

**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 screen is intentionally left blank*

# Table of Content

[Table of Content 4](#_Toc527923265)

[List of Table 9](#_Toc527923266)

[List of Figure 10](#_Toc527923267)

[Definitions, Acronyms, and Abbreviations 12](#_Toc527923268)

[A. Introduction 12](#_Toc527923269)

[1. Project Information 12](#_Toc527923270)

[2. Introduction 12](#_Toc527923271)

[3. Current Situation 13](#_Toc527923272)

[4. Problem Definition 13](#_Toc527923273)

[5. Proposed Solution 13](#_Toc527923274)

[5.1 Feature functions 13](#_Toc527923275)

[5.2 Advantages and Disadvantages 14](#_Toc527923276)

[6. Functional Requirements 14](#_Toc527923277)

[7. Role and Responsibility 15](#_Toc527923278)

[B. Software Project Management Plan 16](#_Toc527923279)

[1. Problem Definition 16](#_Toc527923280)

[1.1 Name of this Capstone Project 16](#_Toc527923281)

[1.2 Problem Abstract 16](#_Toc527923282)

[1.3 Project Overview 16](#_Toc527923283)

[1.3.1 Current Situation 16](#_Toc527923284)

[1.3.2 The Proposed System 17](#_Toc527923285)

[1.3.2.1 Mobile Application 17](#_Toc527923286)

[1.3.2.2 Web Application 17](#_Toc527923287)

[1.3.3 Boundaries of the System 17](#_Toc527923288)

[1.3.4 Future Plan 18](#_Toc527923289)

[1.3.5 Development Environment 18](#_Toc527923290)

[1.3.5.1 Hardware requirements 18](#_Toc527923291)

[1.3.5.2 Software requirements 19](#_Toc527923292)

[2. Project Organization 19](#_Toc527923293)

[2.1 Software Process Model 19](#_Toc527923294)

[2.2 Roles and Responsibilities 20](#_Toc527923295)

[2.3 Tools and Techniques 22](#_Toc527923296)

[3. Project Management Plan 22](#_Toc527923297)

[3.1 Product Backlog 22](#_Toc527923298)

[3.2 Sprint Backlog 22](#_Toc527923299)

[3.3 Meeting Minutes 22](#_Toc527923300)

[4. Coding Convention 23](#_Toc527923301)

[C. Software Requirement Specification 23](#_Toc527923302)

[1. User Requirement Specification 23](#_Toc527923303)

[1.1 Unauthorized User Requirement 23](#_Toc527923304)

[1.2 Authorized User Requirement 23](#_Toc527923305)

[1.3 House Owner Requirement 24](#_Toc527923306)

[1.4 Room Master Requirement 24](#_Toc527923307)

[1.5 Member Requirement 24](#_Toc527923308)

[1.6 Admin Requirement 25](#_Toc527923309)

[2. System Requirement Specification 25](#_Toc527923310)

[2.1 External Interface Requirement 25](#_Toc527923311)

[2.1.1 User Interface 25](#_Toc527923312)

[2.1.2 Hardware Interface. 25](#_Toc527923313)

[2.1.3 Software Interface 26](#_Toc527923314)

[2.1.4 Communication Protocol 26](#_Toc527923315)

[2.2 System Overview Use Case 26](#_Toc527923316)

[2.3 List of Use Case 28](#_Toc527923317)

[2.3.1 <Unauthorized User> Overview Use Case 28](#_Toc527923318)

[2.3.1.1 <Unauthorized User> Register 29](#_Toc527923319)

[2.3.1.2 <Unauthorized User> Login 31](#_Toc527923320)

[2.3.2 <Authorized User> Overview Use 33](#_Toc527923321)

[2.3.2.1 <Authorized User> Logout 33](#_Toc527923322)

[2.3.2.2 <Authorized User> View Notification 35](#_Toc527923323)

[2.3.2.3 <Authorized User> Update Info 36](#_Toc527923324)

[2.3.3 <House Owner> Overview Use Case 40](#_Toc527923325)

[2.3.3.1 <House Owner> Create Room 40](#_Toc527923326)

[2.3.3.2 <House owner> Update Room 42](#_Toc527923327)

[2.3.3.3 <House owner> Delete Room 46](#_Toc527923328)

[2.3.3.4 <House owner> Add Member into Room 48](#_Toc527923329)

[2.3.3.5 <House owner> Delete Member from Room 52](#_Toc527923330)

[2.3.4 <Room Master> Overview Use Case 54](#_Toc527923331)

[2.3.4.1 < Room Master > View Suggested Lis 55](#_Toc527923332)

[2.3.4.2 < Room Master > Create Post 55](#_Toc527923333)

[2.3.4.3 < Room Master > Update Post 58](#_Toc527923334)

[2.3.4.4 < Room Master > Delete Post 62](#_Toc527923335)

[2.3.5 <Member> Overview Use Case 64](#_Toc527923336)

[2.3.5.1 <Member> View Suggested Room Post list 65](#_Toc527923337)

[2.3.5.2 <Member> Create partner Post 65](#_Toc527923338)

[2.3.5.3 <Member> Update partner Post 68](#_Toc527923339)

[2.3.5.4 <Member> Delete partner Post 71](#_Toc527923340)

[2.3.5.5 <Member> Negotiate 71](#_Toc527923341)

[2.3.5.6 <Member> Search Post 71](#_Toc527923342)

[2.3.6 <Admin> Overview Use Case 71](#_Toc527923343)

[2.3.6.1 <Admin> Manage Users 71](#_Toc527923344)

[2.3.6.2 <Admin> Manage Rooms 71](#_Toc527923345)

[2.3.6.3 <Admin> Manage Post 71](#_Toc527923346)

[2.3.6.4 <Admin> Approve Room 71](#_Toc527923347)

[3. Software System Attribute 71](#_Toc527923348)

[3.1 Usability 71](#_Toc527923349)

[3.1.1 Graphic User Interface 71](#_Toc527923350)

[3.1.2 Usability 71](#_Toc527923351)

[3.2 Reliability 72](#_Toc527923352)

[3.3 Availability 72](#_Toc527923353)

[3.4 Security 72](#_Toc527923354)

[3.5 Maintainability 72](#_Toc527923355)

[3.6 Portability 72](#_Toc527923356)

[3.7 Performance 72](#_Toc527923357)

[4. Conceptual diagram 72](#_Toc527923358)

[D. Software Design Description 73](#_Toc527923359)

[1. Design Overview 73](#_Toc527923360)

[2. System Architecture Design 73](#_Toc527923361)

[2.1 Web service Application Architecture Description 73](#_Toc527923362)

[2.2 Mobile Application Architecture Description 73](#_Toc527923363)

[3. Component Diagram 74](#_Toc527923364)

[3.1 Web Application 74](#_Toc527923365)

[3.2 Mobile Application 74](#_Toc527923366)

[4. Detail Description 74](#_Toc527923367)

[4.1 Class Diagram 74](#_Toc527923368)

[4.2 Class Diagram Explanation 74](#_Toc527923369)

[4.3 Interactive diagram 74](#_Toc527923370)

[4.3.1 Sequence Diagram 74](#_Toc527923371)

[4.3.1.1 Register 74](#_Toc527923372)

[4.3.1.2 Login 75](#_Toc527923373)

[4.3.1.3 Create Room 77](#_Toc527923374)

[4.3.1.4 Update Room 78](#_Toc527923375)

[4.3.1.5 Delete Room 79](#_Toc527923376)

[4.3.1.6 View Single Room 80](#_Toc527923377)

[4.3.1.7 Add Member into Room 81](#_Toc527923378)

[4.3.1.8 Delete Member of Room 82](#_Toc527923379)

[4.3.1.9 Create Room Post 83](#_Toc527923380)

[4.3.1.10 Delete Room Post 84](#_Toc527923381)

[4.3.1.11 Update Room Post 85](#_Toc527923400)

[4.3.1.12 Create Partner Post 86](#_Toc527923401)

[4.3.1.13 Update Partner Post 87](#_Toc527923402)

[4.3.1.14 Delete Partner Post 88](#_Toc527923403)

[4.3.1.15 Search Post 89](#_Toc527923404)

[4.3.1.16 View Suggest List 89](#_Toc527923428)

[5. Interface 89](#_Toc527923429)

[5.1 Admin Interface Design 89](#_Toc527923430)

[5.1.1 Login 89](#_Toc527923431)

[5.1.2 Manage User 89](#_Toc527923432)

[5.1.3 View Room List 89](#_Toc527923433)

[5.2 Unauthorized User Interface Design 89](#_Toc527923434)

[5.2.1 Login 89](#_Toc527923435)

[5.3 Authorized User Interface Design 90](#_Toc527923436)

[5.3.1 View Setting 90](#_Toc527923437)

[5.3.2 Update User Information 90](#_Toc527923438)

[5.4 House owner Interface Design 90](#_Toc527923439)

[5.4.1 Create Room 90](#_Toc527923440)

[5.4.2 Update Room 90](#_Toc527923441)

[5.4.3 View Room 90](#_Toc527923442)

[5.5 Room Master 91](#_Toc527923443)

[5.5.1 View suggested partner list 91](#_Toc527923444)

[5.5.2 View suggested room master list 91](#_Toc527923445)

[5.5.3 Create Post 91](#_Toc527923446)

[5.5.4 Update post 91](#_Toc527923447)

[5.6 Member 91](#_Toc527923448)

[5.6.1 View suggested room master list 91](#_Toc527923449)

[5.6.2 Create Post 91](#_Toc527923450)

[5.6.3 Update Post 91](#_Toc527923451)

