

VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY
UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING



PROGRAMMING INTERGRATION PROJECT (CO3103)

Final Report

Fast Coffee Takeaway System

Semester 221 - Class CC02 - Group 1

Instructor: Mr. Trương Tuấn Anh
Students: Nguyễn Duy Bảo - 2052399
Trần Công Huy Hoàng - 2052482
Hoàng Đình Huy - 2053034
Nguyễn Đình Thảo - 2052261
Lê Hoàng - 2052479

HO CHI MINH CITY, JANUARY 2023



Contents

1 Takeaway Coffee Fast Order	3
1.1 Description	3
1.2 Advantages	3
2 Requirements	4
2.1 Functional requirements	4
2.2 Non-functional Requirements	5
3 Use Case diagram	7
3.1 Authentication module	7
3.2 Account management module	8
3.3 Shop management module	9
3.4 Order management module	12
3.5 Customer experience module	13
3.6 Message module and Notification module	15
4 System Modeling	17
4.1 Authentication module	17
4.2 Account management module	20
4.3 Shop Management module	24
4.4 Order Management module	33
4.5 Customer experience module	36
4.6 Message module	39
5 Class Diagram	40
6 Database Design	42
6.1 Enhance Entity Relation Diagram	42
6.2 Relational Schema	43
6.3 Database Management System and Host	44
7 Implementation	45
7.1 Shop Management Web	46
7.2 Client App	51
8 Self Evaluation	61



8.1 Achievements	61
8.2 What are not achieved yet	61
8.3 Future development	61



1 Takeaway Coffee Fast Order

1.1 Description

Application is developed based on practical demand to technicalize the purchase process at take-away coffee points and connect them to make a convenience for customers on purchasing coffee quickly on their way. The system is built with ideas allowing takeaway coffee shops to buy an operating license and register to the system and provide their products to the customer. The customer needs to log in to the system and provide their personal information also their preferred drink, coffee shop or frequent route, etc that supports their order. The customer can place an order directly on the system and the coffee shop - a partner can accept or refuse it. The system will track the customer's position to notice a partner preparing their drink. The system will assist in identifying the right user so that the partner can deliver the correct order to the customer. The system also supports the handling of problems between customers and partners.

The application operates based on the automatic system with tasks to be processed without user interaction. The application is built on the foundation of Web applications, Mobile applications using expo, API, and services to provide independent consumer components...

1.2 Advantages

- Customers can save time buying goods and access preferential coffee shops.
- Customers can set up their favorite orders daily.
- Partner coffee shops can manage the orders fast and effectively, and handle the orders easily.
- Partner coffee shops can have a chance to approach more customers by applying advertisement policies, promotion policies.
- Reduce the overloading orders when buying directly at the coffee shop.
- Manage the coffee shop, open and close time proactively.



2 Requirements

2.1 Functional requirements

2.1.1 Shop and Order Management

- Coffee Sale Staff
 - Be able to use an account to access the software
 - Be able to manage the orders (accept/reject/process/finish the orders) and see the overview of the orders sorted by the system
 - Be able to select the method of payment
 - Be able to raise problem to the shop owner
 - Be able to see the location of the current customer
 - Be notified about the orders from customers
 - Be able to send message to the customer
- Customer
 - Be able to use an account to access the software
 - Be able to make/cancel/view an order
 - Be notified when the order is ready
 - Be able to give feedback about the coffee shop
 - Be able to send message to the shop
 - Be able to manage their profile
 - Be able to mark their favorite coffee shop
 - Be able to do the payment by many methods
 - Be notified about the ads, promotions about near coffee shop
 - Be able to review the order history, make daily orders automatically
 - Be able to see the map including coffee shops

2.1.2 Customer Experience

- Customer
 - Customer can set a route and see nearby coffee shops.
 - Customer can quickly choose a daily route which is created automatically by the system based on recent daily routes.
 - Customer can see the menu of the chosen store and can easily add items into the cart.
 - Customer can manually add notes to each item in an order or choose a quick note created automatically based on recent notes.
 - Customer can pay for an order by many ways (cash, local ATM cards, Visa, Digital Pocket...).



- Customer can cancel the order that has not been made yet and get a refund.
- Customer has a favorite coffee shops list and can easily add new coffee stores to it
- Customer can view order history.
- Customer can view recommended stores based on recent daily routes and recently ordered items.
- Customer can easily report problems with orders or the system.
- Customer can see the order's status.
- Customer can view maps.

2.1.3 Authentication

- Authentication
 - The system requires users to enter a password, username, or coffee shop information before performing certain software functions. The user account needs to be split into 4 types: Admin of the system, partners (The coffee shop owner), Coffee shop's staff, and Customers.
 - The users can log out of the system.
- Account Management
 - Partners can provide their information to register the license for using the system.
 - Users can register an account to log in to the system.
 - The system supplier can check the registration list and accept or refuse based on the provided information.
 - The partner can create and manage accounts of their staff.
 - The system supplier can have an overview of users' accounts.
 - The partner can manage their staff's accounts.
 - The users can change their personal information.

2.2 Non-functional Requirements

- Performance
 - Handle 5000 requests per second
 - Delay of a response after a request has to be less than 1 seconds
 - Precision for locations on the map has to be less than 10m.
 - Can handle 100000 sql queries per second
- Usability
 - Menu should be flexibility scaled with the user's screen resolution and limited horizontal scrolling.
 - Customers are able to order in 3 clicks/finger taps.
 - Staffs can check their tasks in 2 click



- Registration has to be fast and easy (ask only for Name, Age, Username, Password(No need for confirmation), Phone Number), verification via OTP. Can add email later. Considering using login through Google, Facebook, ...
 - Data has to be backed up every 2 days, server has to be backed up everyday.
- Reliability, maintainability, availability
 - Data has to be backed up every 2 days, server has to be backed up everyday.
 - The system must run without failure at 95% of the use-cases during a month.
 - If the server is down, the maximum time to fix is 45 minutes. Available for Vietnamese users 95% of the time in a month during non-working hours (17:00 to 9:00 the next day).
- Security
 - There is a clear distinction between account type Warning for server intrusion
 - Preventing sessions hijacking by sanitizing inputs
 - Prevent data breach by using predefined objects in SQL and by using encryption in this case all the sensitive information (like password) has to be one-way encrypted.

3 Use Case diagram

3.1 Authentication module

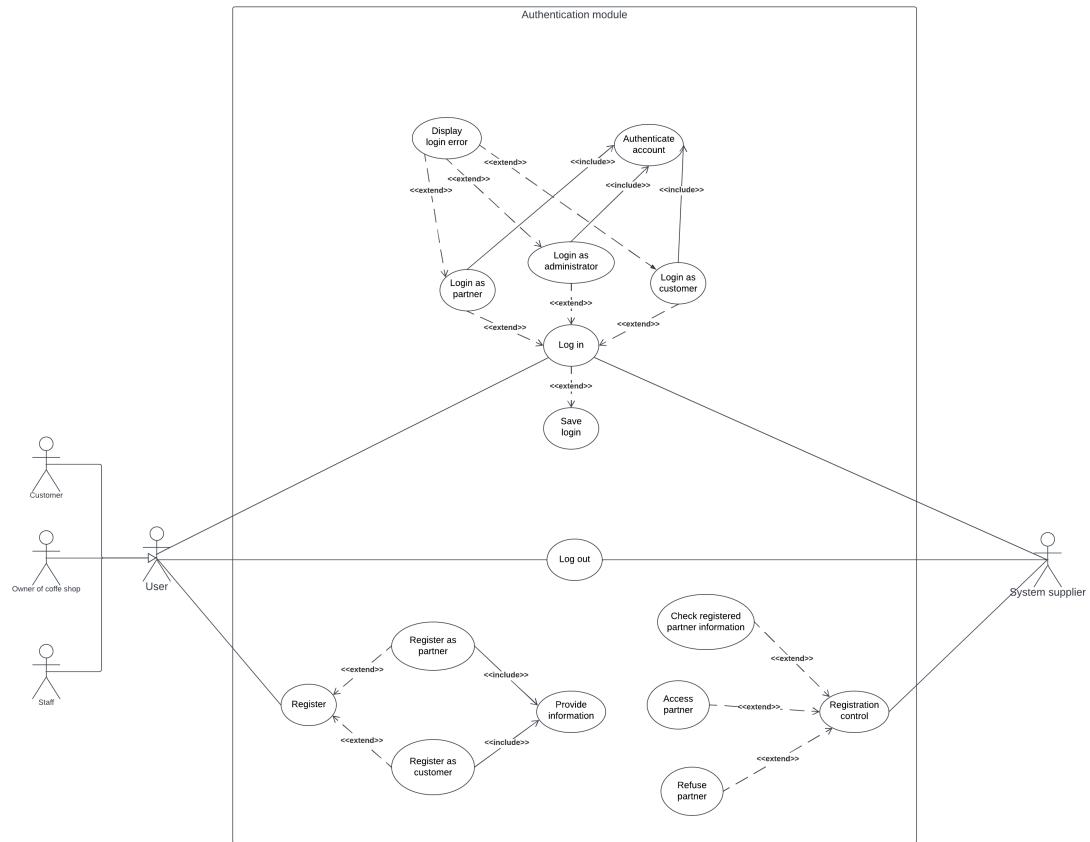


Figure 1: Use Case diagram for Authentication module

Keynotes:

- Users can register, login, logout
- System supplier can control registration

3.2 Account management module

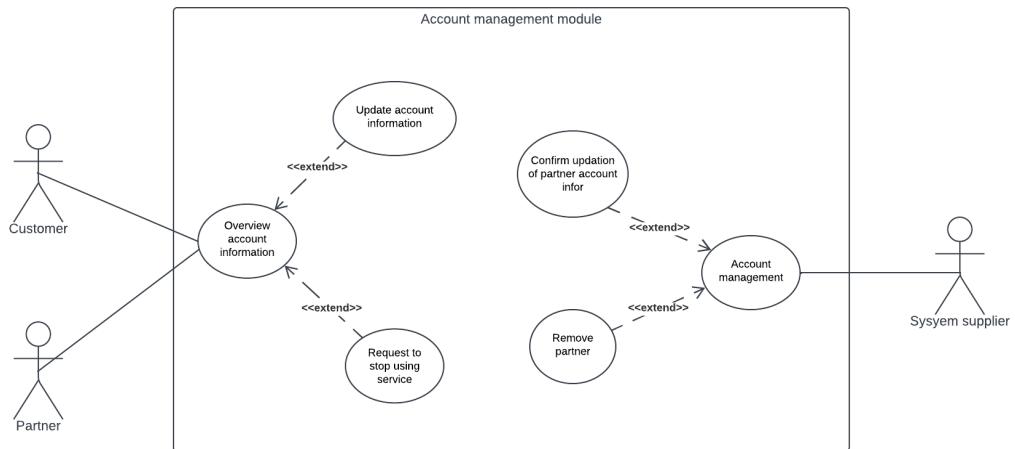


Figure 2: Use Case diagram for Account management module

Keynotes:

- Users can check their own information and other's information
- Users can update their information
- System supplier can remove any accounts
- System supplier can confirm any information update of users

3.3 Shop management module

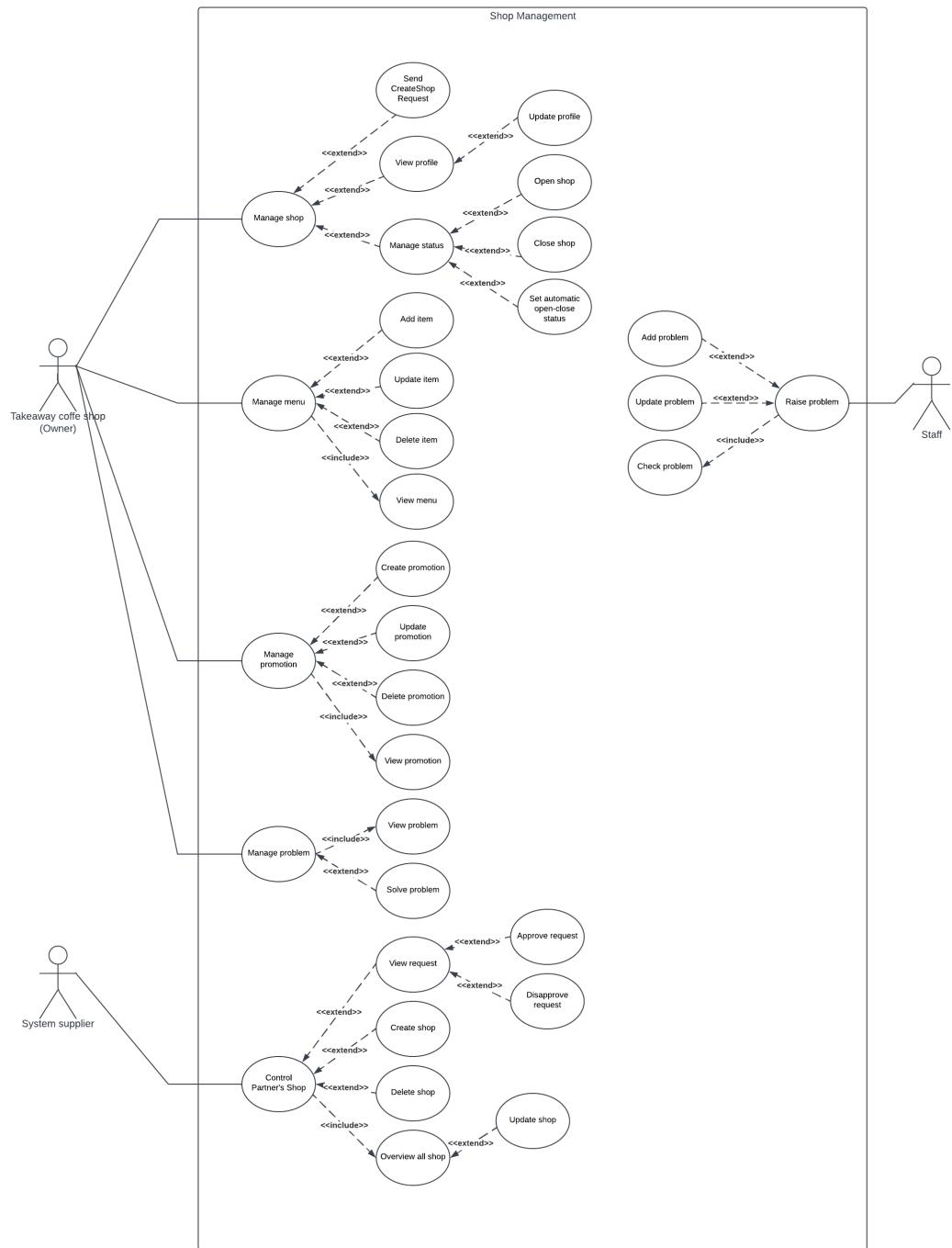


Figure 3: Use Case diagram for Shop management module



Use Case table:

Use-case Name	Manage shop
Actor	Owner
Description	Manage the information about coffee shops. Owner may want to view profile or update status of the coffee shop
Preconditions	Database is available
Normal Flow	<ol style="list-style-type: none">1 Owner sends a request to manage the shop2 Owner chooses a shop to manage3 Owner send CreateShop request4 System asks to choose view profile of update status of the coffee shop5 Owner selects the option.
Exceptions	Shop doesn't belong to the owner
Alternative Flows	None
Use-case Name	Manage menu
Actor	Owner
Description	Manage the menu of the coffee shop. Owner may want to add item, update item
Preconditions	Database is available
Normal Flow	<ol style="list-style-type: none">1 Owner sends a request to manage the menu.2 Owner chooses a shop to manage the menu.3 System display menu.4 Owner selects the item to modify.5 Owner can choose to add, update or delete items.6 System updates data on the menu.
Exceptions	Shop doesn't belong to the owner
Alternative Flows	None



Use-case Name	Manage promotion
Actor	Owner
Description	Manage promotion for coffee shops. Owner may want to create promotion, update promotion, delete promotion and view promotion.
Preconditions	Database is available
Normal Flow	<ol style="list-style-type: none">1 Owner requests to manage promotion.2 Owner chooses a shop to manage promotions.3 System displays a list of current promotions.4 Owner chooses a promotion.5 Owner chooses to add, update or delete the promotion.6 System updates data.
Exceptions	Shop doesn't belong to the owner. Promotion is not appropriate.
Alternative Flows	None
Use-case Name	Raise problem
Actor	Staff
Description	Raise problems that the staff needs a solution from the Owner to solve.
Preconditions	Database is available
Normal Flow	<ol style="list-style-type: none">1 Staff requests to raise a problem2 System show a list of unsolved problem3 Staff can select to add a new problem or update a problem.4 System asks for details of the problem5 Staff enter details and submit6 System update on database.
Exceptions	none
Alternative Flows	3.1 Staff can delete an unsolved problem if it is not important or solved eventually without the need of Owner's solution.
Use-case Name	Manage problem
Actor	Owner
Description	Manage problems raised by staff. Owner can propose a method to solve the problem.
Preconditions	Database is available
Normal Flow	<ol style="list-style-type: none">1 Owner requests to manage problems.2 System displays a list of current problems.3 Owner can choose to propose a solution to solve the problem
Exceptions	Shop doesn't belong to the owner.
Alternative Flows	<ol style="list-style-type: none">3.1 Owner has no intention to solve the problem yet.3.2 The system goes back to the main page.

