

**VIETNAM NATIONAL UNIVERSITY - HOCHIMINH CITY
INTERNATIONAL UNIVERSITY
DEPARTMENT OF PHYSICS**



**GEOLOCATION APP
DEVELOPMENT FOR IOS
FINAL REPORT
Group 3**

Members:

Nguyễn Khánh An - SESEIU21002
Hồ Huy Ngọc Khuê - SESEIU21034
Võ Cự Khôi - SESEIU21020

1. Project Brief Description

1.1. Purposes

- Difficulty in finding reasonable rental accommodations

Following some surveys taken, the demand for renting accommodation in HCMC has increased by 15% compared to the year 2022. This has led to an increase in the rental price, which makes the people in demand find a place with reasonable prices along with the desired house features.

- Difficulty in leasing rental accommodations

For landlords who have accommodations for rent that meet the requirements of a large number of people in demand, public information about their places for rent will be distracted by several profiles from other agencies.

1.2. Solution

Understanding the difficulties and the demand for rent and the availability of houses for rent is crucial for various stakeholders, we aim to develop the MARA application, which will be a safe and convenient platform for searching, renting, and securing rental properties.

1.3. Main functions

Lessees	Lessors	Admin
- Login - Search - Chat - Nearby-places - Payment	- Login - Payment (received payment, return payment) - Keep track of renting Houses/apartments - Add house/apartment for rent properties - Chat	- Announcement - Accounts management - Policy - Payment (Service fees) - Service Centre - Advertisements - Chat - Integration

Table 1: *Main functions table*

1.4. Project charter

Project Name: Mobile Application for Rental Accommodations (MARA) app	
Project Description	The app connects lessees and lessors to provide long-term (>= 6 months) accommodation (apartments, houses) renting services for workers and students in HCMC and Binh Duong. The app will be available on Google Play and App Store.

Project Scope	In-Scope	- Mobile app (iOS, Android)
	Out-Scope	- App maintenance, Model maintenance - Updates
Deliverables		- News on related topics - Tips on related activities
Target Customer		- Students, small families, and single individuals
Stakeholders		- Administrators - Lessees (Users) - Lessors (Users) - Payment service - App developers - Testers
Project Team		- Khánh An, Ngọc Khuê, Cự Khôi

Table 2: *The project charter*

2. Schedule estimation

Phase	16 (Jan)	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31 (Jan)
UI Code							Meeting									
Combination UI + Backend													Meeting			
Docs + Slides																
Presentation Day																

Table 3: *The schedule table*

We planned to finish the project in a total of 16 days, including the presentation day. The project idea was chosen after our discussion on the previous day of the starting day based on our ideas and research following the project's requirements.

The UI coding phase was planned to last for 7 days, which played an important role in setting our general view of the app's UI. A meeting would be held at the end of the phase to conclude our codes and ideas on the interface of the app.

The next phase was to combine our UI codes with the backend to complete the whole UI of the app started right after the previous phase, would last for 6 next days. It is expected to have a complete interface that represents well all the app's functions as our initial expectation at the end of this phase.

After that, we spent 2 days giving all the most detailed content in a document as a report and preparing for the presentation the next day. The presentation was expected to last for about 15-20 minutes and to be well-represented in our outcome from this course.

3. Actors & Actors Functions

Lessees	Lessors	Admin	Payment service
<ul style="list-style-type: none"> - Create a lessee's account - Login - Search for accommodations (with the filtering function provided) - Chat - Nearby-places - Payment - Recommendation on distance and cost - Current renting status/track - Review/Feedback 	<ul style="list-style-type: none"> - Create/Upgrade the lessor's account - Login - Payment - Keep track of renting Houses/apartments - Add house/apartment for rent properties - Chat 	<ul style="list-style-type: none"> - Announcement - Accounts management - Policy - Payment (Service fees) - Feedback/Customer Service Centre - Advertisements - Chat - Integration 	<ul style="list-style-type: none"> - Policy - Service fees - 2-way transaction - Service maintenance

Table 4: *Actors functions table*

4. Business Processes

4.1. Searching Process

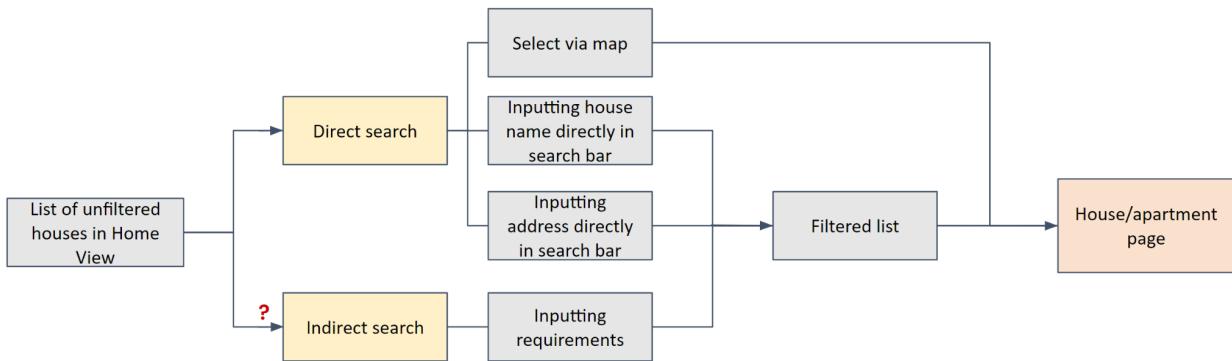


Figure 1: *Searching Process*

At the Home View, users can search for suitable houses that meet their needs. There are 2 methods, which are direct searching and indirect searching. Direct searching can be used via the search bar or the ‘Select via map’ button on the Home View. Using the search bar can identify and give results of a list of houses in the database that have names or address information matching the given input. On the other hand, users can activate the ‘Select via map’ button to open a Map View showing their current location, which has markers on the location of houses that contain a direct navigation link to their House View details. Besides, Map View also comprises marked nearby locations for further references of filtering. The indirect searching will be applied as a function of the filter button in the upper toolbar of the Home View. An algorithm for processing input requirements to suggest a list of suitable house recommendations will be developed.

