

**MINISTRY OF**

**EDUCATION AND TRAINING**

**FPT UNIVERSITY**

Capstone Project Document

**Accommodation Sharing**

|  |  |
| --- | --- |
| **GROUP 9** | |
| **Group members** | Nguyễn Quang Nhật - SE62245 (Leader)  Nguyễn Đỗ Minh Đức - SE61767  Vương Minh Thông - SE61739  Hồ Công Trình - SE61998 |
| **Supervisor** | Nguyễn Huy Hùng |
| **Ext. Supervisor** | N/A |
| **Capstone Project code** | ASP |

- Ho Chi Minh City, **September 11th*, 2018*** –

*This page is intentionally left blank*

# Table of Content

[Table of Content 3](#_Toc532081702)

[List of Table 5](#_Toc532081703)

[List of Figure 5](#_Toc532081704)

[Definitions, Acronyms, and Abbreviations 6](#_Toc532081705)

[A. Introduction 7](#_Toc532081706)

[1. Project Information 7](#_Toc532081707)

[2. Introduction 7](#_Toc532081708)

[3. Current Situation 7](#_Toc532081709)

[4. Problem Definition 7](#_Toc532081710)

[5. Proposed Solution 7](#_Toc532081711)

[5.1 Feature functions 8](#_Toc532081712)

[5.2 Advantages and Disadvantages 8](#_Toc532081713)

[6. Functional Requirements 8](#_Toc532081714)

[7. Role and Responsibility 9](#_Toc532081715)

[B. Software Project Management Plan 9](#_Toc532081716)

[1. Problem Definition 9](#_Toc532081717)

[1.1 Name of this Capstone Project 9](#_Toc532081718)

[1.2 Problem Abstract 9](#_Toc532081719)

[1.3 Project Overview 10](#_Toc532081720)

[1.3.1 Current Situation 10](#_Toc532081721)

[1.3.2 The Proposed System 10](#_Toc532081722)

[1.3.2.1 Mobile Application 10](#_Toc532081723)

[1.3.2.2 Web Application 11](#_Toc532081724)

[1.3.3 Boundaries of the System 11](#_Toc532081725)

[1.3.4 Future Plan 11](#_Toc532081726)

[2. Project Organization 11](#_Toc532081727)

[2.1 Software Process Model 11](#_Toc532081728)

[3. Project Management Plan 12](#_Toc532081729)

[3.1 Product Backlog 12](#_Toc532081730)

[3.2 Sprint Backlog 14](#_Toc532081731)

[3.3 Meeting Minutes 14](#_Toc532081732)

[C. Software Requirement Specification 14](#_Toc532081733)

[1. User Requirement Specification 14](#_Toc532081734)

[1.1 Unauthorized User Requirement 14](#_Toc532081735)

[1.2 Authorized User Requirement 14](#_Toc532081736)

[1.3 House Owner Requirement 14](#_Toc532081737)

[1.4 Room Master Requirement 14](#_Toc532081738)

[1.5 Member Requirement 15](#_Toc532081739)

[1.6 Admin Requirement 15](#_Toc532081740)

[2. System Requirement Specification 15](#_Toc532081741)

[2.1 External Interface Requirement 15](#_Toc532081742)

[2.1.1 User Interface 15](#_Toc532081743)

[2.1.2 Hardware Interface. 15](#_Toc532081744)

[2.1.3 Software Interface 15](#_Toc532081745)

[2.1.4 Communication Protocol 15](#_Toc532081746)

[2.2 System Overview Use Case 16](#_Toc532081747)

[2.3 List of Use Case 17](#_Toc532081748)

[2.3.1 <Authorized User> Overview Use 17](#_Toc532081749)

[2.3.1.1 <Authorized User>View Notification 18](#_Toc532081750)

[2.3.2 <House Owner> Overview Use Case 19](#_Toc532081751)

[2.3.2.1 <House Owner> Create Room 20](#_Toc532081752)

[2.3.2.2 <House owner> Update Room 21](#_Toc532081753)

[2.3.2.3 <House owner> Add Member into Room 23](#_Toc532081754)

[2.3.2.4 <House owner> Delete Member from Room 25](#_Toc532081755)

[2.3.3 <Room Master> Overview Use Case 27](#_Toc532081756)

[2.3.3.1 <Room Master> View Suggested Nearby Room Master 28](#_Toc532081757)

[2.3.3.2 <Room Master> Create Finding Roommate Post 29](#_Toc532081758)

[2.3.3.3 <Room Master> Update Finding Roommate Post 32](#_Toc532081759)

[2.3.4 <Member> Overview Use Case 34](#_Toc532081760)

[2.3.4.1 <Member> View Suggested Finding Roommate Post list 35](#_Toc532081761)

[2.3.4.2 <Member> Create Finding Room Post 37](#_Toc532081762)

[2.3.5 <Admin> Overview Use Case 39](#_Toc532081763)

[2.3.5.1 <Admin> Approve Room 39](#_Toc532081764)

[2.3.5.2 <Admin> View Users 41](#_Toc532081765)

[3. Conceptual diagram 42](#_Toc532081766)

[D. Software Design Description 43](#_Toc532081767)

[1. System Architecture Design 43](#_Toc532081768)

[1.1 System Architecture Design 43](#_Toc532081769)

[2. Component Diagram 43](#_Toc532081770)

[3. Detail Description 44](#_Toc532081771)

[3.1.1 Class Diagram 44](#_Toc532081772)

[3.2 Interactive diagram 46](#_Toc532081773)

[3.2.1 Sequence Diagram 46](#_Toc532081774)

[3.2.1.1 Create Room 46](#_Toc532081775)

[3.2.1.2 Update Room 46](#_Toc532081776)

[3.2.1.3 Delete Room 46](#_Toc532081777)

[3.2.1.4 View Single Room 47](#_Toc532081778)

[3.2.1.5 Add Member into Room 47](#_Toc532081779)

[3.2.1.6 Delete Member of Room 48](#_Toc532081780)

[3.2.1.7 Create Finding Roommate Post 48](#_Toc532081781)

[3.2.1.8 Delete Finding Roommate Post 48](#_Toc532081782)

[4.3.1.11 Update Finding Roommate Post 49](#_Toc532081801)

[4.3.1.12 View Suggest List Overview 49](#_Toc532081821)

[4.3.1.13 <Suggest> Member/Room Master has reference and has no posts. 50](#_Toc532081822)

[4.3.1.14 <Suggest> Room Master has posts 50](#_Toc532081823)

[4.3.1.15 <Suggest>Member/Room Master has no reference but allow access current location 51](#_Toc532081824)

[4.3.1.16 <Suggest> Member/Room Master has no reference and not allow access current location 51](#_Toc532081825)

[4.3.1.17 Create Finding Room Post 52](#_Toc532081826)

[4.3.1.18 Update Finding Room Post 52](#_Toc532081827)

[4.3.1.19 Delete Finding Room Post 52](#_Toc532081828)

[4.3.1.20 Filter 53](#_Toc532081829)

[4. Database Design 53](#_Toc532081830)

[4.1 Entity relationship diagram (ERD) 53](#_Toc532081831)

[4.2 Entity dictionary 54](#_Toc532081832)

[5. Algorithms 54](#_Toc532081833)

[5.1 Definition 54](#_Toc532081834)

[5.2 Define Problem 54](#_Toc532081835)

[5.3 Solution 54](#_Toc532081836)

[E. System Implementation & Test 55](#_Toc532081837)

[1. Database Relationship Diagram 55](#_Toc532081838)

[1.1 Physical Diagram 55](#_Toc532081839)

[1.2 Data dictionary 56](#_Toc532081840)

[G. Appendix 57](#_Toc532081841)

[1. A nautical mile 57](#_Toc532081842)

[2. Spherical law of cosines 57](#_Toc532081843)

# List of Table

[Table 1: Roles and Responsibilities 9](#_Toc532081844)

[Table 6: Product Backlog 14](#_Toc532081845)

[Table 7: <Authorized User> View Notification specification 19](#_Toc532081846)

[Table 8: <House Owner> Create Room specification 21](#_Toc532081847)

[Table 9: <House owner> Update Room specification 23](#_Toc532081848)

[Table 10: <House owner> Add Member into Room specification 25](#_Toc532081849)

[Table 11: <House owner> Delete Member from Room specification 27](#_Toc532081850)

[Table 12: <Room Master> View Suggested Nearby Room Master Post Specification 29](#_Toc532081851)

[Table 13: <Room Master> Create Finding Roommate Post specification 32](#_Toc532081852)

[Table 14: <Room Master> Update Finding Roommate Post specification 34](#_Toc532081853)

[Table 15: <Member> View Suggested Finding Roommate Posts List Specification 37](#_Toc532081854)

[Table 16: <Member> Create Finding Room Post specification 39](#_Toc532081855)

[Table 17: <Admin> Approve room specification 41](#_Toc532081856)

[Table 18: <Admin> View User specification 42](#_Toc532081857)

[Table 19: Conceptual Entity dictionary 43](#_Toc532081858)

[Table 20: Component Diagram Explanation 44](#_Toc532081859)

[Table 21: Class Diagram Explanation 46](#_Toc532081860)

[Table 22: Entity Data dictionary 54](#_Toc532081861)

[Table 23: Data dictionary 57](#_Toc532081862)

# List of Figure

[Figure 1: Scrum Methodology 12](#_Toc532081863)

[Figure 2: System Overview Use Case (1) 16](#_Toc532081864)

[Figure 3: System Overview Use Case (2) 17](#_Toc532081865)

[Figure 4: <Authorized User> Overview Use Case 18](#_Toc532081866)

[Figure 5: <Authorized User> View Notification Use Case 18](#_Toc532081867)

[Figure 6: <House Owner> Overview Use Case 19](#_Toc532081868)

[Figure 7: <House Owner> Create Room Use Case 20](#_Toc532081869)

[Figure 8: <House owner> Update Room Use Case 22](#_Toc532081870)

[Figure 9: <House owner> Add Member into Room Use Case 24](#_Toc532081871)

[Figure 10: <House owner> Delete Member from Room Use Case 26](#_Toc532081872)

[Figure 11: <Room Master> Overview Use Case 28](#_Toc532081873)

[Figure 12: <Room Master> View Suggested Nearby Room Master Post Use Case 28](#_Toc532081874)

[Figure 13: <Room Master> Create Finding Roommate Post Use Case 30](#_Toc532081875)

[Figure 14: <Room Master> Update Finding Roommate Post Use Case 32](#_Toc532081876)

[Figure 15: <Member> Overview Use Case 35](#_Toc532081877)

[Figure 16: <Member> View Suggested Finding Roommate Posts List Use Case 36](#_Toc532081878)

[Figure 17:< Member> Create Finding Room Post Use Case 37](#_Toc532081879)

[Figure 18: <Admin> Overview Use Case 39](#_Toc532081880)

[Figure 19: <Admin> Approve Room Use Case 40](#_Toc532081881)

[Figure 20: <Admin> View User Use Case 41](#_Toc532081882)

[Figure 21: Conceptual Diagram 42](#_Toc532081883)

[Figure 22: System Architecture Design (MVC) 43](#_Toc532081884)

[Figure 23: Component Diagram 44](#_Toc532081885)

[Figure 24: Class Diagram 45](#_Toc532081886)

[Figure 25: Sequence Diagram - <House owner> Create Room 46](#_Toc532081887)

[Figure 26: Sequence Diagram - <House owner> Update Room 46](#_Toc532081888)

[Figure 27: Sequence Diagram - <House owner> Delete Room 47](#_Toc532081889)

[Figure 28: Sequence Diagram - <House owner> View Single Room 47](#_Toc532081890)

[Figure 29: Sequence Diagram - <House owner> Add Member into Room 48](#_Toc532081891)

[Figure 30: Sequence Diagram - <House owner> Delete Members of Room 48](#_Toc532081892)

[Figure 31: Sequence Diagram - <Room Master> Create Finding Roommate Post 48](#_Toc532081893)

[Figure 32: Sequence Diagram - <Room Master> Delete Finding Roommate Post 49](#_Toc532081894)

[Figure 33: Sequence Diagram - <Room Master> Update Finding Roommate Post 49](#_Toc532081895)

[Figure 34: Sequence Diagram - <Member/Room Master> View Suggest List 50](#_Toc532081896)

[Figure 35: Sequence Diagram - <Member/Room Master> Has reference and has no posts. 50](#_Toc532081897)

[Figure 36: Sequence Diagram - <Room Master> Room Master has posts 51](#_Toc532081898)

[Figure 37: Sequence Diagram - <Member/Room Master> Member/Room Master has no reference but allow access current location. 51](#_Toc532081899)

[Figure 38: Sequence Diagram - <Member/Room Master> Member/Room Master has no reference and not allow access current location. 52](#_Toc532081900)

[Figure 39: Sequence Diagram - <Member> Create Finding Room Post 52](#_Toc532081901)

[Figure 40: Sequence Diagram - <Member> Update Finding Room Post 52](#_Toc532081902)

[Figure 41: Sequence Diagram - <Member> Delete Finding Room Post 53](#_Toc532081903)

[Figure 42: Sequence Diagram - <Authorized User> Filter 53](#_Toc532081904)

[Figure 43: Entity Relationship diagram 53](#_Toc532081905)

[Figure 44: Physical diagram 56](#_Toc532081906)

# Definitions, Acronyms, and Abbreviations

|  |  |
| --- | --- |
| Name | Definition |
| ASP | Accommodation Sharing |

# A. Introduction

## Project Information

* Project Name: **Accommodation Sharing Platform**
* Project Code: **ASP**
* Project Type: **Mobile Application**
* Start Date: **September 11th, 2018**
* End Date: **N/A**

## Introduction

Nowadays, finding an accommodation is really a difficult and exhausting process, especially for province people who are working in the city recently. Searching for an accommodation information is so difficult, choosing a good and safe accommodation at reasonable cost is more difficult, not to mention scams, and other bad situations. When searching for an accommodation around the internet, the informations are not usually accurately. As a matter of fact, the Accommodation Sharing Platform with a simple interface, easy to use, users do not have to spend much time to post or search for an accommodation.

## Current Situation

Currently there are many mobile applications that help people to find an accommodation. However, they only show places that are match with users. When users who have already rented an accommodation, looking for roommates, they have to search for all the users who requested for room/house to stay, but not the ones who are also in the same situation, so that they can discuss to move in together.

## Problem Definition

We have research on some current mobile applications and websites in Vietnam such as: Ohana, phongtro123.com, thuephongtro.com… Some of them have some limits:

* Not support suggesting for Room Master (people who are looking for roommates) in the same area move in together.
* Not support finding nearby Room Master.
* Not support suggesting best match for those who looking for a room/house to stay.

## Proposed Solution

Our proposed solution is to build a system named “ASP”. User can add information about their requests in system. The system processes the requests and matches the Room Master with the Member appropriately. Besides that, 2 Room Master can be suggested to move in if they are in the same area.

ASP system includes a mobile application for Room Master and Member and a web application for administrator and householder.

### Feature functions

* Mobile application
* User can register a new account to find an accommodation or post for shared room.
* User can manage their own information.
* House owner can create room and manage members in room.
* Room Master can create and manage their own post.
* System matching Room Master with utilities and suggest appropriate room for member.
* Member can view information of room master’s post.
* Room Master and member can negotiate with feature to send text or call.
* House owner can rate for member in their room.
* Member can rate their current room.
* Web application (for admin)
* Manage users.
* Manage rooms.

### Advantages and Disadvantages

* Advantages:
* Members can easily find the place they want.
* Accommodation will be verified by staff.
* House-owner can manage their room.
* Room master can get suggested nearby other room master to move in if they are in the same area.
* Member can get suggested nearby their current location.
* Disadvantages:
* Trust issues is top priority problem.

## Functional Requirements

Function requirement of the system are listed at below:

* Mobile application (for unauthorized user)
* Login form.
* Register.
* Mobile application (for authorized user)
* Negotiate.
* View and manage information.
* Get notification.
* Logout.
* Mobile application (for House Owner)
* Manage rooms.
* Manage members of room.
* Mobile application (for Room Master)
* View suggested nearby other room master’s posts.
* Manage post.
* View finding room post.
* View finding roommate post.
* Manage references.
* Mobile application (for Member)
* View suggest finding roommate posts base one current reference.
* View suggest finding roommate posts base one current location.
* Manage information.
* Manage references.
* Web application (for admin)
* View and manage user.
* View and manage rooms.

## Role and Responsibility

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Full name** | **Role** | **Position** | **Contact** |
| 1 | Nguyễn Huy Hùng | Project Manager | Supervisor | Hungnh@fpt.edu.vn |
| 2 | Nguyễn Quang Nhật | Scrum Master | Leader | [Nhatnqse62245@fpt.edu.vn](mailto:Nhatnqse62245@fpt.edu.vn) |
| 3 | Nguyễn Đỗ Minh Đức | Developer | Member | Ducndmse61767@fpt.edu.vn |
| 4 | Vương Minh Thông | Developer | Member | Thongvmse61739@fpt.edu.vn |
| 5 | Hồ Công Trình | Developer | Member | Trinhhcse61998@fpt.edu.vn |

Table 1: Roles and Responsibilities

# B. Software Project Management Plan

## Problem Definition

### Name of this Capstone Project

* Official name: Accommodation Sharing
* Vietnamese name: Ứng dụng hỗ trợ tìm người ở ghép
* Abbreviation: ASP

### Problem Abstract

First, Vietnam is currently in the integration and development process, so developed countries are always focus on this potential market. Foreign companies invest a lot in Vietnam and it always have number of employee working onsite.

Second, young generation used to move on to the big city to study and work. Besides that, traveling is one of the most popular in their mind right now, so that we providing the best accommodation for living, short term travelling and studying.

From above, we have thought about an idea of an application that can help those who travel, live and study in Vietnam have the opportunity to easily find suitable accommodation.

### Project Overview

#### Current Situation

Below are some problems that we encountered in this project:

* **Business logic:** Some scopes of this project are not clear in early stage of development. For example, room master and member share the same interface because users can be different roles in different situations.
* **Scheduler of team members:** Team members can have conflicts in meeting schedule because of sick, study and work schedule, etc.
* **Framework study:** Team members have a problem when applying the play framework into project. The team needs an amount of time to get familiar new techniques.
* **Lack of UI and UX design skill:** Our team members all study IS major, and no one has studied UI, UX design. Therefore, that some UI may misunderstand or hard to use with normal user.