Use-case Name	Control Partner's Shop
Actor	System Supplier
Description	Manage problems raised by staff. Owner can propose a method to solve the problem.
Preconditions	Database is available
Normal Flow	1 System supplier requests to control other shops. 2 System displays all the shops which the Owner is in charge of. 3 System supplier can choose to create a shop or delete a shop. 4 System supplier confirms the action. 5 System updates data.
Exceptions	none.
Alternative Flows	3.1 System suppliers can choose to update the information of a shop. 3.2 System asks for details. 3.3 System supplier enters details and submit 3.4 System updates Data

3.4 Order management module

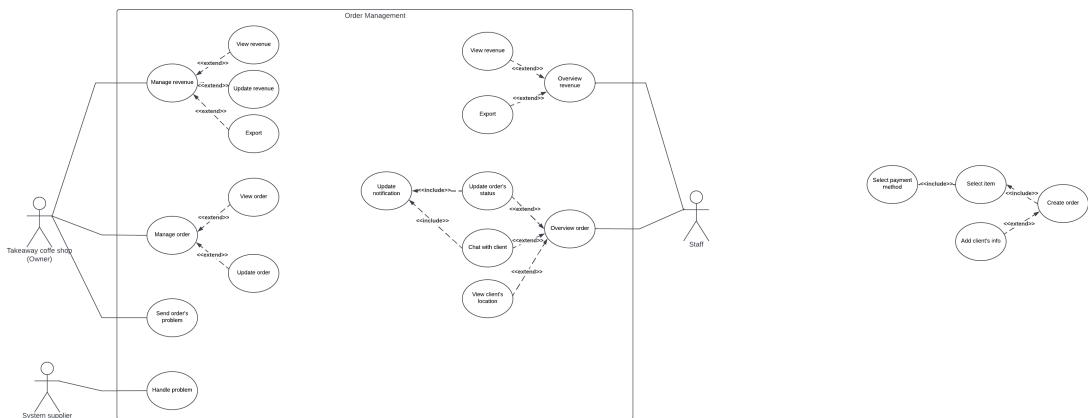


Figure 4: Use Case diagram for Order management module

Keynote:

- Owner can manage order (view and update order), manage revenue
- Staff can view the revenue and order
- System supplier handle problem

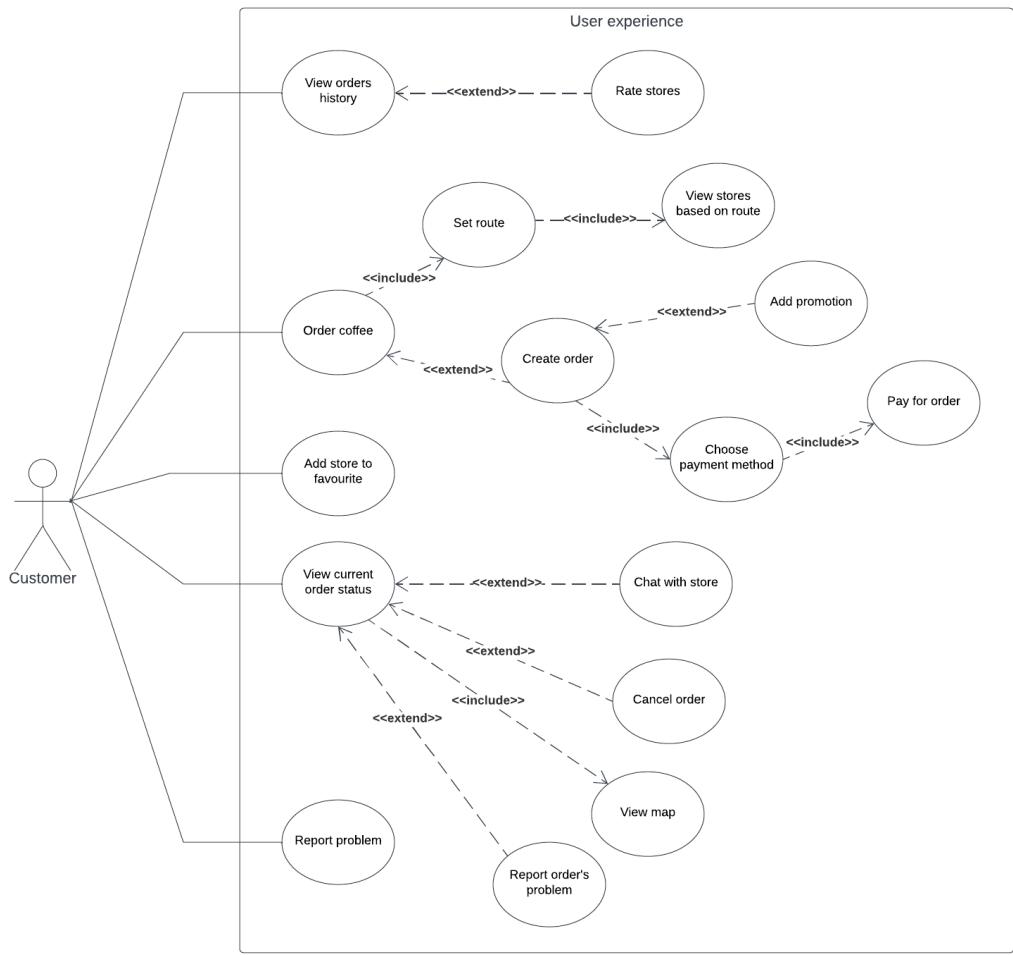


Figure 5: Use Case diagram for Customer experience module

3.5 Customer experience module

Use case tables:



Use-case Name	View order history
Actor	Customer
Description	The costumers can see their order history.
Preconditions	The customer has logged in
Normal Flow	<ol style="list-style-type: none">1. The customer requests order history.2. System shows a list of order history.3. The customer can see detailed information of an order in the list.4. The customer ends the action.
Exceptions	none.
Alternative Flows	3.1 The customers can rate the store in an order they have made.
Use-case Name	Order coffee
Actor	Customer
Description	The customers orders coffee based on their route..
Preconditions	The customer has logged in
Normal Flow	<ol style="list-style-type: none">1. The customer needs to create a route.2. The system, based on the route, recommends takeaway coffee stores near the route.3. The customer chooses a store, chooses items, promotions, a payment method and pays for the order.4. The system updates a new route through the store
Exceptions	none.
Alternative Flows	<ol style="list-style-type: none">2.1 The customer can choose to see his/her favorite stores list or can search for a specific store.3.1 The customer can add notes for an item.3.2 The customer can add notes for the order.
Use-case Name	Add store to favorite
Actor	Customer
Description	The customer can add a store to their favorite list.
Preconditions	The customer has logged in
Normal Flow	<ol style="list-style-type: none">1. The customer chooses a specific store.2. The customer adds the chosen store to his/her favorite list.
Exceptions	none.
Alternative Flows	none



Use-case Name	View current order status
Actor	Customer
Description	The customers can track their current order status.
Preconditions	The customer has logged in and has created an order.
Normal Flow	<ol style="list-style-type: none">1. The customer requests to see current order status.2. The system shows the order status and a map showing the current customer's location, store location, customer's destination and a route through these places.3. The customer ends this action or the order is finished.
Exceptions	none.
Alternative Flows	<ol style="list-style-type: none">2.1 The customer can choose to chat with the store.2.2 The customer can report the order's problem.2.3 The customer can cancel the order which is in "Accepted" state or "Pending" state.

Use-case Name	Report problem
Actor	Customer
Description	The customer can report problems to the system staff.
Preconditions	The customer has logged in
Normal Flow	<ol style="list-style-type: none">1. The customer requests a problem report.2. The customer writes the report and submits it to the system.3. The system sends the report to the system staff.4. Action ends.
Exceptions	none.
Alternative Flows	none

