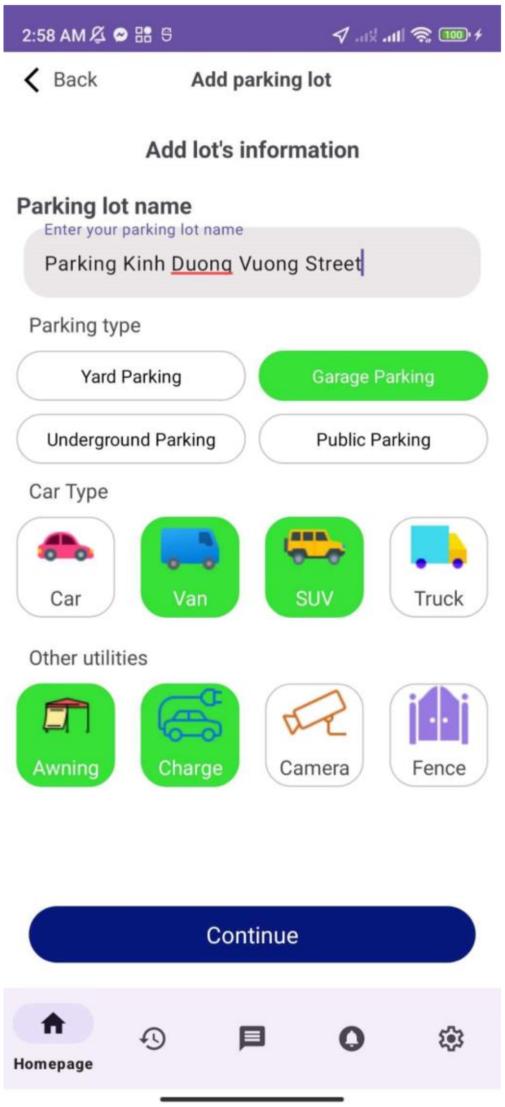
Payment methods

Check your added payment method.

MOMO

Save info

Figure 3.19. View and change personal information



Add lot's information

Parking lot name

Enter your parking lot name: Parking Kinh Duong Vuong Street

Parking type: Yard Parking [✓ Garage Parking] Underground Parking Public Parking

Car Type: Car [✓ Van] [✓ SUV] Truck

Other utilities: [✓ Awning] [✓ Charge] Camera Fence

Figure 3.20. Add new parking lot (1)