#### The Proposed System

The system contains a web application for admin and a mobile application for room master, member, house owner.

##### Mobile Application

**- House-owner:**

* House-owner can create rooms and system can upload their room for room master create a post if they want.
* House-owner can add room master and other members to their room.

**- Room Master:**

* Room Master can create new post for member to know which one they should contact.
* Room Master can manage their own information, post they have made and negotiate with member.

**- Member:**

* Member can create utilities and system can match their utilities with location room master to make a suggestion.
* Member can manage their own information, their utilities.

##### Web Application

**- Admin:**

* Admin manages user informations, rooms.

#### Boundaries of the System

This section supposes that users of the system have stable internet connection while using.

Language of application is Vietnamese.

* **The system can:**
* Allow room master search for partner.
* Allow member search for rom master.
* Matching post and utilities then suggest to member.
* Matching 2 accommodation if in the same area.
* Send notification to users.
* **The system cannot:**
* The application can not run in another platform except IOS.

#### Future Plan

The current system only support for IOS, we recommend some features for future plan:

* Mobile application will be available on Android OS.

## Project Organization

### Software Process Model

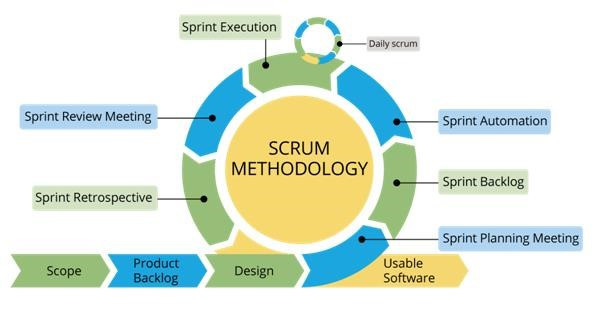


Figure 1: Scrum Methodology

For more information: https://www.belatrixsf.com/

This project is developed under Scrum model. We apply Scrum model to capable with current situation of our team. We choose this model due to following reasons:

* **Greater flexibility:** Though traditional methodologies require the customer to provide detail idea requirements, scrum software developers are more flexible because of their iterative style of work.
* **Quick responding to changing needs:** Since scrum software development goes through multiple rollout cycles, it has a better chance of providing exactly what the customers need at any given point in time. These cycles are designed not only to add new features but also to make adjustments to what has already been deployed.
* **Continuous feedback and easy to track:** It allow continuous feedbacks in bill to shape the final design.

## Project Management Plan

### Product Backlog

|  |  |  |
| --- | --- | --- |
| **ID** | **Features** | **User Story** |
| 1 | Register | Registering to become member |
| 2 | Login/Logout | Accessing the account to use their features |
| 3 | View all rooms | Admin want to view all rooms |
| 4 | Approve room | Admin want to remove room which changed house owner or not used |
| 5 | Remove room | Admin want to remove a room |
| 6 | View all users | Admin want to view all users |
| 7 | Approve house owner | Admin want to promote member to house owner |
| 8 | Remove user | Admin want to remove an user |
| 9 | Create room | House Owner want to add new room |
| 10 | Update account information | House Owner want to update informations |
| 11 | Update room information | House Owner want to update their room's informations |
| 12 | Remove room | House Owner want to delete room |
| 13 | Assign room master | House Owner want to assign room master for room |
| 14 | Add room's member | House Owner want to add room's members |
| 15 | View all room's information | House Owner want to view all room's informations |
| 16 | Remove room's member | House Owner want to remove member of room |
| 17 | Update account information | Room master want to update account information |
| 18 | Create finding roommate post | Room master want to create finding roommate post |
| 19 | Update finding roommate post | Room master want to update finding roommate post |
| 20 | Remove finding roommate post | Room master want to remove finding roommate post |
| 21 | Get notification about verified post | Room master want to get notified about verified post |
| 22 | Get notification about declined post | Room master want to get notified about declined post |
| 23 | Suggested member | Room master want to get suggested member |
| 24 | Suggested nearby room master | Room master want to get suggested nearby room master |
| 25 | Search for roommate | Room master want to search for roommate |
| 26 | Suggested room master | Member want to get suggested nearby rooms |
| 27 | Suggested by utilities | Member want to get suggested rooms by utilities |
| 28 | Search for room master | Member want to search rooms by address or utilities |
| 29 | Bookmark room | Member want to add bookmark |
| 30 | View all available room | Member want to view all available rooms |
| 31 | Filter room by utilities | Member want to filter room by utilities |
| 32 | Update account information | Member want to update account information |

Table 2: Product Backlog

### Sprint Backlog

Refer to “Sprint Backlog” folder.

### Meeting Minutes

Refer to “Meeting Minutes” folder.

# C. Software Requirement Specification

## User Requirement Specification

### Unauthorized User Requirement

Unauthorized user is a person who does not access to the system. Unauthorized user can use some functions in the system, to use all functions, they must login. These are functions that they can use:

* Register
* Login

### Authorized User Requirement

Authorized user is a person who logins in the system and will be separated into House owner, Room master, Member depend on what they want to do in the application. Authorized user can do the following functions, such as:

* Manage profile
* Search
* Negotiate
* View notification
* Logout

### House Owner Requirement

House owner is an authorized user who want to upload their room so that room master can create a post about that room in order to find roommates.

House owner can do the following functions:

* Create room
* Update room
* View room
* Delete room
* Add room members
* Remove room members

### Room Master Requirement

Room master is an authorized user who is looking for roommate.

Room master can do the following functions:

* Create finding roommate post
* Update finding roommate post
* View finding roommate post
* Delete finding roommate post
* View suggested nearby room master’s posts.

### Member Requirement

Member is an authorized user who is looking for a room

Member can do the following functions:

* Create finding room post
* Update finding room post
* View finding room post
* Delete finding room post
* View posts list
* View suggested posts list

### Admin Requirement

Admin is a completely different role form the others. Admin can manage information and tracking if the system work efficiently or not. Admin can do following functions.

* Manage room
* Manage user
* Approve room
* Logout

## System Requirement Specification

### External Interface Requirement

#### User Interface

* The user interface use Vietnamese.
* Use the consistent palette of colors between the text and the background.
* General requirement for graphical user interface of application on the mobile device is that the GUI should be simple, reminiscent.

#### Hardware Interface.

* Smartphone.

#### Software Interface

* Mobile application: IOS operating system (IOS 10 or above).

#### Communication Protocol

* Using Http protocol.

### System Overview Use Case

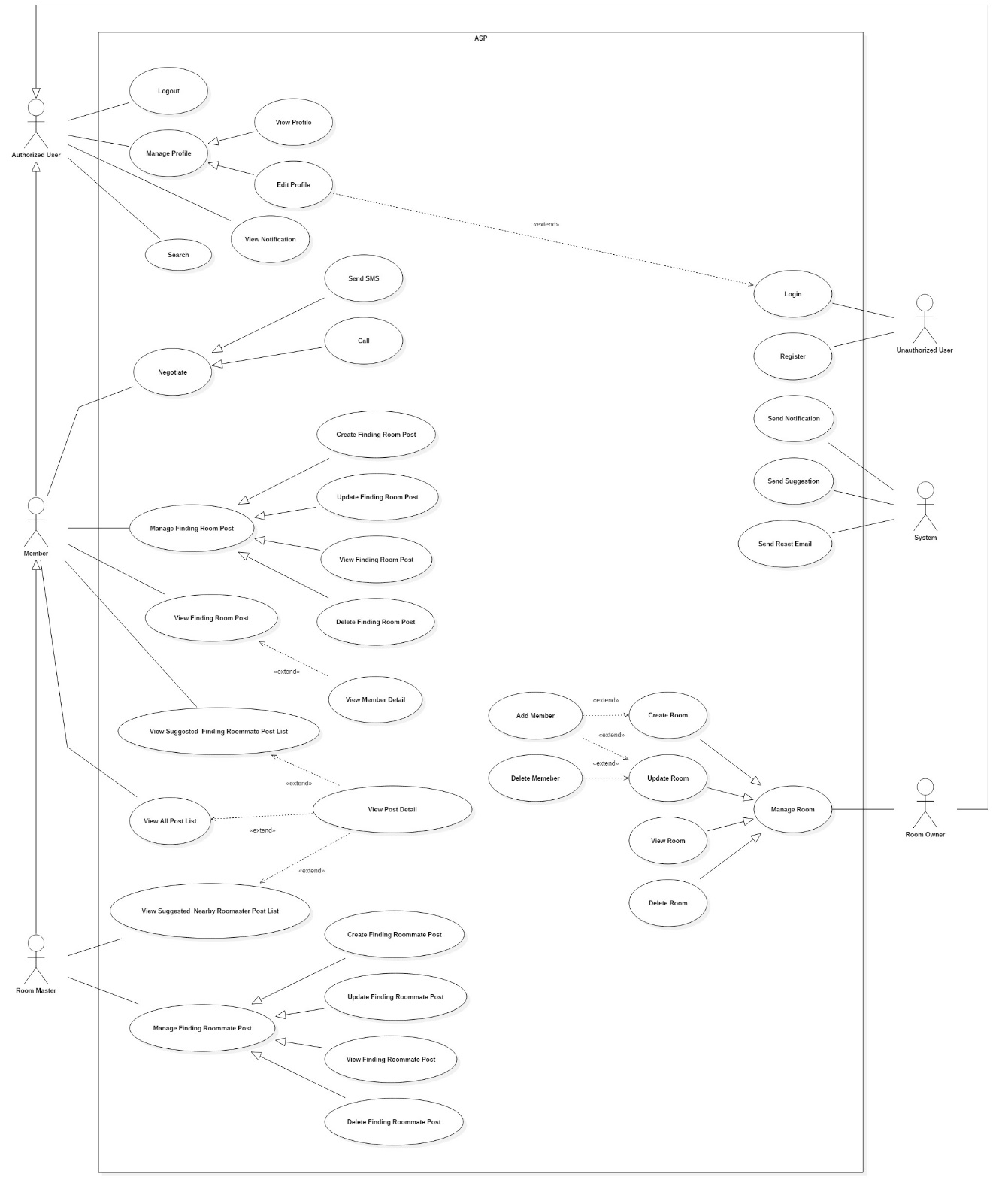


Figure 2: System Overview Use Case (1)



Figure 3: System Overview Use Case (2)

### List of Use Case

#### <Authorized User> Overview Use

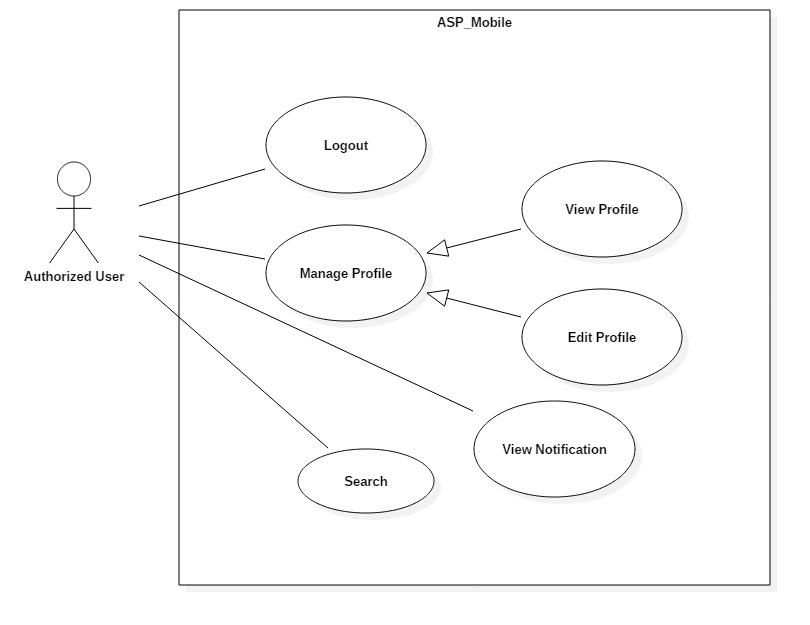


Figure 4: <Authorized User> Overview Use Case

##### <Authorized User>View Notification

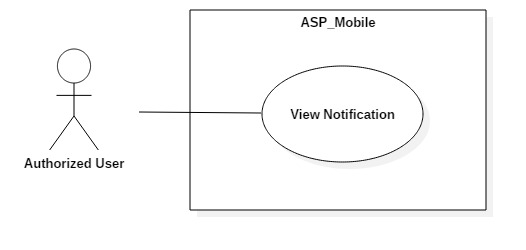


Figure 5: <Authorized User> View Notification Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP\_4** | | | |
| **Use Case No.** | UC\_ ASP\_4 | **Use Case Version** | 0.2 |
| **Use Case Name** | View Notification | | |
| **Author** | Hồ Công Trình | | |
| **Date** | 17/10/2018 | **Priority** | Medium |
| **Actor:**   * Authorized User   **Summary:**   * This use case allows user to view notification.   **Goal:**   * Get notification.   **Triggers:**   * From home screen, press on notification button in navigator bar.   **Preconditions:**   * User logged in the system.   **Post conditions:**   * Success: User get list of notifications. * Fail: System displays error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Actor presses on notification button. | System navigates to notification screen with list of notifications. |   **Alternative Scenario:**N/A  **Exceptions:** N/A  **Relationships:** UC\_ ASP\_24, UC\_ ASP\_9, UC\_ ASP\_10  **Business Rules:**   * User has already accessed to the system. * User can be any role to get notified. * If there is no notification, display “Hiện tại bạn chưa có thông báo mới”. * If house owner’s room approved by admin, display “Phòng của bạn đã được xác thực.” * If members are added into room, display “Bạn đã được thêm vào phòng.” * If members are removed out of room, display “Bạn đã bị xóa ra khỏi phòng.” | | | |

Table 3: <Authorized User> View Notification specification

#### <House Owner> Overview Use Case

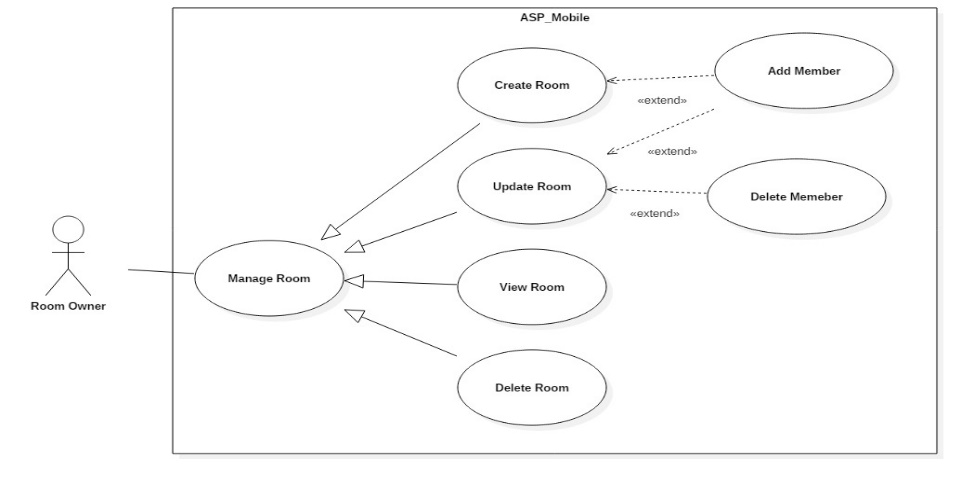


Figure 6: <House Owner> Overview Use Case

##### <House Owner> Create Room

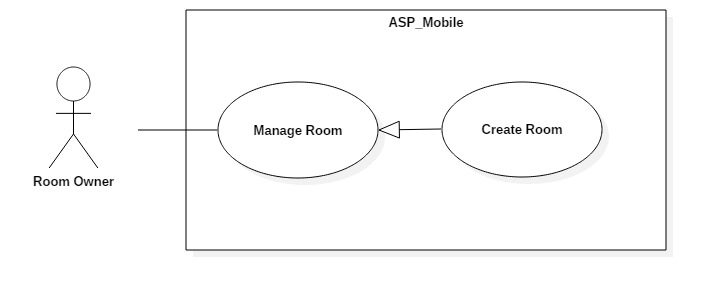


Figure 7: <House Owner> Create Room Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP\_6** | | | |
| **Use Case No.** | UC\_ ASP\_6 | **Use Case Version** | 0.2 |
| **Use Case Name** | Create Room | | |
| **Author** | Vương Minh Thông | | |
| **Date** | 17/10/2018 | **Priority** | High |
| **Actor:**   * House Owner   **Summary:**   * This use case allows house owner to create new room.   **Goal:**   * To create new room.   **Triggers:**   * Press create room button in home screen.   **Preconditions:**   * User is house owner.   **Post conditions:**   * Success: User can create new room. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Actor presses on create room button. | System displays create room screen with:  - “Tên Phòng”: Text input.  - “Thành phố”: Dropdown list.  - “Quận”: Dropdown list.  - “Giá”: Text input.  - “Diện tích”: Text input.  - “Địa chỉ”: Text input.  - “Số khách tối đa”: Text input.  - “Mô tả”: Text input.  - “Tiện ích”: Checkmark.  - “Hình ảnh”: Image upload.  - “Thêm thành viên vào phòng” : Button  - “Tạo phòng” : Button | | 2 | Actor fills form and presses on save button. | [Exception no.1]  System creates room successfully and navigate user back to home screen.  System will display latest information. |   **Alternative Scenario:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | 1 | User does not input required fields. | System notices that user need to input all these field:  - “Tên phòng”: System displays warning message: “Tên phòng không được để trống”.  - “Giá”: System displays warning message: “Giá không được để trống”.  - “Địa chỉ”: System displays warning message: “Địa chỉ không được để trống”. |   **Relationships:** N/A  **Business Rules:**   * User already logged in the system. * Room name must not be empty. * Room price must not be empty. * Address must not be empty. * User must be house owner role to create room. * After house owner create room, redirect to home screen. | | | |