3.6 Message module and Notification module

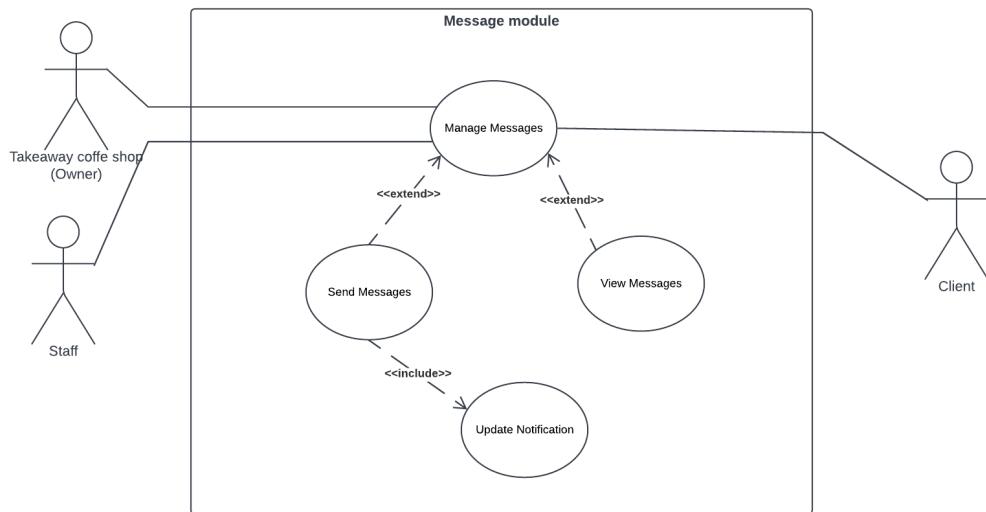


Figure 6: Use Case diagram for Message management module

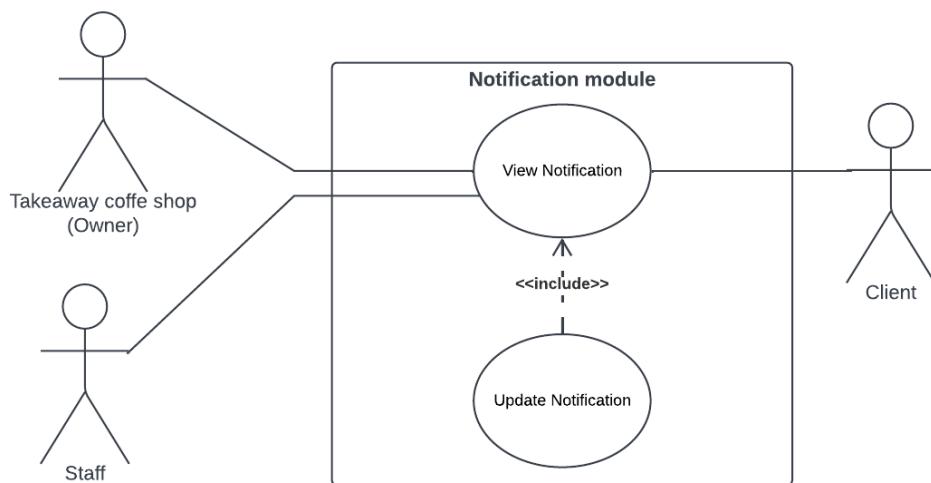


Figure 7: Use Case diagram for Notification management module

4 System Modeling

4.1 Authentication module

4.1.1 Activity diagram

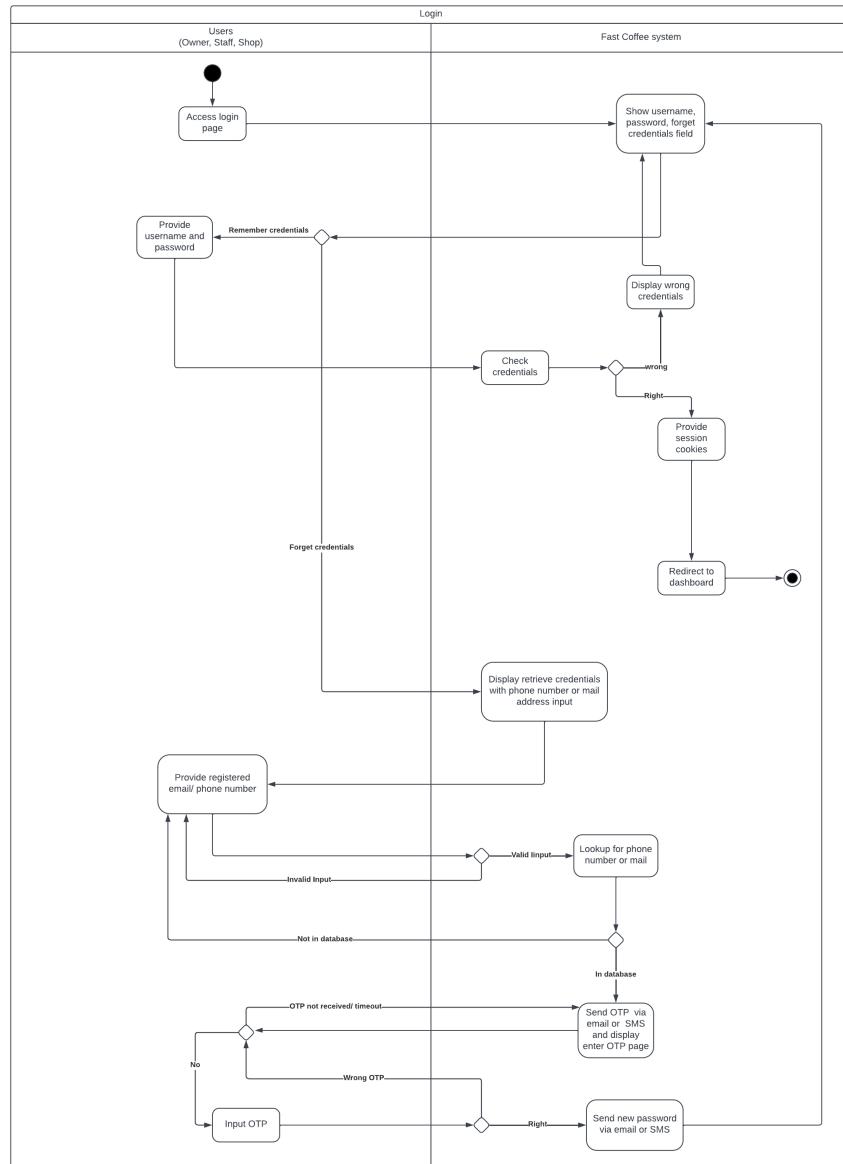


Figure 8: Activity diagram of Authentication module

4.1.2 Sequence diagram

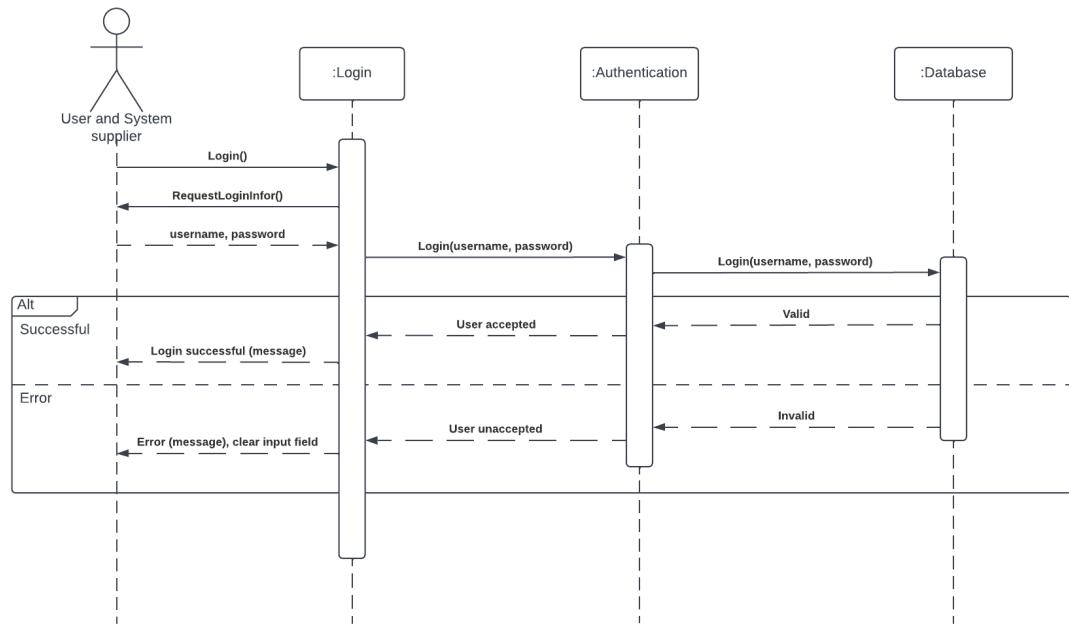


Figure 9: Sequence diagram of Authentication module - Login

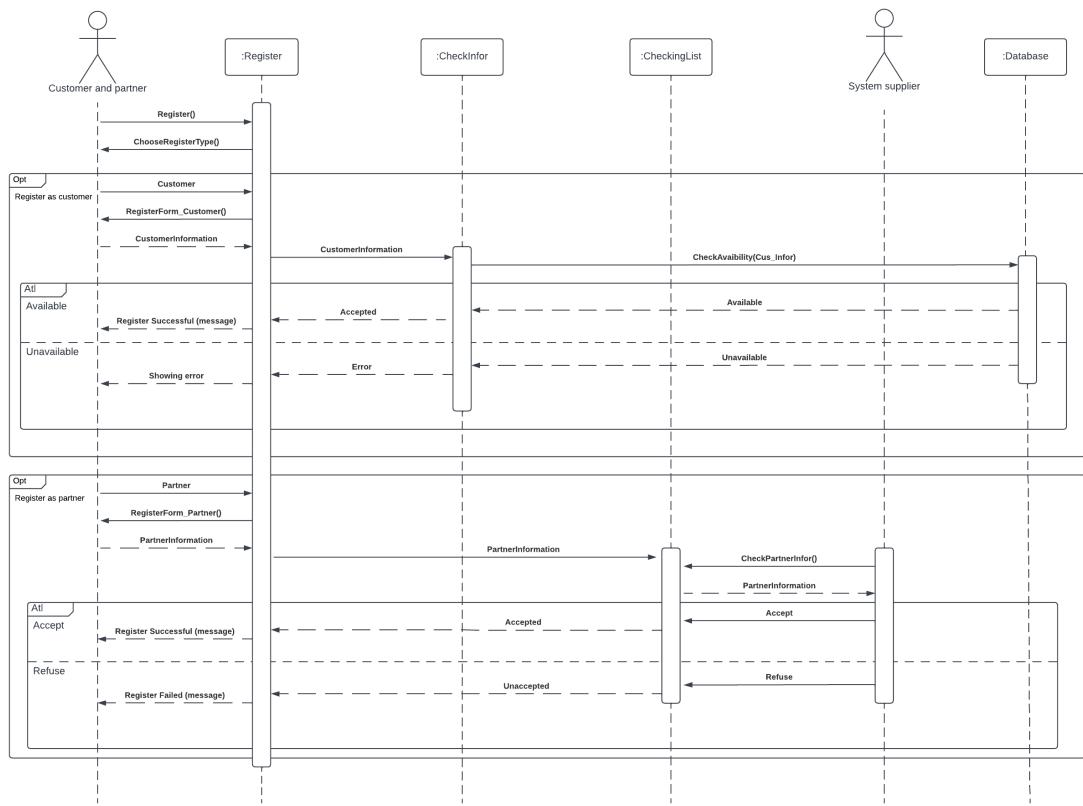


Figure 10: Sequence diagram of Authentication module - Register

4.2 Account management module

4.2.1 Activity diagram

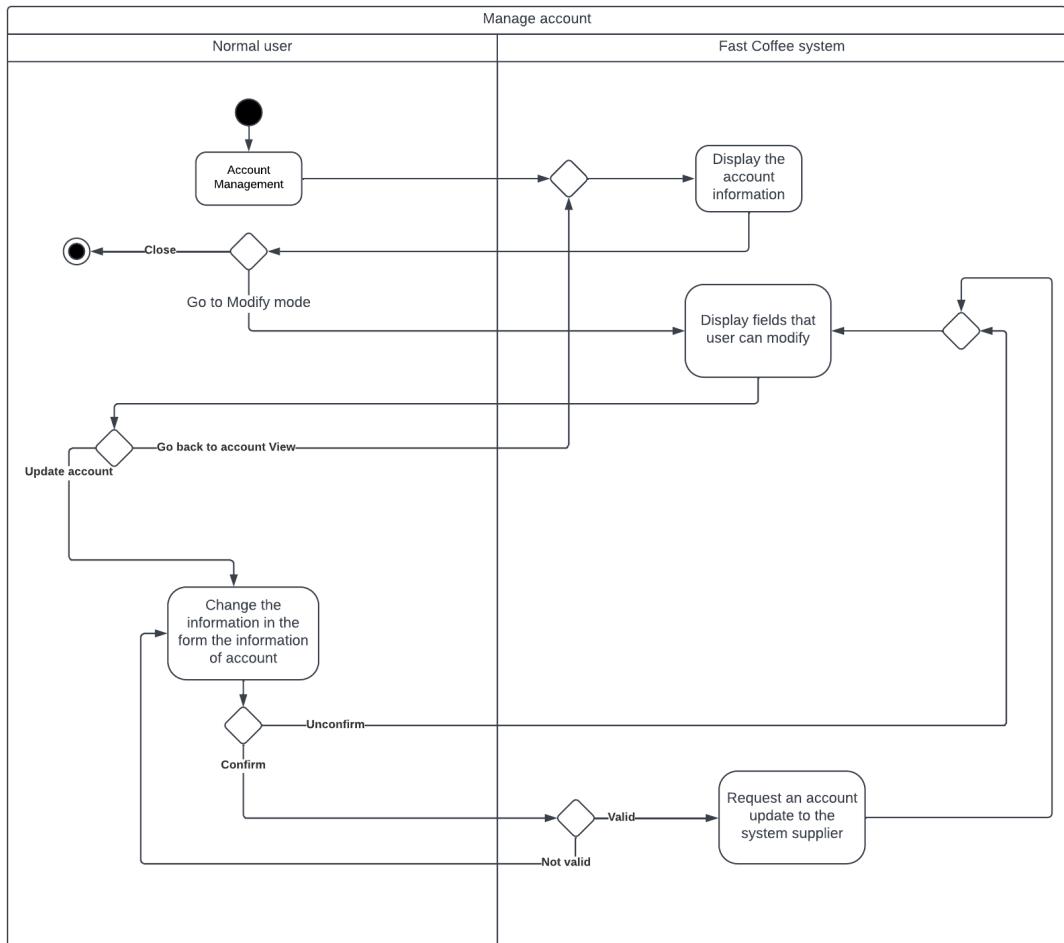


Figure 11: Activity diagram of Account management module - Manage account

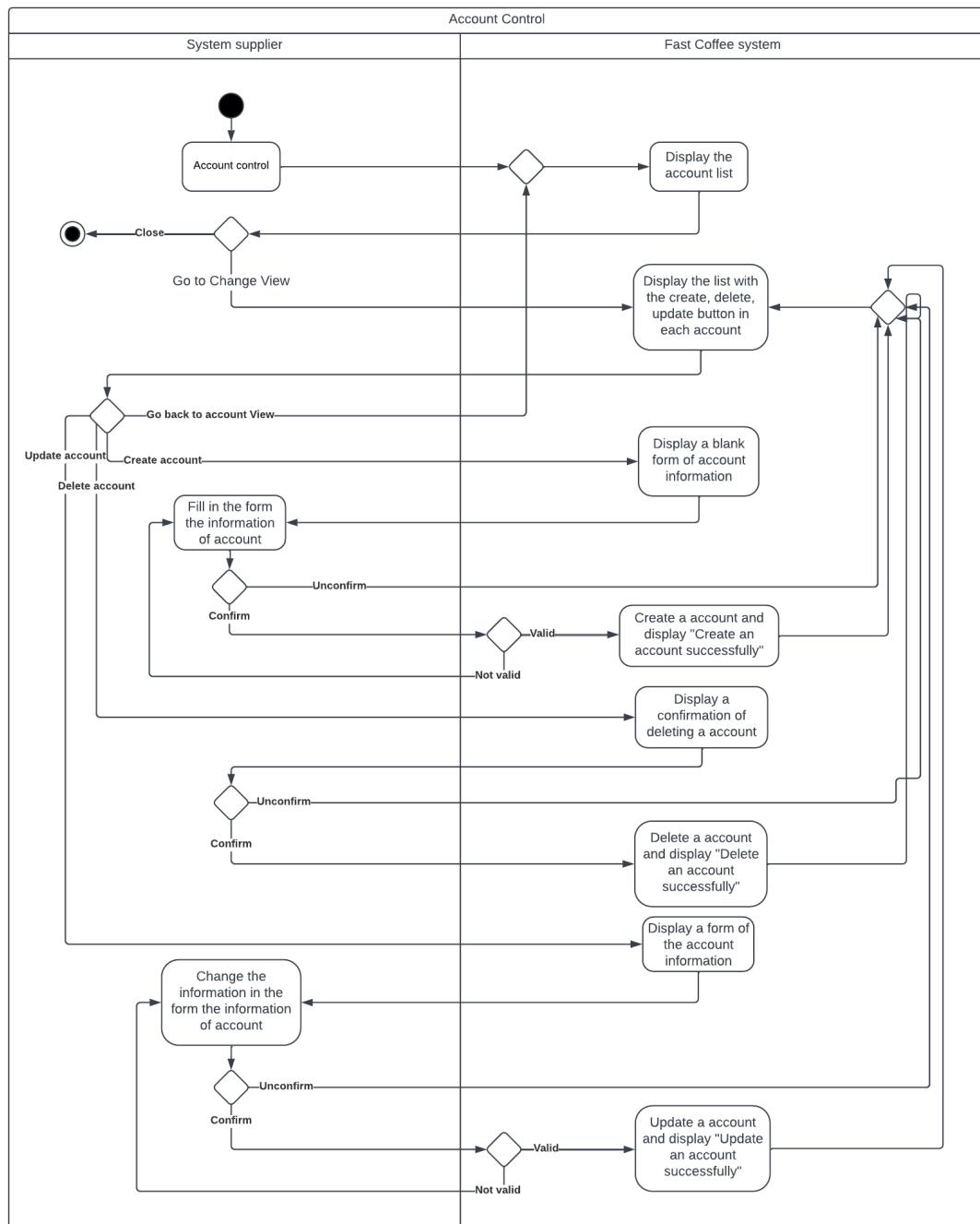


Figure 12: Activity diagram of Account management module - Account Control

4.2.2 Sequence diagram

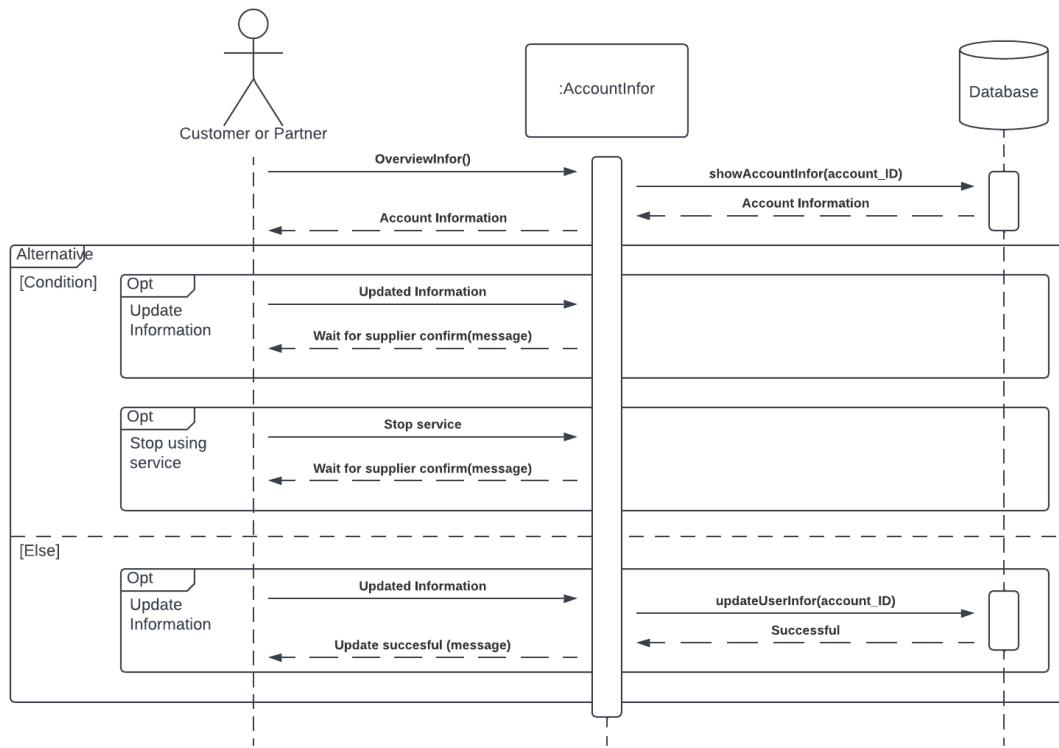