[6. Database Design 91](#_Toc527923452)

[6.1 Entity relationship diagram (ERD) 91](#_Toc527923453)

[6.2 Entity dictionary 93](#_Toc527923454)

[7. Algorithms 93](#_Toc527923455)

[7.1 Definition 93](#_Toc527923456)

[7.2 Define Problem 93](#_Toc527923457)

[7.3 Solution 93](#_Toc527923458)

[7.4 Flowchart 93](#_Toc527923459)

# List of Table

[Table 1: Roles and Responsibilities 15](#_Toc527923460)

[Table 2: Hardware Requirements for Server 18](#_Toc527923461)

[Table 3: Hardware Requirement for Mobile 19](#_Toc527923462)

[Table 4: Software Requirements 19](#_Toc527923463)

[Table 5: Roles and Responsibilities 22](#_Toc527923464)

[Table 6: Tools and Techniques 22](#_Toc527923465)

[Table 7: <Unauthorized User> Register specification 31](#_Toc527923466)

[Table 8: <Unauthorized User> Login specification 33](#_Toc527923467)

[Table 9: <Authorized User> Logout specification 35](#_Toc527923468)

[Table 10: <Authorized User> View Notification specification 36](#_Toc527923469)

[Table 11: <Authorized User> Update Information specification 40](#_Toc527923470)

[Table 12: <House Owner> Create Room specification 42](#_Toc527923471)

[Table 13: <House owner> Update Room specification 46](#_Toc527923472)

[Table 14: <House owner> Delete Room specification 48](#_Toc527923473)

[Table 15: <House owner> Add Member into Room specification 52](#_Toc527923474)

[Table 16: <House owner> Delete Member from Room specification 54](#_Toc527923475)

[Table 17: <Room Master> Create Post specification 58](#_Toc527923476)

[Table 18: <Room Master> Update Post specification 62](#_Toc527923477)

[Table 19: <Room Master> Delete Post specification 64](#_Toc527923478)

[Table 20: <Member> Create partner Post specification 68](#_Toc527923479)

[Table 21: <Member> Update partner Post specification 71](#_Toc527923480)

[Table 22: Class Diagram Explanation 74](#_Toc527923481)

[Table 23: Admin login - Field, Buttons, Hyperlinks 89](#_Toc527923482)

[Table 24: Admin Manage User - Field, Buttons, Hyperlinks 89](#_Toc527923483)

[Table 25: Admin View Order List - Field, Buttons, Hyperlinks 89](#_Toc527923484)

[Table 26: Login - Field, Buttons, Hyperlinks 89](#_Toc527923485)

[Table 27: View Setting - Field, Buttons, Hyperlinks 90](#_Toc527923486)

[Table 28: Update User Information - Field, Buttons, Hyperlinks 90](#_Toc527923487)

[Table 29: Create Room - Field, Buttons, Hyperlinks 90](#_Toc527923488)

[Table 30: Update Room - Field, Buttons, Hyperlinks 90](#_Toc527923489)

[Table 31: View Room - Field, Buttons, Hyperlinks 91](#_Toc527923490)

[Table 32: Entity Data dictionary 93](#_Toc527923491)

# List of Figure

[Figure 1: Scrum Methodology 20](#_Toc527923492)

[Figure 2: System Overview Use Case (1) 28](#_Toc527923493)

[Figure 3: System Overview Use Case (2) 28](#_Toc527923494)

[Figure 4: <Unauthorized User> Overview Use Case 29](#_Toc527923495)

[Figure 5: <Unauthorized User> Register Use Case 29](#_Toc527923496)

[Figure 6: <Unauthorized User> Login Use Case 31](#_Toc527923497)

[Figure 7: <Authorized User> Overview Use Case 33](#_Toc527923498)

[Figure 8: <Authorized User> Logout Use Case 34](#_Toc527923499)

[Figure 9: <Authorized User> View Notification Use Case 35](#_Toc527923500)

[Figure 10: <Authorized User> Update Information Use Case 37](#_Toc527923501)

[Figure 11: <House Owner> Overview Use Case 40](#_Toc527923502)

[Figure 12: <House Owner> Create Room Use Case 40](#_Toc527923503)

[Figure 13: <House owner> Update Room Use Case 43](#_Toc527923504)

[Figure 14: <House owner> Delete Room Use Case 46](#_Toc527923505)

[Figure 15: <House owner> Add Member into Room Use Case 48](#_Toc527923506)

[Figure 16: <House owner> Delete Member from Room Use Case 52](#_Toc527923507)

[Figure 17: <Room Master> Overview Use Case 55](#_Toc527923508)

[Figure 18: < Room Master > Create Post Use Case 56](#_Toc527923509)

[Figure 19: < Room Master > Update Post Use Case 59](#_Toc527923510)

[Figure 20: < Room Master > Delete Post Use Case 62](#_Toc527923511)

[Figure 21: <Member> Overview Use Case 65](#_Toc527923512)

[Figure 22:< Member> Create partner Post Use Case 66](#_Toc527923513)

[Figure 23:< Member> Update partner Post Use Case 68](#_Toc527923514)

[Figure 24: Web Server Application Architure Design 73](#_Toc527923515)

[Figure 25: Mobile Application Architecture Design 74](#_Toc527923516)

[Figure 26: Component Diagram - Web Application 74](#_Toc527923517)

[Figure 27: Component Diagram - Mobile Application 74](#_Toc527923518)

[Figure 28: Class Diagram 74](#_Toc527923519)

[Figure 29: Sequence Diagram - <Unauthorized> Register 75](#_Toc527923520)

[Figure 30: Sequence Diagram - <Unauthorized> Login 76](#_Toc527923521)

[Figure 31: Sequence Diagram - <House owner> Create Room 77](#_Toc527923522)

[Figure 32: Sequence Diagram - <House owner> Update Room 79](#_Toc527923523)

[Figure 33: Sequence Diagram - <House owner> Delete Room 79](#_Toc527923524)

[Figure 34: Sequence Diagram - <House owner> View Single Room 80](#_Toc527923525)

[Figure 35: Sequence Diagram - <House owner> Add Member into Room 81](#_Toc527923526)

[Figure 36: Sequence Diagram - <House owner> Delete Members of Room 82](#_Toc527923527)

[Figure 37: Sequence Diagram - <Room Master> Create Room Post 83](#_Toc527923528)

[Figure 38: Sequence Diagram - < Room Master > Delete Room 84](#_Toc527923529)

[Figure 39: Sequence Diagram - <Room Master> Update Room Post 85](#_Toc527923530)

[Figure 40: Sequence Diagram - <Member> Create Partner Post 86](#_Toc527923531)

[Figure 41: Sequence Diagram - < Member > Update Partner Post 87](#_Toc527923532)

[Figure 42: Sequence Diagram - < Member > Delete Partner Post 88](#_Toc527923533)

[Figure 43: Sequence Diagram - <Authorized> Search Post 89](#_Toc527923534)

[Figure 44: Sequence Diagram - <Authorized> View Suggest List 89](#_Toc527923535)

[Figure 45: <Admin> Login Interface 89](#_Toc527923536)

[Figure 46: <Admin> Manage User Interface 89](#_Toc527923537)

[Figure 47: <Admin> View Room List Interface 89](#_Toc527923538)

[Figure 48: <Unauthorized User> Login Interface 89](#_Toc527923539)

[Figure 49: <Authorized User> View Setting Interface 90](#_Toc527923540)

[Figure 50: <Authorized User> Update Information Interface (1) 90](#_Toc527923541)

[Figure 51: <Authorized User> Update Information Interface (2) 90](#_Toc527923542)

[Figure 52: <House owner> Create Room Interface 90](#_Toc527923543)

[Figure 53: < House owner > Update Room Interface 90](#_Toc527923544)

[Figure 54: < House owner > View Room Interface 91](#_Toc527923545)

[Figure 55: Entity Relationship diagram 92](#_Toc527923546)

[Figure 56: Flowchart 93](#_Toc527923547)

# 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 support for suggesting places that is best match for 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 RoomMaster (people who are looking for roommates) in the same area move in together.
* Not support finding nearby RoomMaster.
* 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 RoomMaster with the Guest appropriately. Besides that, 2 RoomMaster can be suggested to move in if they are in the same area.

ASP system includes a mobile application for RoomMaster and Guest 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.
* RoomMaster can create and manage their own post.
* System matching RoomMaster with utilities and suggest appropriate room for guest.
* Guest can view information of room of lodger.
* RoomMaster and Guests can negotiate with message feature to send text or call.
* Mobile application (for house-owner)
* Manage rooms.
* Web application (for admin)
* Manage users.
* Manage posts.
* Manage rooms.

### Advantages and Disadvantages

* Advantages:
* Guests can easily find the place they want.
* Accommodation will be verified by staff.
* House-owner can manage their room
* 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.
* Notification.
* Logout.
* Mobile application (for RoomMaster)
* View suggested partner list.
* Manage post.
* View partner list.
* View post.
* Mobile application (for guest)
* View suggest RoomMaster posts.
* Manage information.
* Manage utilities.
* Web application (for admin)
* View and manage user.
* View and manage rooms.
* View and manage posts.

## 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 | Baonqse61774@fpt.edu.vn |
|  | Nguyễn Đỗ Minh Đức | Developer | Member | Ducndmse61767@fpt.edu.vn |
| 3 | Vương Minh Thông | Developer | Member | Hungndse61819@fpt.edu.vn |
| 5 | Hồ Công Trình | Developer | Member | Vuttse60983@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 for studying and working . Beside 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, roommaster and guest 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 rommaster, guest, house-owner.