4.2. Uploading Process from Lessors

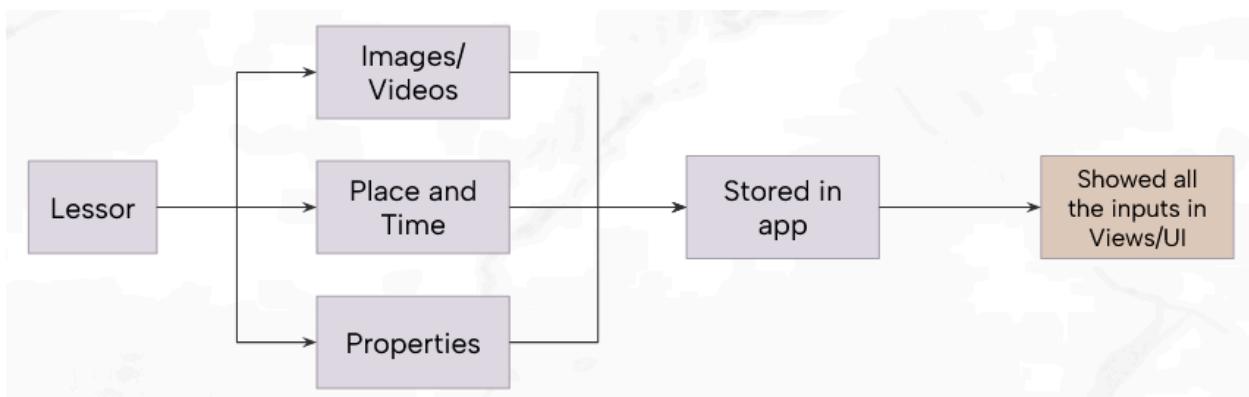


Figure 2: *Uploading Process*

4.3. Payment Process

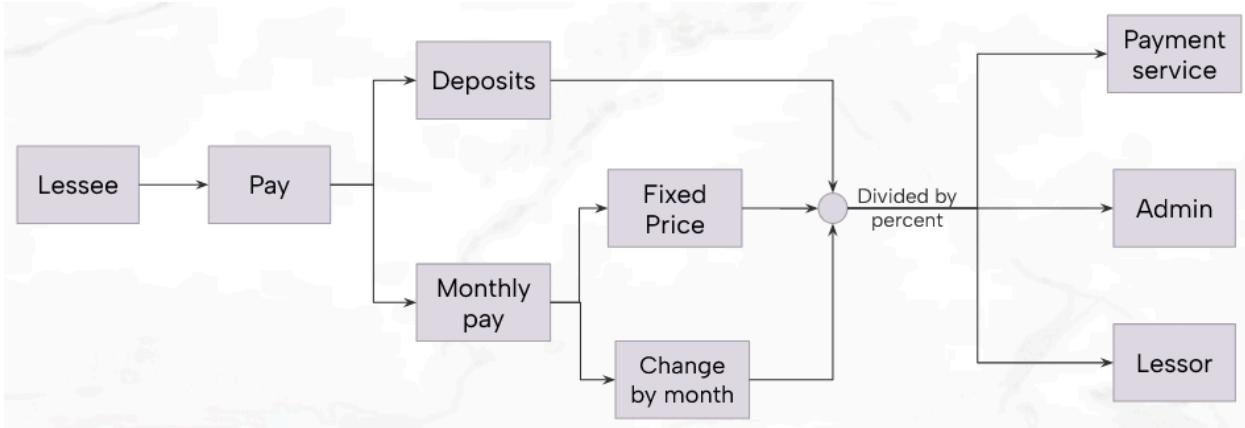


Figure 3: Payment Process

The Purchase View will show a few options to pay based on the rental house progress. If it is the first time purchase since the start of the rental contract, a deposit amount is required to pay depending on the lessor's requirement (from 1-2 times the monthly rental fee). During the renting time, users as lessees have to pay monthly payments including the fixed price (room/house/apartment) and the price that can be changed each month (i.e. electricity, water, etc.) The payment process is conducted through the payment service provider. The final in-money will be divided into 3 parts of different percentages depending on the demand of the payment service, the app, and the lessors.