Figure 13: Sequence diagram of Account management module - Overview information

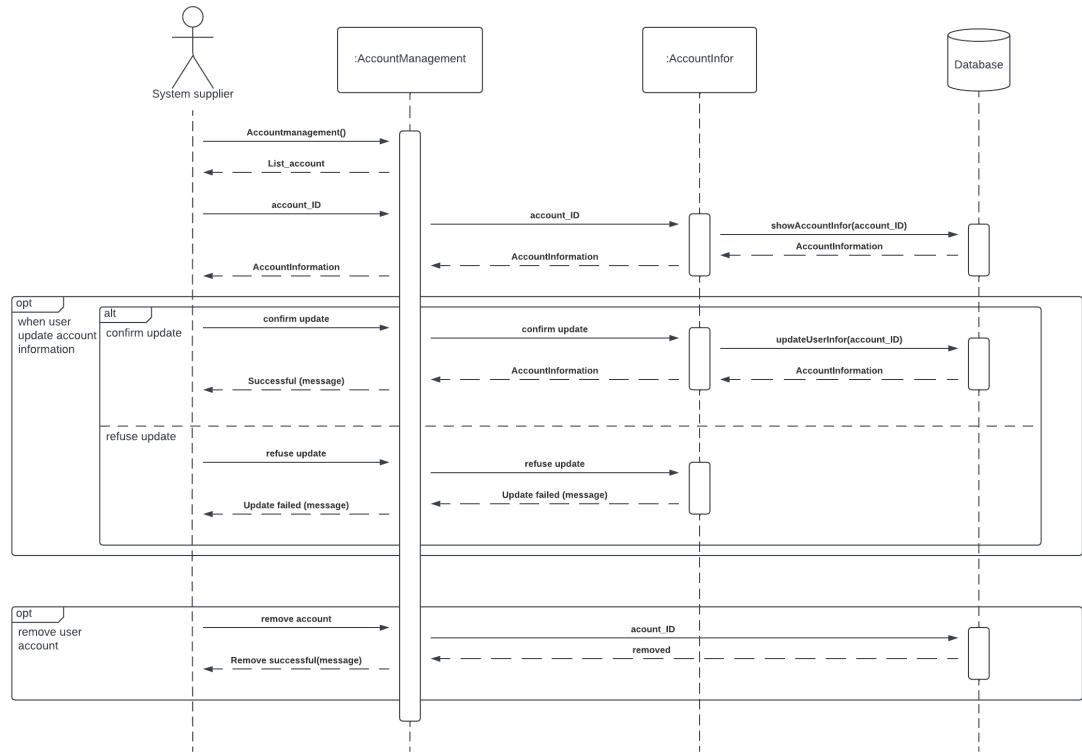


Figure 14: Sequence diagram of Account management module - Account management

4.3 Shop Management module

4.3.1 Activity Diagram

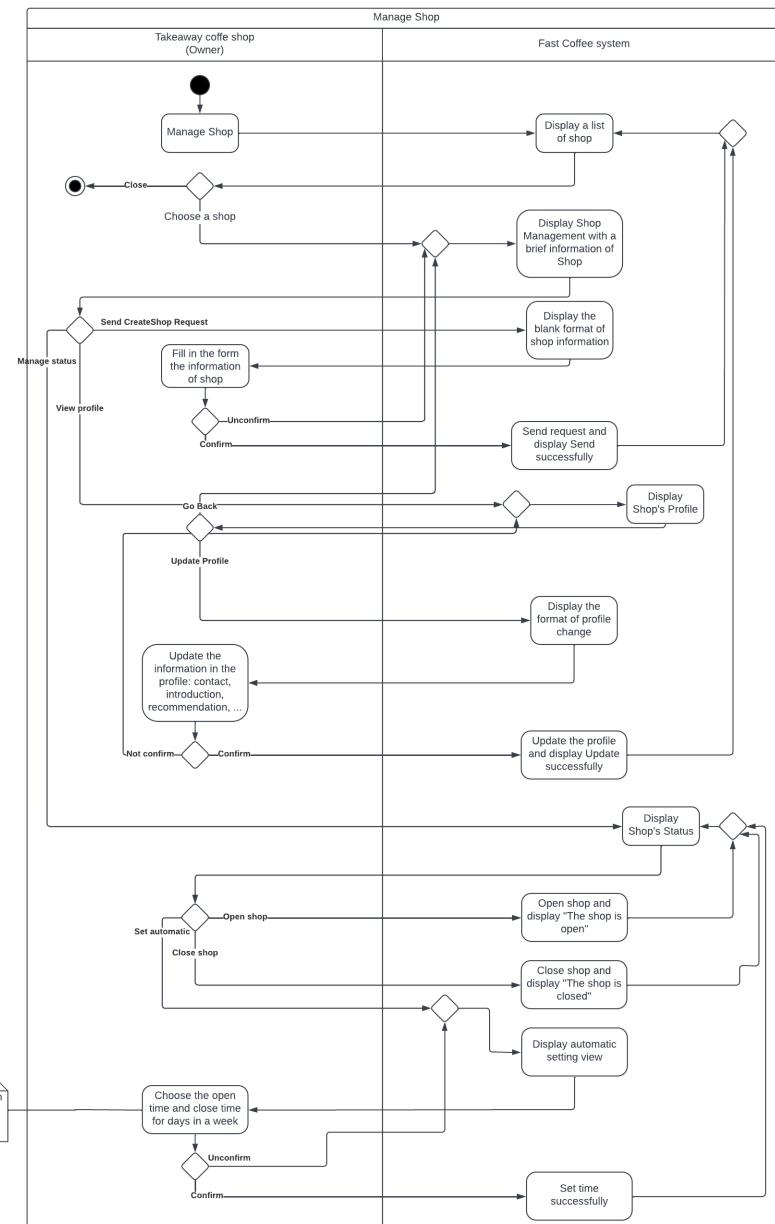


Figure 15: Activity diagram of Shop Management module - Manage Shop

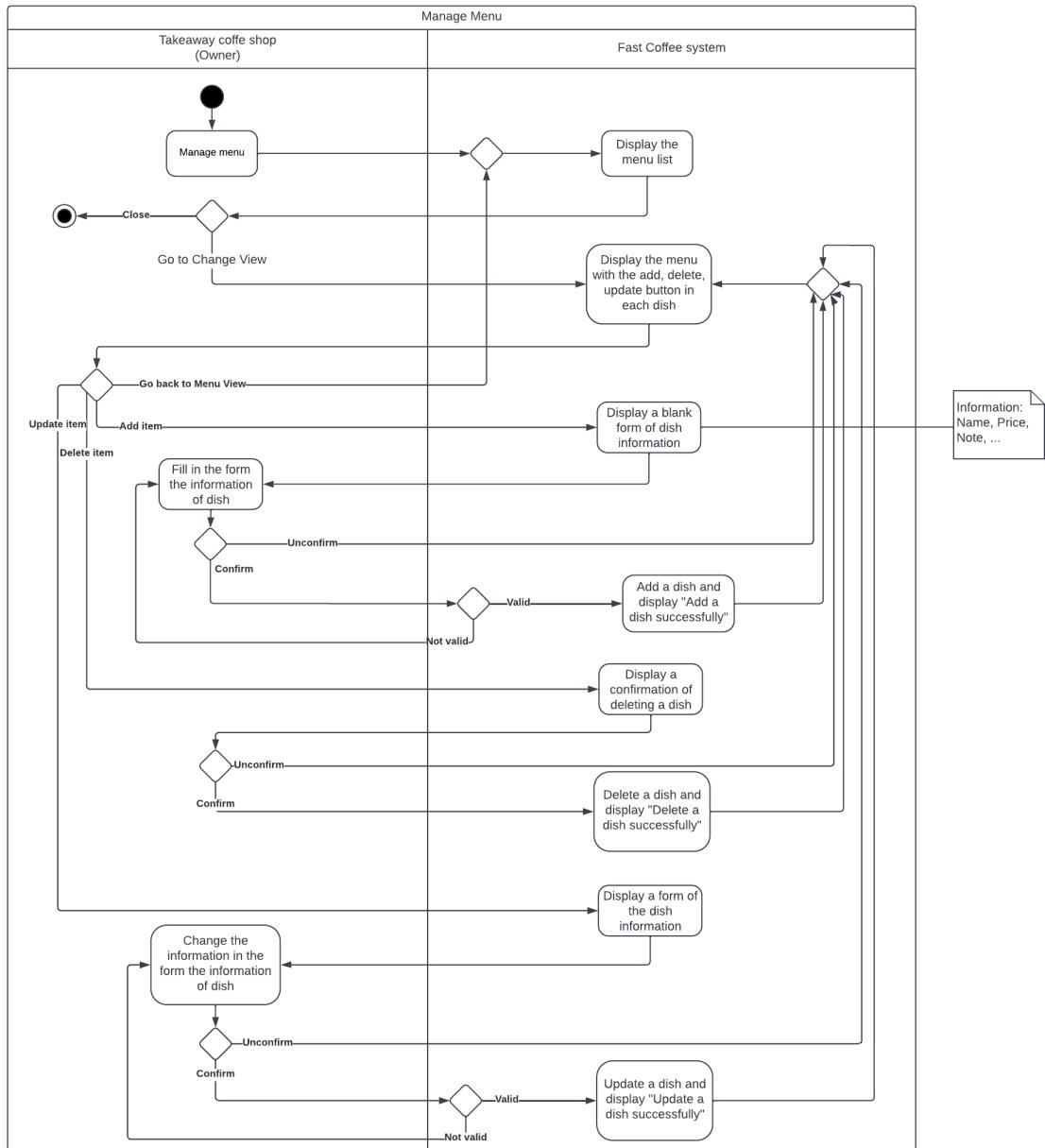


Figure 16: Activity diagram of Shop Management module - Manage Menu

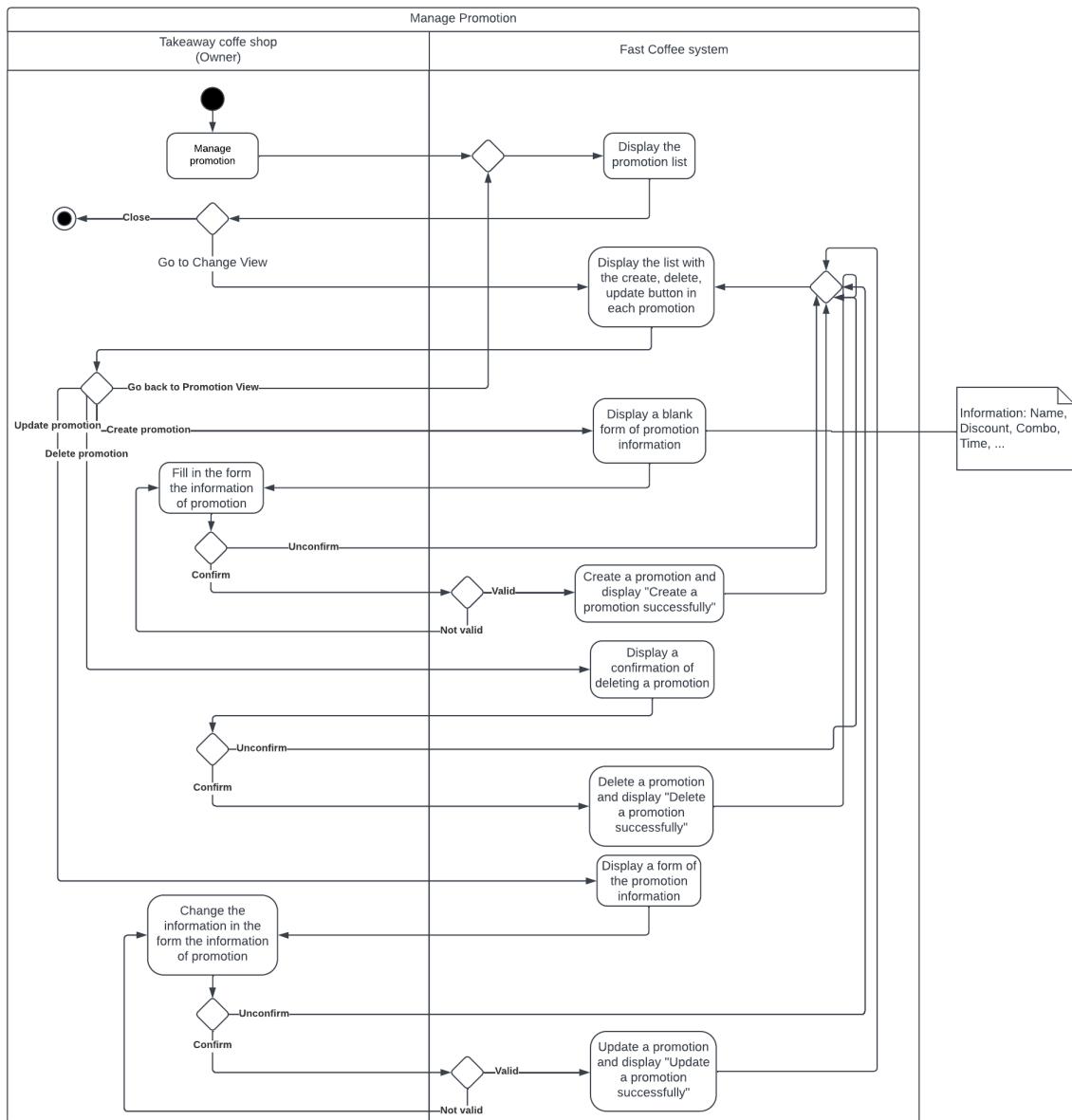


Figure 17: Activity diagram of Shop Management module - Manage Promotion

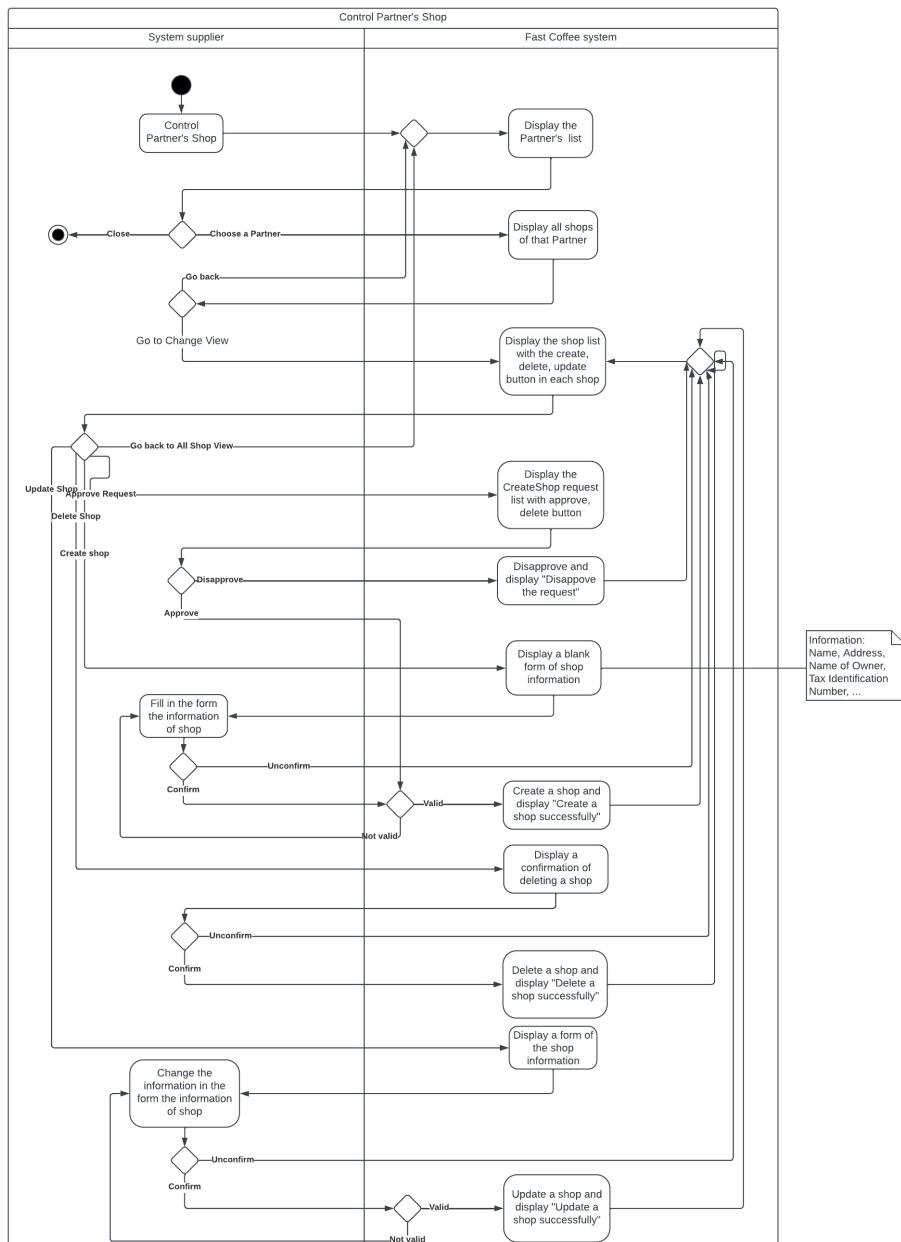


Figure 18: Activity diagram of Shop Management module - Control Partner's shop

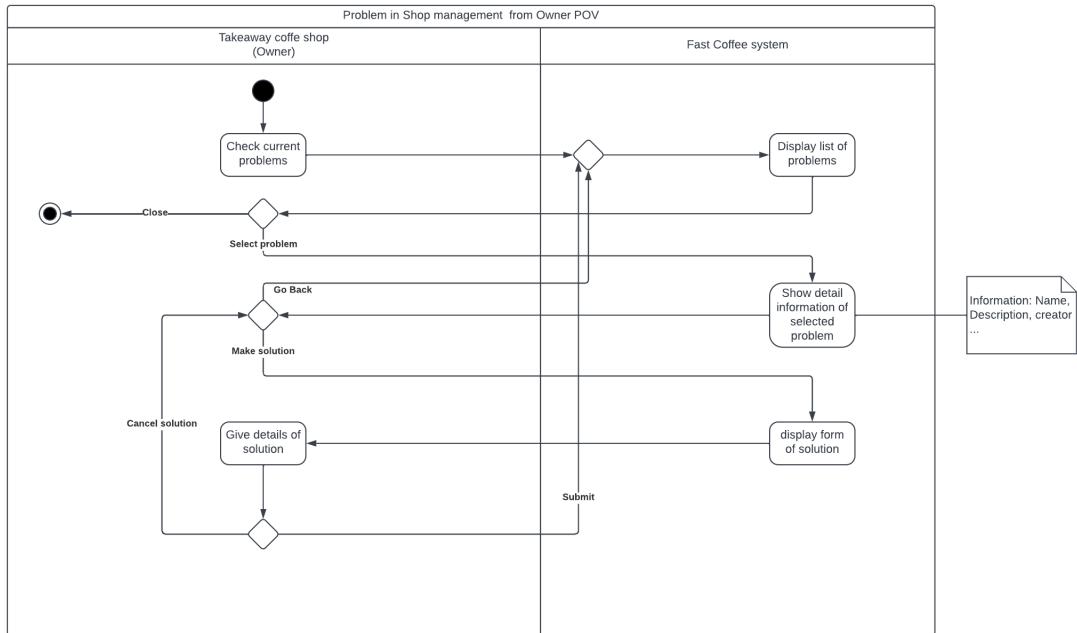


Figure 19: Activity diagram of Shop Management module - Problem management from Owner View

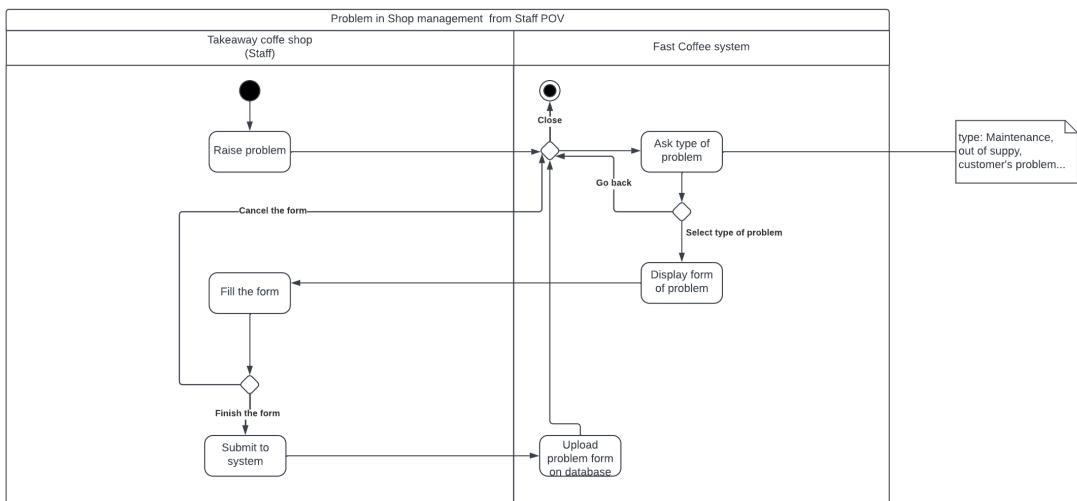


Figure 20: Activity diagram of Shop Management module - Problem management from Staff View