##### Mobile Application

**- House-owner:**

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

**- RoomMaster:**

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

**- Guest:**

* Guest can create utilities and system can match their utilities with location roommaster go to make a suggestion.
* Guest can manage their own information, their utilities.

##### Web Application

**- Admin:**

* Admin manages user informations, rooms, posts.

#### 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 roommaster search for partner.
* Allow guest search for rommaster.
* Matching post and utilities then suggest to guest.
* 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.
* The application can not scrap data from all e-commerce website.

#### Future Plan

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

* Mobile application will be available on Android OS.
* Application can scrap data from more e-commerce website.

#### Development Environment

##### Hardware requirements

**For Server**

|  |  |  |
| --- | --- | --- |
| Windows | Minimum Requirements | Recommend Requirements |
| OS | Window Server 2008 | Window Server 2012 |
| Internet Connection | Cable, Wifi (8 Mbps) | Cable, Wifi (16 Mbps) |
| Processor | Intel® Xeon ® 1.4GHz | Intel® Xeon ® Quad-Core (12M Cache, 2.50 GHz) |
| RAM | 1GB | 2GB |
| I/O Speed | 120MB/s | 350MB/s |

Table 2: Hardware Requirements for Server

**For Mobile**

|  |  |  |
| --- | --- | --- |
| Mobile | Minimum Requirements | Recommend Requirements |
| OS | IOS 10.2 | IOS 11.4 |
| Internet Connection | 2 Mbps | 4 Mbps |
| Memory | N/A | N/A |

Table 3: Hardware Requirement for Mobile

##### Software requirements

|  |  |  |
| --- | --- | --- |
| Software | Name/Version | Description |
| Environment | Windows 7 or above | Window 10 |
| Modeling Tool | StarUML v2.0 |  |
| IDE | Intellij Idea, X-Code 9.0 |  |
| DBMS | Microsoft SQL Server 2014 |  |
| Source Control | Github |  |
| Web Browser | Chrome |  |

Table 4: Software Requirements

## Project Organization

### Software Process Model

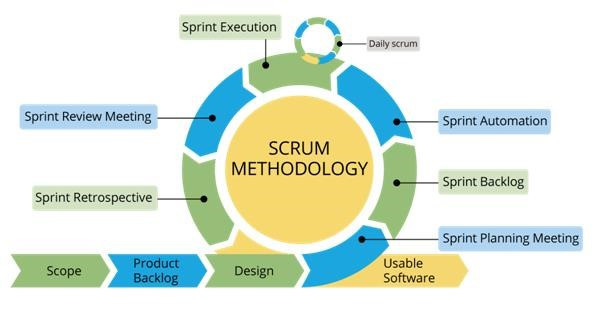


Figure 1: Scrum Methodology

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.

### Roles and Responsibilities

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Full name** | **Role** | **Responsibilities** |
| 1 | Nguyễn Huy Hùng | Project Manager | - Specify user requirement  - Control development process  - Give out technique and business analysis support  - Review document and product |
| 2 | Nguyễn Quang Nhật | Team Leader, BA, Developer | - Managing process  - Planning project  - Distribute ASPks  - Designing database  - Clarifying requirements  - Prepare document  - Create test plan  - Coding  - Testing  - Deploy product |
| 3 | Nguyễn Đỗ Minh Đức | Team Member, BA, Developer, Tester | - Designing database  - Clarifying requirements  - Prepare document  - Create test plan  - Coding  - Testing |
| 4 | Vương Minh Thông | Team Member, BA, Developer, Tester | - Designing database  - Clarifying requirements  - Prepare document  - GUI design  - Create test plan  - Coding  - Testing |
| 5 | Hồ Công Trình | Team Member, BA, Developer, Tester | Designing database  - Clarifying requirements  - Prepare document  - GUI design  - Create test plan  - Coding  - Testing |

Table 5: Roles and Responsibilities

### Tools and Techniques

|  |  |  |
| --- | --- | --- |
|  | **Tools** | **Techniques** |
| Front-end | Sketch, X-Code 9.0 |  |
| Back-end | Intellij |  |
| Web server | IIS 10 |  |
| Application | X-Code 9.0 |  |
| DBMS | MYSQL Workbench 6.3 |  |
| Source control | Github |  |
| Modeling tool | StarUML v2.0 |  |
| Document tool | Microsoft Word 2013 |  |

Table 6: Tools and Techniques

## Project Management Plan

### Product Backlog

All product backlogs could be found here:

### Sprint Backlog

All print backlogs could be found here:

### Meeting Minutes

All meeting documents could be found here:

## Coding Convention

We use followings naming convention from the reference to capable with the current situation in our team:

* Naming Convention:
* Follows camelcase syntax for naming the class, interface, method and variable.
* Method names should be verbs.
* Class names should be nouns, in mixed case with first letter of each internal word capitalized.
* Constant names should be all uppercase with words separated by underscore.
* Comment:
* Using /\* \*/ for block comments.
* Using // for line comments.
* Follow Field Naming Conventions
* Other fields start with a lowercase letter.
* Public static final fields (constants) are ALL\_CAPS\_WITH\_UNDERSCORES.

Using Java coding convention from:

<http://www.oracle.com/technetwork/java/codeconvtoc-136057.html>

# C. Software Requirement Specification

## User Requirement Specification

### Unauthorized User Requirement

Unauthorized user is a person who doesn’t access to the system. Unauthorized user can use some functions in the system, to use all functions, they must login. These are one function they can use:

* Register
* Login

### Authorized User Requirement

Authorized user is a person after they login in the system and they will separate into House owner, Room master, Member depend on what they want to do in our application. Authorized user can do some functions, such as:

* Manage profile
* View notification
* Logout

### House Owner Requirement

House owner is authorized user which want to post their room in room to room master can create a finding roommate post.

House owner can do following functions:

* Manage room
* Add room member
* Remove room member

### Room Master Requirement

Room master is authorized user which want to try to looking for roommate and make deal with them.

Room master can do following functions:

* Manage finding roommate post
* View suggested partner list
* View partner list

### Member Requirement

Member is authorized user which want to try to looking for room and make deal with room-master.

Member can do following functions:

* Search
* Manage partner post
* View post list
* View suggested post list
* Negotiates

### Admin Requirement

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

* Search
* Manage room
* Manage finding room post
* Manage finding roommate post
* Manage user
* 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 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

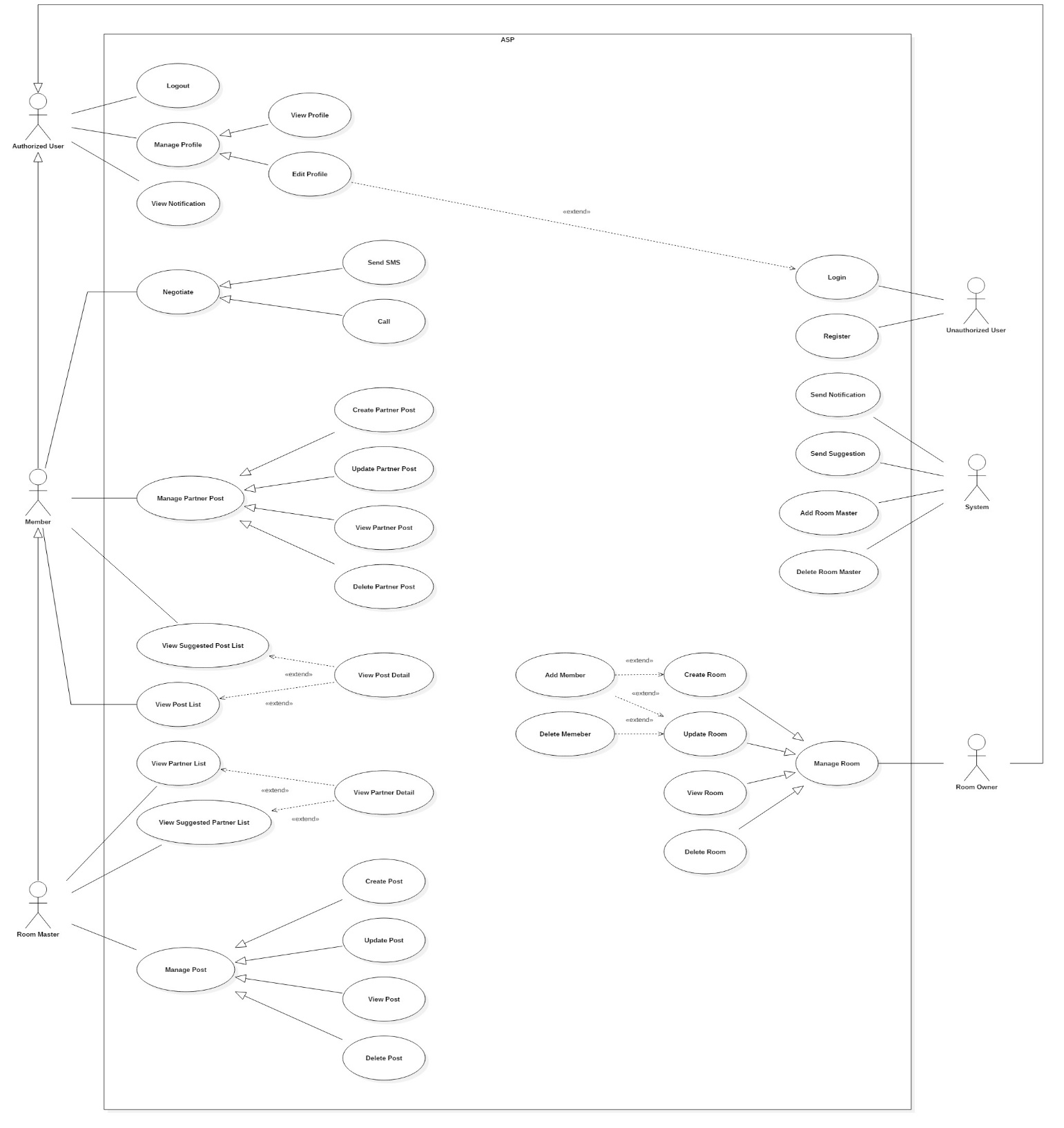


Figure 2: System Overview Use Case (1)