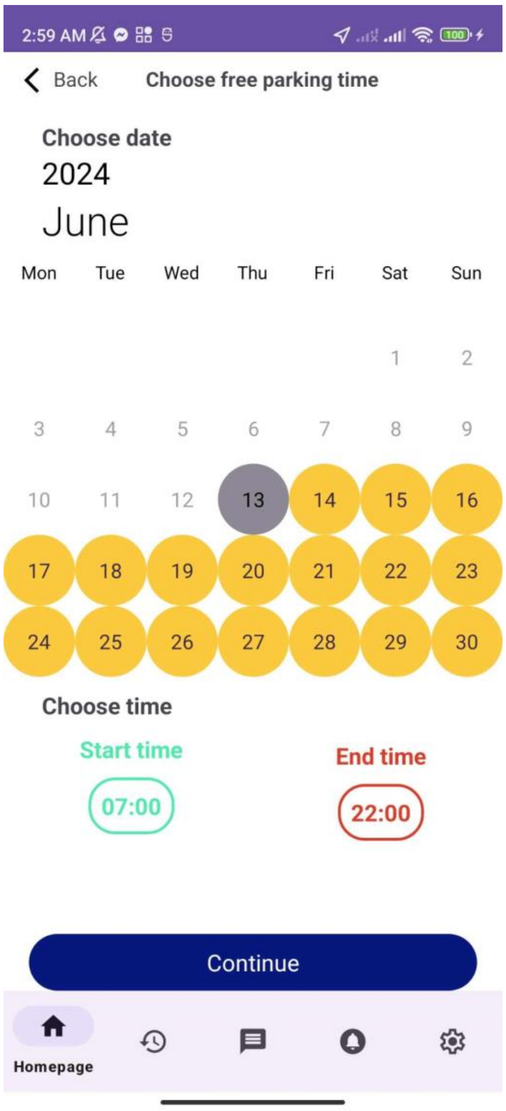
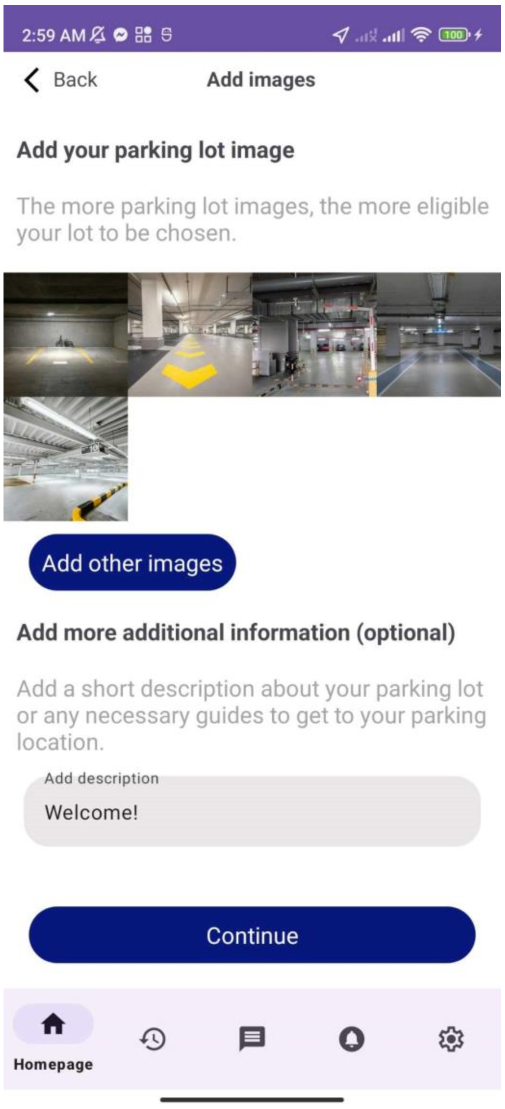


Figure 3.21. Add new parking lot (2)


Back
Add images

Add your parking lot image

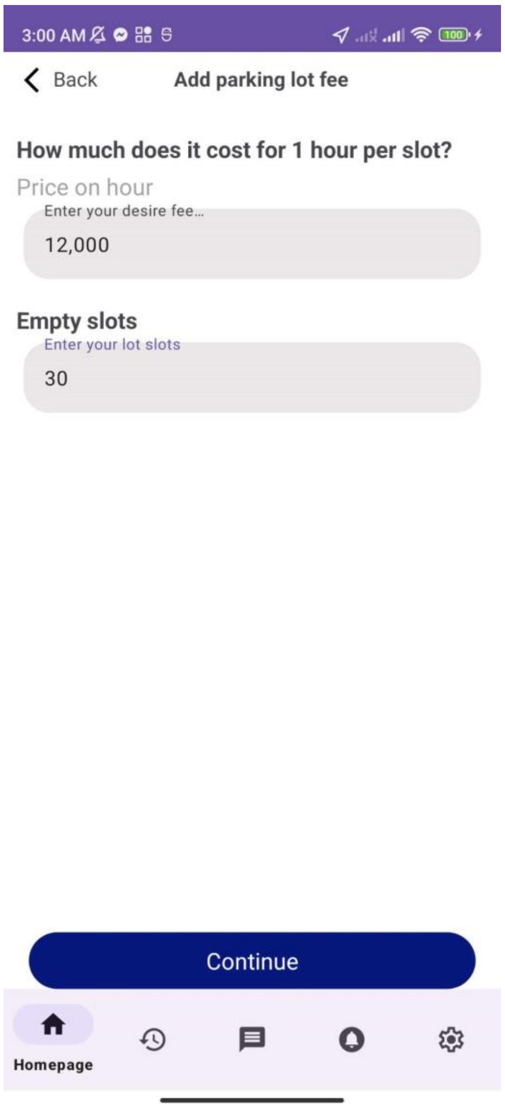
The more parking lot images, the more eligible your lot to be chosen.