4.4. Profile Editing Process

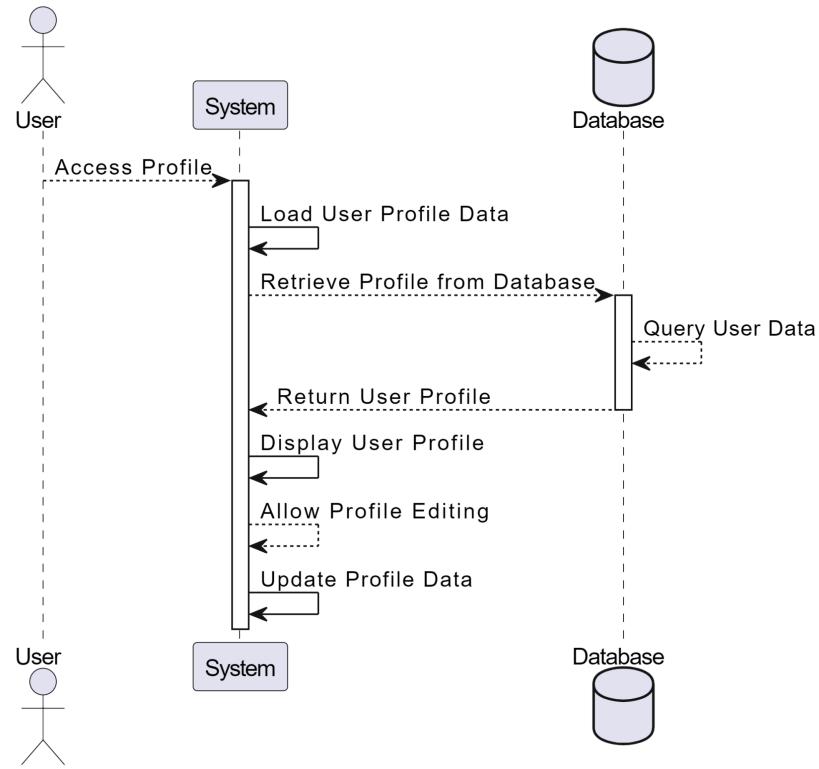


Figure 4: Profile Editing Process

- Actors and Participants:
 - `UserActor` represents the user interacting with the system.
 - `UserProfileSystem` is the system managing user profiles.
 - `UserDatabase` represents the database storing user profile data.
- Process Steps:
 - The user accesses their profile (`UserActor --> UserProfileSystem: Access Profile`).
 - The system loads user profile data from the database (`UserProfileSystem -> UserProfileSystem: Load User Profile Data`).
 - The system retrieves the user profile from the database (`UserProfileSystem --> UserDatabase: Retrieve Profile from Database`).
 - The database queries and returns user data (`UserDatabase --> UserDatabase: Query User Data, UserDatabase --> UserProfileSystem: Return User Profile`).
 - The system displays the user profile (`UserProfileSystem -> UserProfileSystem: Display User Profile`).

- The system allows the user to edit their profile (`UserProfileSystem --> UserProfileSystem: Allow Profile Editing`).
- If the user updates the profile, the system updates the data and saves it to the database (`UserProfileSystem -> UserProfileSystem: Update Profile Data, UserProfileSystem --> UserDatabase: Save Updated Profile to Database`).
- The database updates the user data (`UserDatabase --> UserDatabase: Update User Data`).
- The system notifies the user that the profile has been updated (`UserProfileSystem --> UserProfileSystem: Notify User Profile Updated`).

5. Use case diagram

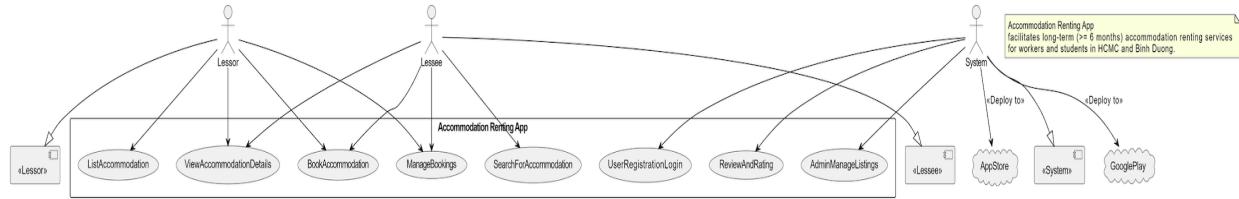


Figure 5: The Use case diagram

- Actors:
 - `LesseeActor` represents individuals looking for accommodation.
 - `LessorActor` represents individuals or entities offering accommodations for rent.
 - `SystemActor` represents the application system.
- Use Cases:
 - Each actor has associated use cases listed in a rectangle representing the "Accommodation Renting App".
 - For example, `LesseeActor` can perform actions like searching for accommodation, viewing details, booking, and managing bookings.
- Relationships:
 - Arrows indicate associations and inheritance relationships between actors and use cases.
 - For example, `LesseeActor` is associated with "SearchForAccommodation" and other use cases.
- Cloud Elements:
 - `GooglePlayStore` and `AppStore` are represented as cloud elements, indicating the deployment platforms.

- Deployment Arrows:
 - Deployment arrows show the deployment of the system to Google Play Store and App Store.
- Note:
 - The note describes the app's purpose, indicating that it facilitates long-term accommodation renting services for workers and students in HCMC and Binh Duong.