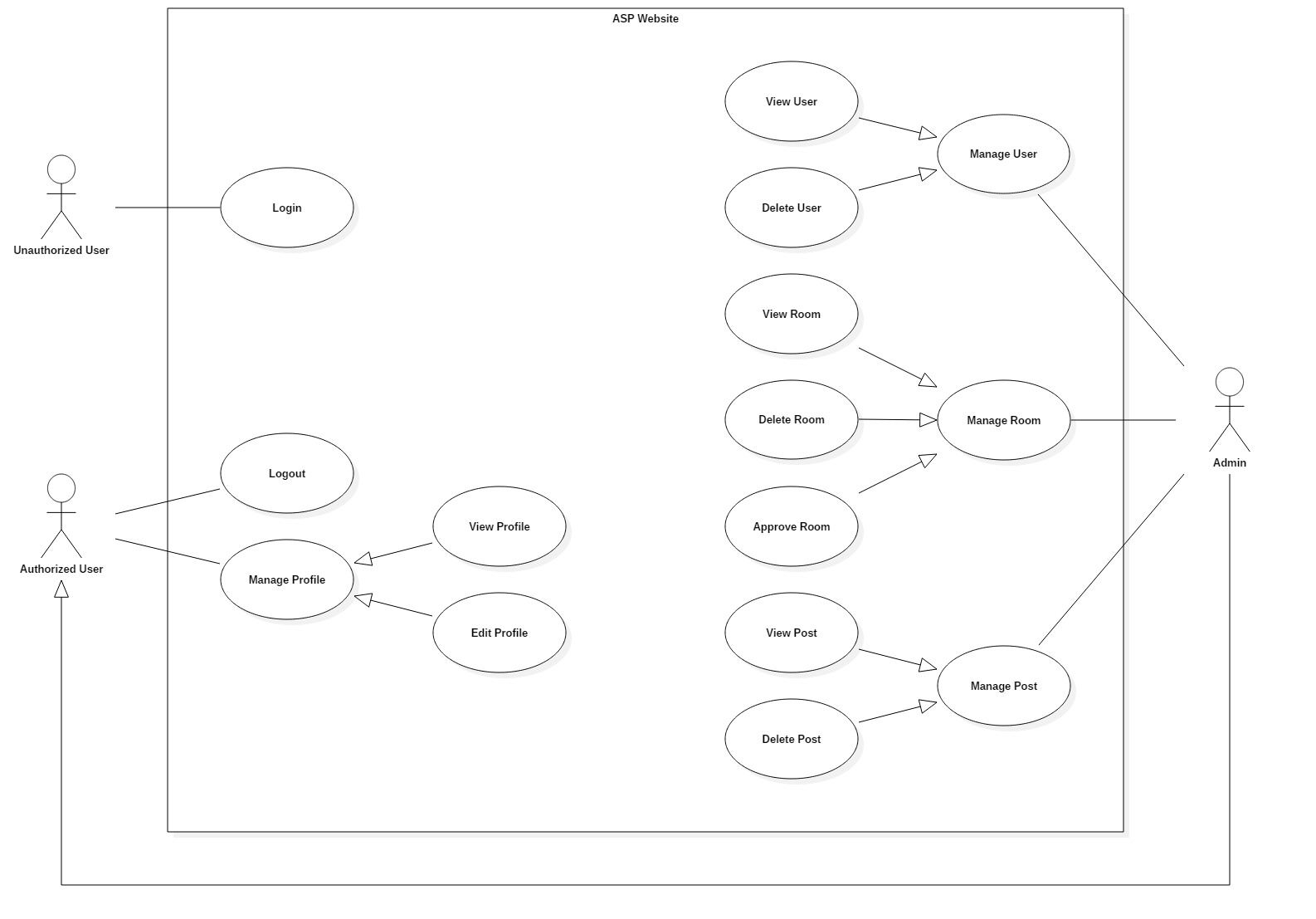


Figure 3: System Overview Use Case (2)

### List of Use Case

#### <Unauthorized User> Overview Use Case

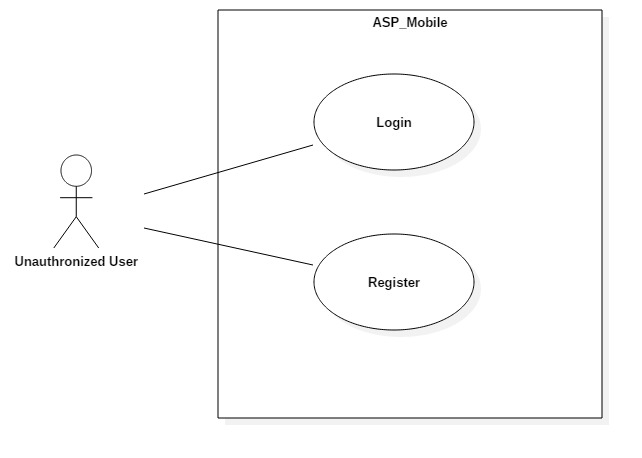


Figure 4: <Unauthorized User> Overview Use Case

##### <Unauthorized User> Register

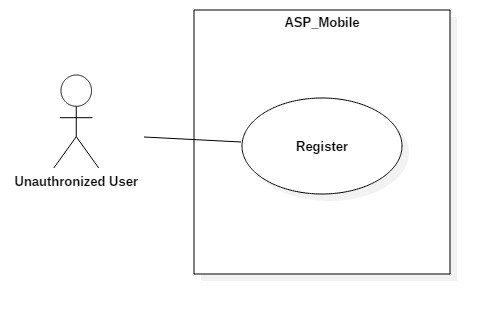


Figure 5: <Unauthorized User> Register Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP1** | | | |
| **Use Case No.** | UC\_ ASP1 | **Use Case Version** | 0.2 |
| **Use Case Name** | Register | | |
| **Author** | Vương Minh Thông | | |
| **Date** | 17/10/2018 | **Priority** | High |
| **Actor:**   * Unauthorized User   **Summary:**   * This use case allows unauthorized user to create account.   **Goal:**   * To create account then login to the system.   **Triggers:**   * Actor sends register command.   **Preconditions:** N/A  **Post conditions:**   * Success: Actor accesses to system successfully. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Actor goes to register screen | System requires information:   * “Tên tài khoản”: Text input * “Mật khẩu”: Text input * “Email”: Text input * “Giới tính”: Switch button * “Số điện thoại”: Text input * “Ngày sinh”: Date time picker * “Ảnh đại diện”: Image upload * Register button. | | 2 | Actor inputs informations | [Exception no.1] | | 3 | Actor sends command to login  (Click “Register” button after input login form). | [Exception no.2]  System navigates to home screen. |   **Alternative Scenario:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | 1 | Actor inputs wrong information. | System shows message:“Thông tin bạn nhập không hợp lệ” | | 2 | Actor left blank to required fields. | System shows message: “Vui lòng điền đầy đủ thông tin”. |   **Relationships:** N/A  **Business Rules:**   * User have not logged in to system. * Username must be in range of 5-35 characters * Password must be in range of 4-30 characters. * Password must be encrypted before being sent to server. | | | |

Table 7: <Unauthorized User> Register specification

##### <Unauthorized User> Login

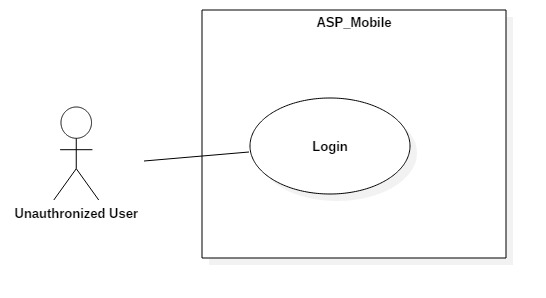


Figure 6: <Unauthorized User> Login Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP2** | | | |
| **Use Case No.** | UC\_ ASP2 | **Use Case Version** | 0.2 |
| **Use Case Name** | Login | | |
| **Author** | Vương Minh Thông | | |
| **Date** | 17/10/2018 | **Priority** | High |
| **Actor:**   * Unauthorized User   **Summary:**   * This use case allows unauthorized user to login to system with username and password.   **Goal:**   * To login to the system.   **Triggers:**   * Actor sends login command.   **Preconditions:** N/A  **Post conditions:**   * Success: Actor accesses to system successfully. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Actor goes to login screen | System requires information:   * “Username”: Text input * “Password”: Text input * Login button. | | 2 | Actor inputs information  (username and password) | [Exception no.1] | | 3 | Actor sends command to login  (Click “Login” button after input login form). | [Exception no.2]  System navigates to home screen. |   **Alternative Scenario:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | 1 | Actor inputs wrong information. | System shows message:“Thông tin đăng nhập không hợp lệ” | | 2 | Actor left blank to required fields. | System shows message: “Vui lòng điền đầy đủ thông tin”. |   **Relationships:** N/A  **Business Rules:**   * User have not logged in to system. * Username must be in range of 5-35 characters * Password must be in range of 4-30 characters. * Password must be encrypted before being sent to server. | | | |