4.3.2 Sequence Diagram

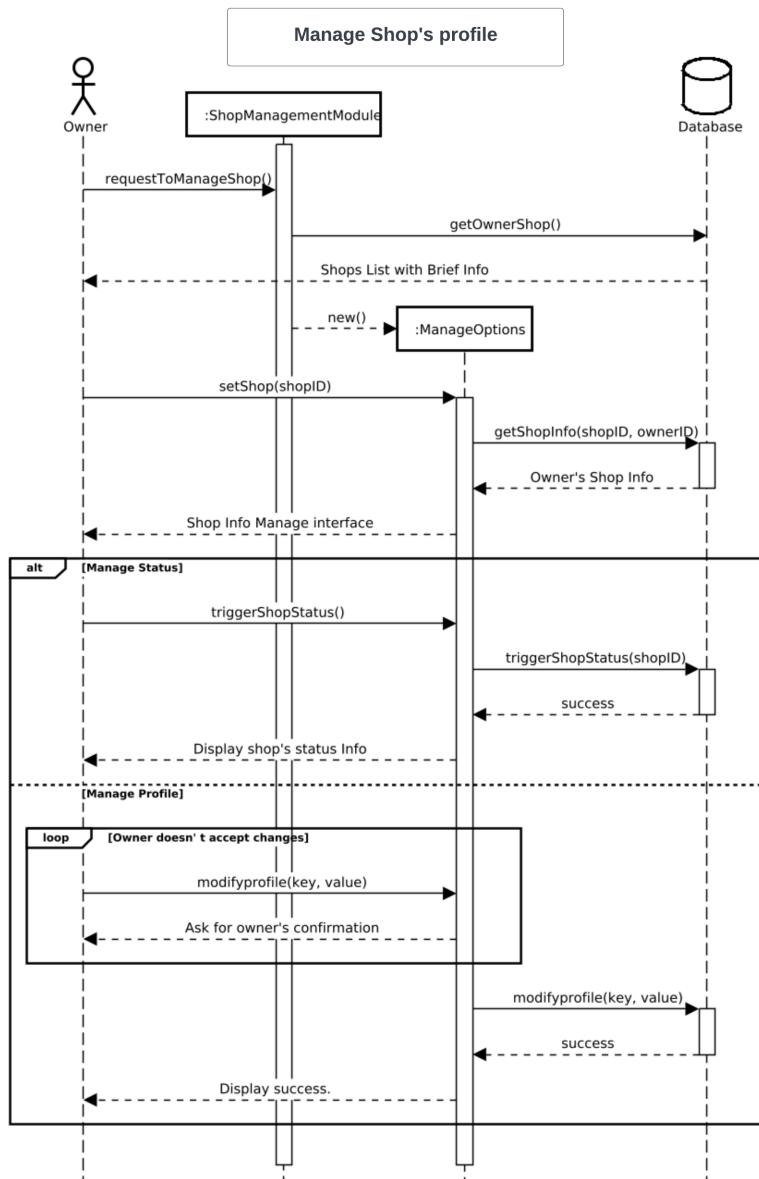


Figure 21: Activity diagram of Shop Management module - Manage Shop's profile

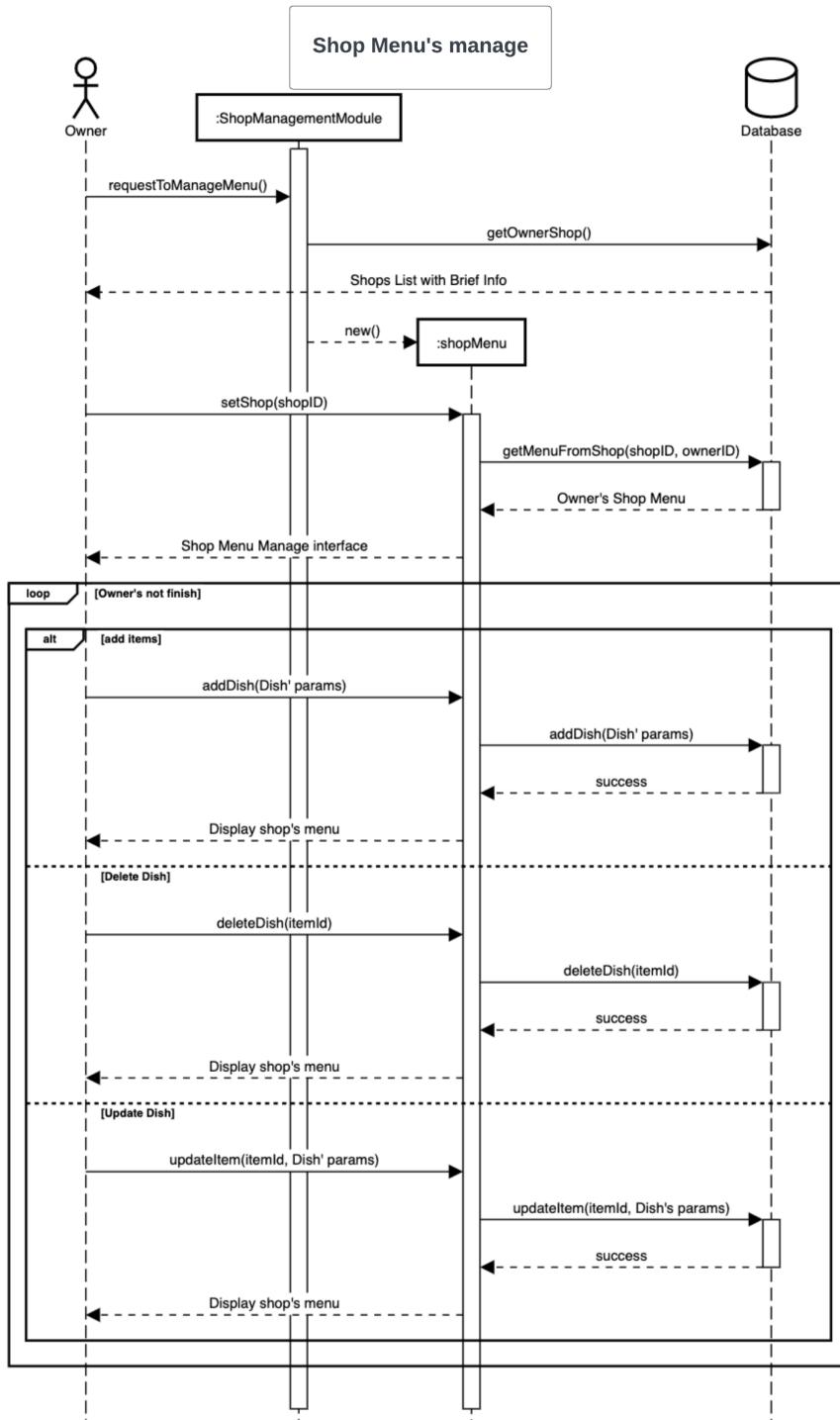


Figure 22: Activity diagram of Shop Management module - Manage Shop's menu

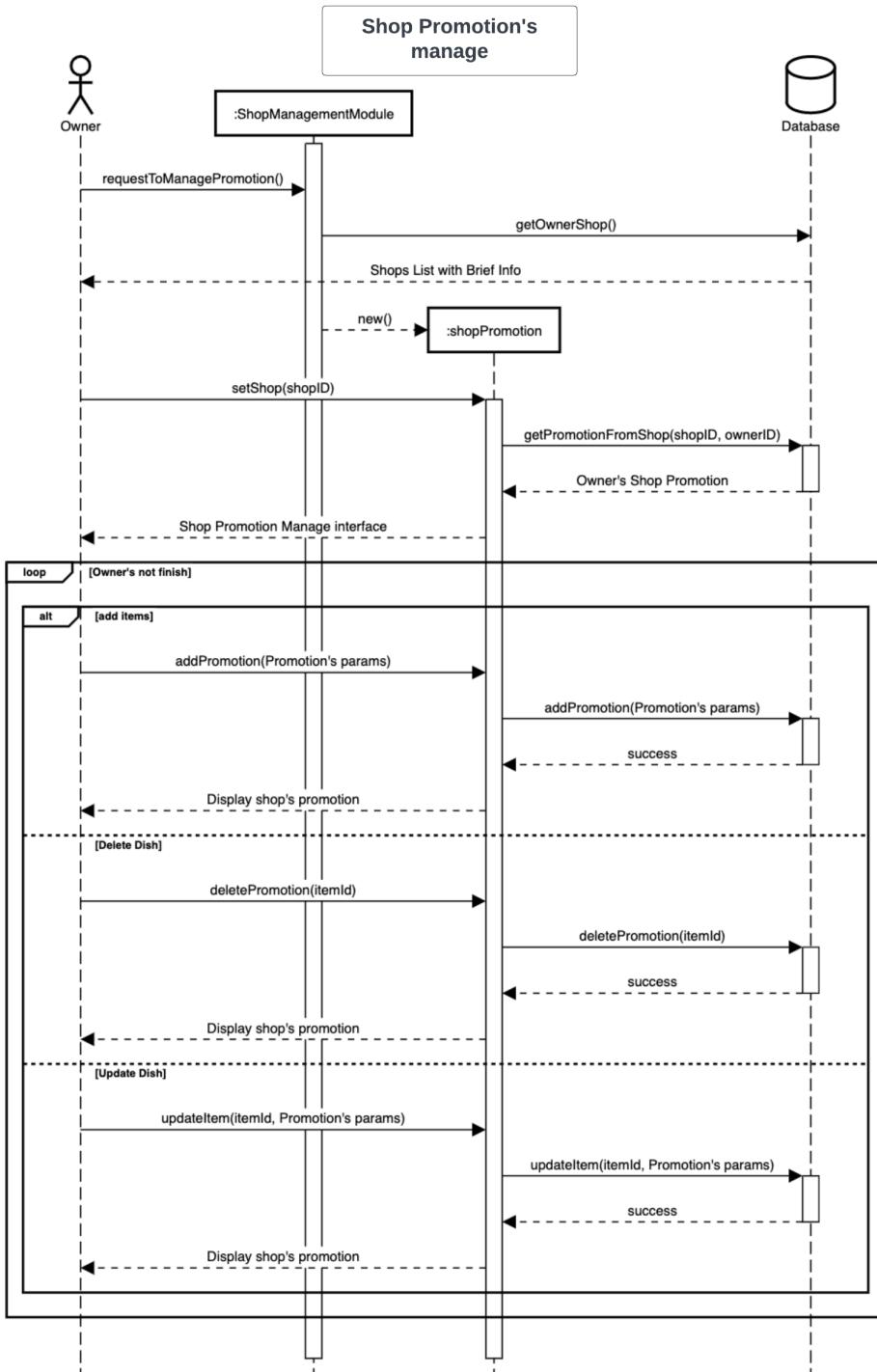


Figure 23: Activity diagram of Shop Management module - Manage Shop's promotion

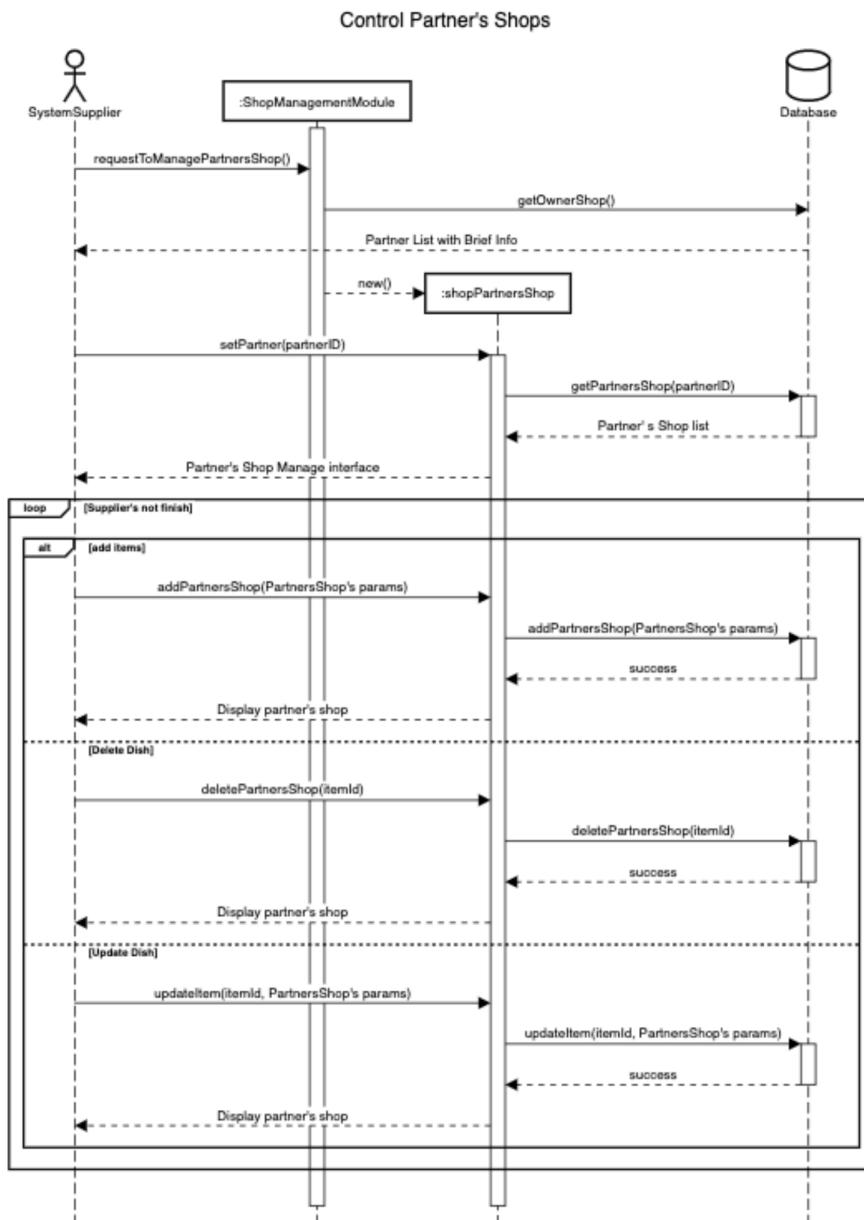


Figure 24: Activity diagram of Shop Management module - Control partner's shop

4.4 Order Management module

4.4.1 Activity diagram

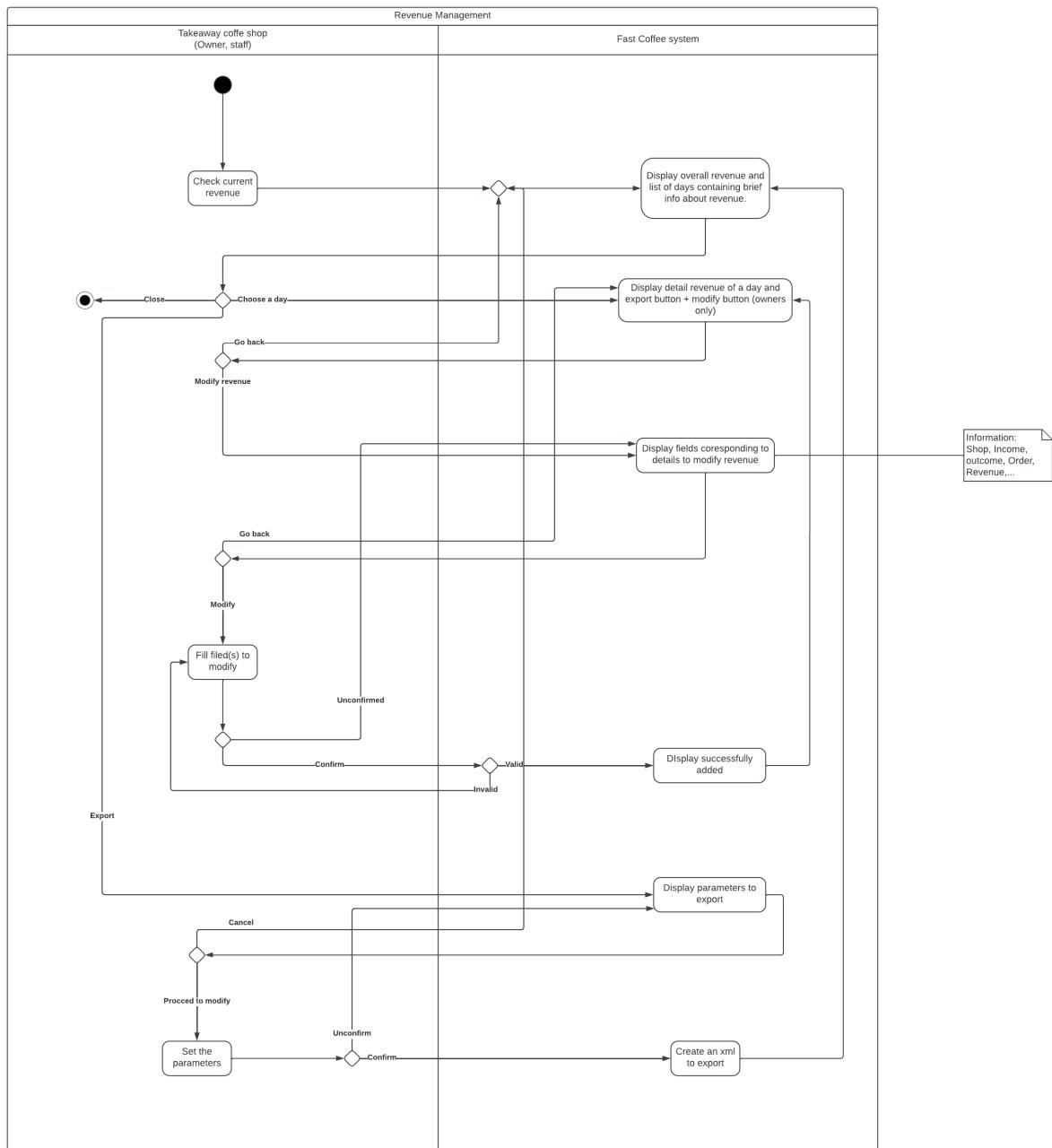


Figure 25: Activity diagram of Order Management module - Revenue Management

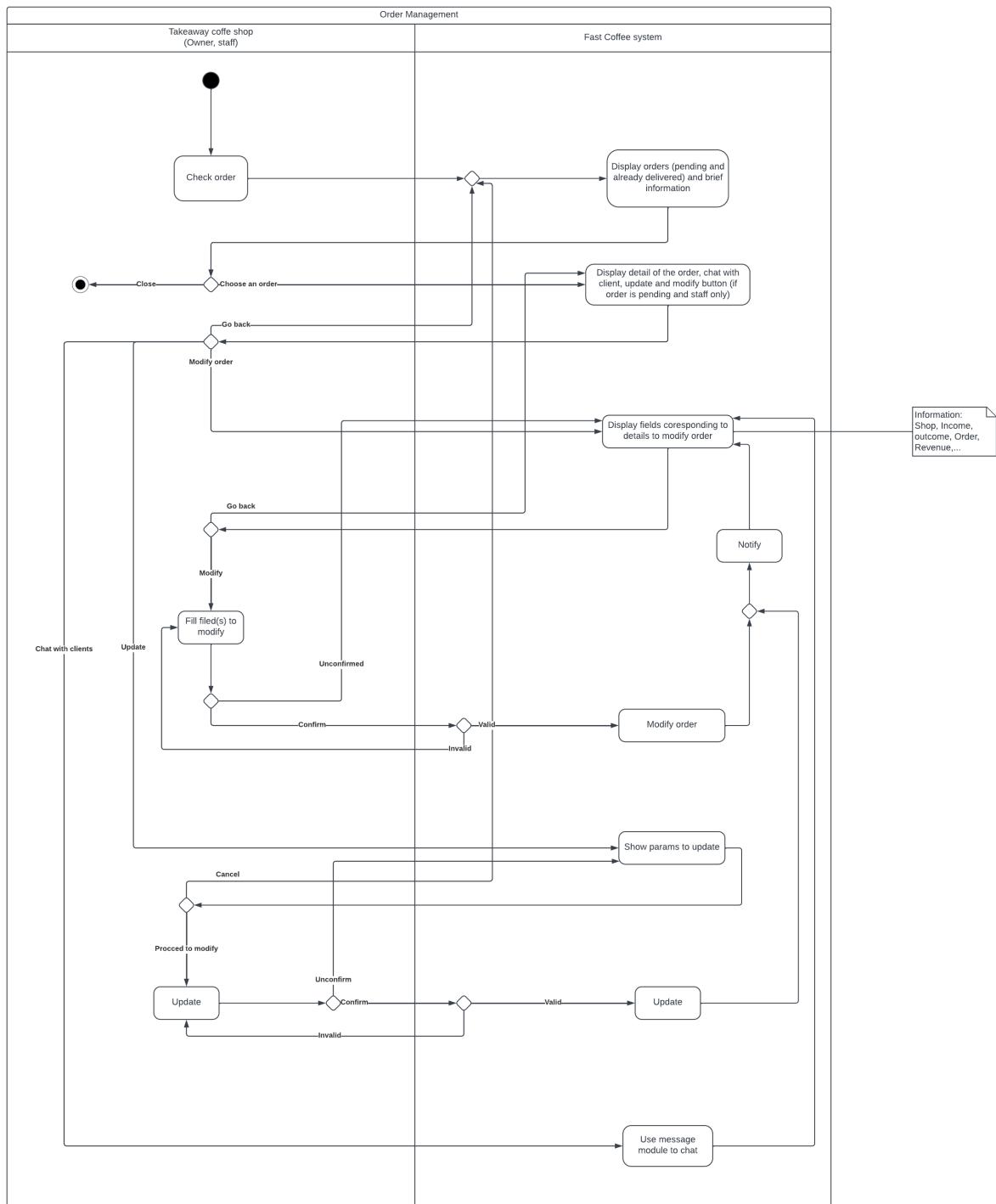


Figure 26: Activity diagram of Order Management module - Order Management

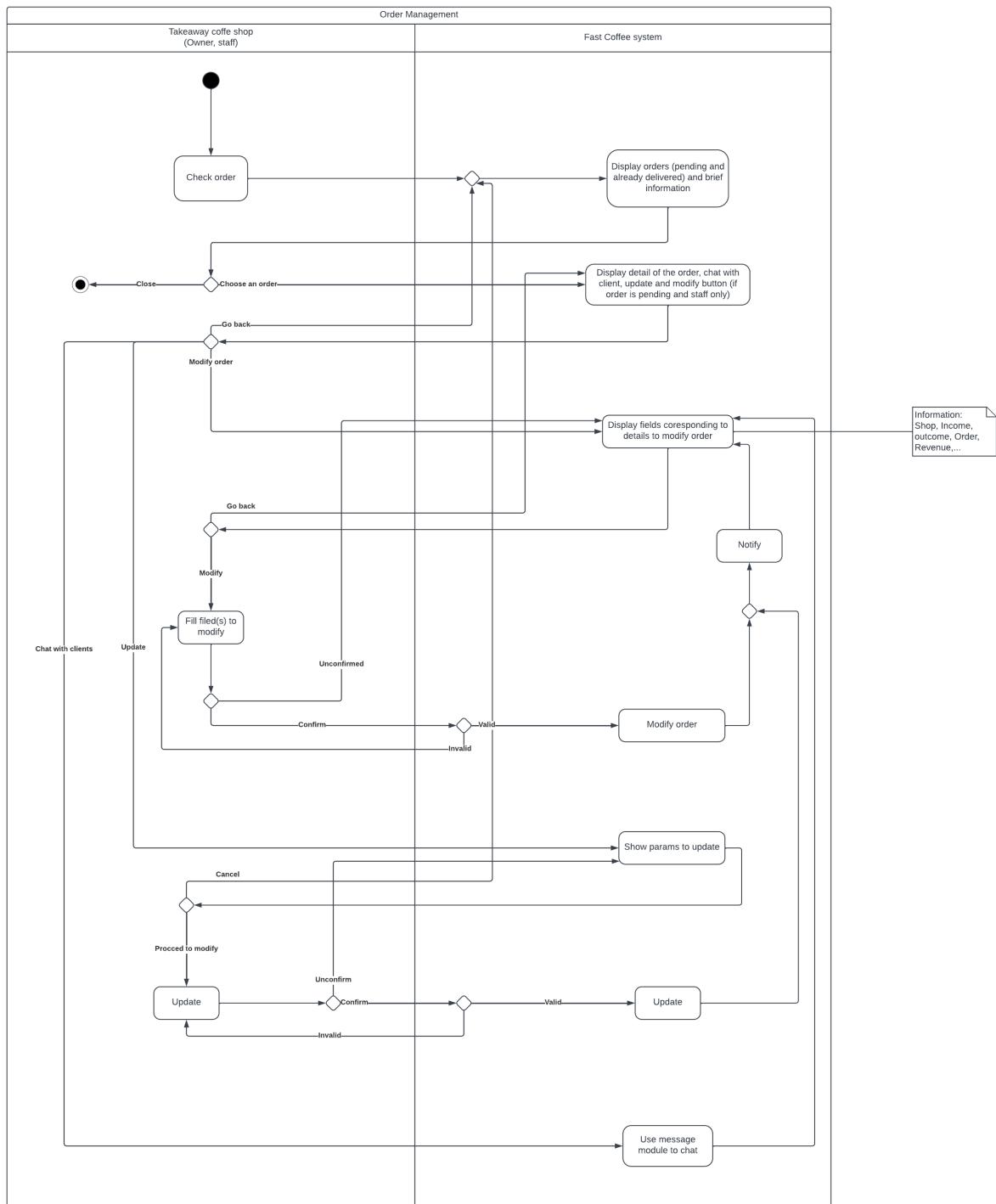