6. Sequence diagram

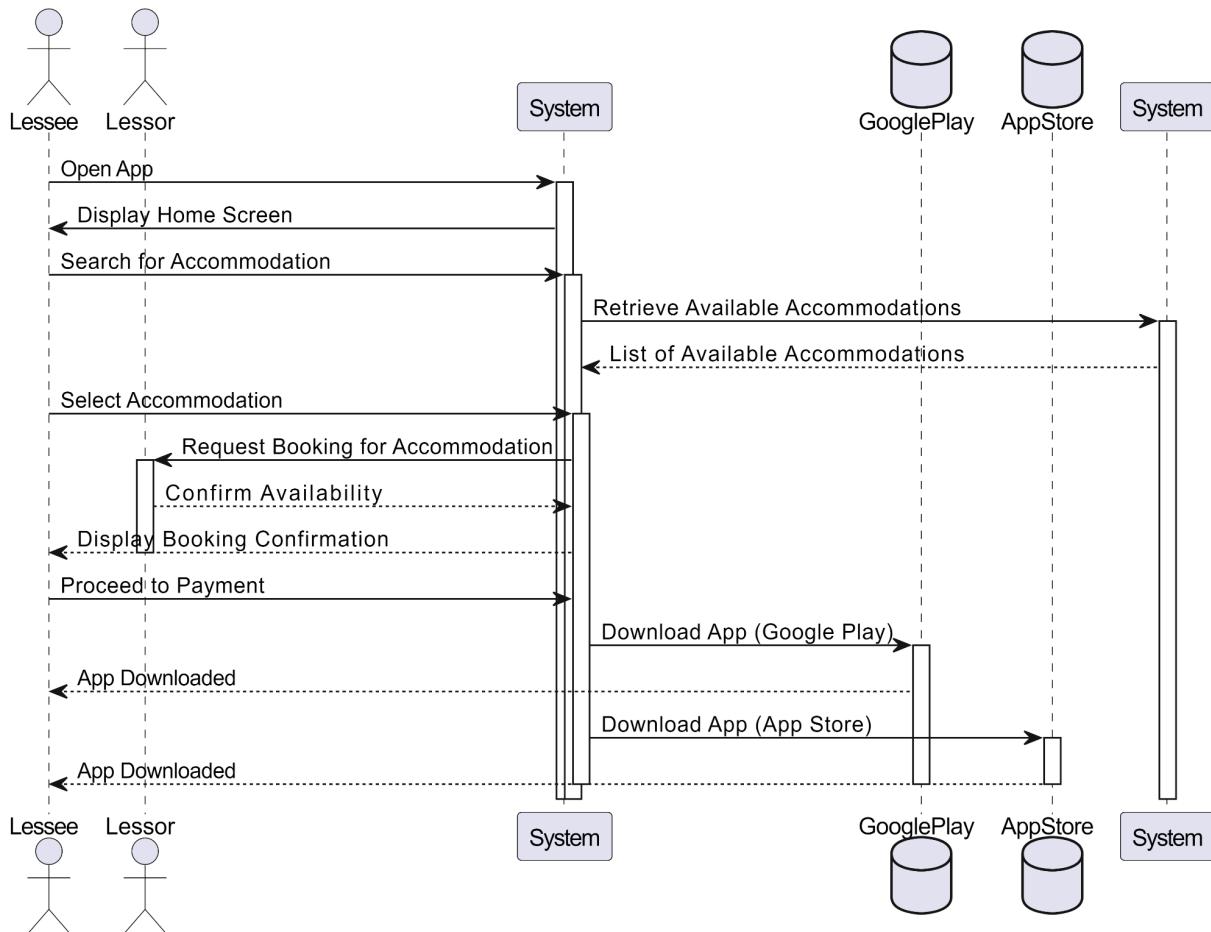


Figure 6: The Sequence diagram

- Actors and Participants:
 - `LesseeActor` and `LessorActor` represent the lessee and lessor, respectively.
 - `AccommodationApp` is the application facilitating accommodation services.
 - `GooglePlayStore` and `AppStore` are databases representing the app download stores.

- The sequence of Interactions:
 - The sequence starts with the lessee opening the app (`LesseeActor -> AccommodationApp: Open App`).
 - The app displays the home screen to the lessee (`AccommodationApp -> LesseeActor: Display Home Screen`).
 - The lessee searches for accommodation, leading to the retrieval of available options from the system (`LesseeActor -> AccommodationApp: Search for Accommodation, AccommodationApp -> System: Retrieve Available Accommodations`).
 - After selecting accommodation, a booking request is sent to the lessor (`AccommodationApp -> LessorActor: Request Booking for Accommodation`).
 - The lessor confirms the availability, and the app displays the booking confirmation to the lessee (`LessorActor --> AccommodationApp: Confirm Availability, AccommodationApp --> LesseeActor: Display Booking Confirmation`).
 - The lessee proceeds to payment (`LesseeActor -> AccommodationApp: Proceed to Payment`).
 - Finally, the app can be downloaded from both Google Play and the App Store (`AccommodationApp -> GooglePlayStore: Download App (Google Play), AccommodationApp -> AppStore: Download App (App Store)`).
- Activation and Deactivation:
 - Activation and deactivation markers (`activate` and `deactivate`) are used to indicate when participants are actively involved in the sequence.
- Aesthetics:
 - Aesthetic improvements include consistent use of colors and shapes for better visual appeal and clarity.