Add more additional information (optional)

Add a short description about your parking lot or any necessary guides to get to your parking location.

Add description
Welcome!

Figure 3.22. Add new parking lot (3)



How much does it cost for 1 hour per slot?

Price on hour

Enter your desire fee...

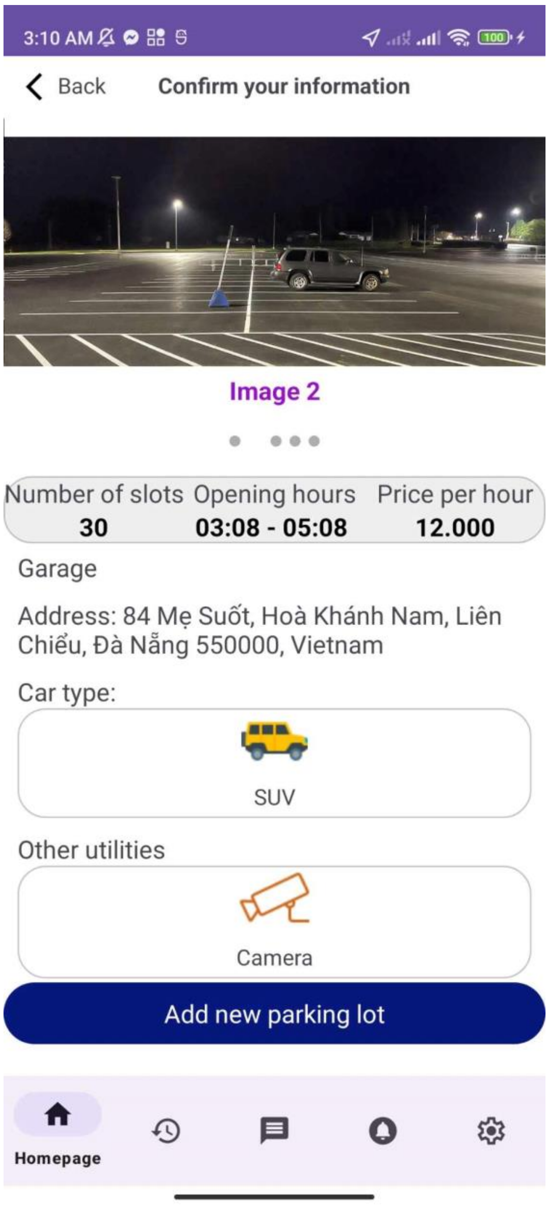
12,000

Empty slots

Enter your lot slots

30

Figure 3.23. Add new parking lot (4)



3:10 AM

Back Confirm your information

Image 2

Number of slots	Opening hours	Price per hour
30	03:08 - 05:08	12.000

Garage

Address: 84 Mẹ Suốt, Hoà Khánh Nam, Liên Chiểu, Đà Nẵng 550000, Vietnam

Car type:

SUV

Other utilities

Camera

Figure 3.24. Add new parking lot (5)

3.2. Application testing

Order	Function	Input	Output	Result
1	Login	Enter valid information and an account matching the database with role is user.	Login successful and navigate to the home screen.	Pass
		Enter valid information and an account matching the database with role is owner.	Login successful and navigate to the home screen.	
		Missing input of multiple fields in the form.	Showing pop-up error message.	
		Enter wrong email and password	Showing pop-up error message.	
2	View lot detail information	Choose a parking lot	Navigate to the lot detail information screen	Pass
3	Reserve a parking space	Click button “Order lot”	Navigate to the confirmation screen with detail order information.	Pass
4	View parking history	Click button “History” at the bottom navigation	Show the list of active parking order and completed order.	Pass
5	Add new parking lot	Add location	Show map and location	Pass