Figure 27: Activity diagram of Order Management module - Order Management

4.5 Customer experience module

4.5.1 Activity diagram

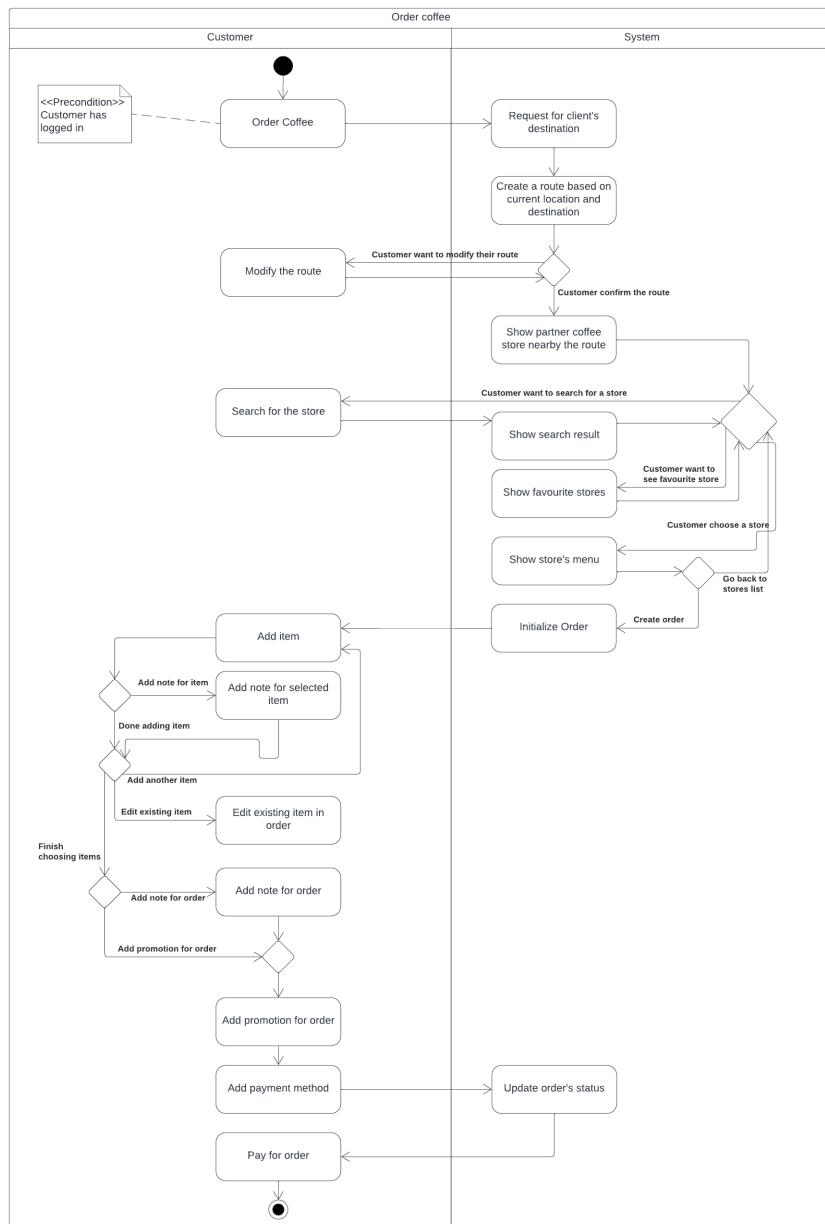


Figure 28: Activity diagram of Customer experience module - Order coffee

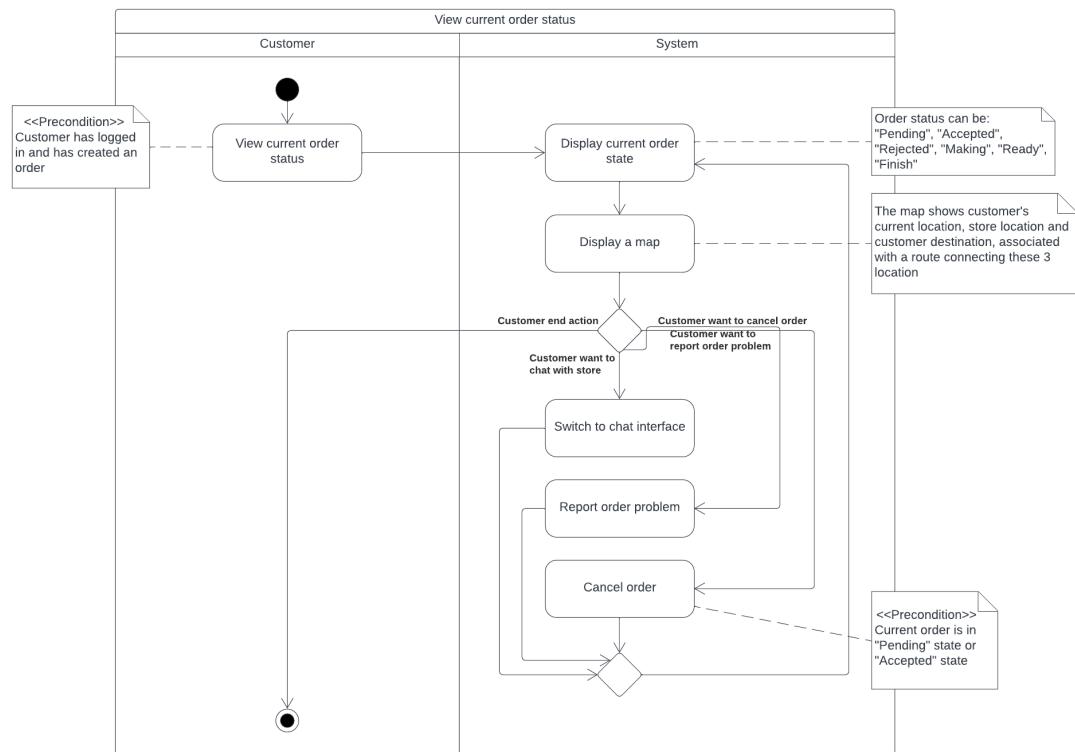


Figure 29: Activity diagram of Customer experience module - View current order status

4.5.2 Sequence diagram

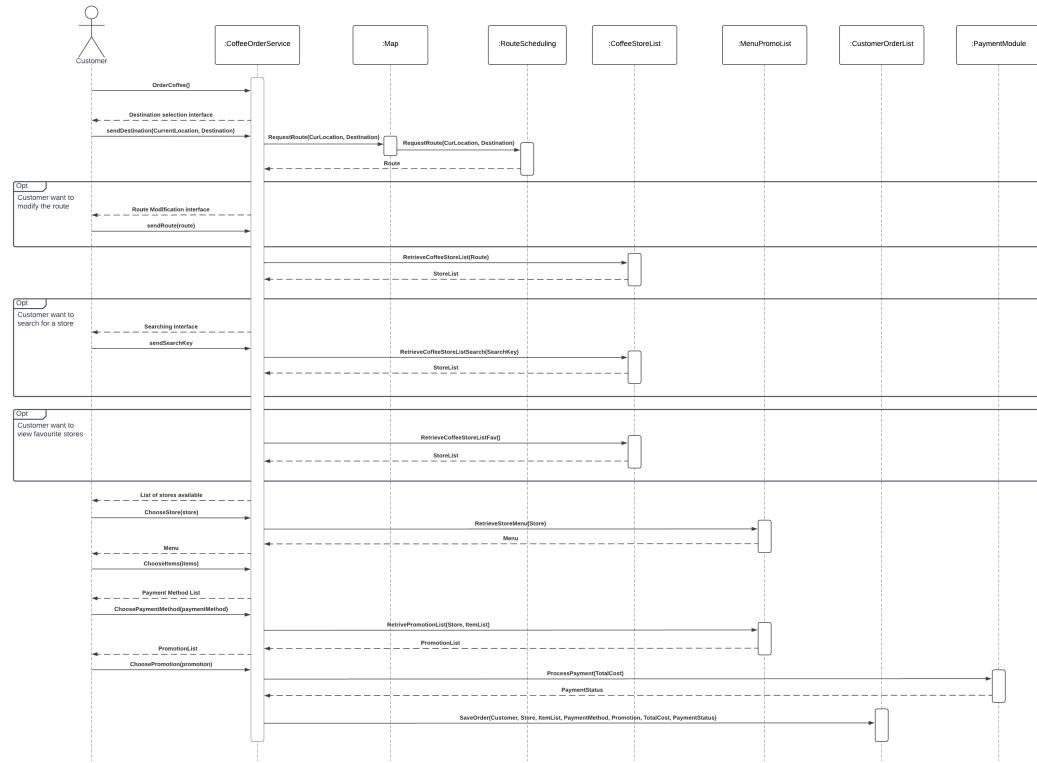


Figure 30: Sequence diagram of Customer experience module - Order coffee

4.6 Message module

4.6.1 Activity diagram

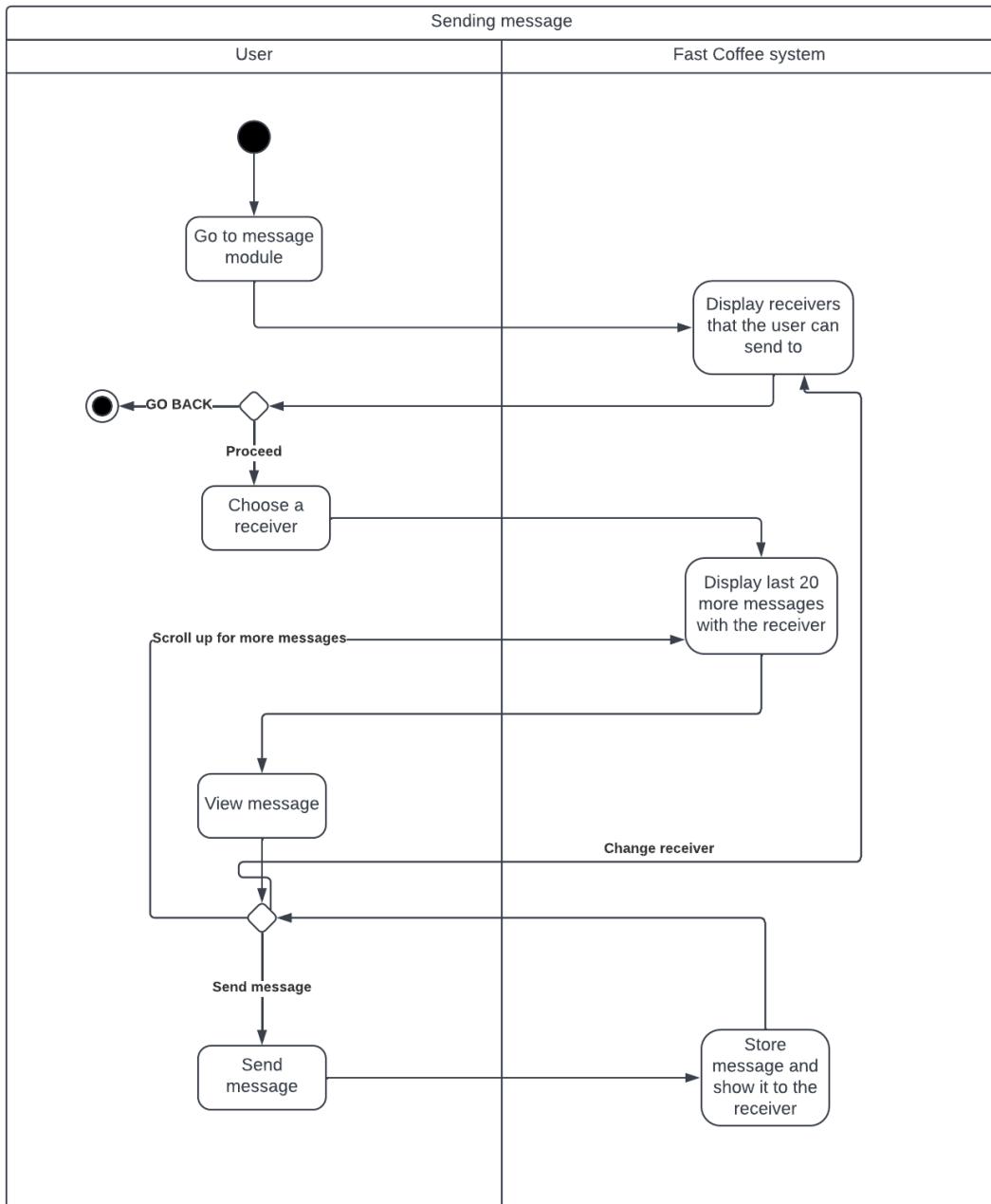


Figure 31: Sequence diagram of Message module

5 Class Diagram

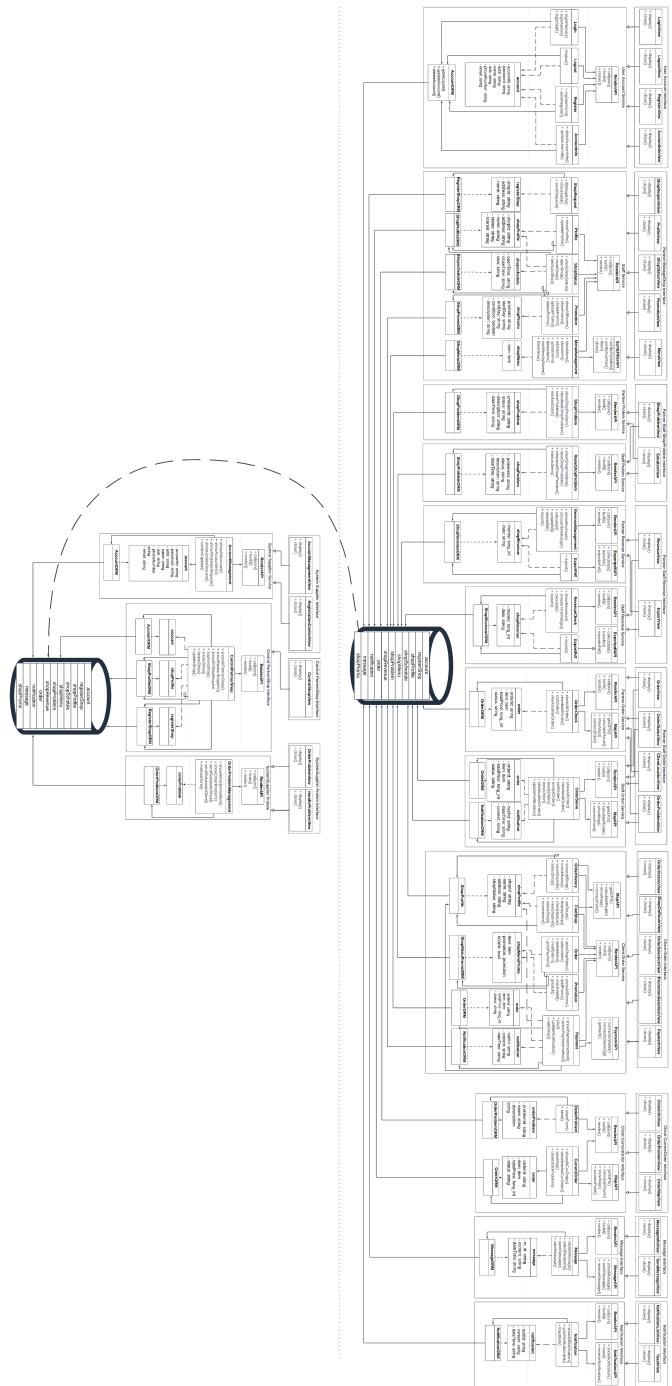


Figure 32: Class diagram and Architecture of entire system



Link for class diagram:
https://drive.google.com/file/d/1QzXWJ8CGdz65zpc-gHZC_C4-4772lGcQ/view?usp=sharing