7. User Interface Wireframes

7.1. Home View - House View

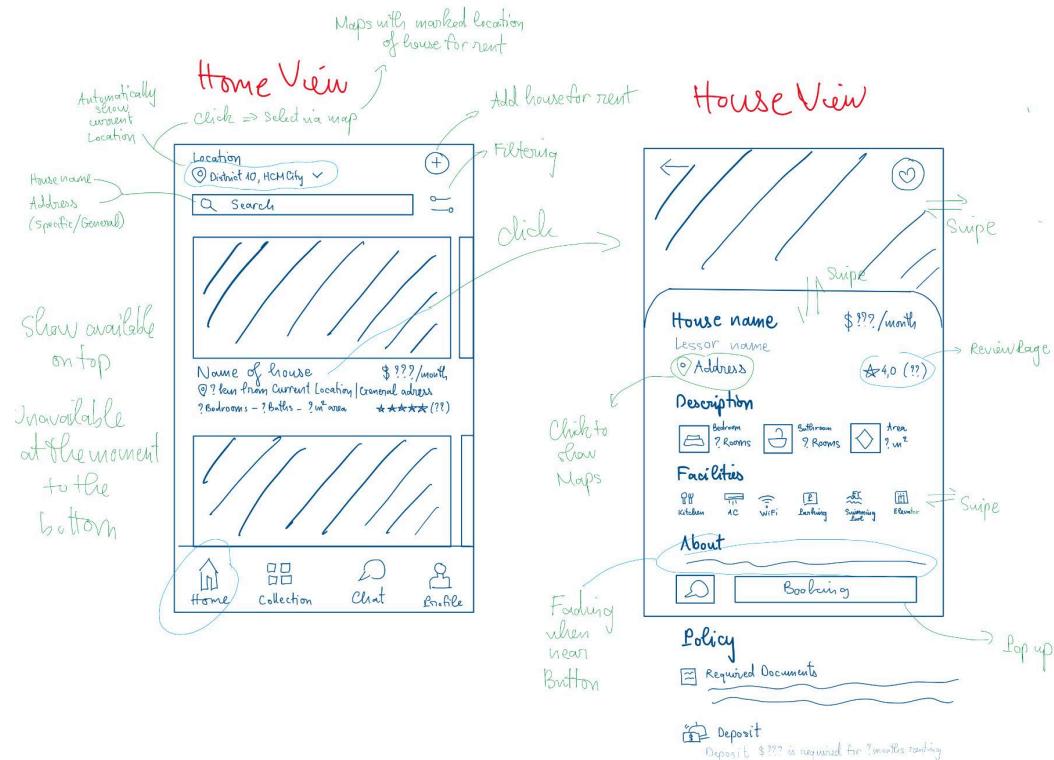


Figure 7: Home View and House View wireframes

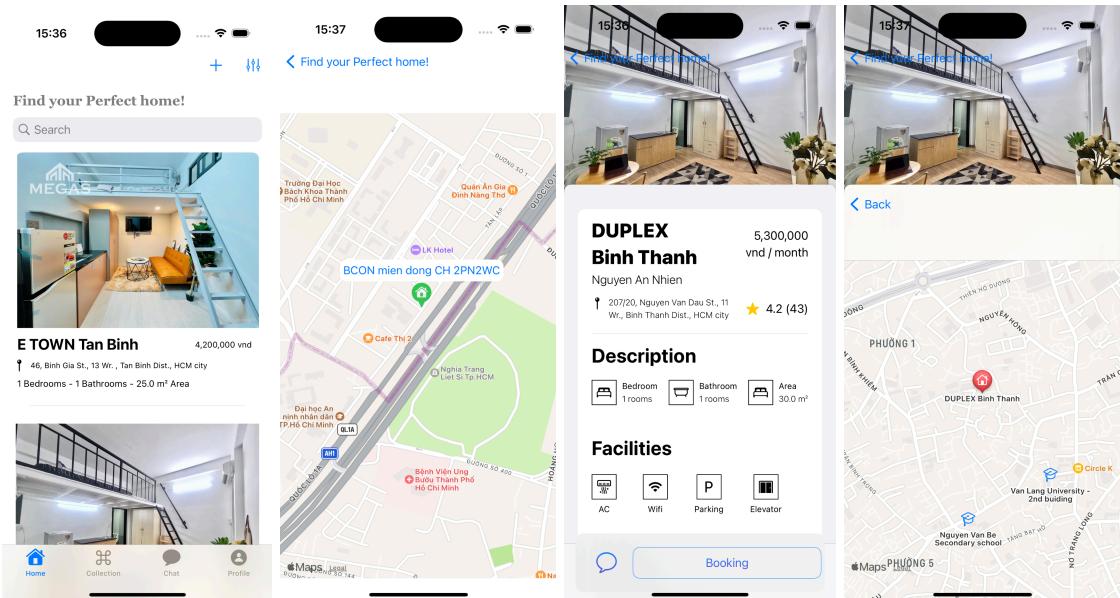


Figure 8: Home View, House View and MapViews UIs in simulation

Home View and House View are the main features of the app which are also code-implemented at the early stages of the development. According to the figures, the wireframes and UI simulations show significant correspondences. The bottom toolbar contains 4 main Views, which are also identified in the UI below. The Home View shows a list of house instances, which can transfer to their details House View when activated by clicking in each block. The search bar and filtering button are included in the UI with changes in distribution for coherent design. Instead of a location block in the wireframe, the activation for showing a map view of the current location will show as a button when the search bar is activated. The House View UI delivers thoroughly the elements and design of its wireframe, although there is a missing button for adding to the Favorite collection. In addition, a map view showing the house's location is implemented as a navigation link in the address text block and shows when activated.

7.2. Collection View

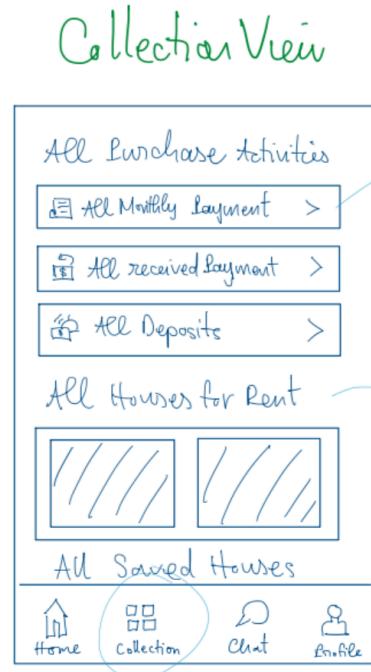


Figure 9: Collection View wireframe

Although the code-implemented Collection View is under development, the wireframe of its design is useful for illustrating the function. Collection View consists of 3 main elements: Purchases activities, a collection of houses for rent as a lessor, and saved houses collection. All Monthly Payment button will show the Purchase View of the user with a list of all current houses rented by them. In contrast, the All Received Payment button will show a different Purchase View of a list of their owned houses that are being rented by other users and their rent income each month. Correspondingly, the

All Houses for Rent section will present a list of houses added for rent by the user. All Saved Houses is a collection of houses being saved by a Favorite button placed at the corresponding House Views.