Table 4: <House Owner> Create Room specification

##### <House owner> Update Room

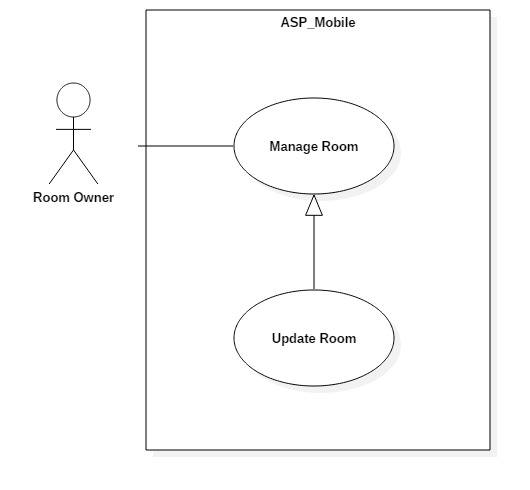


Figure 8: <House owner> Update Room Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP\_7** | | | |
| **Use Case No.** | UC\_ ASP\_7 | **Use Case Version** | 0.2 |
| **Use Case Name** | Update Room | | |
| **Author** | Vương Minh Thông | | |
| **Date** | 17/10/2018 | **Priority** | High |
| **Actor:**   * House owner   **Summary:**   * This use case allows House owner update room.   **Goal:**   * To update room.   **Triggers:**   * Actor press on update room button.   **Preconditions:**   * User logged in the system. * Room have been uploaded.   **Post conditions:**   * Success: User can update new information to room. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Actor presses on update room button. | Systems navigate to update room screen.  System displays update screen with:  - “Tên Phòng”: Text input.  - “Thành phố”: Dropdown list.  - “Quận”: Dropdown list.  - “Giá”: Text input.  - “Diện tích”: Text input.  - “Địa chỉ”: Text input.  - “Số khách tối đa”: Text input.  - “Mô tả”: Text input.  - “Tiện ích”: Checkmark.  - “Hình ảnh”: Image upload.  - “Thành viên trong phòng” table  - “Thêm thành viên vào phòng” button  - “Lưu” button | | 2 | Actor fills form presses on save button. | [Exception no.1]  Room info is updated successfully and system refreshes the current screen with new information. |   **Alternative Scenario:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | 1 | User does not input required fields. | System notices that user need to re-input all these fields:  - “Tên phòng”: System displays warning message: “Tên phòng không được để trống”.  - “Giá”: System displays warning message: “Giá không được để trống”.  - “Địa chỉ”: System displays warning message: “Địa chỉ không được để trống”. |   **Relationships:** N/A  **Business Rules:**   * User already logged in the system. * User must have role as House Owner to update room. * Room’s information is updated must be matched the defined room’s information structure. * Room name must not be empty. * Room price must not be empty. * Room address must not be empty. | | | |

Table 5: <House owner> Update Room specification

##### <House owner> Add Member into Room

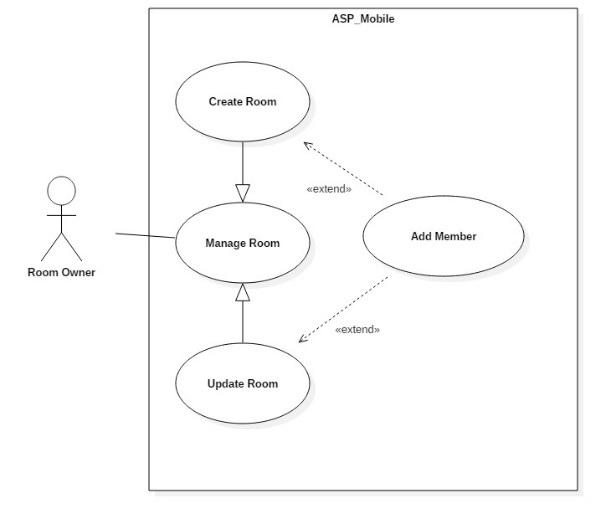


Figure 9: <House owner> Add Member into Room Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP\_9** | | | |
| **Use Case No.** | UC\_ ASP\_9 | **Use Case Version** | 0.2 |
| **Use Case Name** | Add Member into Room | | |
| **Author** | Hồ Công Trình | | |
| **Date** | 17/10/2018 | **Priority** | High |
| **Actor:**   * House owner   **Summary:**   * This use case allows house owner add member into Room.   **Goal:**   * To add member into room.   **Triggers:**   * Actor press on certain room in account screen. * Actor press add member button.   **Preconditions:**   * User logged in the system. * Room have been uploaded.   **Post conditions:**   * Success: House owner can add member into Room. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Actor presses on certain room. | System will navigate to this room screen with common detail.  - “Hình ảnh”  - “Tên phòng”  - “Giá phòng”  - “Diện tích”  - “Địa chỉ”  - “Ngày đăng bài”  - “Mô tả”  - “Chỉnh sửa” button  - “Xóa” button | | 2 | Actor presses on edit member button. | Systems navigate to add member into room screen.  System displays add member into room screen with:  - “Tài khoản”: Text input  - “Thêm” button. | | 3 | Actor presses on add button. | [Exception no.1]  [Exception no.2]  Room info is updated successfully and system refreshes the “Thêm thành viên vào phòng” screen with new information. |   **Alternative Scenario:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | 1 | User does not input required fields. | System notices that user need to re-input all these fields:  - “Tài khoản”: System displays warning message: “Tài khoản không được để trống”. | | 2 | User inputs wrong information in some fields. | System notices that user need to re-input all these fields:  - “Tài khoản”: System displays warning message: “Tài khoản không tồn tại hoặc bị trùng”. |   **Relationships:** Extend Create/Update room  **Business Rules:**   * User already logged in the system. * User must be house owner role. * Username must have been in system. * Date in must be later than date create room. * House owner enter username of member then press add button. * House owner must choose one room master before submit. | | | |

Table 6: <House owner> Add Member into Room specification

##### <House owner> Delete Member from Room

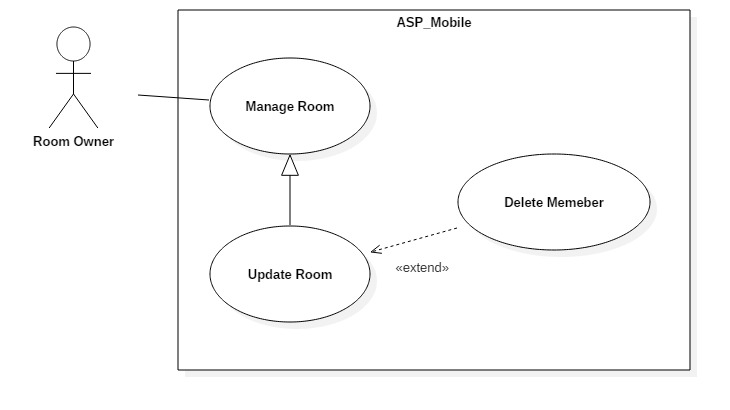


Figure 10: <House owner> Delete Member from Room Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP\_10** | | | |
| **Use Case No.** | UC\_ ASP\_10 | **Use Case Version** | 0.2 |
| **Use Case Name** | Delete Member from Room | | |
| **Author** | Hồ Công Trình | | |
| **Date** | 17/10/2018 | **Priority** | High |
| **Actor:**   * House owner   **Summary:**   * This use case allows house owner delete member from room.   **Goal:**   * To delete member from room.   **Triggers:**   * Actor presses on certain room in account screen. * Actor presses delete member button.   **Preconditions:**   * User logged in the system. * Member already added into room.   **Post conditions:**   * Success: House owner can delete member of room. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Actor presses on certain room. | System will navigate to this room screen with common detail.  - “Hình ảnh”  - “Tên phòng”  - “Giá”  - “Diện tích”  - “Địa chỉ”  - “Ngày đăng bài”  - “Mô tả”  - “Chỉnh sửa” button  - “Xóa” button | | 2 | Actor chooses 1 member of room and press delete button | Systems display popup “Xoá thành công”.  System displays the rest of members of the room. | | 3 | Actor presses on save button. | Room info is updated successfully and system refreshes the update room screen with new information. |   **Alternative Scenario:** N/A  **Relationships:** Extend Update room  **Business Rules:**   * User already logged in the system. * Member already added into room. * User must be house owner role. * If Room Master was removed, system will promote the next member to room master automatically. | | | |

Table 7: <House owner> Delete Member from Room specification

#### <Room Master> Overview Use Case

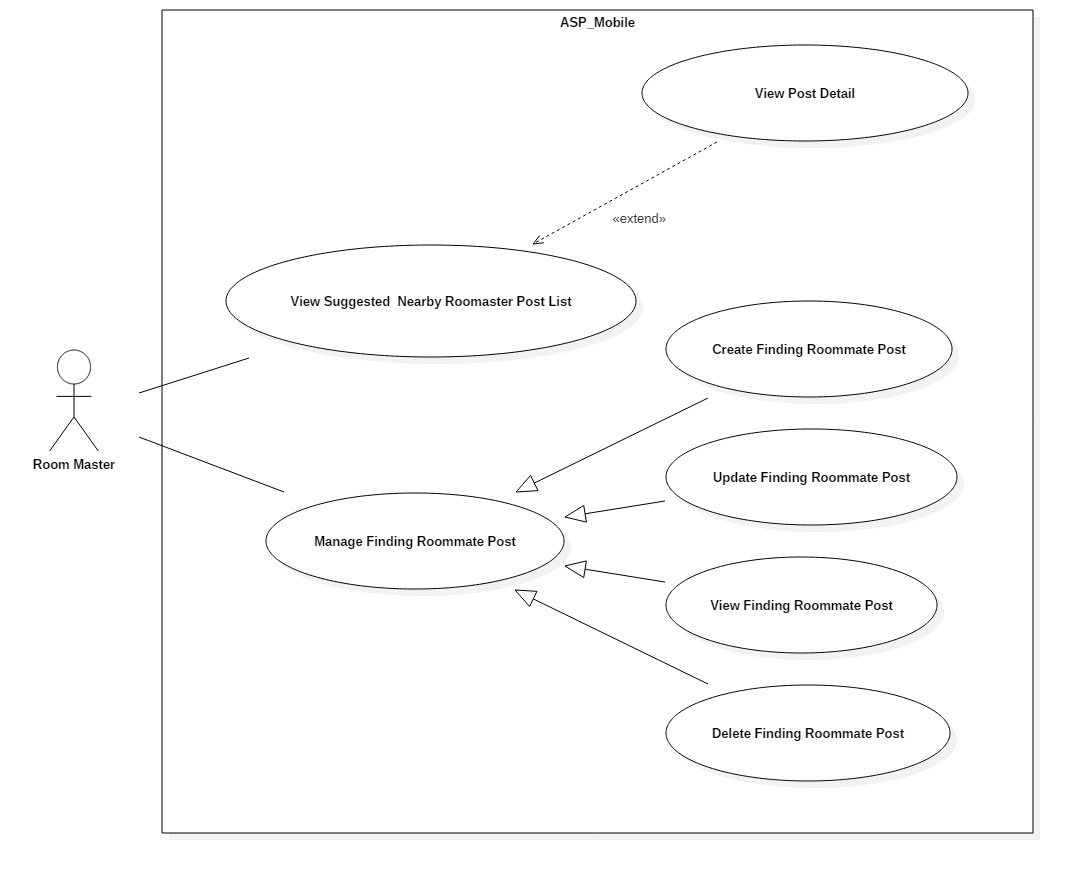


Figure 11: <Room Master> Overview Use Case

##### <Room Master> View Suggested Nearby Room Master

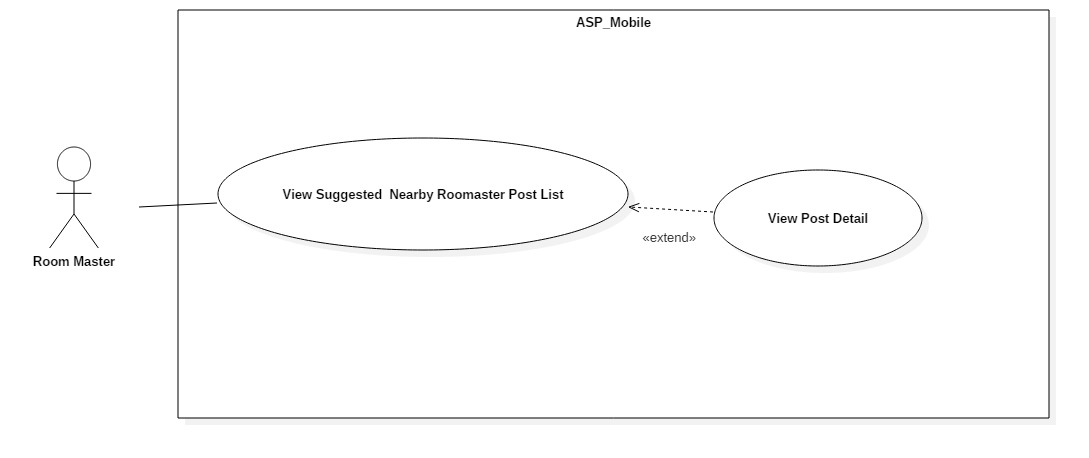


Figure 12: <Room Master> View Suggested Nearby Room Master Post Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP\_11** | | | |
| **Use Case No.** | UC\_ ASP\_11 | **Use Case Version** | 0.2 |
| **Use Case Name** | View Suggested Nearby Room master Post | | |
| **Author** | Nguyễn Đỗ Minh Đức | | |
| **Date** | 17/10/2018 | **Priority** | High |
| **Actor:**   * Room Master   **Summary:**   * This use case allows room master to view suggested nearby room master posts.   **Goal:**   * To get suggested nearby room master posts.   **Triggers:**   * Actor logged into system and goes to home screen.   **Preconditions:**   * User logged in the system.   **Post conditions:**   * Success: User can view suggested finding nearby room master posts. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Actor access application. | System displays home screen with:  - “Những bài gần bạn”: Horizontal scroll view. |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:** N/A  **Business Rules:**   * User already logged in the system. * System get all posts which are in the same city as room master’s current finding roommate post. * System calculate distance between them based on their longitude and latitude which stored in database. * System then shows nearby finding roommate posts. | | | |

Table 8: <Room Master> View Suggested Nearby Room Master Post Specification

##### <Room Master> Create Finding Roommate Post

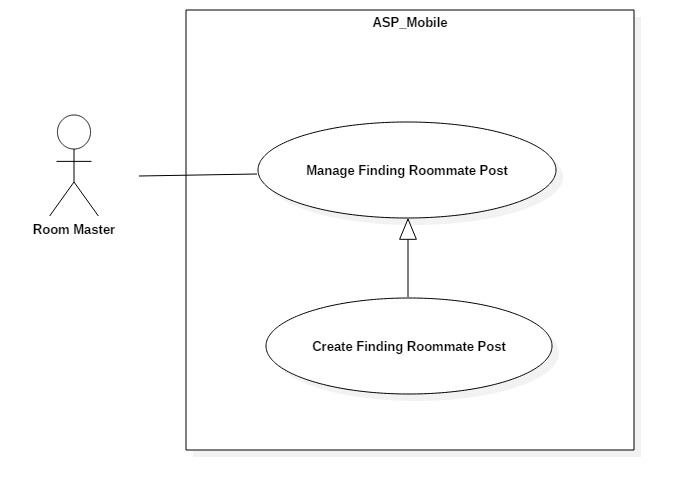


Figure 13: <Room Master> Create Finding Roommate Post Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP\_12** | | | |
| **Use Case No.** | UC\_ ASP\_12 | **Use Case Version** | 0.2 |
| **Use Case Name** | Create Finding Roommate Post | | |
| **Author** | Nguyễn Quang Nhật | | |
| **Date** | 17/10/2018 | **Priority** | High |
| **Actor:**   * Room Master   **Summary:**   * This use case allows room master to create new finding roommate post.   **Goal:**   * To create new post.   **Triggers:**   * Actor press on create post button in home screen.   **Preconditions:**   * User logged in the system. * The first member was added into room by house owner.   **Post conditions:**   * Success: User can create new post. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Actor press on create post button. | System displays create post screen with:  - “Tên bài”: Text input.  - “Giá”: Text input.  - “Số điện thoại”: Text input  - “Diện tích”  - “Địa chỉ”  - “Số bạn ở ghép”: Text input  - “Mô tả”: Text input.  - “Giới tính”: Radio button  - “Đăng bài”: Button | | 3 | Fill form and press on submit button. | [Exception no.1]  [Exception no.2]  System creates post successfully and navigate user back to home screen.  System displays latest information. |   **Alternative Scenario:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | 1 | User does not input required fields. | System notices that user need to input all these field:  - “Giá”: System displays warning message: “Giá không được để trống”.  - “Số bạn ở ghép”: System displays warning message: ““Số bạn ở ghép không được để trống”. | | 2 | User input wrong information in some fields. | System notices that user need to re-input all these field:  - “Giá”: System displays warning message: “Không được sử dụng chữ hoặc ký tự đặc biệt !@#$%^&\*()-=\_+...”.  - “Số bạn ở ghép”: System displays warning message: “Không được sử dụng chữ hoặc ký tự đặc biệt !@#$%^&\*()-=\_+... Và phải ít hơn số thành viên tối đa của phòng”. |   **Relationships:** N/A  **Business Rules:**   * User already logged in the system. * Number of partner must not be empty. * Price must not be empty. * User must be room master. * User must be added into room. | | | |