6 Database Design

6.1 Enhance Entity Relation Diagram

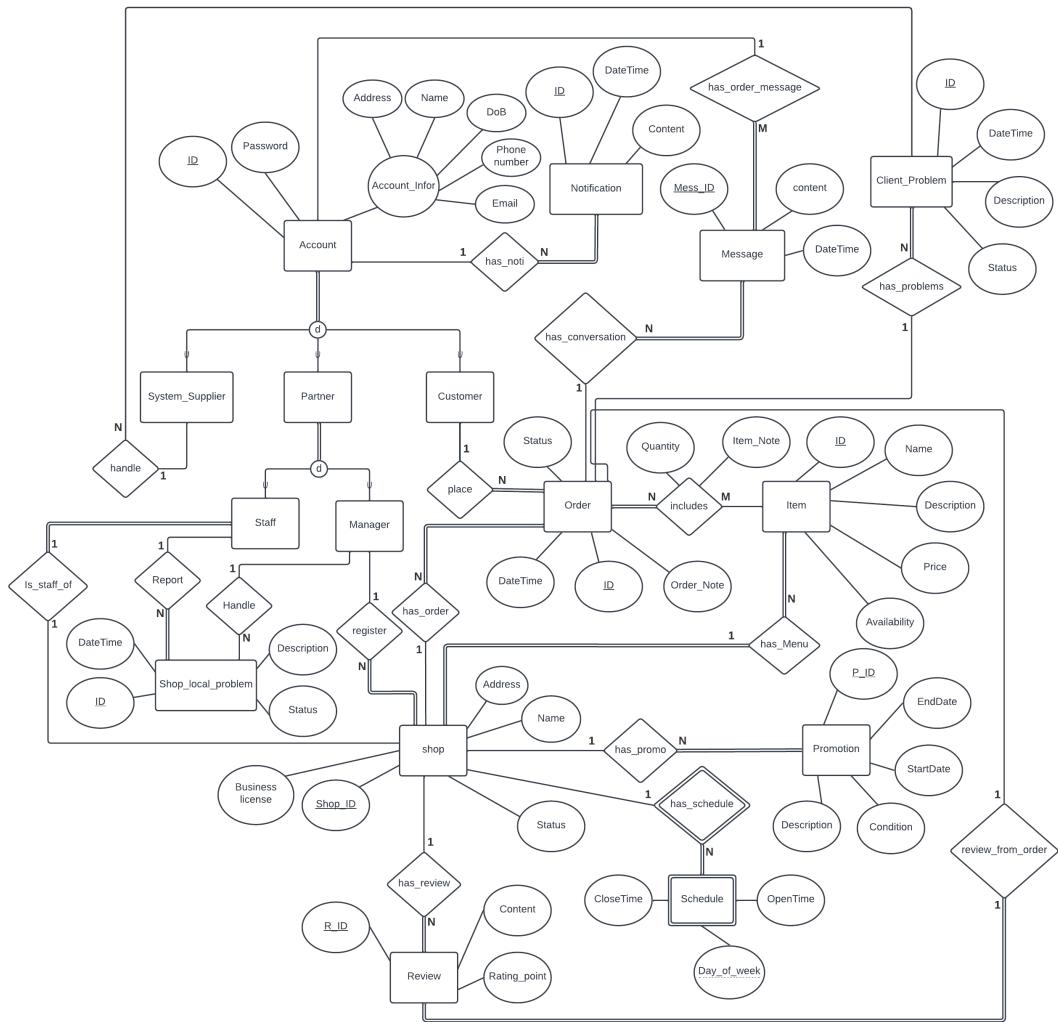


Figure 33: EERD of the system

6.2 Relational Schema

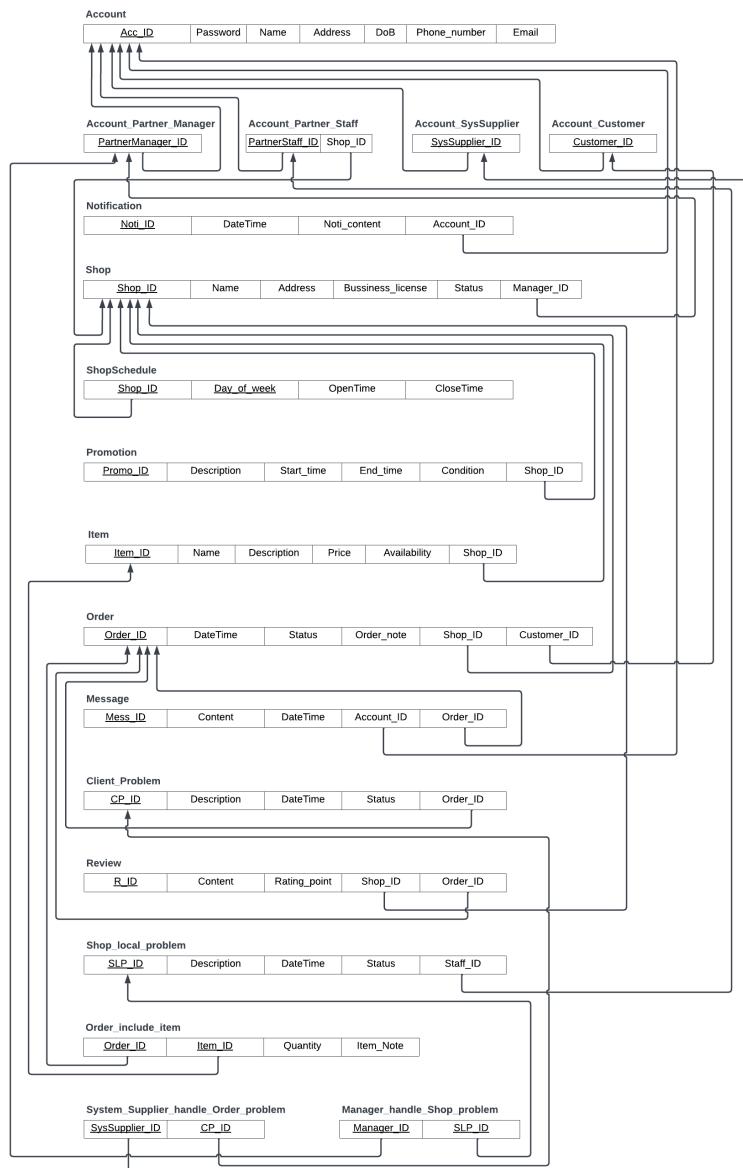


Figure 34: EERD of the system

6.3 Database Management System and Host

In this project, our group decide to use [MySQL Workbench 8.0 CE](#) as Database Management System. The Database is hosted on [Railway](#). It is a free server platform for database hosting. We can connect through Railway host by using MySQL Workbench with the provided credentials.

- mysqldatabase: railway
- host: containers-us-west-173.railway.app
- password: S8fxWJbYCTCSvqRGICzH
- port: 5631
- user: root

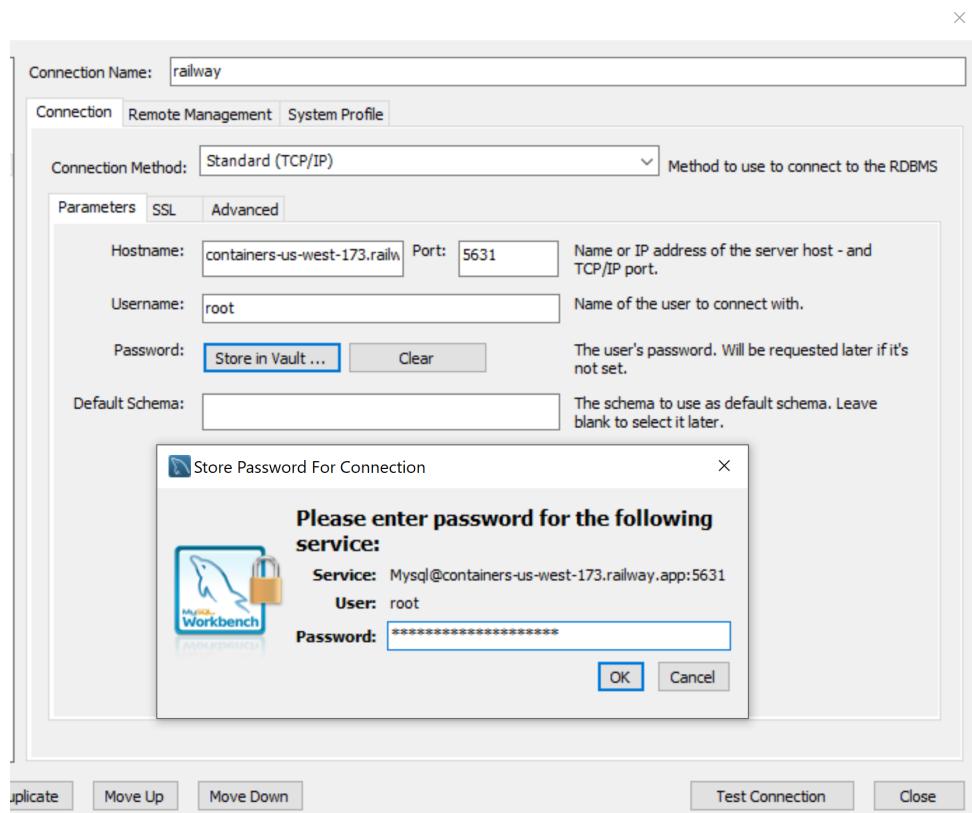


Figure 35: Example of creating connection to host



7 Implementation

- Database We have hosted the database in railway.app

- Server We have hosted the server in the vercel.com

This is our source code for Back-end (Server): <https://github.com/baonguyenduy2002/FastCoffeeBE>
The link to the server: <https://fast-coffee-be.vercel.app/>

All of the above hosting service we use is free for student. So there is some limit in bandwidth or disconnection.

If you meet some problem when loading the data in the UI, please call us: 0963991765



7.1 Shop Management Web

This is our source code for Shop Front-end:

<https://github.com/baonguyenduy2002/FastCoffee>

We use:

- Language: Javascript, html, css
- Framework: Reactjs, Bootstrap
- And some libraries for fetching data

To deploy our Shop Management Web, you can follow these step:

1. Clone repository from this github to your VSCode:
<https://github.com/baonguyenduy2002/FastCoffee>

2. Open the terminal and download the libraries in our project by typing "npm install"

```
PS C:\Users\Lenovo\Desktop\HCMUT221\fastcoffee\FastCoffee> npm install
```

Wait a moment for the libraries installment.

3. Go to "fastcoffee" folder, then run the code by this:

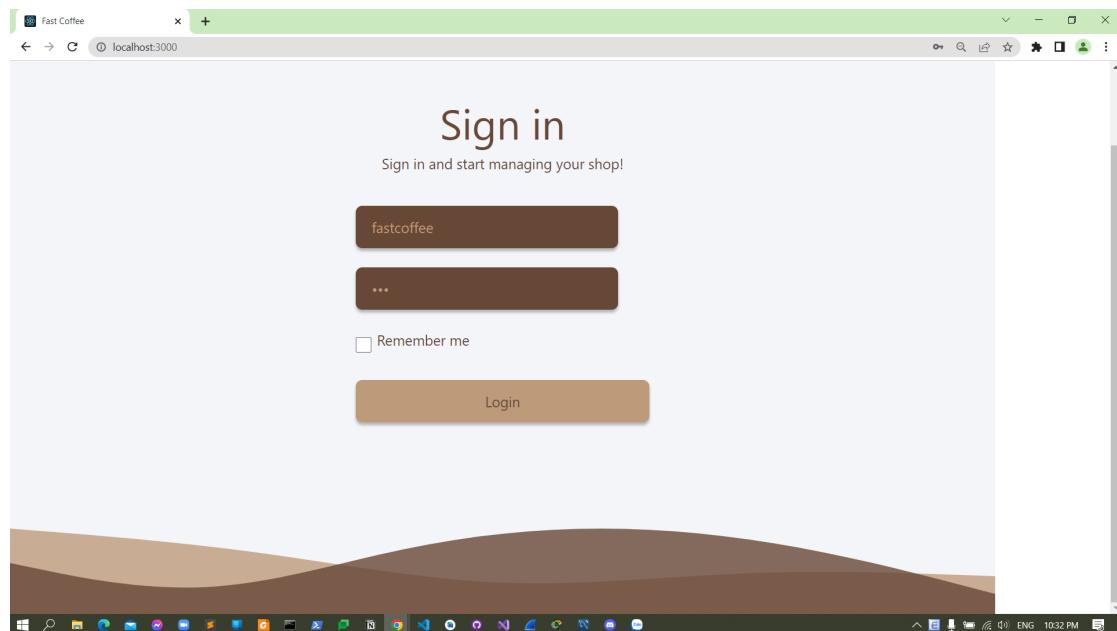
```
PS C:\Users\Lenovo\Desktop\HCMUT221\fastcoffee\FastCoffee> cd fastcoffee
```

```
PS C:\Users\Lenovo\Desktop\HCMUT221\fastcoffee\FastCoffee\fastcoffee> npm start
```

After run "npm start", the web will be run on your localhost:3000 and automatically opened in your default browser. The terminal will display this when loading the page. Just wait a

moment **Starting the development server...**

After the above steps, the login UI will appear:



The account is "fastcoffee" and the password is "123". You can set Remember me for no need for register in the next time.

7.1.1 Dashboard Page

The Dashboard page contains basic information of the shop like shop name, address, menu,... Only the shop owner can change the information on the dashboard:



7.1.2 Menu Management Service

At the menu page, the shop owner and the employee can edit the menu, how each drink will be called, how they cost and the descriptor of the drink. The shop owner can choose to add or delete

Highlands Coffee
0.2km - XL Ha Noi, Thảo Điền, Quận 2

Menu

Ice Black Coffee
Classic Black Coffee with Ice
49.000đ

Ice Black Coffee With Condensed Milk
Classic Black Coffee with Ice and Condensed Milk
59.000đ

Cappuccino
A combination of a single espresso shot and hot milk, with the surface topped with foamed milk
55.000đ

Mango Milk Tea
Gongcha Milk Tea with Mango and black bubble
59.000đ

Jasmine Milk Tea
Milk Tea with Jasmine flavor
55.000đ

Honey Lemon Tea
Honey and Lemon flavored tea.
55.000đ

Dark Chocolate Macchiato
Bitter Dark Chocolate with Coffee
62.000đ

hello cafe
fffffffffsdfsdfasdfasdfs
15.000đ



7.1.3 Order Management Service

The Order Management page:

This screenshot shows a web browser window titled "Fast Coffee" at "localhost:3000/order". The main content area is titled "Your current ORDERS" and contains a list of pending orders. To the left of the list is a sidebar with icons for home, refresh, search, and other navigation. Below the icons is a list of order stages: Pending, Accepted, Processing, Ready, Finished, Denied, and Cancelled. The "Pending" stage is currently selected, indicated by a blue background.

You can see the order in the stage according to the collapse container by clicking the content of that stage of order:

This screenshot shows a detailed view of a pending order. The order number is "# 3" and it was placed on "20/11/2022 at 7:0:0". The order status is "Pending". The order details show three items: "hello-world-coffee" (Quantity: 1), "fullstack-coldbrew" (Quantity: 2), and "python-smoothie" (Quantity: 1). Each item has a note field: "Note:" for coffee, "Note:" for coldbrew, and "Note: I want it sweeter" for smoothie. The sidebar on the left shows the collapsed stages: Pending, Accepted, Processing, Ready, Finished, Denied, and Cancelled. The "Pending" stage is currently expanded.

The staff can change the stage or the order by choose the appropriate button in each order. Then the order will moved into next stage. In this the database we host in railway.com is limit. If the order is not moved automactically, we can reload the page to see the result.



The screenshot shows a web application interface for managing orders. At the top, there's a header with a logo and the text "Fast Coffee". Below it, the URL "localhost:3000/order" is visible. The main content area displays two orders:

- # 2**
2/11/2022 at 7:0:0
Note: I want more milk in my coffee
Accept Deny
- # 7**
20/11/2022 at 7:0:0
Note:
Accept Deny

Each order entry includes a thumbnail image of a coffee cup, the item name, quantity, and a note. The bottom of the screen shows a Windows taskbar with various icons.

7.1.4 Profile Page

The profile of the shop fetched from the data will be displayed like this:

The screenshot shows a web application interface for viewing a shop profile. At the top, there's a header with a logo and the text "Fast Coffee". Below it, the URL "localhost:3000/profile" is visible. The main content area displays the following information:

- A large image of a coffee cup with the text "Highlands Coffee" and "Open".
- A graphic representation of a "BUSINESS LICENSE" card with placeholder text.
- A photo of a smiling man standing in a shop, identified as the "Manager Information".
- Manager Information details:
 - Name: Dinh Thao
 - Address: Street 1A
 - Date of Birth: 5/0/2023

The bottom of the screen shows a Windows taskbar with various icons.



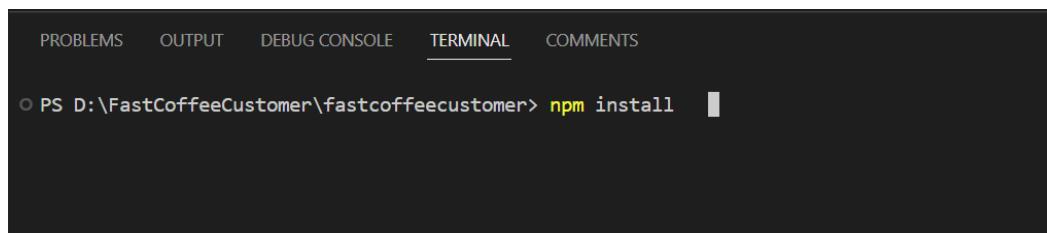
7.2 Client App

This is our source code for the Customer Android Mobile app:
<https://github.com/baonguyenduy2002/FastCoffeeCustomer.git>
We use

- Language: Javascript
- Framework: React Native, Expo
- And some libraries for fetching data

To deploy our Customer Android Mobile app, you can follow these steps:

1. Clone repository from this GitHub to your machine:
<https://github.com/baonguyenduy2002/FastCoffeeCustomer.git>
2. Open "fastcoffeecustomer" folder, right-click, and open the folder with VSCode.
3. Open the terminal and download the libraries in our project by typing "npm install"



A screenshot of the VS Code terminal interface. The tab bar at the top shows 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL' (which is underlined), and 'COMMENTS'. In the terminal window, there is a single line of text: 'PS D:\FastCoffeeCustomer\fastcoffeecustomer> npm install'. The background of the terminal is dark.