7.3. Chat View

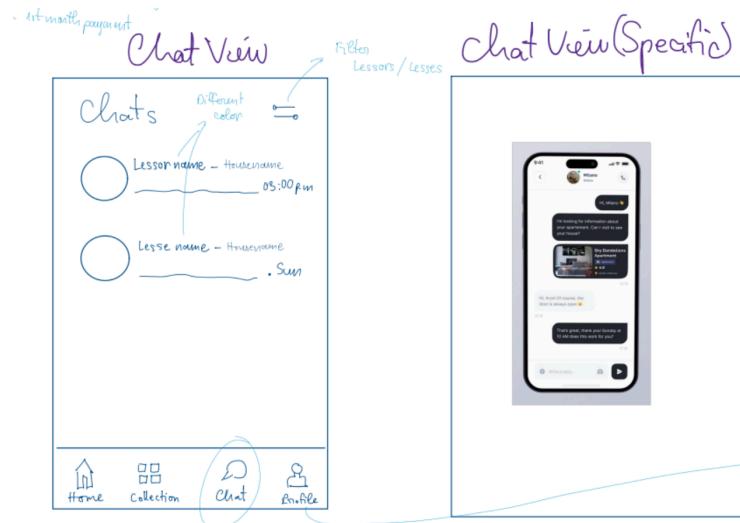


Figure 10: Chat Views wireframes

- **Chat View Description:**

The Chat View serves as the dynamic hub for communication within our accommodation rental app, fostering seamless interactions between lessees and lessors. This interface prioritizes user engagement and responsiveness, ensuring a robust and user-friendly platform for effective communication.

- **Intuitive Messaging Interface:**

At the core of the Chat View is an intuitive messaging interface, designed to facilitate fluid conversations. Users can engage in real-time discussions, share information, and coordinate details effortlessly, enhancing the overall user experience.

- **User Avatars and Names:**

Within the chat, user avatars and names are prominently displayed, providing a visual context for participants in the conversation. This feature adds a personalized touch, making interactions more relatable and user-friendly.

- **Message Timestamps:**

To keep conversations organized and coherent, each message is accompanied by a timestamp. This allows users to track the chronology of the conversation, aiding in contextual understanding and reference.

- **Media Sharing:**

The Chat View supports multimedia sharing, enabling users to exchange images, documents, and other relevant files directly within the chat. This feature enhances the communication experience, especially when conveying detailed information or property-related documents.

- **Typing Indicators:**

To keep participants informed about ongoing interactions, typing indicators are incorporated. Users are notified when the other party is composing a message, fostering a sense of real-time engagement.

- **Read Receipts:**

Read receipts provide transparency by indicating when a message has been viewed by the recipient. This feature is instrumental in confirming the acknowledgment of critical information, ensuring effective communication between lessees and lessors.

- **Notification Integration:**

Seamless integration with the notification system ensures that users are promptly alerted to new messages, promoting timely responses and facilitating efficient communication.

The Chat View is designed to be a central hub for effective communication, encouraging lessees and lessors to interact seamlessly. With its user-friendly features and thoughtful design, the Chat View enhances the collaborative experience within our accommodation rental app in HCMC and Binh Duong.