Table 8: <Unauthorized User> Login specification

#### <Authorized User> Overview Use

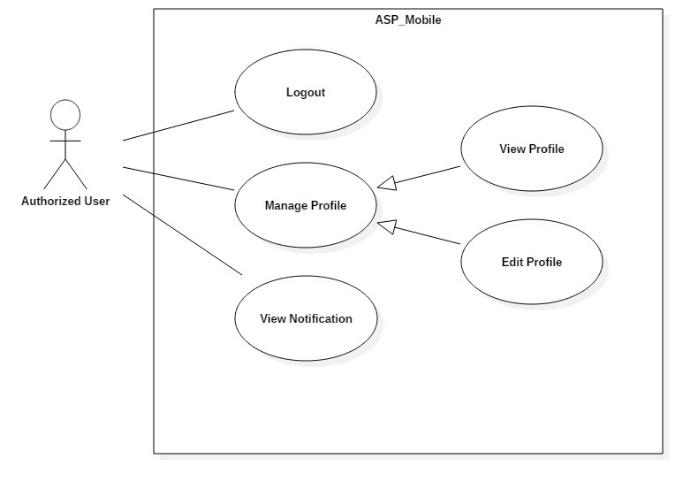


Figure 7: <Authorized User> Overview Use Case

##### <Authorized User> Logout

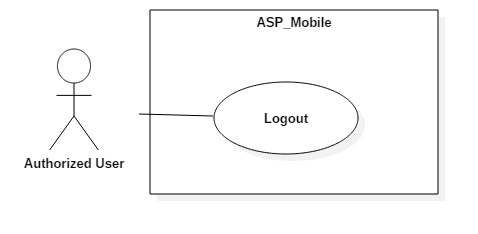


Figure 8: <Authorized User> Logout Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP3** | | | |
| **Use Case No.** | UC\_ ASP3 | **Use Case Version** | 0.2 |
| **Use Case Name** | Logout | | |
| **Author** | Vương Minh Thông | | |
| **Date** | 17/10/2018 | **Priority** | High |
| **Actor:**   * Authorized User   **Summary:**   * This use case allows authorized user signs out from the system.   **Goal:**   * Authenticated user stops accessing to the system. * User’s session is removed from the system.   **Triggers:**   * Actor sends signs out command.   **Preconditions:**   * User logged in the system.   **Post conditions:**   * Success: User session is removed, and system displays login screen. * Fail: System displays error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Actor press “Cài Đặt” button. | User navigated to setting screen | | 2 | Actor press “Đăng xuất” button. | User session is removed, and system displays login screen. |   **Alternative Scenario:**N/A  **Exceptions:** N/A  **Relationships:** N/A  **Business Rules:**   * Actor has already accessed to the system. * After signing out, Authorized User will become “Unauthorized User” | | | |

Table 9: <Authorized User> Logout specification

##### <Authorized User> View Notification

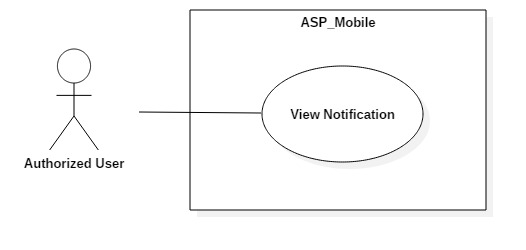


Figure 9: <Authorized User> View Notification Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP4** | | | |
| **Use Case No.** | UC\_ ASP4 | **Use Case Version** | 0.2 |
| **Use Case Name** | View Notification | | |
| **Author** | Vương Minh Thông | | |
| **Date** | 17/10/2018 | **Priority** | Medium |
| **Actor:**   * Authorized User   **Summary:**   * This use case allows user to view notification.   **Goal:**   * To view notification.   **Triggers:**   * Actor sends notification command.   **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 press on “Thông báo” button. | System navigates to notification screen with list of notifications. |   **Alternative Scenario:**N/A  **Exceptions:** N/A  **Relationships:** N/A  **Business Rules:**   * Actor has already accessed to the system. | | | |

Table 10: <Authorized User> View Notification specification

##### <Authorized User> Update Info

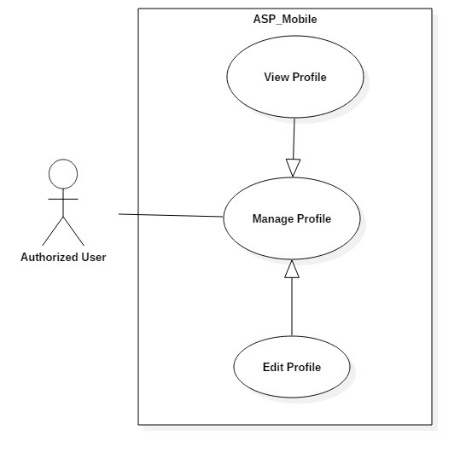


Figure 10: <Authorized User> Update Information Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP5** | | | |
| **Use Case No.** | UC\_ ASP5 | **Use Case Version** | 0.2 |
| **Use Case Name** | Update User Info | | |
| **Author** | Vương Minh Thông | | |
| **Date** | 22/7/2018 | **Priority** | High |
| **Actor:**   * Authorized User   **Summary:**   * This use case allows authorized user to update their own information.   **Goal:**   * To update information of user.   **Triggers:**   * Actor sends update user info command.   **Preconditions:**   * User logged in the system.   **Post conditions:**   * Success: System navigates back to user information screen and displays new information of the current user. * Fail: System displays error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Actor press “Cài đặt” button. | System navigates to setting screen. | | 2 | Actor press on current user name section. | System displays profile screen with:   * “Ảnh đại diện”: Image Upload * “Họ và tên”: Text input * “Email”: Text input * “Số điện thoại”: Text input * “Địa chỉ”: Text input * “Giới thiệu bản thân”: Text input * “Tiện ích”: Multiple choose button | | 3 | Actor press “Lưu” button. | [Exception no.1]  [Exception no.2]  User info is updated successfully and system refreshes the profile 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:  - “Email”: System displays warning message: “Email không được để trống”. | | 2 | User input wrong some fields with requirement. | System notices that guest need to re-input all these field:  - “Họ và tên”: System displays warning message: “Không được sử dụng ký tự đặc biệt !@#$%^&\*()-=\_+...”.  - “Email”: System displays warning message: “Sai định dạng, thiếu @”  - “Số điện thoại”: System displays warning message: “Không được sử dụng chữ hoặc ký tự đặc biệt !@#$%^&\*()-=\_+...”. |   **Relationships:** N/A  **Business Rules:**   * Actor has already accessed to the system. * Name is not null and do not use special character. * Email can not be wrong format or not contained @ character. * Phone can not be alphabet character. | | | |

Table 11: <Authorized User> Update Information specification

#### <House Owner> Overview Use Case

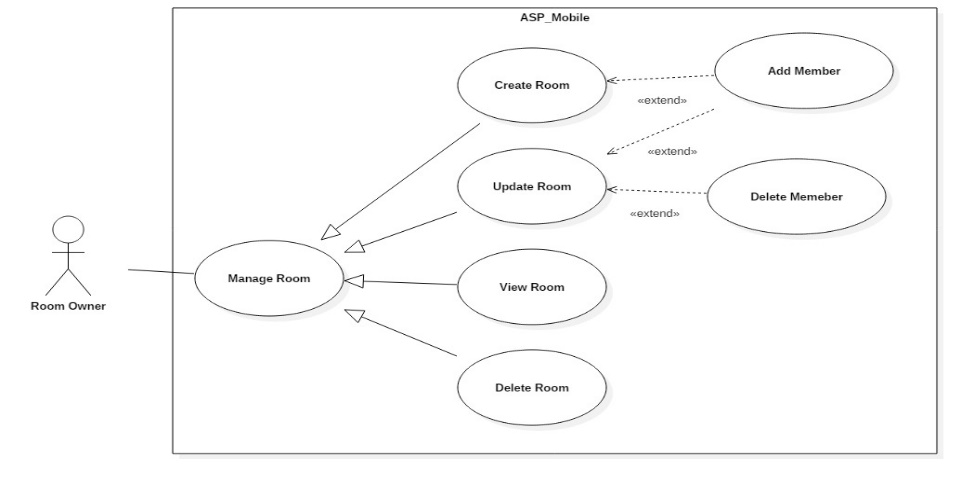


Figure 11: <House Owner> Overview Use Case

##### <House Owner> Create Room

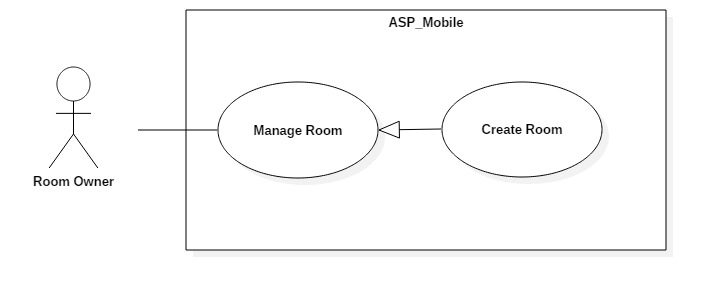


Figure 12: <House Owner> Create Room Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP6** | | | |
| **Use Case No.** | UC\_ ASP6 | **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:**   * Actor sends create room command.   **Preconditions:**   * User logged in the system.   **Post conditions:**   * Success: User can create new room. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Actor press on “Tạo phòng” button. | System displays create room screen with:  - “Tên Phòng”: Text input.  - “Giá”: Text input.  - “Diện tích”: Text input.  - “Địa chỉ”: Text input.  - “Số khách tối đa”: Text input.  - “Mô tả”: Text input.  - “Thêm thành viên vào phòng” button  - “Tạo phòng” button | | 3 | Actor press on “Tạo phòng” 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 is not null. * Room price is not null. * Address is not null. | | | |