Table 9: <Room Master> Create Finding Roommate Post specification

##### <Room Master> Update Finding Roommate Post

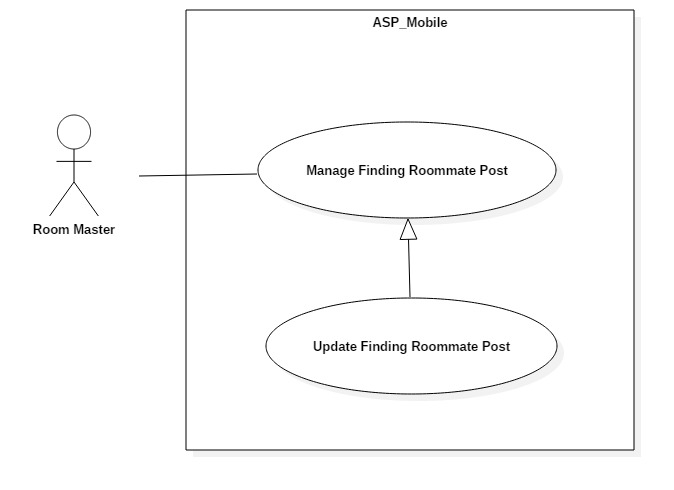


Figure 14: <Room Master> Update Finding Roommate Post Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP\_13** | | | |
| **Use Case No.** | UC\_ ASP\_13 | **Use Case Version** | 0.2 |
| **Use Case Name** | Update Finding Roommate Post | | |
| **Author** | Nguyễn Quang Nhật | | |
| **Date** | 17/10/2018 | **Priority** | High |
| **Actor:**   * Room Master   **Summary:**   * This use case allows room master to update finding roommate post.   **Goal:**   * To update post.   **Triggers:**   * Actor press on edit post button.   **Preconditions:**   * User logged in the system. * Post have been uploaded.   **Post conditions:**   * Success: User can update post. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Actor presses on edit post button. | Systems navigate to update room screen.  System displays update screen with:  - “Hình ảnh”  - “Tên phòng” ”: Text input  - “Số điện thoại”: Text input  - “Giá phòng” ”: Text input  - “Diện tích”  - “Địa chỉ”  - “Ngày đăng bài”  - “Mô tả” ”: Text input  - “Lưu” button | | 2 | Actor fills form and presses on save button. | [Exception no.1]  [Exception no.2]  Room info is updated successfully and system refreshes the “Đã đăng” screen with new information. |   **Alternative Scenario:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | 1 | User does not input required fields. | System notices that user need to input all these field:  - “Giá”: System displays warning message: “Giá không được để trống”.  - “Số bạn ở ghép”: System displays warning message: ““Số bạn ở ghép không được để trống”. | | 2 | User input wrong information in some fields. | System notices that user need to re-input all these field:  - “Giá”: System displays warning message: “Không được sử dụng chữ hoặc ký tự đặc biệt !@#$%^&\*()-=\_+...”.  - “Số bạn ở ghép”: System displays warning message: “Không được sử dụng chữ hoặc ký tự đặc biệt !@#$%^&\*()-=\_+... Và phải ít hơn số thành viên phòng tối đa”. |   **Relationships:** N/A  **Business Rules:**   * User already logged in the system. * Post number of partner is not null. * Post price is not null. * User must be room master role. * Information are updated must not be in wrong format. | | | |

Table 10: <Room Master> Update Finding Roommate Post specification

#### <Member> Overview Use Case



Figure 15: <Member> Overview Use Case

##### <Member> View Suggested Finding Roommate Post list

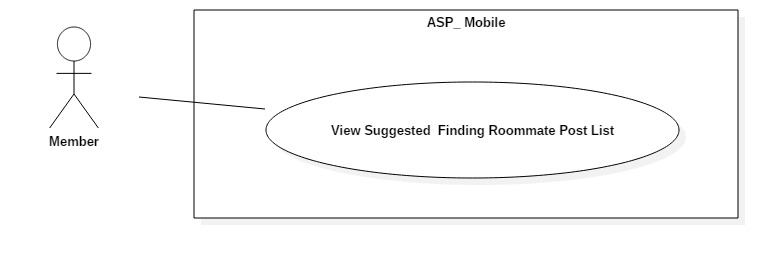


Figure 16: <Member> View Suggested Finding Roommate Posts List Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP\_16** | | | |
| **Use Case No.** | UC\_ ASP\_16 | **Use Case Version** | 0.2 |
| **Use Case Name** | View Suggested Finding Roommate Posts | | |
| **Author** | Nguyễn Quang Nhật | | |
| **Date** | 17/10/2018 | **Priority** | High |
| **Actor:**   * Member   **Summary:**   * This use case allows member to view suggested finding roommate posts.   **Goal:**   * To get suggested post list.   **Triggers:**   * User logged into system and goes to home screen.   **Preconditions:**   * User logged in the system.   **Post conditions:**   * Success: User can view suggested finding roommate posts. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Actor access application. | System displays home screen with:  - “Những bài gần bạn”: Horizontal scroll view. |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:** N/A  **Business Rules:**   * User already logged in the system. * User must accept location access request from the application. * At the time user access the application, system automatically calculate the distance based on the current location. * If user does not accept location access request from the application, display all newest accommodations in home screen. * If user has references suggest posts by references, otherwise suggest newest posts. | | | |

Table 11: <Member> View Suggested Finding Roommate Posts List Specification

##### <Member> Create Finding Room Post

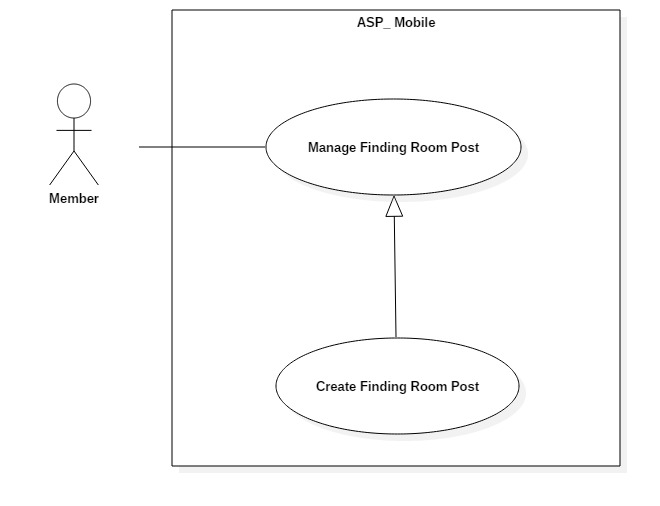


Figure 17:< Member> Create Finding Room Post Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP\_17** | | | |
| **Use Case No.** | UC\_ ASP\_17 | **Use Case Version** | 0.2 |
| **Use Case Name** | Create Finding Room Post | | |
| **Author** | Nguyễn Quang Nhật | | |
| **Date** | 17/10/2018 | **Priority** | High |
| **Actor:**   * Member   **Summary:**   * This use case allows member to create finding room post.   **Goal:**   * To create new finding room post.   **Triggers:**   * Actor sends create finding room post command.   **Preconditions:**   * User logged in the system.   **Post conditions:**   * Success: User can create new post. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Actor presses on create post button. | System displays create post screen with:  - “Tên bài”: Text input.  - “Khoảng Giá”: Slide choose.  - “Diện tích”  - “Quận”: Checkmark  - “Mô tả”: Text input.  - “Số điện thoại”: Text input  - “Giới tính” switch button  - “Đăng bài” button | | 2 | Actor fills form and presses on submit button. | [Exception no.1]  [Exception no.2]  System creates post successfully and navigate user back to home screen.  System will display latest information. |   **Alternative Scenario:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | 1 | User input wrong information in some fields. | System notices that user need to re-input all these field:  - “Giá”: System displays warning message: “Không được sử dụng chữ hoặc ký tự đặc biệt !@#$%^&\*()-=\_+...”.  - “Diện tích”: System displays warning message: “Không được sử dụng chữ hoặc ký tự đặc biệt !@#$%^&\*()-=\_+...”. | | 2 | User does not input fields with requirement. | System notices that user need to re-input all these field:  - “Quận”: System displays warning message: “Bạn phải chọn ít nhất 1 quận.” .  - “Số điện thoại”: System displays warning message: “Số điện thoại không được để trống”. |   **Relationships:** N/A  **Business Rules:**   * User already logged in the system. * User must be member role. * District must not be empty. * Phone number must not be empty. * After member create finding rom post. System get user’s utilities and apply in account setting utilities. | | | |

Table 12: <Member> Create Finding Room Post specification

#### <Admin> Overview Use Case

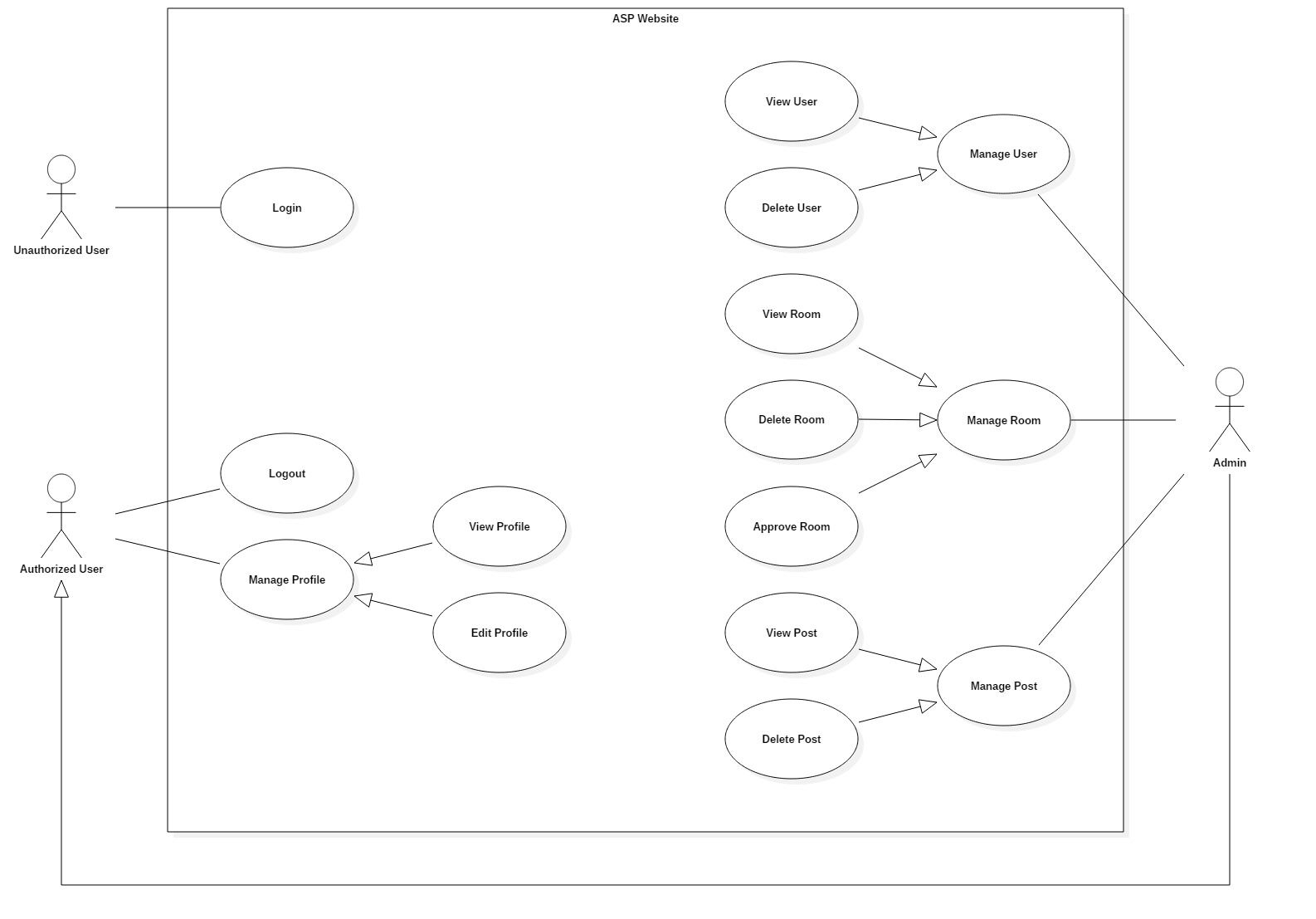


Figure 18: <Admin> Overview Use Case

##### <Admin> Approve Room

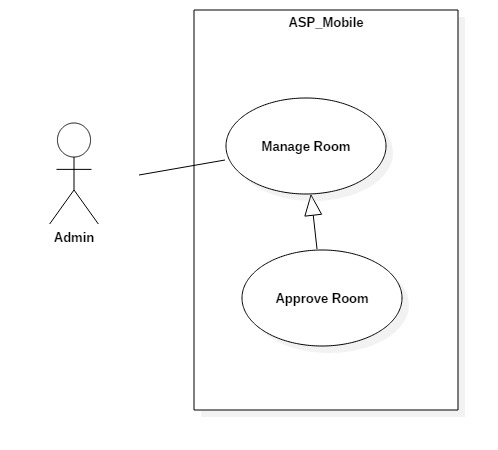


Figure 19: <Admin> Approve Room Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP\_23** | | | |
| **Use Case No.** | UC\_ ASP\_23 | **Use Case Version** | 0.2 |
| **Use Case Name** | Approve room | | |
| **Author** | Vương Minh Thông | | |
| **Date** | 17/10/2018 | **Priority** | High |
| **Actor:**   * Admin   **Summary:**   * This use case allows admin to approve room.   **Goal:**   * To approve room.   **Triggers:**   * Actor click “Approve” button.   **Preconditions:**   * User logged in the system.   **Post conditions:**   * Success: User can approve room. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Actor clicks “Approve” button. | System displays “Phòng” page with last informations.  - List of rooms.  - Manage buttons in each row of table room list. |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:** N/A  **Business Rules:**   * Admin already logged in the system. * House owner must upload their house’s license in their account setting. Depending on that, admin decide whether to confirm or not. * The approved rooms will appear on the mobile application. | | | |

Table 13: <Admin> Approve room specification

##### <Admin> View Users

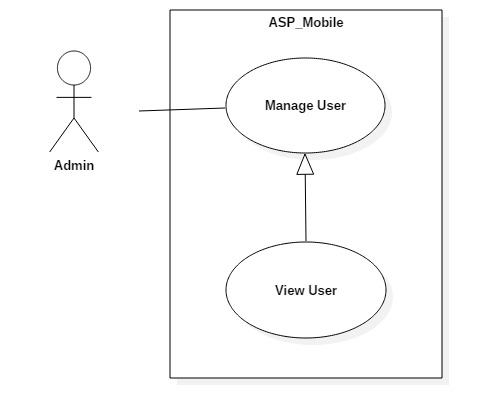


Figure 20: <Admin> View User Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP\_25** | | | |
| **Use Case No.** | UC\_ ASP\_25 | **Use Case Version** | 0.2 |
| **Use Case Name** | View User | | |
| **Author** | Nguyễn Quang Nhật | | |
| **Date** | 17/10/2018 | **Priority** | High |
| **Actor:**   * Admin   **Summary:**   * This use case allows admin to view user list.   **Goal:**   * To view user list.   **Triggers:**   * Actor click on users tab in web admin.   **Preconditions:**   * User logged in the system. * User must be admin.   **Post conditions:**   * Success: Admin can see all user information. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Actor clicks “Người dùng” tab. | System navigates to “Người dùng” page.  System displays “Người dùng” page with:  - List of users.  - “Quản lý” button.  - “Refresh” button. |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:** N/A  **Business Rules:**   * Admin already logged in the system. * Users exist in database. | | | |

Table 14: <Admin> View User specification

## Conceptual diagram

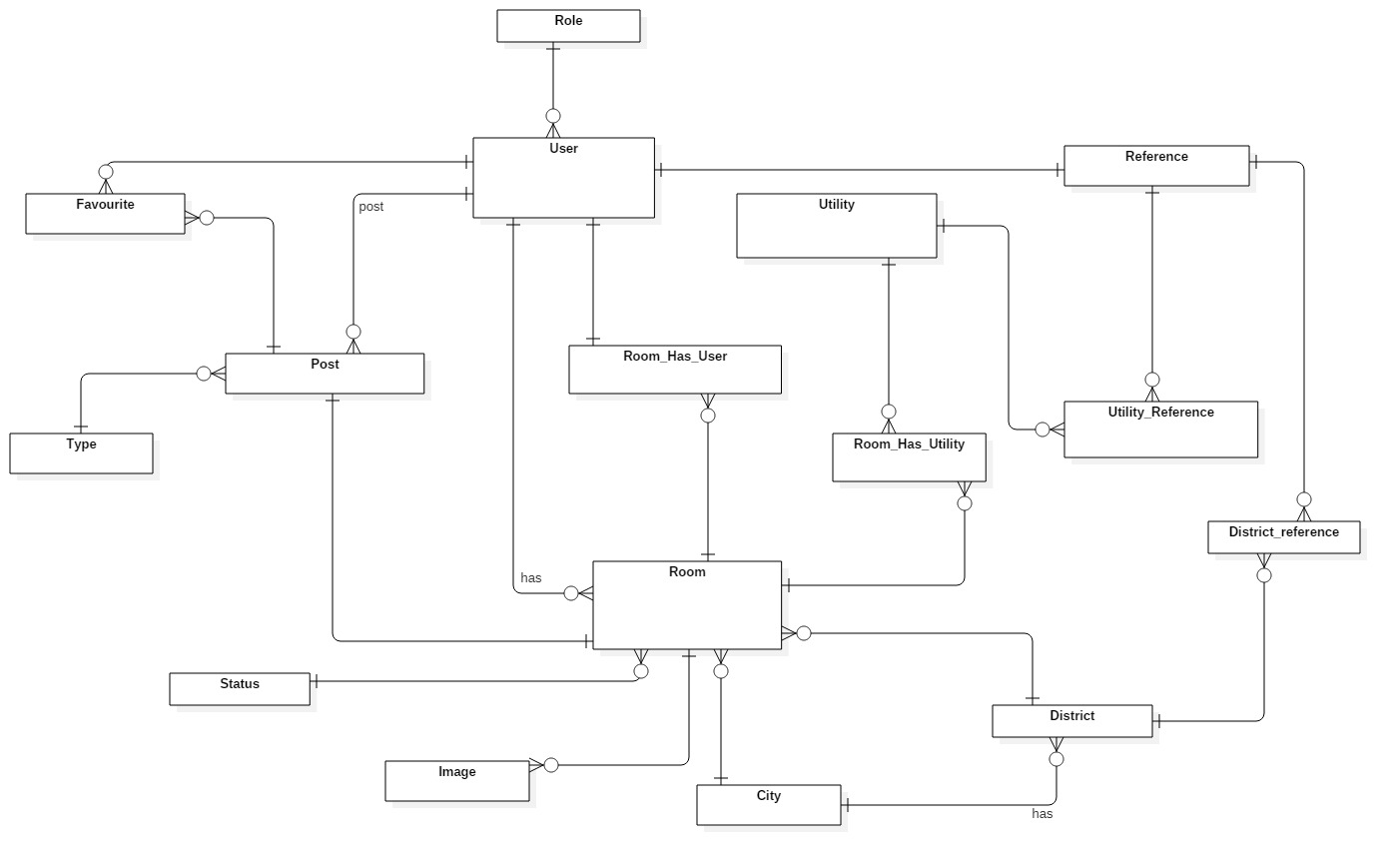


Figure 21: Conceptual Diagram

|  |  |
| --- | --- |
| Entity Data dictionary: describe content of all entities | |
| Entity Name | **Description** |
| User | Contains users information |
| Room | Contains rooms information |
| Post | Contains posts information |
| Role | Describe roles in system. |
| Favorites | Contains favorite information |
| Utilities | Contains utilities information |
| Reference | Contains references information |
| District | Contains districts information |
| City | Contains city information |
| Image | Contains images information |
| Type | Describe type of post in system. |
| Status | Describe status of room in system. |
| Room\_Has\_Utilities | Describe utilities of room in system |
| Utilities\_Reference | Describe utilities of user in system |
| District\_Reference | Describe districts that user prefer to search |
| Room\_Has\_User | Contains users in room information |