7.4. Purchase View

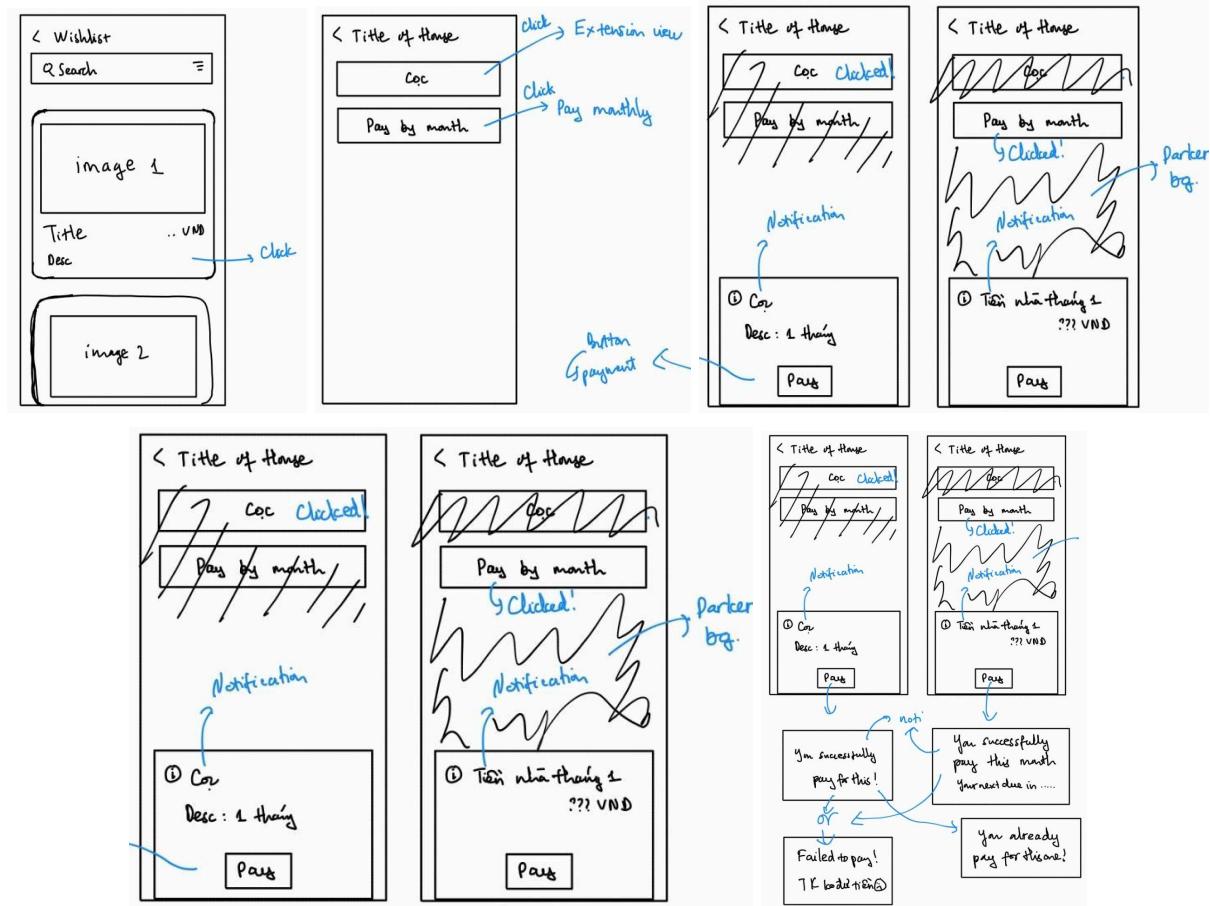


Figure 11: Purchase Views wireframes

7.5. Profile View

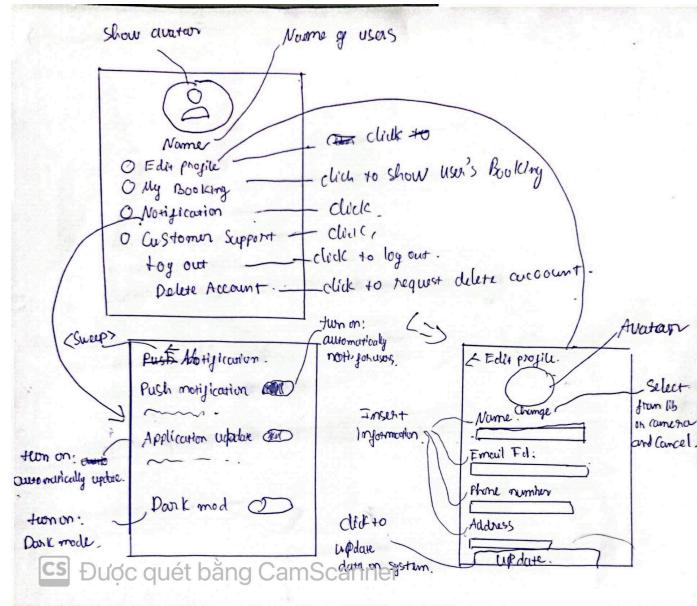


Figure 12: Profile View wireframes

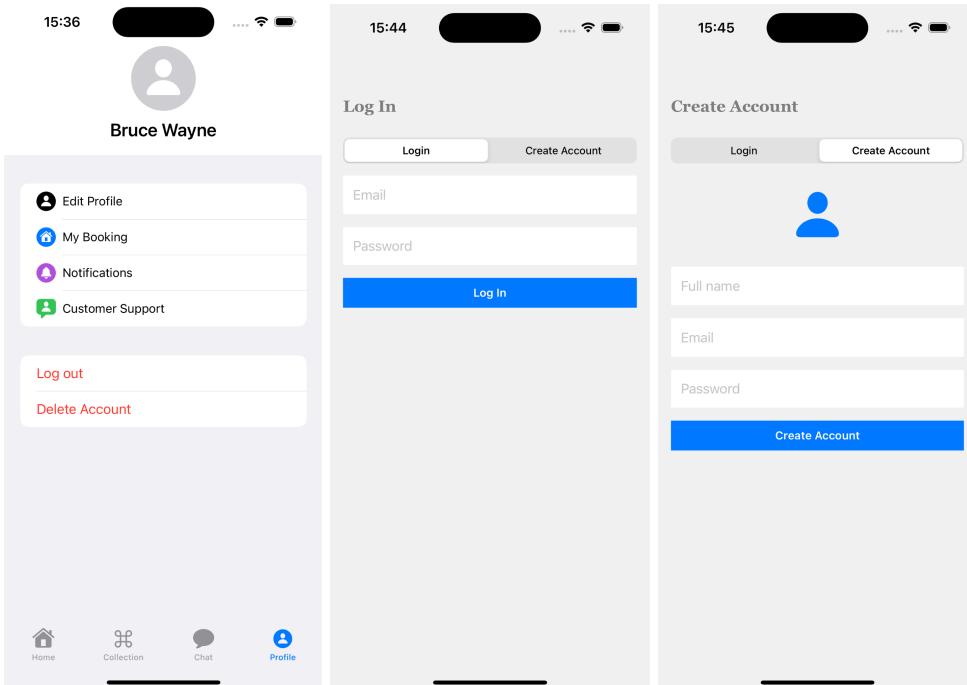


Figure 13: Profile Views and Log In, Sign In UIs in simulation

Within the Profile View interface, user information is elegantly presented with the avatar and name positioned prominently at the top. Six distinct buttons are provided, each serving a specific functionality to enhance user engagement and control. These buttons include "Edit Profile," "My Booking," "Notification," "Customer Support," "Log Out," and "Delete Account."