Table 12: <House Owner> Create Room specification

##### <House owner> Update Room

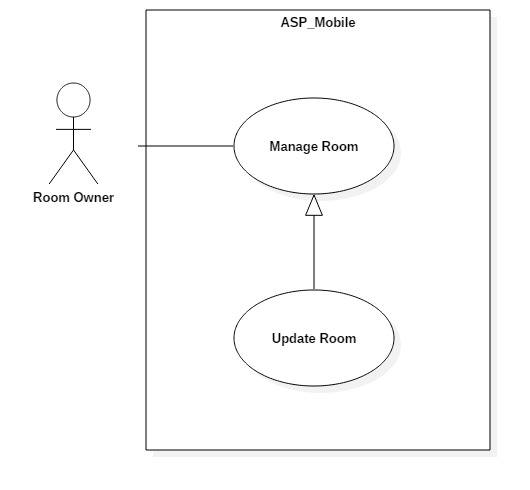


Figure 13: <House owner> Update Room Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP7** | | | |
| **Use Case No.** | UC\_ ASP7 | **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 sends update room command.   **Preconditions:**   * User logged in the system. * Rooms were 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 press “Tài khoản” button. | System navigates to “Tải khoản” screen.  System displays “Tài khoản” screen with:  - “Đã đăng” tab. | | 2 | Actor press 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”  - “Thành viên trong phòng” table  - “Chỉnh sửa” button  - “Xóa” button | | 5 | Actor press on “Chỉnh sửa” button. | Systems navigate to update room screen.  System displays update screen with:  - “Hình ảnh”  - “Tên phòng”  - “Giá phòng”  - “Diện tích”  - “Địa chỉ”  - “Ngày đăng bài”  - “Mô tả”  - “Thành viên trong phòng” table  - “Thêm thành viên vào phòng” button  - “Lưu” button | | 6 | Actor press on “Lưu” button. | [Exception no.1]  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 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. * Room name is not null. * Room price is not null. * Address is not null. | | | |

Table 13: <House owner> Update Room specification

##### <House owner> Delete Room

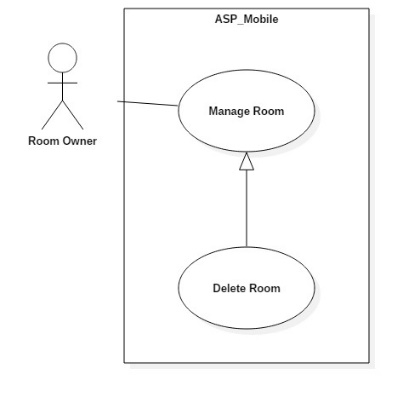


Figure 14: <House owner> Delete Room Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP8** | | | |
| **Use Case No.** | UC\_ ASP8 | **Use Case Version** | 0.2 |
| **Use Case Name** | Delete Room | | |
| **Author** | Vương Minh Thông | | |
| **Date** | 17/10/2018 | **Priority** | High |
| **Actor:**   * House owner   **Summary:**   * This use case allows House owner delete room.   **Goal:**   * To delete room.   **Triggers:**   * Actor sends delete room command.   **Preconditions:**   * User logged in the system. * Rooms were uploaded.   **Post conditions:**   * Success: User can delete room. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Actor press “Tài khoản” button. | System navigates to “Tải khoản” screen.  System displays “Tài khoản” screen with:  - “Đã đăng” tab. | | 2 | Actor press 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 | | 5 | Actor press on “Xoá” button. | Room info is deleted successfully and system refreshes the “Đã đăng” screen with new information. |   **Relationships:** N/A  **Business Rules:**   * User already logged in the system. | | | |

Table 14: <House owner> Delete Room specification

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

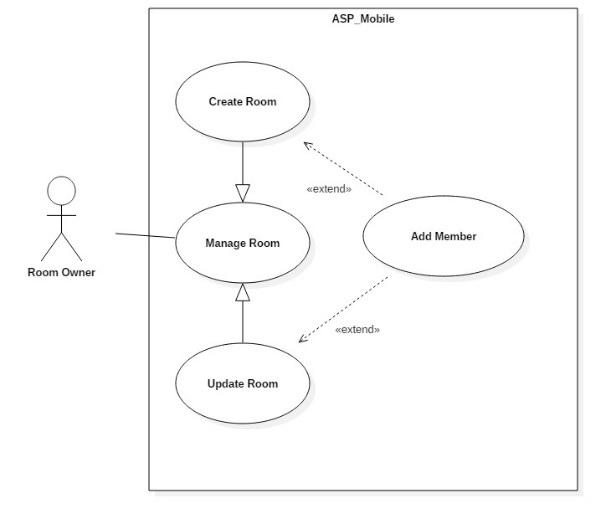


Figure 15: <House owner> Add Member into Room Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP9** | | | |
| **Use Case No.** | UC\_ ASP9 | **Use Case Version** | 0.2 |
| **Use Case Name** | Add Member into Room | | |
| **Author** | Vương Minh Thông | | |
| **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 sends Add Member into Room command.   **Preconditions:**   * User logged in the system.   **Post conditions:**   * Success: User can add Member into Room. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Actor press “Tài khoản” button. | System navigates to “Tải khoản” screen.  System displays “Tài khoản” screen with:  - “Đã đăng” tab. | | 2 | Actor press 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 | | 5 | Actor press on “Chỉnh sửa” button. | Systems navigate to update room screen.  System displays update screen with:  - “Hình ảnh”  - “Tên phòng”  - “Giá phòng”  - “Diện tích”  - “Địa chỉ”  - “Ngày đăng bài”  - “Mô tả”  - “Thành viên trong phòg” table  - “Thêm thành viên vào phòng” button  - “Lưu” button | | 6 | Actor press on “Thêm thành viên vào phòng” button. | Systems navigate to add member into room screen.  System displays add member into room screen with:  - “Tài khoản”: Text input  - “Ngày thuê”: Date time picker  - “Ngày trả phòng”: Date time picker  - “Thêm” button. | | 6 | Actor press on “Thêm” 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”.  - “Ngày thuê”: System displays warning message: “Ngày thuê không được để trống”. | | 2 | User input wrong 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”.  - “Ngày thuê”: System displays warning message: “Ngày thuê không được sớm hơn ngày đăng phòng”. |   **Relationships:** Extend Create/Update room  **Business Rules:**   * User already logged in the system. * Room name is not null. * Room price is not null. * Address is not null. | | | |

Table 15: <House owner> Add Member into Room specification

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

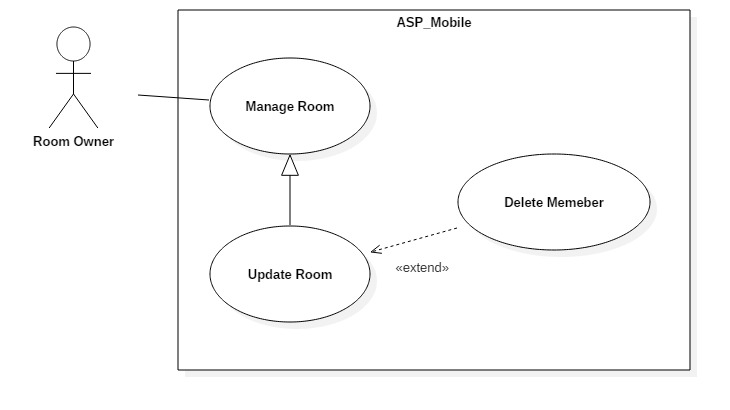


Figure 16: <House owner> Delete Member from Room Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP10** | | | |
| **Use Case No.** | UC\_ ASP10 | **Use Case Version** | 0.2 |
| **Use Case Name** | Delete Member from Room | | |
| **Author** | Vương Minh Thông | | |
| **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 sends delete member from room command.   **Preconditions:**   * User logged in the system. * Member already added into room.   **Post conditions:**   * Success: User can delete member from room. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Actor press “Tài khoản” button. | System navigates to “Tải khoản” screen.  System displays “Tài khoản” screen with:  - “Đã đăng” tab. | | 2 | Actor press 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 | | 5 | Actor press on “Chỉnh sửa” button. | Systems navigate to update room screen.  System displays update screen with:  - “Hình ảnh”  - “Tên phòng”  - “Giá phòng”  - “Diện tích”  - “Địa chỉ”  - “Ngày đăng bài”  - “Mô tả”  - “Thành viên trong phòng” table  - “Thêm thành viên vào phòng” button  - “Lưu” button | | 6 | Actor choose 1 member of room and press “Xoá” button | Systems display popup “Xoá thành công”.  System displays the rest of members of the room. | | 6 | Actor press on “Lưu” 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. | | | |