Table 15: Conceptual Entity dictionary

# D. Software Design Description

## System Architecture Design

### System Architecture Design

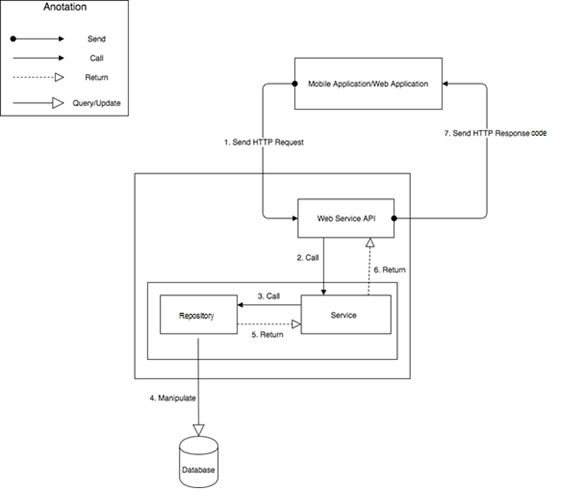


Figure 22: System Architecture Design (MVC)

## Component Diagram

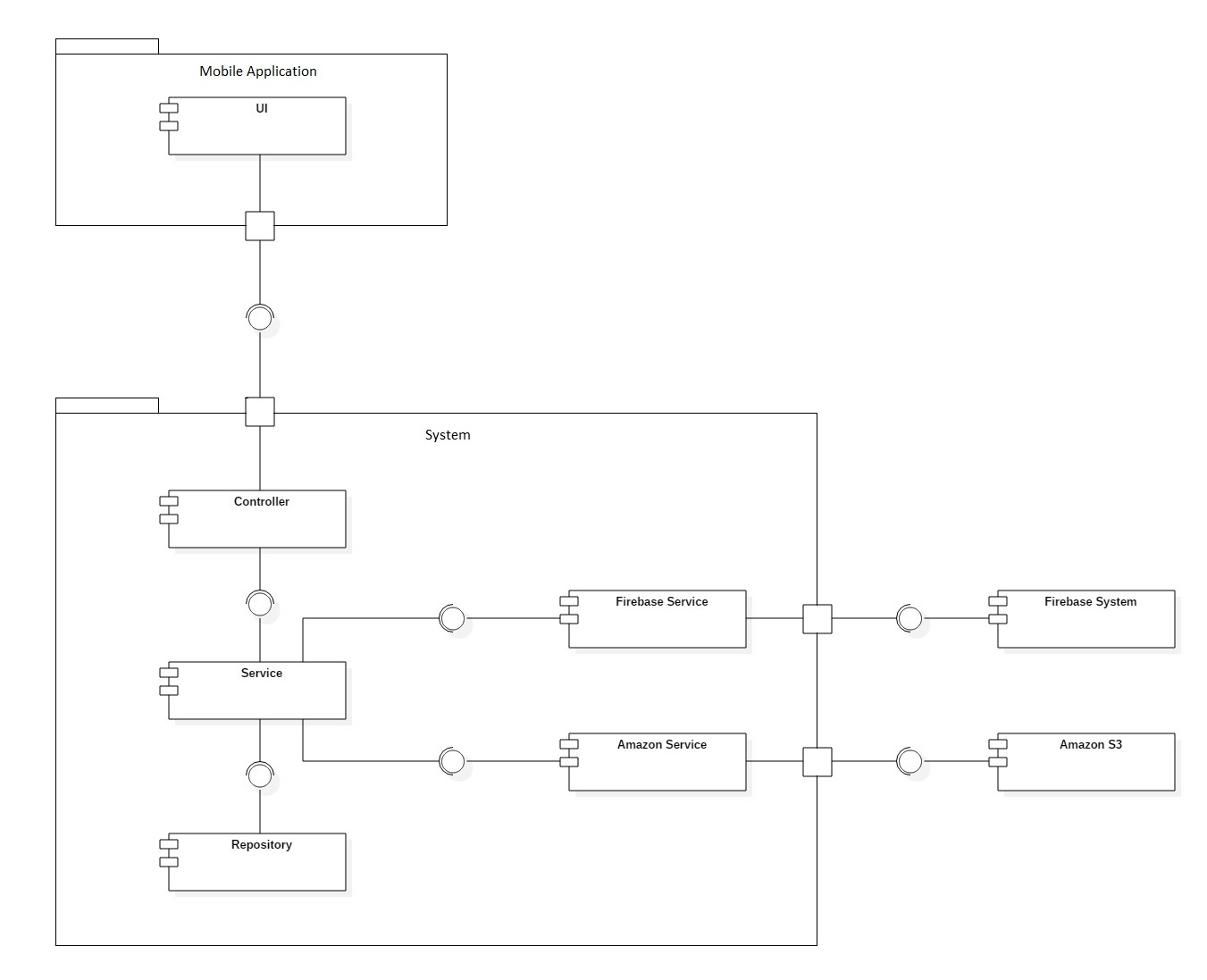


Figure 23: Component Diagram

|  |  |
| --- | --- |
| **COMPONENT DICTIONARY: DESCRIBES COMPONENT** | |
| **Component Name** | **Description** |
| UI | Mobile application package: View, Controller |
| Controller | Component to handle HTTP request |
| Service | Component to handle business logic |
| Repository | Component to handle data |
| Firebase Service | Component to handle push notification |
| Amazon Service | Component to handle upload image |
| Firebase System | Component to handle access to real time database |
| Amazon S3 | Component to handle image storage |

Table 16: Component Diagram Explanation

## Detail Description

### Class Diagram

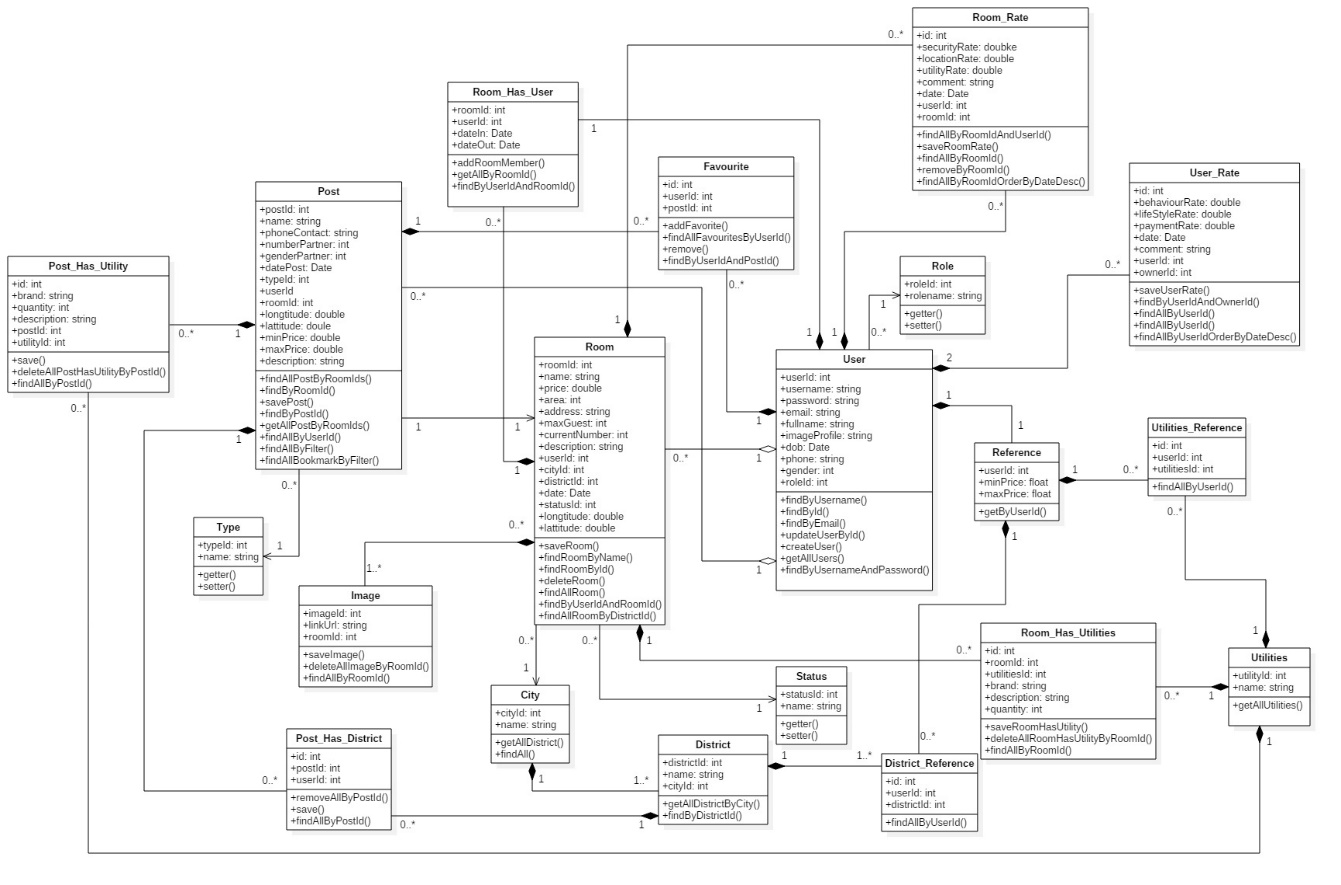


Figure 24: Class Diagram

|  |  |  |
| --- | --- | --- |
| **CLASS DICTIONARY: DESCRIBE CLASS** | | |
| **Class Name** | **Mapping column with Conceptual diagram** | **Description** |
| User | User | Contain the user information |
| Room | Room | Contain the room information |
| Post | Post | Contain the post information |
| Role | Role | Contain the role information |
| Favourites | Favourites | Contain the bookmark information |
| Utilities | Utilities | Contain the utilities information |
| Reference | Reference | Contain the reference profile information |
| District | District | Contain the district information |
| City | City | Contain the city information |
| Image | Image | Contain the image information |
| Type | Type | Contain the type information |
| Status | Status | Contain the status information |
| Room\_Has\_Utilities | Room\_Has\_Utilities | Contain the room’s utilities information |
| Utilities\_Reference | Utilities\_Reference | Contain the user’s utilities information |
| District\_Reference | District\_Reference | Contain districts that user prefer to search |
| Room\_Has\_User | Room\_Has\_User | Contains users in room information |
| Room\_Rate | N/A | Contain the room rate information |
| User\_Rate | N/A | Contain the user rate information |
| Post\_Has\_Utility | N/A | Contain the post’s utilities |
| Post\_Has\_District | N/A | Contain the post’s districts |