4. When the installation finish, run "npm install -g expo-cli eas-cli" with terminal
5. Run "eas build --profile development --platform android", an EAS build is created so that you can run on your Android device/emulator. Type "y" when it asks for keystone and anything else.
6. When it completes creating the build, you can use your android device to scan QR code display on the console, download and install the app, or you can choose to open it with your android emulator.
7. Once you finished installing the demo app, run "expo start -dev-client" command to start the Metro Bundler server for the demo app.

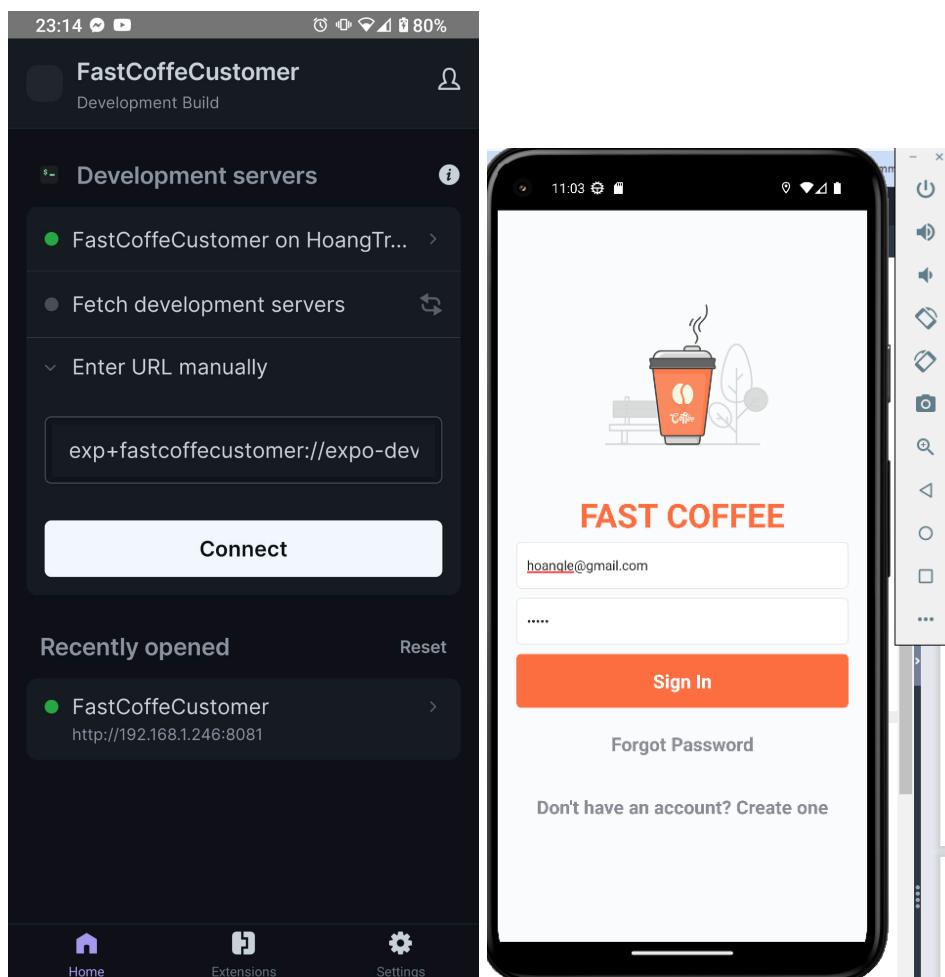


```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL COMMENTS

Starting project at D:\FastCoffeeCustomer\fastcoffeecustomer
Some dependencies are incompatible with the installed expo package version:
- react-native-reanimated - expected version: <2.12.0 - actual version installed: 2.13.0
Your project may not work correctly until you install the correct versions of the packages.
Install the correct versions of these packages, please run: expo doctor --fix-dependencies,
or install individual packages by running expo install [package-name ...]
Starting Metro Bundler


> Metro waiting on expo+fastcoffeecustomer://expo-development-client/?url=http%3A%2F%2F192.168.1.246%3A8081
> Scan the QR code above with Expo Go (Android) or the Camera app (iOS)
> Press a | open Android
> Press w | open web
> Press r | reload app
> Press m | toggle menu
> Press ? | show all commands
Logs for your project will appear below. Press Ctrl+C to exit.
```

8. If you run your app on your android device, you can scan the QR code to run the app or you can copy the url shown on the console and paste it into your app to run. If you run the app on the android emulator, simply press a to start the app.

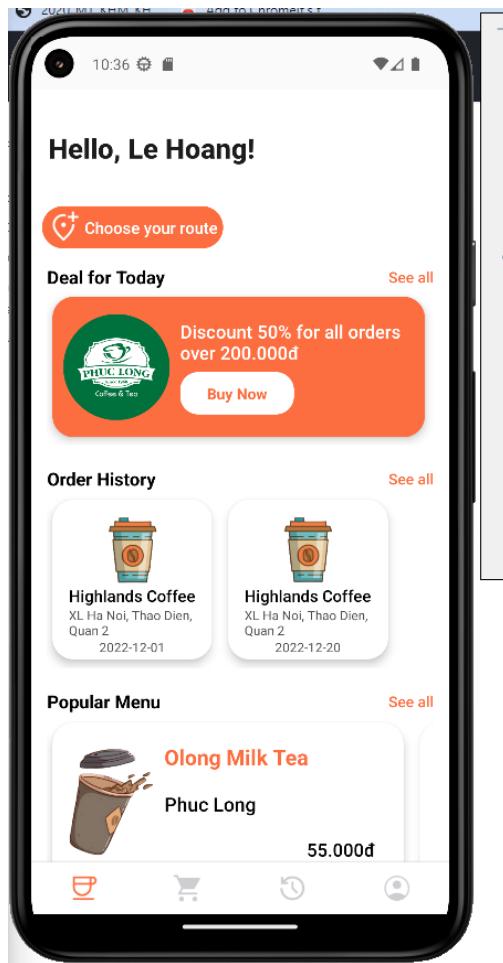


After the above steps, the login UI will appear
The account is "hoangle@gmail.com" and the password is "lh123".



7.2.1 Dashboard Page

When you logged in to the application, the Dashboard UI will be shown:



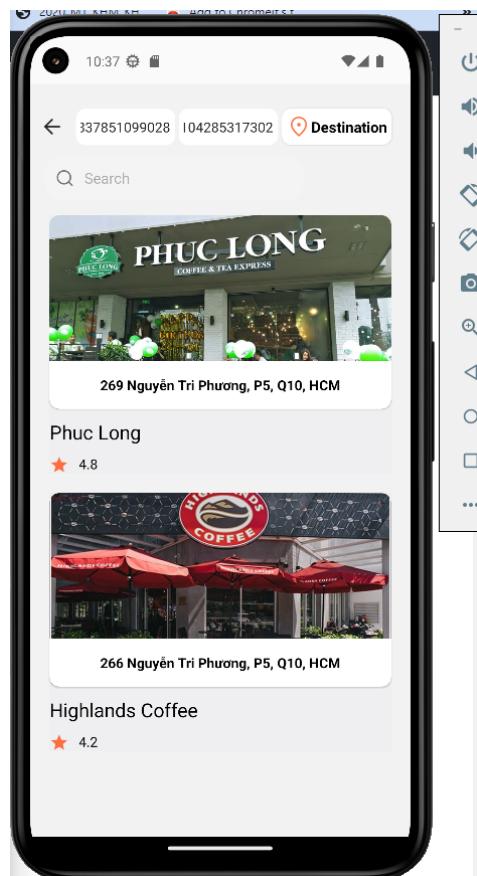
It includes three parts:

- "Deal for today" shows you the Promotions of some shops.
- "Order History" shows show your last order.
- "Popular menu" suggests most favorite drinks. You can press the "see all" button for more details of each part.

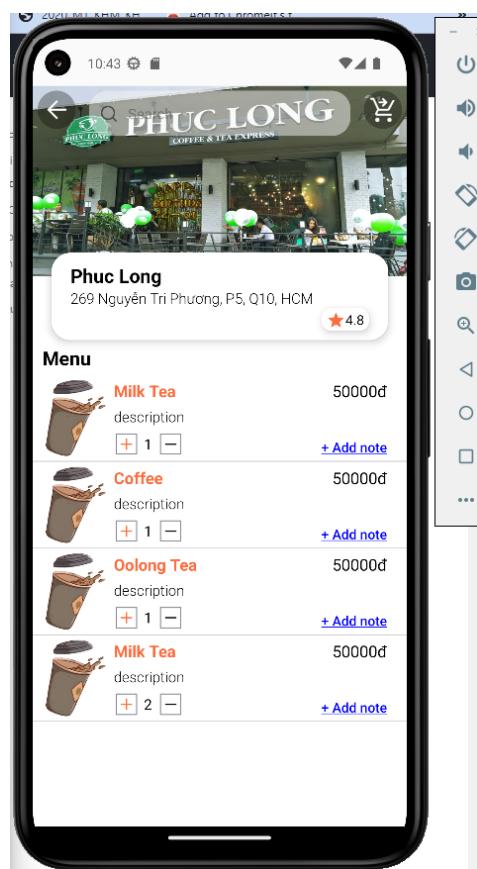
7.2.2 Order Management

Before placing your order, you need to follow these steps:

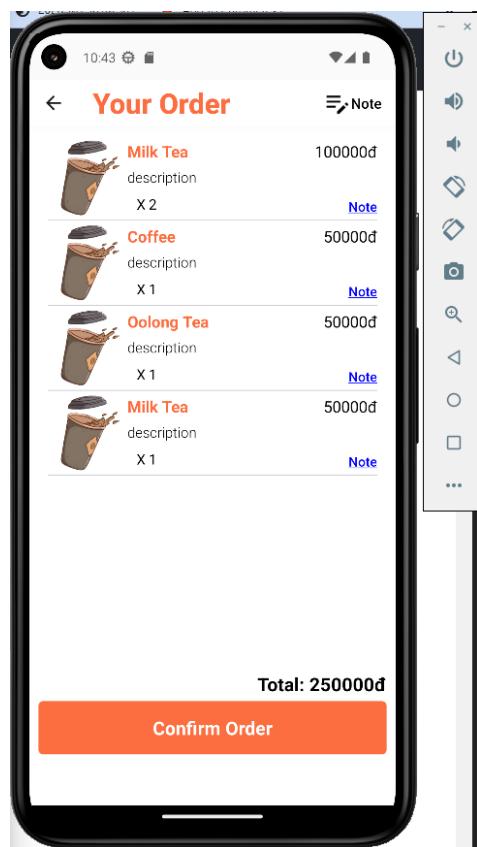
- Modify your destination. The system will suggest some shops based on the route created automatically.
- Choose a shop. Then, the system will show you the shop's menu.



- Place a order. You can place your order by pressing the "add" icon on each item, also modify the amount of it and add some notes.

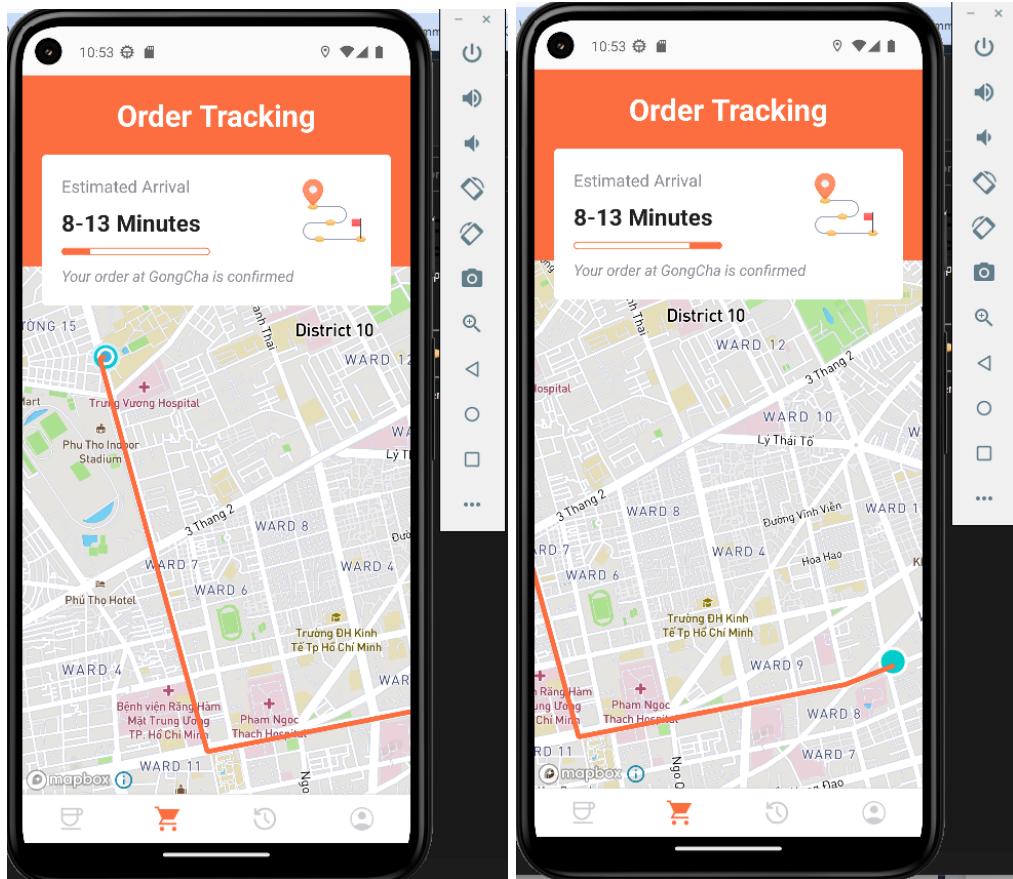


- The final step is checkout and confirm the order.



7.2.3 Order Tracking

After placing the order, you can follow the status of it on "Order Tracking Page":

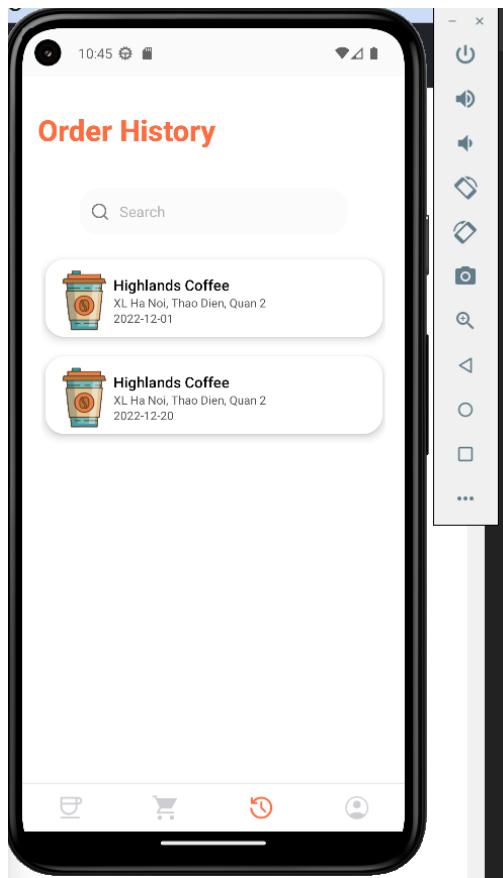


As images shown above, you can keep track the estimated time and the route to your destination.



7.2.4 Order History

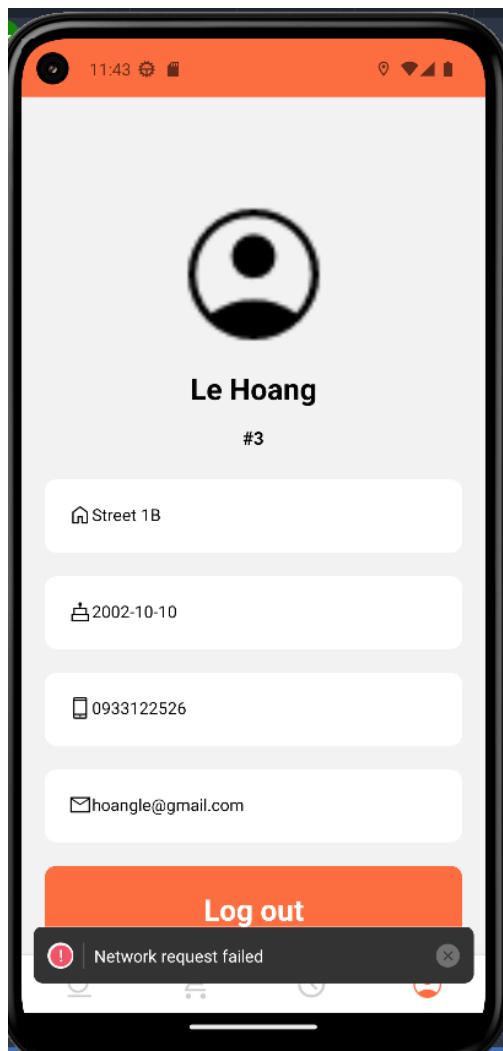
You can keep track your last order on "Order History Page".





7.2.5 User Profile

You can manage your personal information on "User Profile Page"





8 Self Evaluation

8.1 Achievements

After finished the system, our group gets some Achievements according to the planning phase:

- The system is working properly, both Web Application for Shops and Mobile application for Users.
- The Web works well with authentication system and can track for Customer's orders. The data is managed and displayed clearly.
- The Mobile application has been simplified designed for better experience.
- Most of the features are finished and working properly.
- The map tracking system is a great success in our implementation, which helps users tracks for the best available shop as well as helps the shops keep track for customer nearby according to the route.

8.2 What are not achieved yet

Although the system is finished, there are some features we do want to have in our design but not implemented yet. Due to our limitation, we did not yet finish these things:

- The application for services provider is not completed.
- There may be some minor bugs on the applications.
- Some functions on mobile app are not working properly yet.
- The database is hosted on a free server-platform, so the connection may not be unstable and cause bad effects on the system.

8.3 Future development

In future, the first thing our team have to do is to complete the whole system, which also means to implement and deploy the application for services provider. After that, we will have the system tested more properly to make sure that the system work well without many critical bugs. We are also planning for the multiple-language system, to help with the needs of international users.