Table 16: <House owner> Delete Member from Room specification

#### <Room Master> Overview Use Case

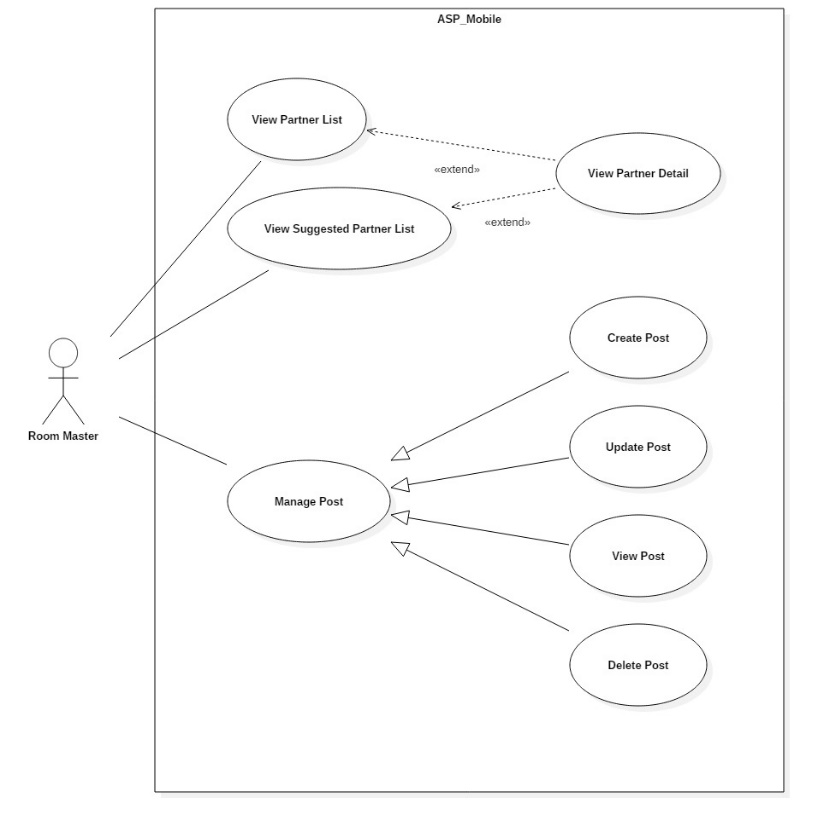


Figure 17: <Room Master> Overview Use Case

##### < Room Master > View Suggested Lis

##### < Room Master > Create Post

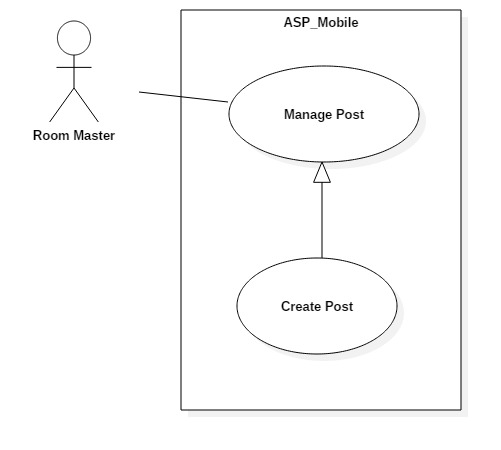


Figure 18: < Room Master > Create Post Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP11** | | | |
| **Use Case No.** | UC\_ ASP11 | **Use Case Version** | 0.2 |
| **Use Case Name** | Create Room | | |
| **Author** | Vương Minh Thông | | |
| **Date** | 17/10/2018 | **Priority** | High |
| **Actor:**   * Room Master   **Summary:**   * This use case allows room master to create new post.   **Goal:**   * To create new post.   **Triggers:**   * Actor sends create post command.   **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 “Tìm ở ghép” button. | System displays create post screen with:  - “Tên bài”: Text input.  - “Giá”: Text input.  - “Diện tích”  - “Địa chỉ”  - “Số bạn ở ghép”: Text input  - “Mô tả”: Text input.  - “Giới tính” switch button  - “Đăng bài” button | | 3 | Actor press on “Đăng bài” 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 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 some fields with requirement. | System notices that guest 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. | | | |

Table 17: <Room Master> Create Post specification

##### < Room Master > Update Post

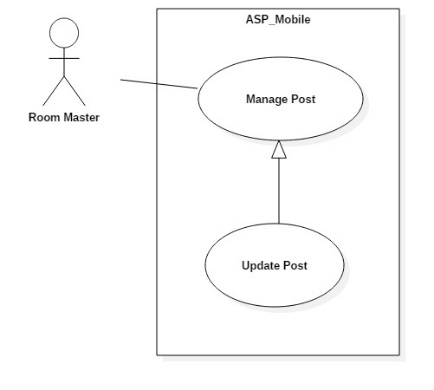


Figure 19: < Room Master > Update Post Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP12** | | | |
| **Use Case No.** | UC\_ ASP12 | **Use Case Version** | 0.2 |
| **Use Case Name** | Update Room | | |
| **Author** | Vương Minh Thông | | |
| **Date** | 17/10/2018 | **Priority** | High |
| **Actor:**   * Room Master   **Summary:**   * This use case allows room master to update post.   **Goal:**   * To update post.   **Triggers:**   * Actor sends update post command.   **Preconditions:**   * User logged in the system. * Post was uploaded.   **Post conditions:**   * Success: User can update post. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Actor press “Tài khoản” button. | System navigates to “Tải khoản” screen.  System displays “Tài khoản” screen with:  - “Đã đăng” tab. | | 2 | Actor press on certain post. | System will navigate to this post screen with common detail.  - “Hình ảnh”  - “Tên phòng”  - “Giá phòng”  - “Diện tích”  - “Địa chỉ”  - “Mô tả”  - “Thành viên trong phòng” table  - “Chỉnh sửa” button  - “Xóa” button | | 5 | Actor press on “Chỉnh sửa” button. | Systems navigate to update room screen.  System displays update screen with:  - “Hình ảnh”  - “Tên phòng”  - “Giá phòng”  - “Diện tích”  - “Địa chỉ”  - “Ngày đăng bài”  - “Mô tả”  - “Lưu” button | | 6 | Actor press on “Lưu” 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 some fields with requirement. | System notices that guest 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. | | | |

Table 18: <Room Master> Update Post specification

##### < Room Master > Delete Post

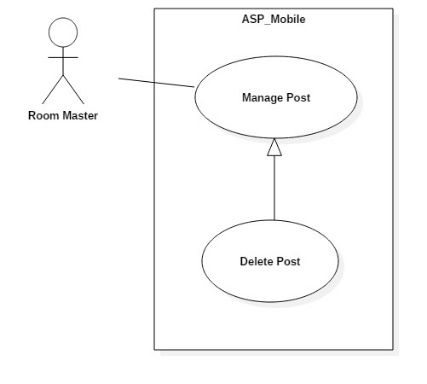


Figure 20: < Room Master > Delete Post Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP13** | | | |
| **Use Case No.** | UC\_ ASP13 | **Use Case Version** | 0.2 |
| **Use Case Name** | Delete Post | | |
| **Author** | Vương Minh Thông | | |
| **Date** | 17/10/2018 | **Priority** | High |
| **Actor:**   * Room Master   **Summary:**   * This use case allows room master to delete post.   **Goal:**   * To delete post.   **Triggers:**   * Actor sends delete post command.   **Preconditions:**   * User logged in the system. * Post was uploaded.   **Post conditions:**   * Success: User can delete post. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Actor press “Tài khoản” button. | System navigates to “Tải khoản” screen.  System displays “Tài khoản” screen with:  - “Đã đăng” tab. | | 2 | Actor press on certain post. | System will navigate to this post screen with common detail.  - “Hình ảnh”  - “Tên phòng”  - “Giá phòng”  - “Diện tích”  - “Địa chỉ”  - “Mô tả”  - “Thành viên trong phòng” table  - “Chỉnh sửa” button  - “Xóa” button | | 6 | Actor press on “Xoá” button. | Post info is deleted successfully and system refreshes the “Đã đăng” screen with new information. |   **Alternative Scenario:** N/A  **Relationships:** N/A  **Business Rules:**   * User already logged in the system. | | | |

Table 19: <Room Master> Delete Post specification

#### <Member> Overview Use Case



Figure 21: <Member> Overview Use Case

##### <Member> View Suggested Room Post list

##### <Member> Create partner Post

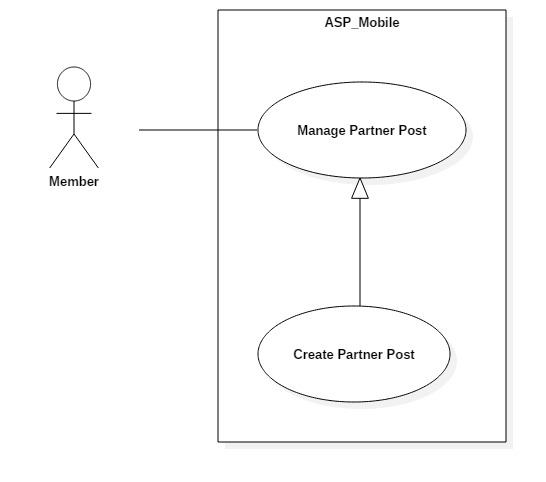


Figure 22:< Member> Create partner Post Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP15** | | | |
| **Use Case No.** | UC\_ ASP15 | **Use Case Version** | 0.2 |
| **Use Case Name** | Create partner Post | | |
| **Author** | Vương Minh Thông | | |
| **Date** | 17/10/2018 | **Priority** | High |
| **Actor:**   * Member   **Summary:**   * This use case allows member to create finding room post.   **Goal:**   * To create new post.   **Triggers:**   * Actor sends create partner 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 press on “Tạo bài tìm phòng” button. | System displays create post screen with:  - “Tên bài”: Text input.  - “Khoảng Giá”: Slide choose.  - “Diện tích”  - “Quận”: Multi choose button  - “Mô tả”: Text input.  - “Giới tính” switch button  - “Đăng bài” button | | 3 | Actor press on “Đăng bài” button. | [Exception no.1]  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 some fields with requirement. | System notices that guest need to re-input all these field:  - “Giá”: System displays warning message: “Không được sử dụng chữ hoặc ký tự đặc biệt !@#$%^&\*()-=\_+...”.  - “Diện tích”: System displays warning message: “Không được sử dụng chữ hoặc ký tự đặc biệt !@#$%^&\*()-=\_+...”. |   **Relationships:** N/A  **Business Rules:**   * User already logged in the system. | | | |