Table 17: Class Diagram Explanation

### Interactive diagram

#### Sequence Diagram

##### Create Room

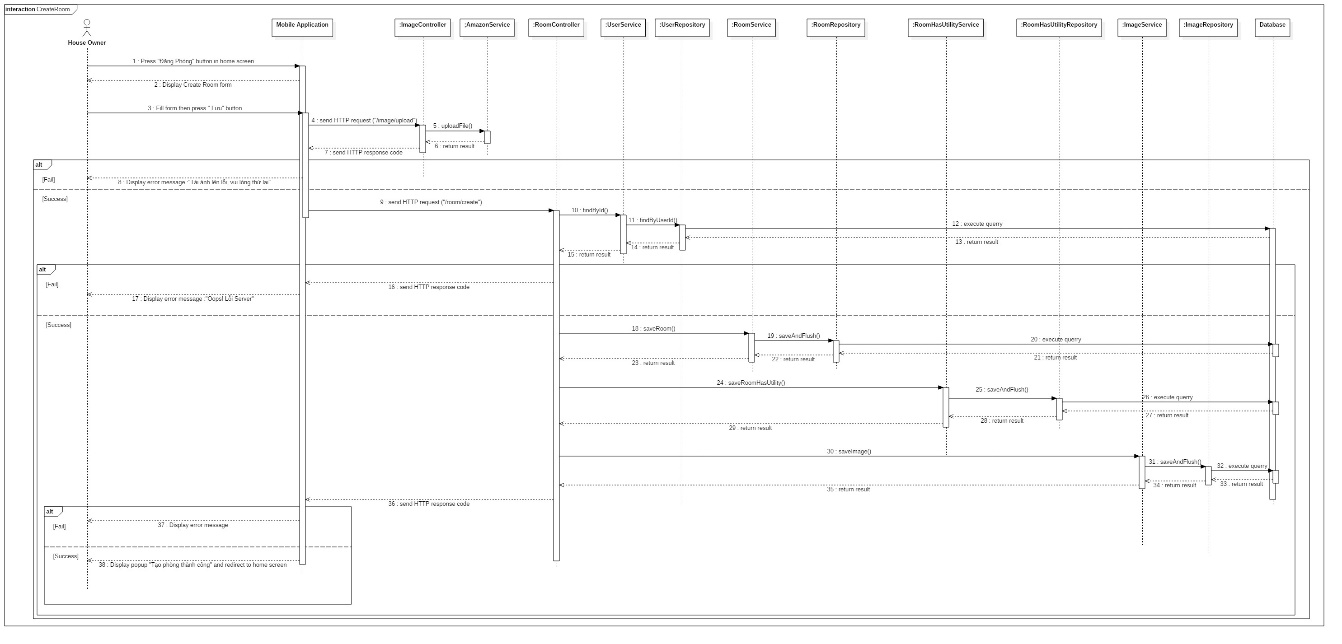


Figure 25: Sequence Diagram - <House owner> Create Room

##### Update Room

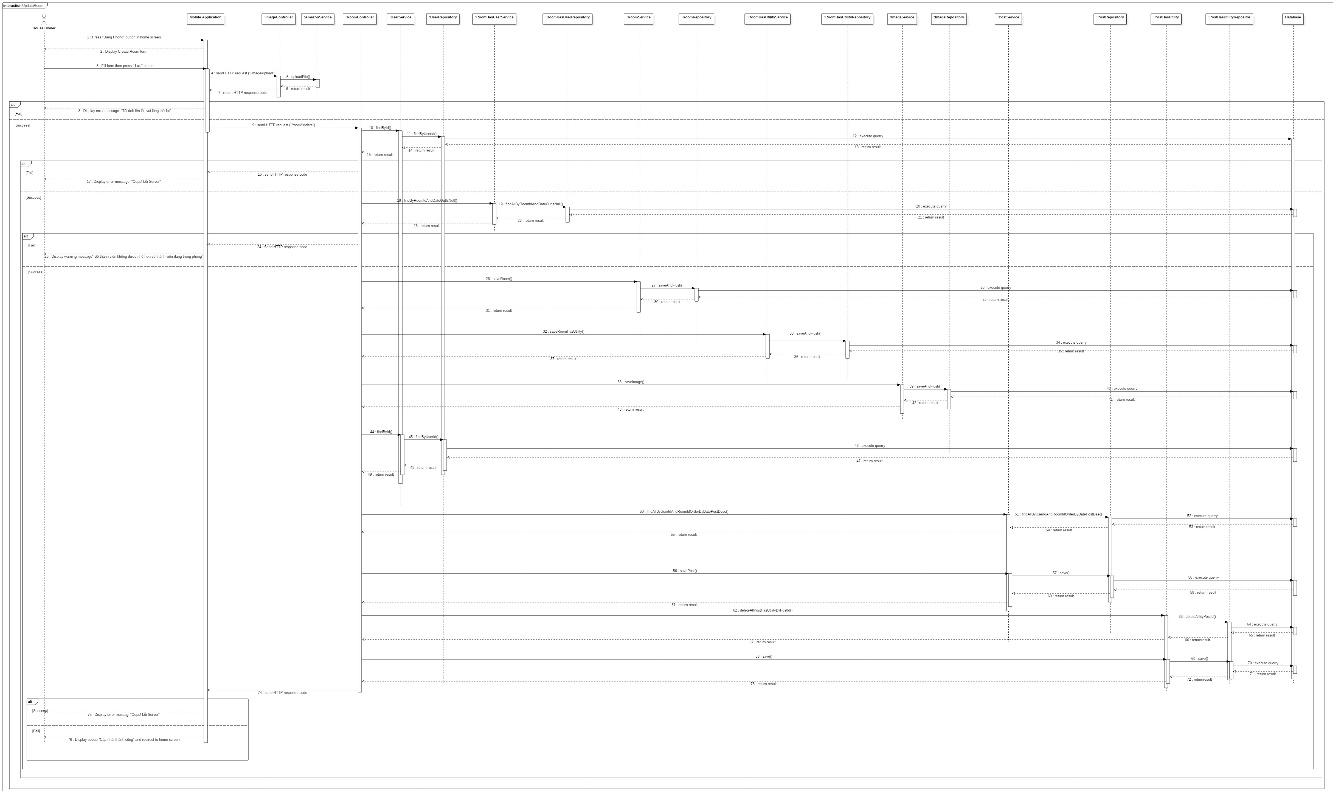


Figure 26: Sequence Diagram - <House owner> Update Room

##### Delete Room

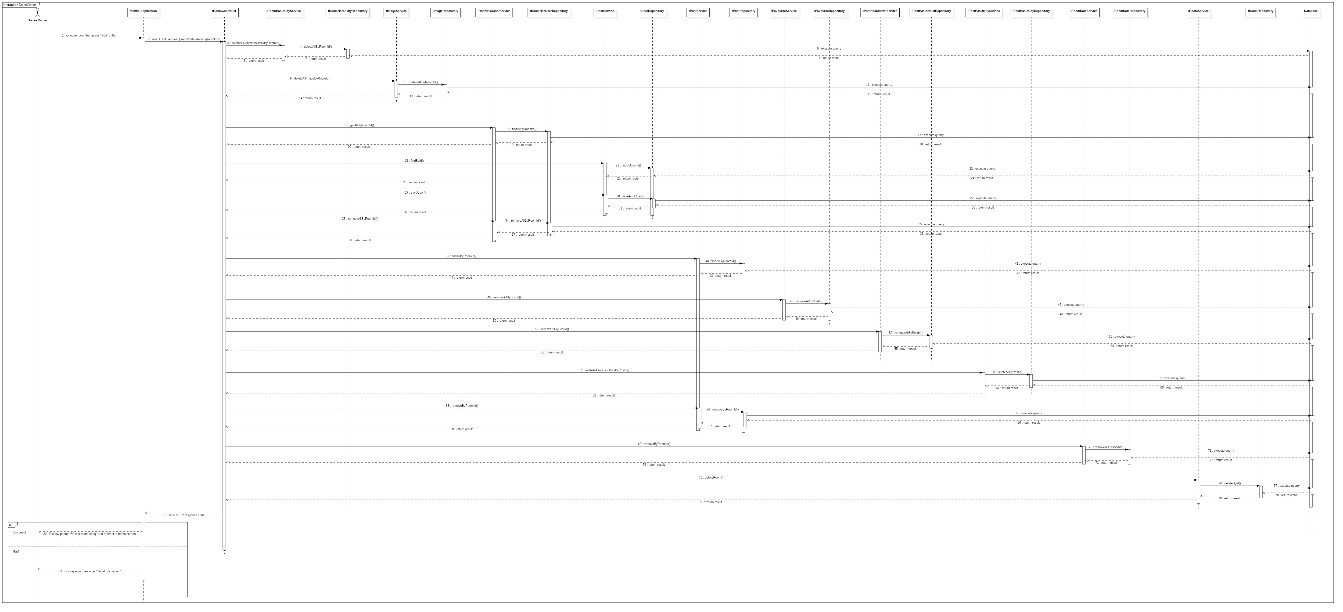


Figure 27: Sequence Diagram - <House owner> Delete Room

##### View Single Room

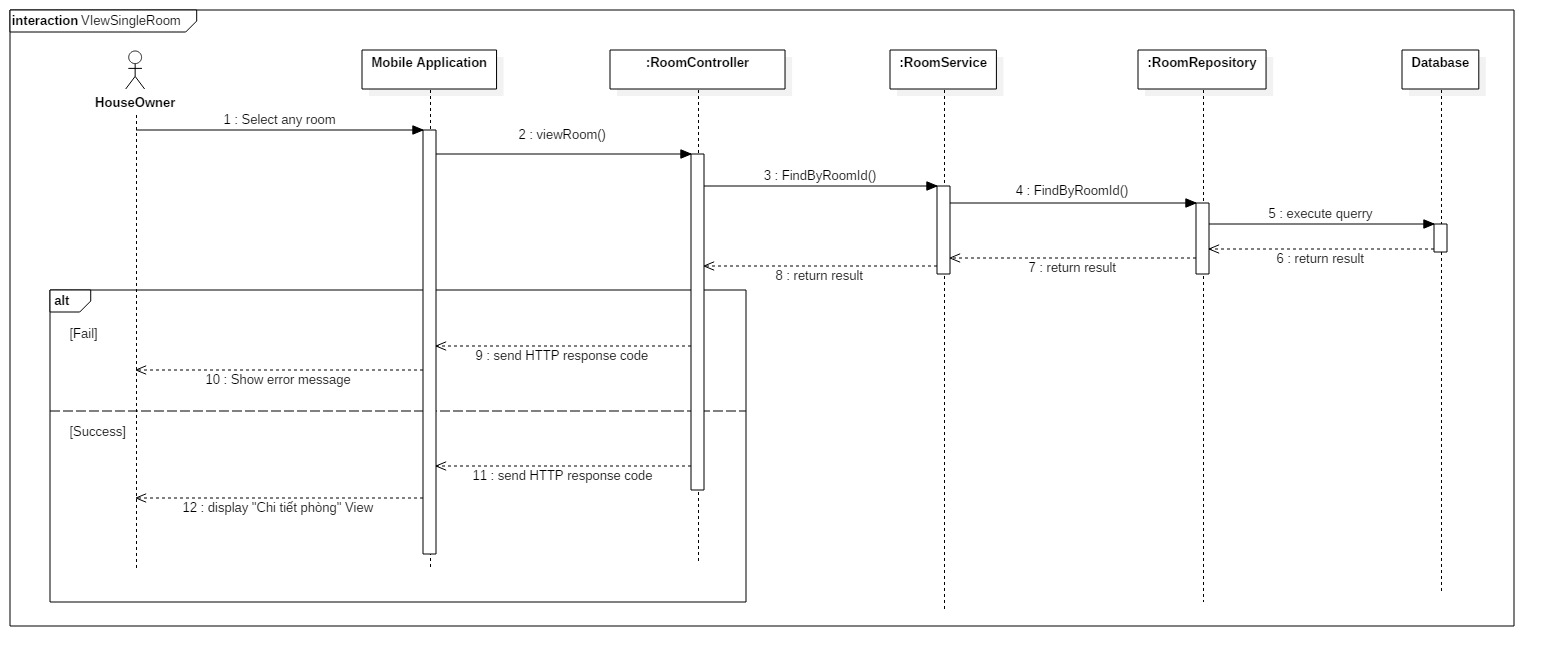


Figure 28: Sequence Diagram - <House owner> View Single Room

##### Add Member into Room

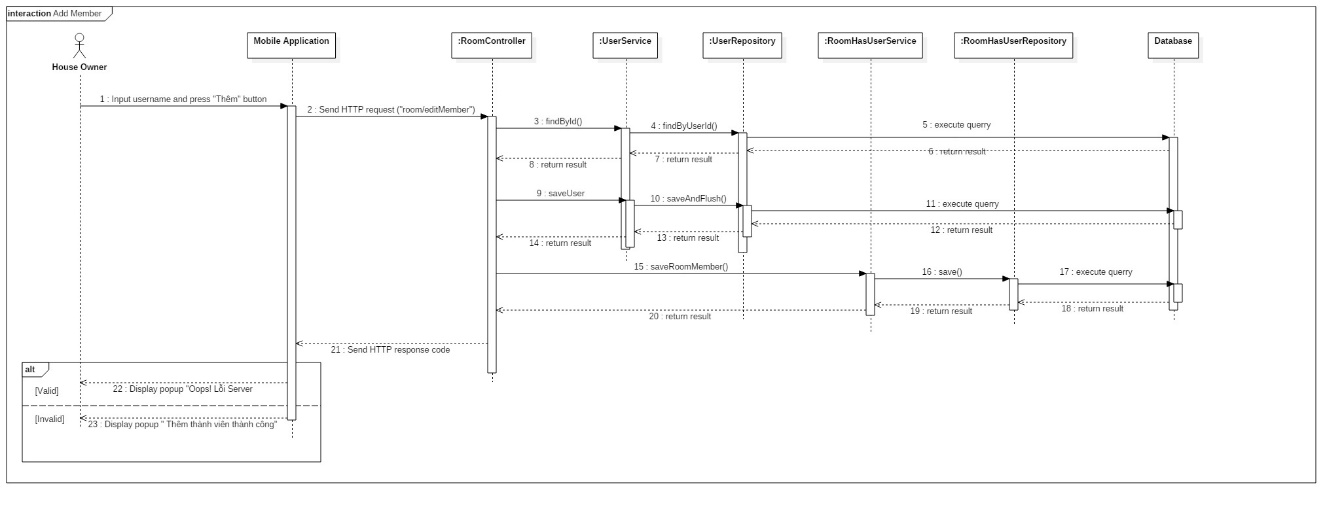


Figure 29: Sequence Diagram - <House owner> Add Member into Room

##### Delete Member of Room

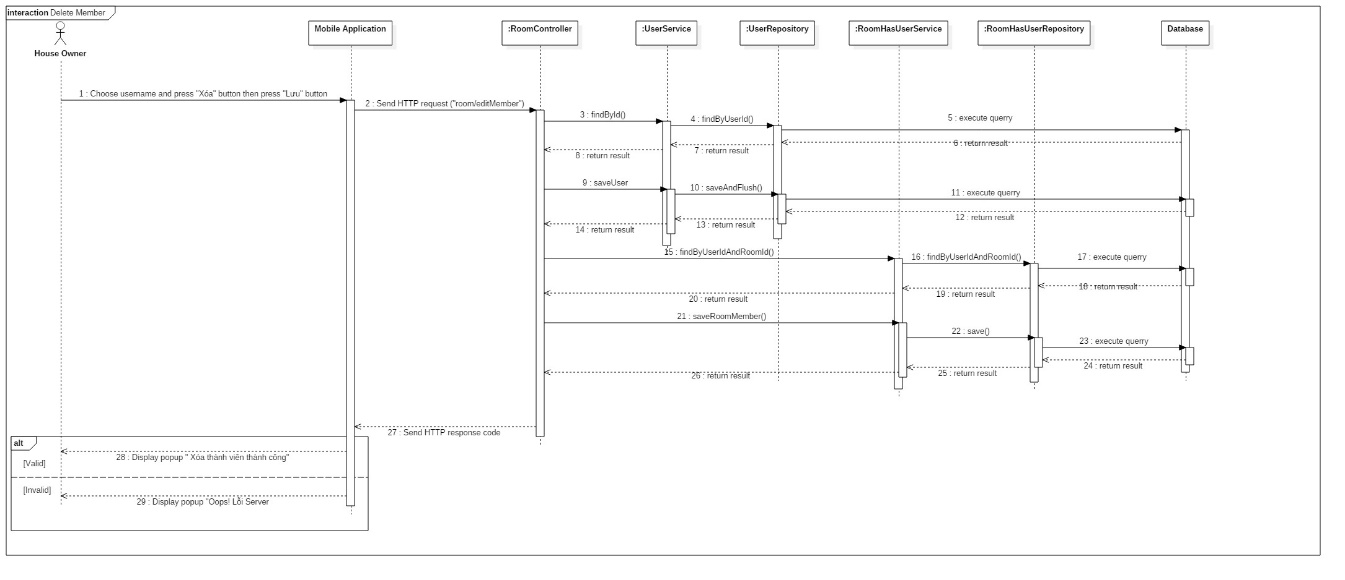


Figure 30: Sequence Diagram - <House owner> Delete Members of Room

##### Create Finding Roommate Post

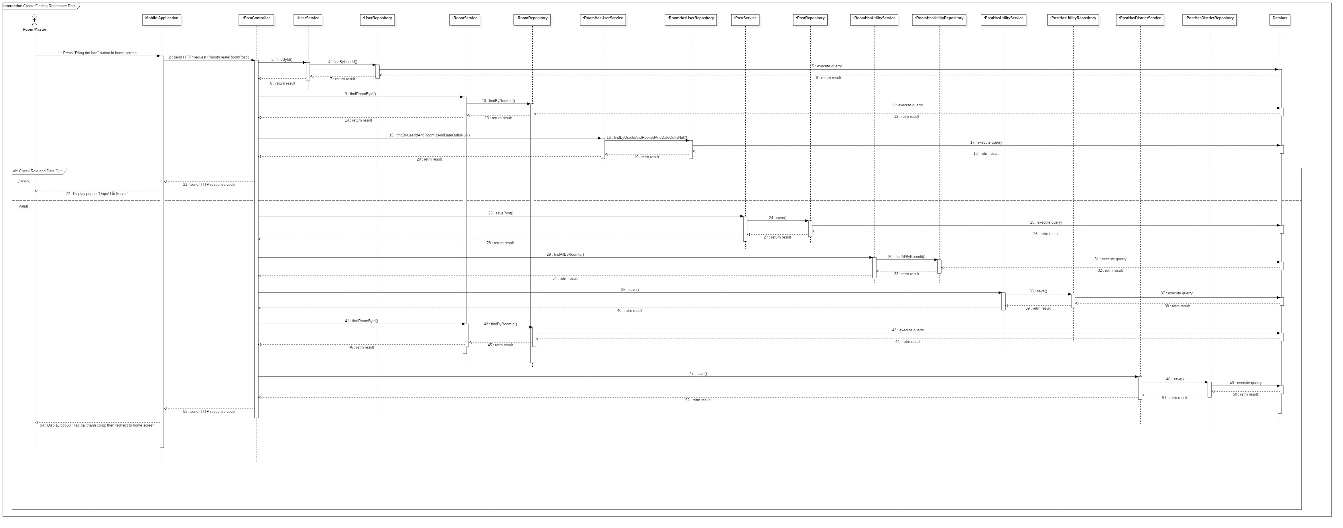


Figure 31: Sequence Diagram - <Room Master> Create Finding Roommate Post

##### Delete Finding Roommate Post

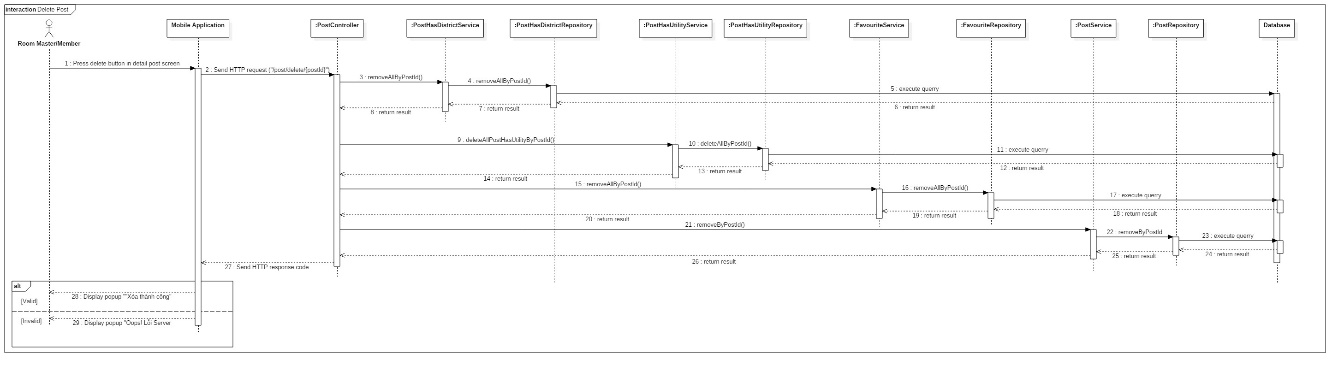


Figure 32: Sequence Diagram - <Room Master> Delete Finding Roommate Post