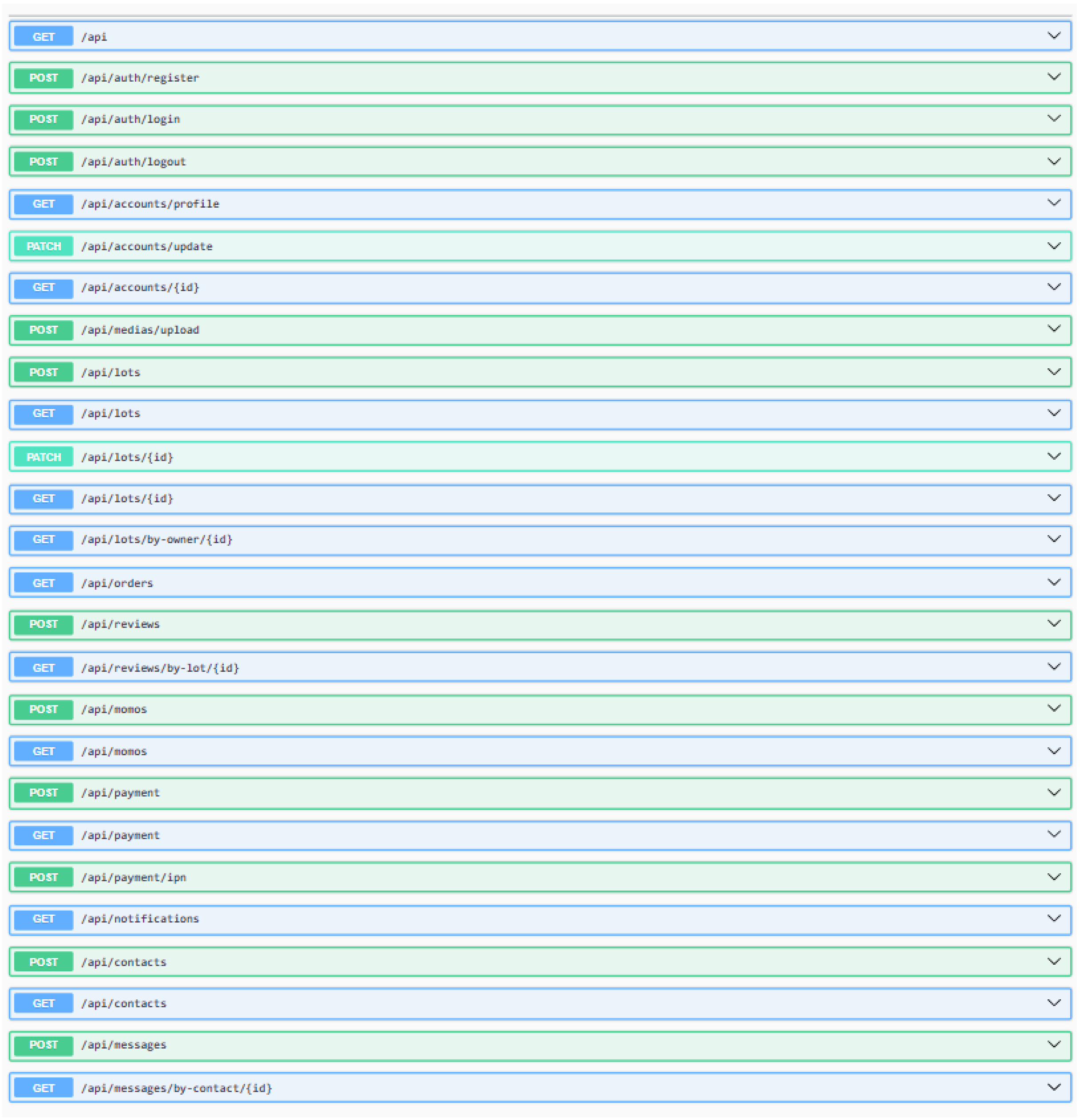
		Add features	Show list of features	
		Add parking time	Show clock to choose start time and end time	
		Add description	Show input field to add description	
		Add price and total slots	Show input field to add price and total slots	
6	Show list of messages	Click on “Message” at bottom navigation	Show the list of messages of specific contacts.	Pass
7	Show notifications	Click on “Notification” at bottom navigation	Show the list of notifications	Pass
8	Logout	Click on “Logout” button.	Navigate to the login screen.	Pass