- **Edit Profile:**

Under the "Edit Profile" section, users are empowered to input and modify personal details such as name, email address, and phone number through designated input boxes. A pivotal "Update" button is seamlessly integrated, allowing users to efficiently save any alterations made, thereby ensuring accurate and up-to-date information in the database.

- **My Booking:**

The "My Booking" feature offers users an insightful overview of their current rental arrangements. Specifically, users can access details regarding the department or accommodation they have leased, providing a centralized platform for managing and tracking their rental history.

- **Notification:**

In the "Notification" section, users are presented with three customizable buttons, each designed to activate specific notification preferences. Beneath each button, a succinct information section is incorporated, offering brief details related to the type of notification it represents. This tailored approach provides users with granular control over their notification settings, enhancing their overall experience within the application.

- **Customer Support:**

For assistance and query resolution, the "Customer Support" button directs users to a dedicated support system, ensuring a seamless avenue for addressing concerns or seeking guidance.

- **Log Out and Delete Account:**

Users are afforded the option to "Log Out" when desired, ensuring a secure and controlled access management process. Additionally, for those contemplating a more permanent exit, the "Delete Account" button is available, allowing users to remove their accounts from the platform.

The Profile View UI, with its strategically placed buttons and user-centric design, aims to provide an academically refined and efficient user experience. By combining intuitive navigation with detailed functionalities, this interface facilitates ease of use, information management, and customization for users engaging with accommodation services in HCMC and Binh Duong.

8. Model diagram

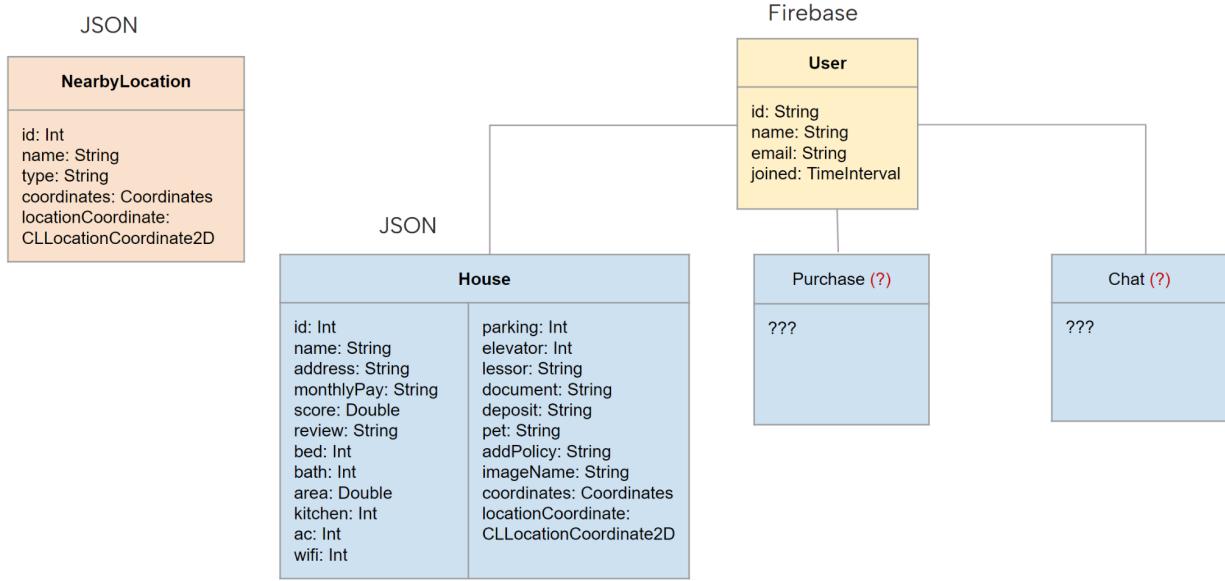


Figure 14: *The Model diagram*

The model diagram illustrates the relationship between data models of the app development. The User model contains users' personal register information, which is connected and stored in Firebase cloud storage. The House model consists of details of the house and its available properties. Its correlation with the User model can be described in the process of renting house action as a lessee and adding a new house for rent as a lessor. A House instance always has an ownership relationship with a User instance that creates it as its lessor. Additionally, they can have a permanent leasehold relationship with another User instance as a lessee. Since the app's function includes purchase and chat methods to make financial transactions and interact between lessor and lessee, further data models of Purchase and Chat will be developed and applied to store bills and messages. Besides, the NearbyLocation model is an independent model for further optimization algorithms of filtering house results. Although House and NearbyLocation instances are currently manually written in JSON for visualization, the upcoming development of the house-adding method will enable the transition of manual JSON written to cloud Firebase for data storage.

9. Conclusion

9.1. Accomplishment

The app's general workflow has been mainly established to visualize the intuitive user experience. Key user interfaces, including HomeView, ProfileView, and LogInRegisterView, have been designed and implemented. Sub-interfaces like HouseView, HouseMapView, and MapView for the searching process contribute to the app's overall usability. Furthermore, the utilization of Firebase for cloud storage has been applied for storing user registration, while local storage of JSON files is currently implied to manage house details and nearby locations.

9.2. Following plans for future development

Main interfaces such as CollectionView and ChatView will be introduced to complete the main View functions. Sub-interfaces like HouseUploadView, PurchaseView, EditProfileView, NotificationSettingView, and CustomerSupportView are in the pipeline, promising a comprehensive user experience. The implication of essential functionalities, including booking, house uploading, search filtering, chatting, and purchasing, are necessary to the app's utility. In addition, the seamless connection of user data with house information, messages, and purchase history is one of the crucial aspects of upcoming developments. Moreover, the transition from manually written JSON to cloud storage via Firebase is expected to improve scalability and data management efficiency.