##### Update Finding Roommate Post

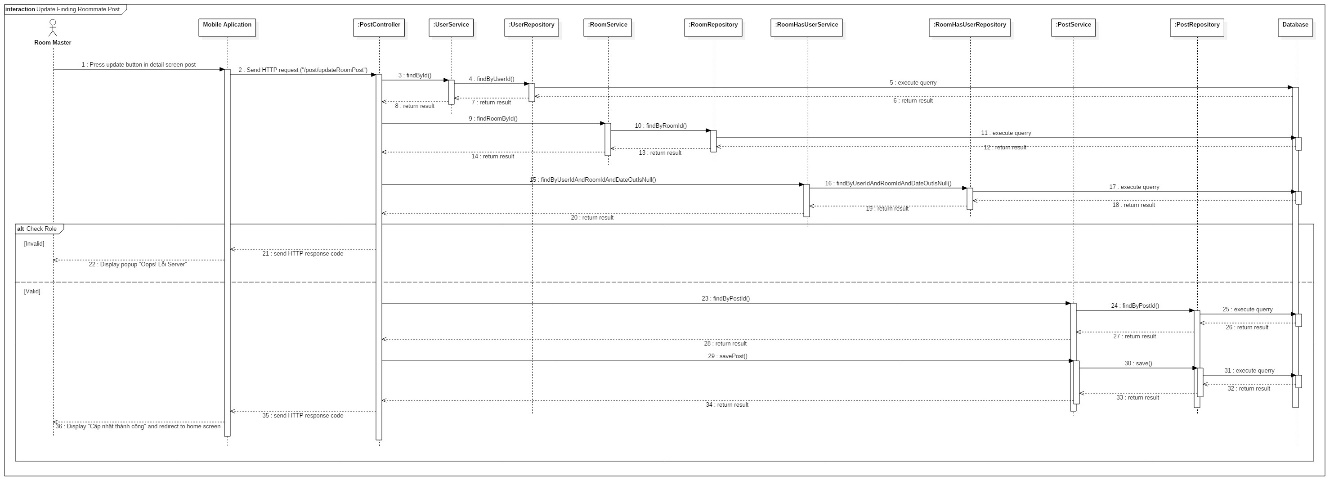


Figure 33: Sequence Diagram - <Room Master> Update Finding Roommate Post



##### View Suggest List Overview

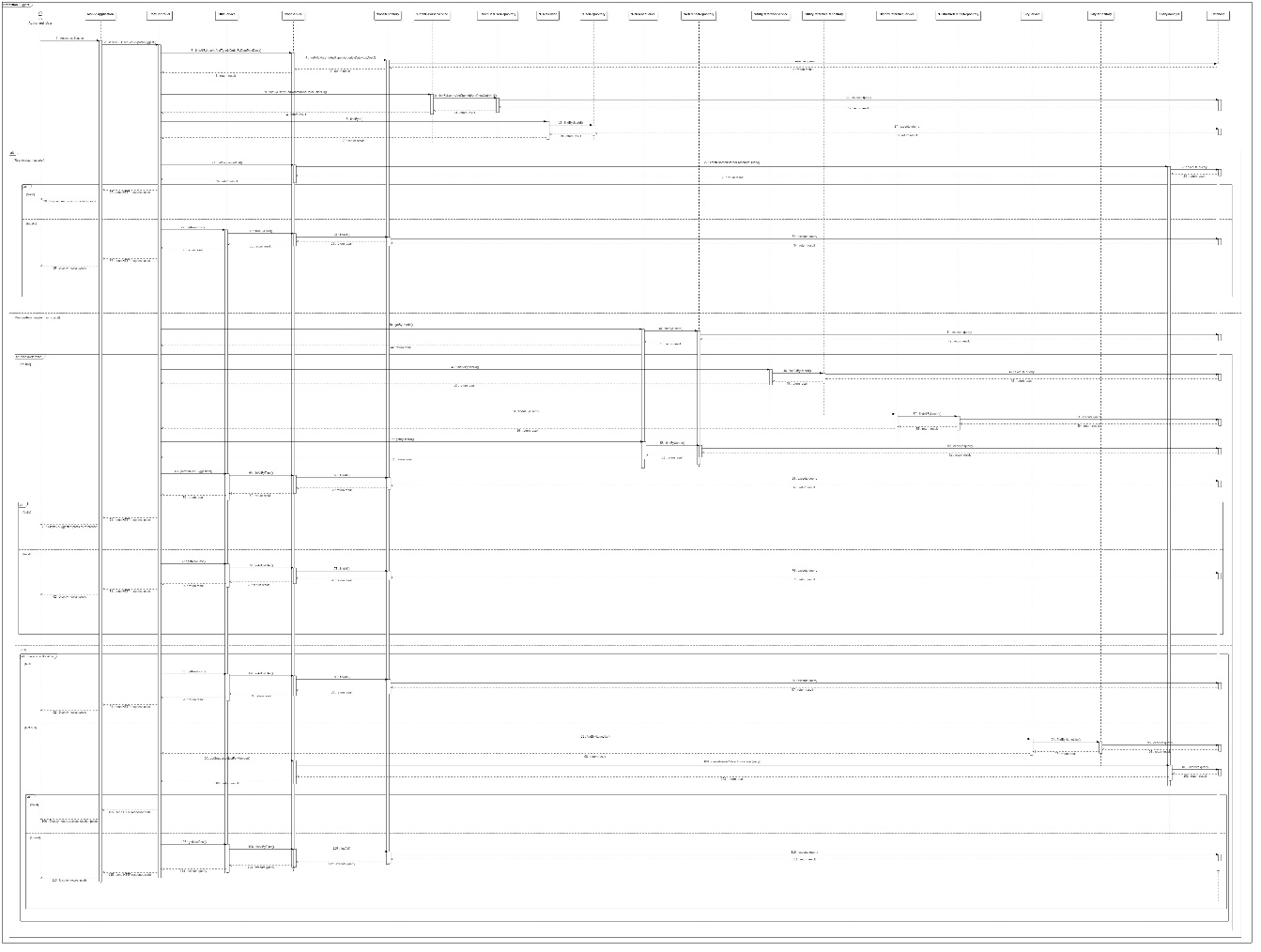


Figure 34: Sequence Diagram - <Member/Room Master> View Suggest List

##### <Suggest> Member/Room Master has reference and has no posts.

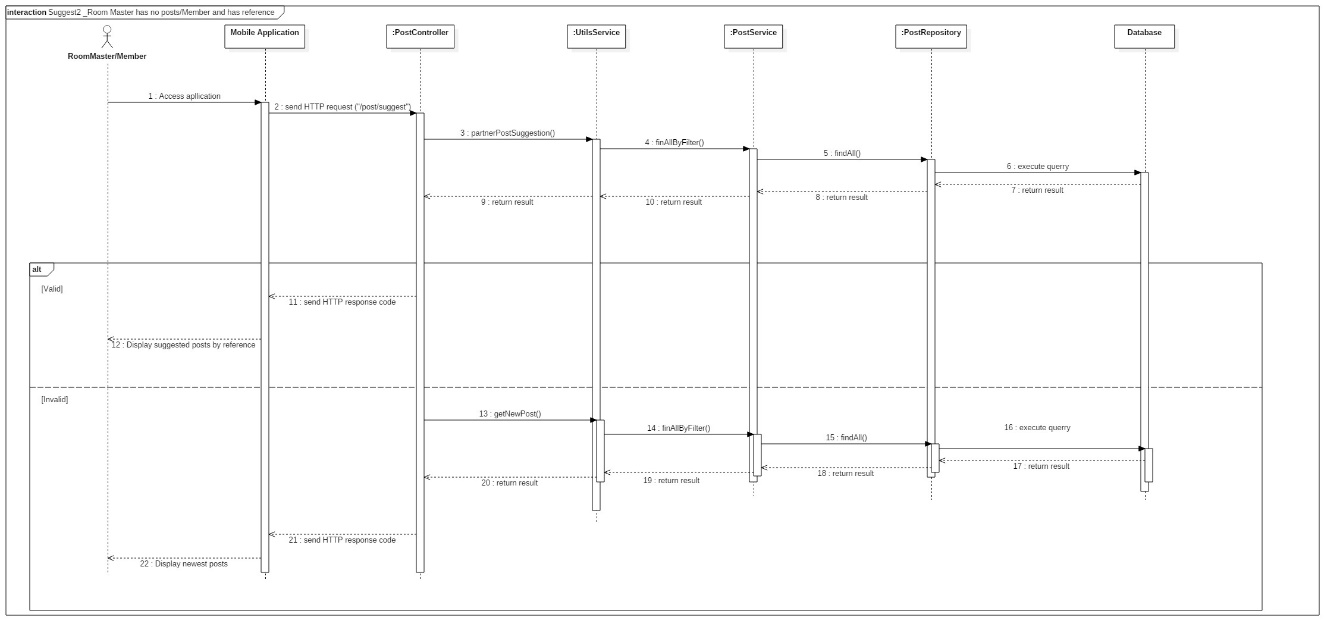


Figure 35: Sequence Diagram - <Member/Room Master> Has reference and has no posts.

##### <Suggest> Room Master has posts

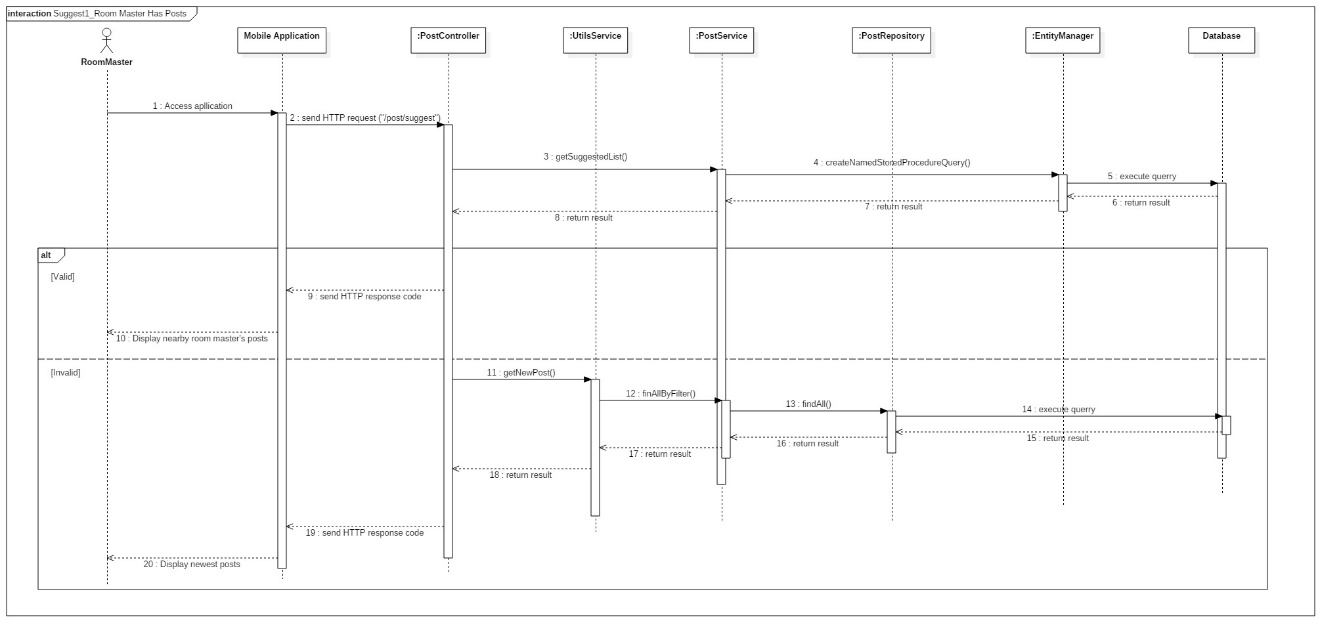


Figure 36: Sequence Diagram - <Room Master> Room Master has posts

##### <Suggest>Member/Room Master has no reference but allow access current location

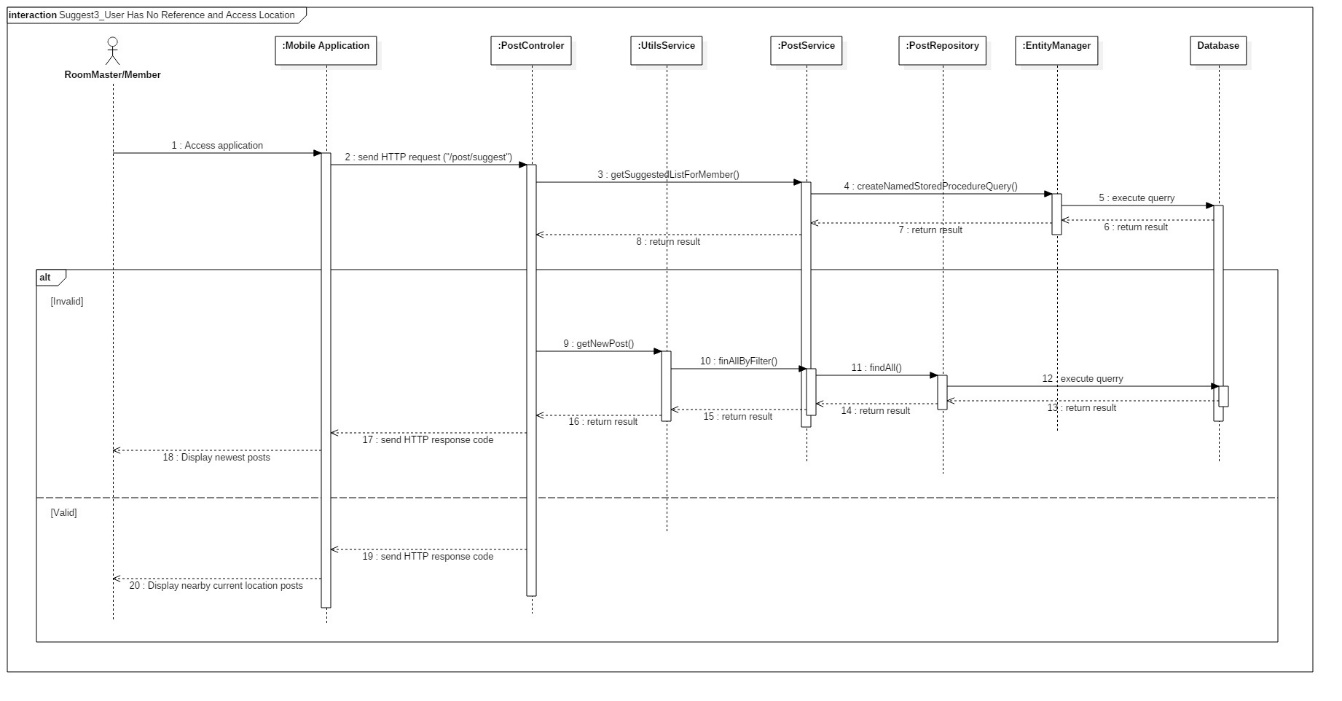


Figure 37: Sequence Diagram - <Member/Room Master> Member/Room Master has no reference but allow access current location.

##### <Suggest> Member/Room Master has no reference and not allow access current location

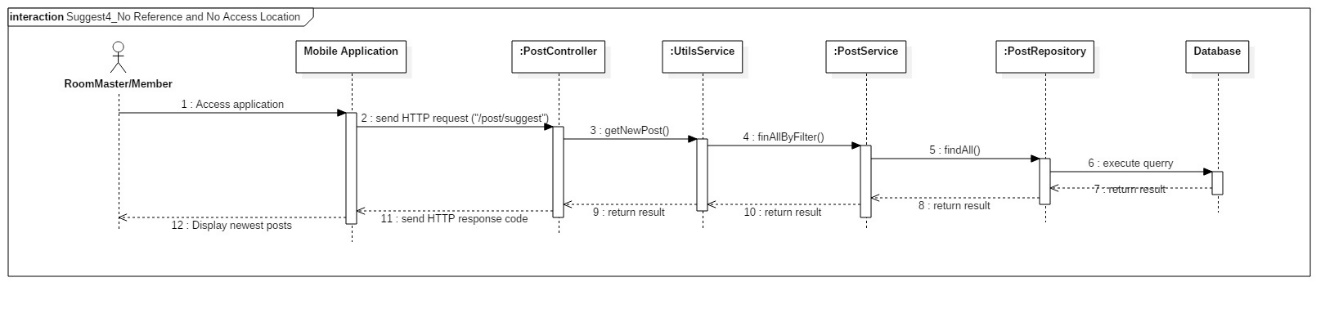


Figure 38: Sequence Diagram - <Member/Room Master> Member/Room Master has no reference and not allow access current location.