Table 3.2. Table Application testing

3.3. Deployment

To run and test the application, you simply need to install the APK file of the application.

3.4. List of APIs



GET /api

POST /api/auth/register

POST /api/auth/login

POST /api/auth/logout

GET /api/accounts/profile

PATCH /api/accounts/update

GET /api/accounts/{id}

POST /api/medias/upload

POST /api/lots

GET /api/lots

PATCH /api/lots/{id}

GET /api/lots/{id}

GET /api/lots/by-owner/{id}

GET /api/orders

POST /api/reviews

GET /api/reviews/by-lot/{id}

POST /api/momos

GET /api/momos

POST /api/payment

GET /api/payment

POST /api/payment/ipn

GET /api/notifications

POST /api/contacts

GET /api/contacts

POST /api/messages

GET /api/messages/by-contact/{id}

Figure 3.25. List of APIs

CHAPTER 4: CONCLUSION AND FUTURE ORIENTATION

4.1. Achieved results

4.1.1. In terms of theoretical

During the process of researching and implementing the project, I have gained a solid understanding of the fundamental knowledge needed to build an Android application using the Kotlin language. I have also applied micro-service and SOA models and used APIs to develop applications. Moreover, I have learned how to analyze business processes and systems that meet specific requirements.

4.1.2. In terms of application

Basically, the application has met the basic requirements such as:

- Finding and booking parking spots within the nearest range, providing detailed information about parking spots.
- Helping users book parking spots quickly and easily.
- Allowing users to add new parking locations, update information of parking spots, view parking history, contact parking spot owners.
- Receiving real-time notifications when someone books a spot.
- Sending messages between users and parking spot owners.
- Integrating online payment with MOMO.

4.2. Drawbacks

Although basic features have been completed, due to limited experience and knowledge as well as time constraints, there are inevitably some shortcomings. Some issues that I have identified after using the application for a while are that the app interface is not attractive, the app performance is not optimal, and there are instances of lag when interacting with screens.

4.3. Future development

The parking application will be further developed to address these drawbacks and provide the best utility for user experience. In the upcoming versions, I plan to add features such as integrating login with Facebook, Google; I will also strive to create a more appealing interface and optimize the app performance.

REFERENCES

[1] "Kotlin Android tutorial," 15 May 2023. [Online]. Available: https://www.geeksforgeeks.org/kotlin-android-tutorial/. [Accessed 10 April 2024].

[2] K. Vasava, "Retrofit in Android," 8 November 2023. [Online]. Available: https://medium.com/@KaushalVasava/retrofit-in-android-5a28c8e988ce. [Accessed 10 April 2024].

[3] "Firebase Cloud Messaging," Google, [Online]. Available: https://firebase.google.com/docs/cloud-messaging. [Accessed 20 April 2024].

[4] "Understand Firebase for Android," Google, [Online]. Available: https://firebase.google.com/docs/android/learn-more. [Accessed 12 April 2024].

[5] "Places SDK for Android," Google, [Online]. Available: https://developers.google.com/maps/documentation/places/android-sdk. [Accessed 10 April 2024].

[6] N. Kanezawa, "Native Socket.IO and Android," 20 January 2015. [Online]. Available: https://socket.io/blog/native-socket-io-and-android/. [Accessed 3 May 2024].