##### Create Finding Room Post

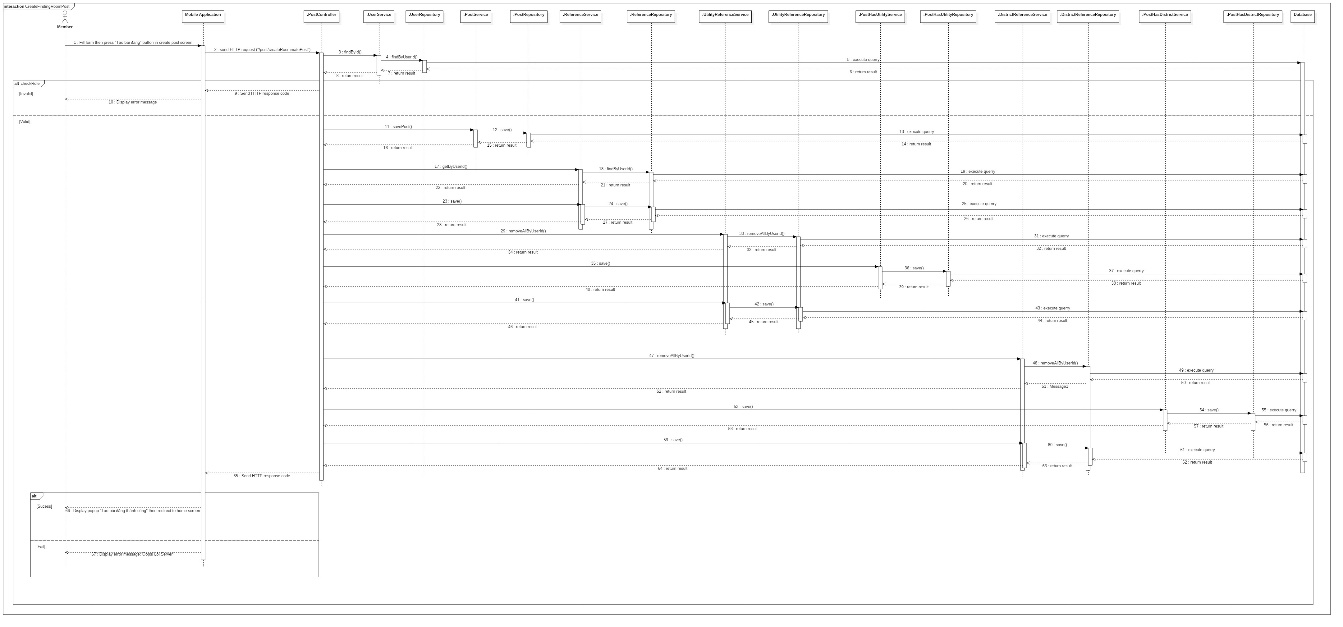


Figure 39: Sequence Diagram - <Member> Create Finding Room Post

##### Update Finding Room Post

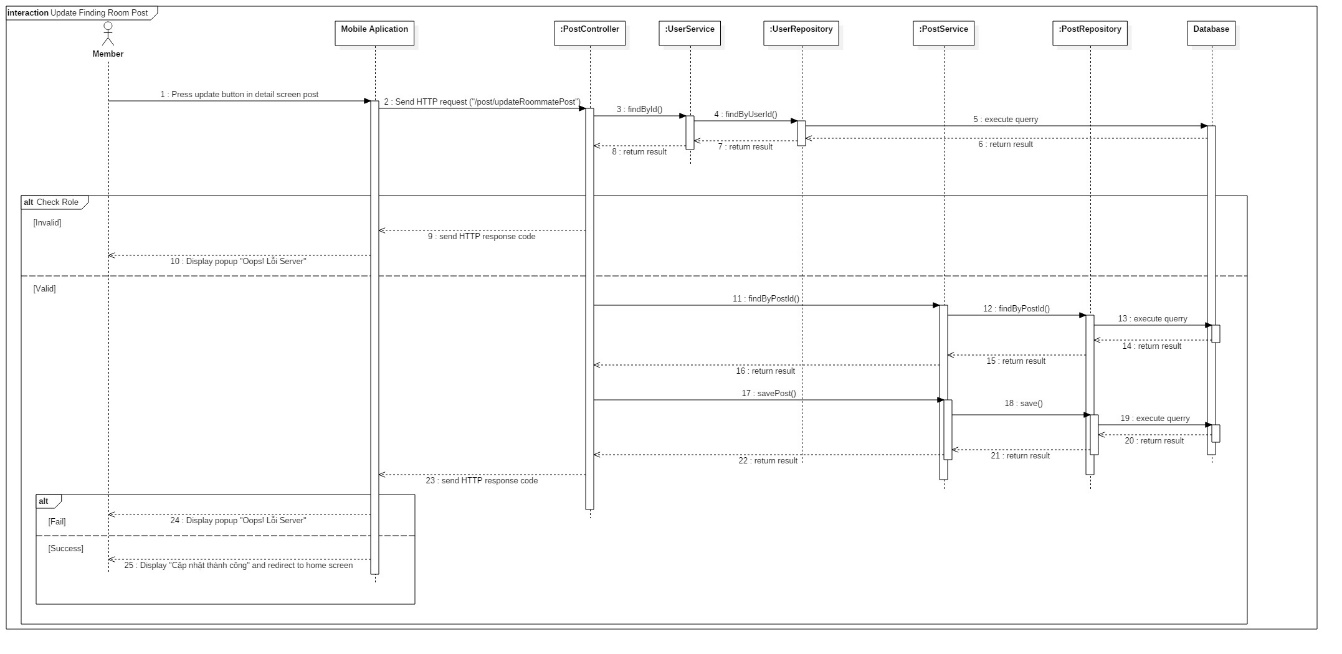


Figure 40: Sequence Diagram - <Member> Update Finding Room Post

##### Delete Finding Room Post

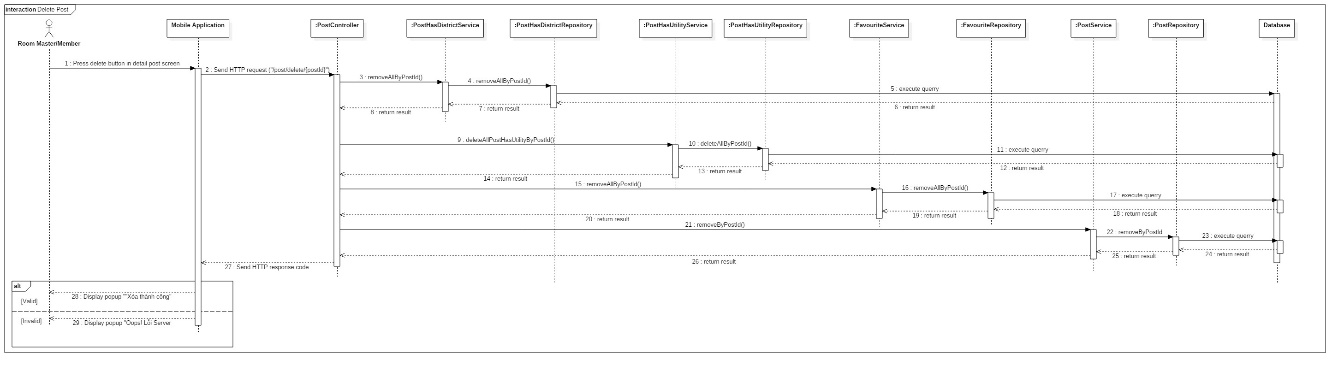


Figure 41: Sequence Diagram - <Member> Delete Finding Room Post

##### Filter

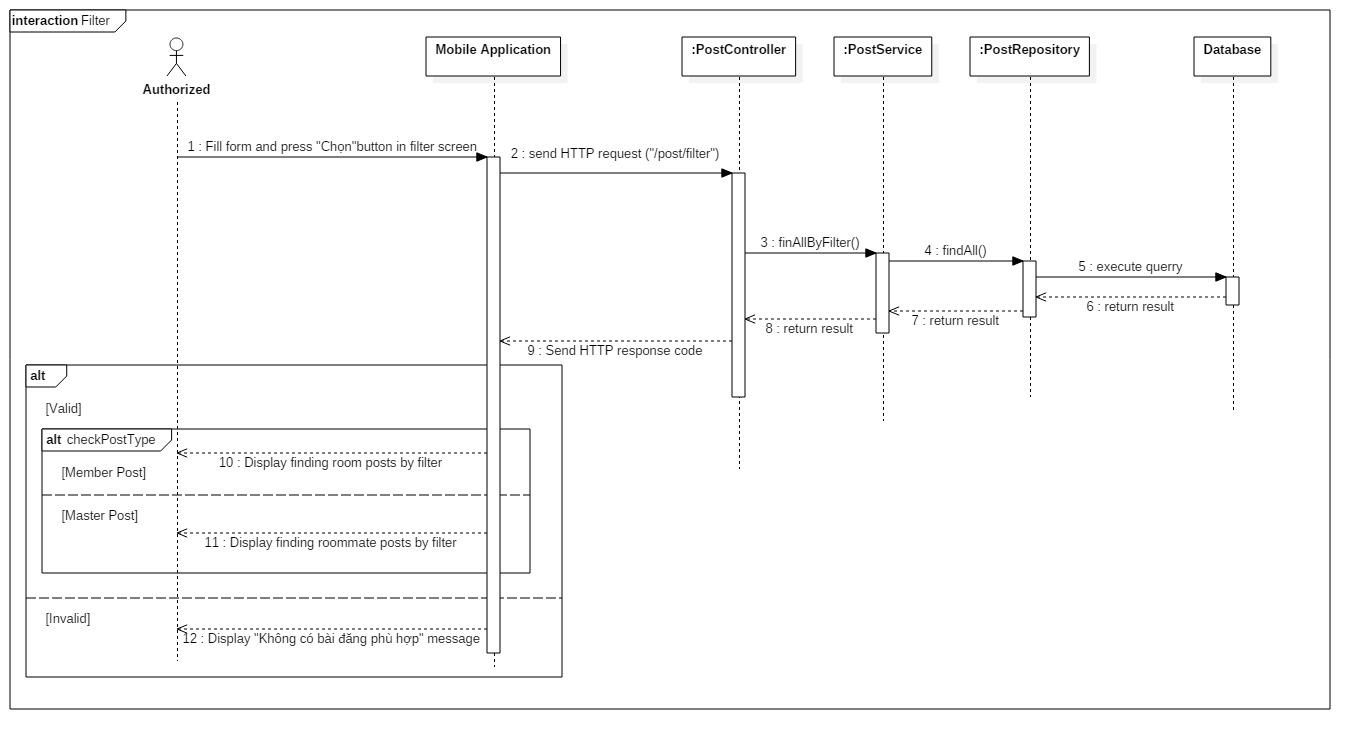


Figure 42: Sequence Diagram - <Authorized User> Filter

## Database Design

### Entity relationship diagram (ERD)

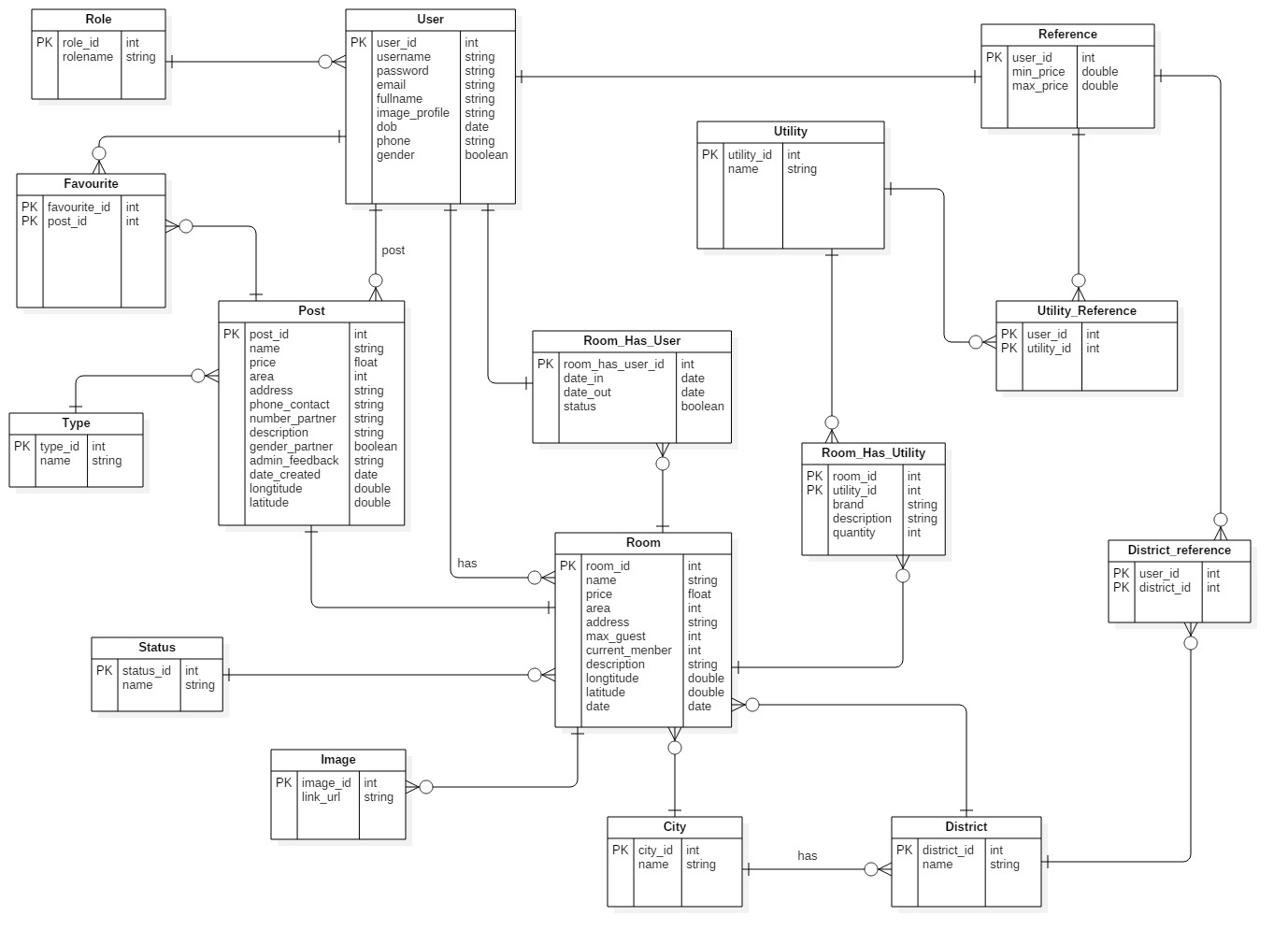


Figure 43: Entity Relationship diagram

### Entity dictionary

|  |  |
| --- | --- |
| **ENTITY DATA DICTIONARY: DESCRIBE CONTENT OF ALL ENTITIES** | |
| **Entity Name** | **Description** |
| **User** | Contains users information |
| **Room** | Contains rooms information |
| **Post** | Contains posts information |
| **Role** | Describe roles in system. |
| **Favorites** | Contains favorite information |
| **Utilities** | Contains utilities information |
| **Reference** | Contains references information |
| **District** | Contains districts information |
| **City** | Contains city information |
| **Image** | Contains images information |
| **Type** | Describe type of post in system. |
| **Status** | Describe status of room in system. |
| **Room\_Has\_Utilities** | Describe utilities of room in system |
| **Utilities\_Reference** | Describe utilities of user in system |
| **District\_Reference** | Describe districts that user prefer to search |
| **Room\_Has\_User** | Contains users in room information |

Table 18: Entity Data dictionary

## Algorithms

### Definition

In ASP system, suggest nearby posts function is a main scope.

As a room master who has room post, this function will suggest nearby other room master's posts.

As a member who does not have a reference and allows to access location, this function will suggest room posts around.

### Define Problem

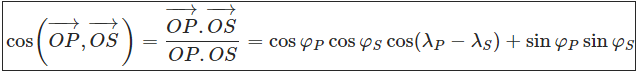
When room master posts a post or member allows accessing location, the system will calculate the distance based on longitude and latitude. Because the earth is a spherical shape, so we need an algorithm to do this.

### Solution

View the Earth as a spherical shape with center is O. City S is located 11° latitude, 106°30' longitude.

City P is located 49° latitude, 2°30' longitude.

With Spherical Law of Cosines, the system can calculate the angle of



Where φ is latitude (positive when the point is in the Northern Hemisphere, negative in Southern Hemisphere), λ is the longitude (positive sign in the Eastern Hemisphere, negative sign in the Western Hemisphere)

cos() = cos49°cos11°cos104°+sin11°sin49° ≈ −0,01179

So that,   ≈ 90°40'32"

A nautical mile is based on the circumference of the planet Earth. If you were to cut the Earth in half at the equator, you could pick up one of the halves and look at the equator as a circle. You could divide that circle into 360 degrees. You could then divide a degree into 60 minutes. A minute of arc on the planet Earth is 1 nautical mile.

So that, we convert distances from degrees to nautical miles by multiply by 60.

The system converts nautical mile to statute mile by multiply by 1.1515.

Finally, the system converts statute mile to kilometer by multiply by 1.609344.

Base on the result above, the system takes 90°40'32" multiply by 60, multiply by 1.1515 and multiply by 1.609344, and get the result is 10082.8746 km.

# E. System Implementation & Test

## Database Relationship Diagram

### Physical Diagram

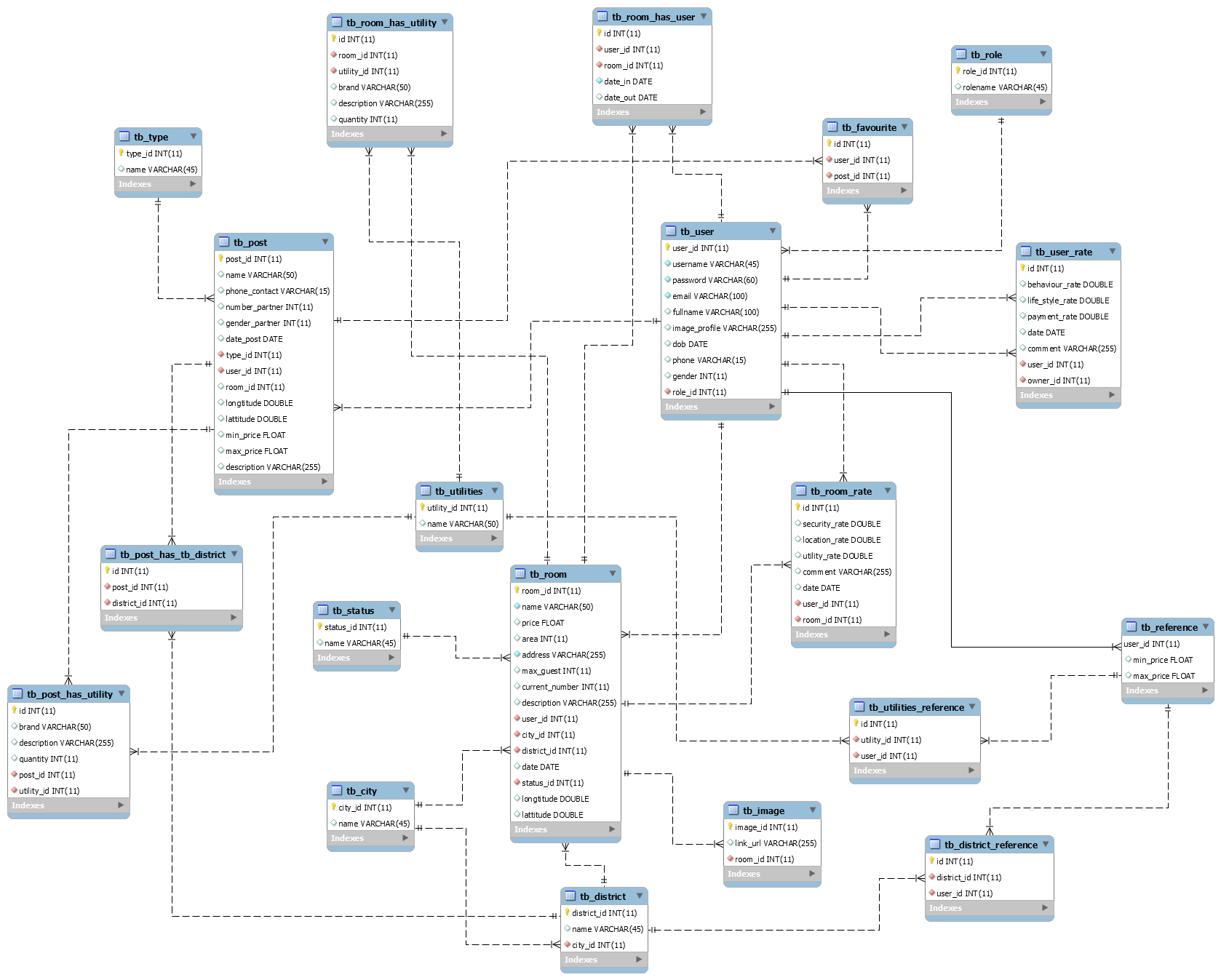


Figure 44: Physical diagram

### Data dictionary

|  |  |
| --- | --- |
| **DATA DICTIONARY: DESCRIBE CONTENT OF ALL TABLES** | |
| **Table name** | **Description** |
| **User** | Contains users information |
| **Room** | Contains rooms information |
| **Post** | Contains posts information |
| **Role** | Describe roles in system. |
| **Favorites** | Contains favorite information |
| **Utilities** | Contains utilities information |
| **Reference** | Contains references information |
| **District** | Contains districts information |
| **City** | Contains city information |
| **Image** | Contains images information |
| **Type** | Describe type of post in system |
| **Status** | Describe status of room in system |
| **User\_Rate** | Contains user rate information |
| **Room\_Rate** | Contains room rate information |
| **Room\_Has\_Utilities** | Describe utilities of room in system |
| **Utilities\_Reference** | Describe utilities of user in system |
| **District\_Reference** | Describe districts that user prefer to search |
| **Room\_Has\_User** | Contains users in room information |
| **Post\_Has\_Utility** | Describe utilities of post in system |
| **Post\_Has\_District** | Describe districts of post in system |

Table 19: Data dictionary

# G. Appendix

## 1. A nautical mile

<https://science.howstuffworks.com/innovation/science-questions/question79.htm>

Accessed on: Oct. 2018.

## 2. Spherical law of cosines

<https://en.wikipedia.org/wiki/Spherical_law_of_cosines>

Accessed on: Oct. 2018.