Table 20: <Member> Create partner Post specification

##### <Member> Update partner Post

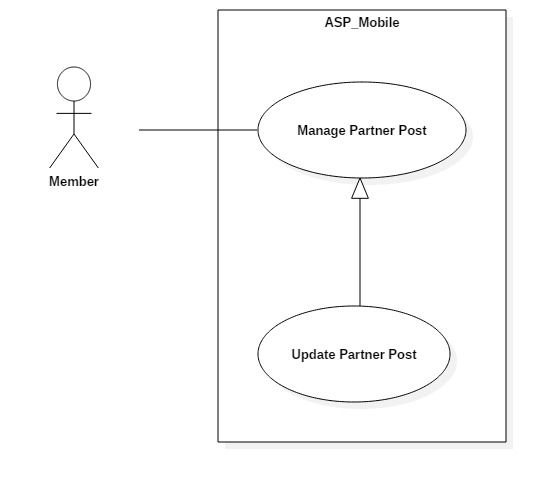


Figure 23:< Member> Update partner Post Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP16** | | | |
| **Use Case No.** | UC\_ ASP16 | **Use Case Version** | 0.2 |
| **Use Case Name** | Update partner Post | | |
| **Author** | Vương Minh Thông | | |
| **Date** | 17/10/2018 | **Priority** | High |
| **Actor:**   * Member   **Summary:**   * This use case allows member to update finding room post.   **Goal:**   * To update partner post.   **Triggers:**   * Actor sends update partner post command.   **Preconditions:**   * User logged in the system. * Post already uploaded.   **Post conditions:**   * Success: User can update partner post. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Actor press “Tài khoản” button. | System navigates to “Tải khoản” screen.  System displays “Tài khoản” screen with:  - “Đã đăng” tab. | | 2 | Actor press on certain post. | System will navigate to this post screen with common detail.  - “Tên bài đăng”  - “Khoảng giá”  - “Diện tích”  - “Mô tả”  - “Quận”  - “Chỉnh sửa” button  - “Xóa” button | | 5 | Actor press on “Chỉnh sửa” button. | Systems navigate to update partner post screen.  System displays update screen with:  - “Tên bài đăng”  - “Khoảng giá”  - “Diện tích”  - “Mô tả”  - “Quận”  - “Lưu” button | | 6 | Actor press on “Lưu” button. | [Exception no.1]  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 input wrong some fields with requirement. | System notices that guest need to re-input all these field:  - “Diện tích”: System displays warning message: “Không được sử dụng chữ hoặc ký tự đặc biệt !@#$%^&\*()-=\_+...”. |   **Relationships:** N/A  **Business Rules:**   * User already logged in the system. | | | |

Table 21: <Member> Update partner Post specification

##### <Member> Delete partner Post

##### <Member> Negotiate

##### <Member> Search Post

#### <Admin> Overview Use Case

##### <Admin> Manage Users

##### <Admin> Manage Rooms

##### <Admin> Manage Post

##### <Admin> Approve Room

## Software System Attribute

### Usability

#### Graphic User Interface

* All the texts, labels, alerts and messages will be written in Vietnamese.
* GUI for mobile application is designed base on material design with Navigation Bar at the top contains the main functions of the application.

#### Usability

* The system usability is easy to use that users generally don’t need to spend too much time looking around.
* Icons that indicate the actions should be easy to understand and users will not meet any troubles to recognize the feature of screen.

### Reliability

* The data should be backed up every day.

### Availability

* The IOS application must be available 24/7.
* System reply in maximum 3 seconds.

### Security

* Users is authentication/authorization for all users when they login to the system.

### Maintainability

* The system is divided into separated modules for easy maintain.

### Portability

* The software is a mobile application so it can be access anywhere with internet connection.
* User can use the mobile application running IOS 10 or above.
* Web application can be run on Chrome browser version 42 or later.

### Performance

* Requests from users are responded in less than 10 seconds at 5 Mbps bandwidth speed.
* System can handle 100 requests at one time.
* If the system crash, it shall ensure security less than 30 minutes and fixing hardware less than 2 days.

## Conceptual diagram

# D. Software Design Description

## Design Overview

* This document describes the technical and user interface design of **ASP**. It includes the architectural design, the detailed design of common functions and business functions and the design of database model.
* The architectural design describes the overall architecture of the system and the architecture of each main component and subsystem.
* The detailed design describes static and dynamic structure for each component and functions. It includes class diagrams, class explanations and sequence diagrams for each use cases.
* The database design describes the relationships between entities and details of each entity.
* Document overview:
* Section 2: gives an overall description of the system architecture design.
* Section 3: gives component diagrams that describe the connection and integration of the system.
* Section 4: gives the detail design description which includes class diagram, class explanation, and sequence diagram to details the application functions.
* Section 5: describe screens design.
* Section 6: describe a fully attributed ERD.
* Section 7: describe algorithms.

## System Architecture Design

### Web service Application Architecture Description

Figure 24: Web Server Application Architure Design

### Mobile Application Architecture Description

Figure 25: Mobile Application Architecture Design

## Component Diagram

#### Web Application

Figure 26: Component Diagram - Web Application

#### Mobile Application

Figure 27: Component Diagram - Mobile Application

## Detail Description

### Class Diagram

Figure 28: Class Diagram

|  |  |  |
| --- | --- | --- |
|  | | |
| **Class Name** | **Mapping column with Conceptual diagram** | **Description** |
|  |  |  |

Table 22: Class Diagram Explanation

### Class Diagram Explanation

### Interactive diagram

#### Sequence Diagram

##### Register

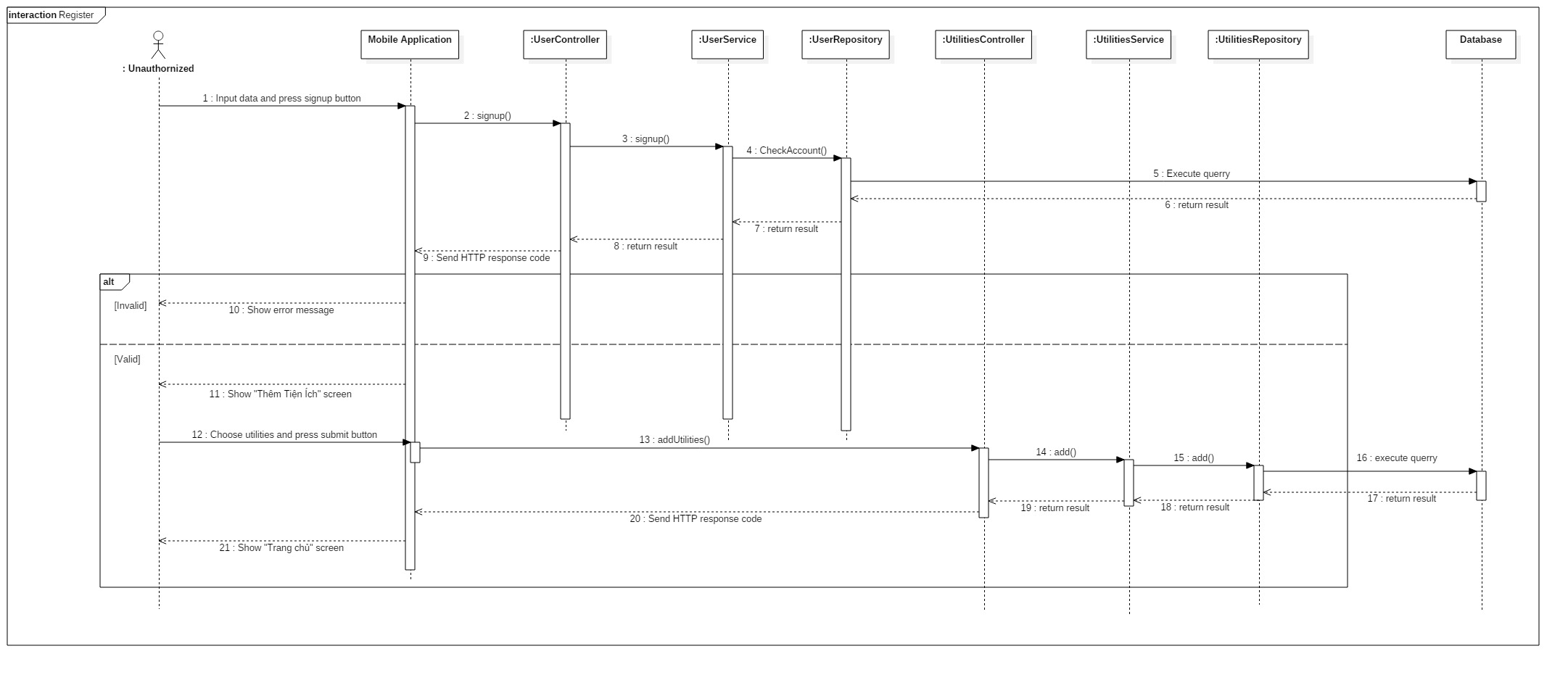


Figure 29: Sequence Diagram - <Unauthorized> Register

##### Login

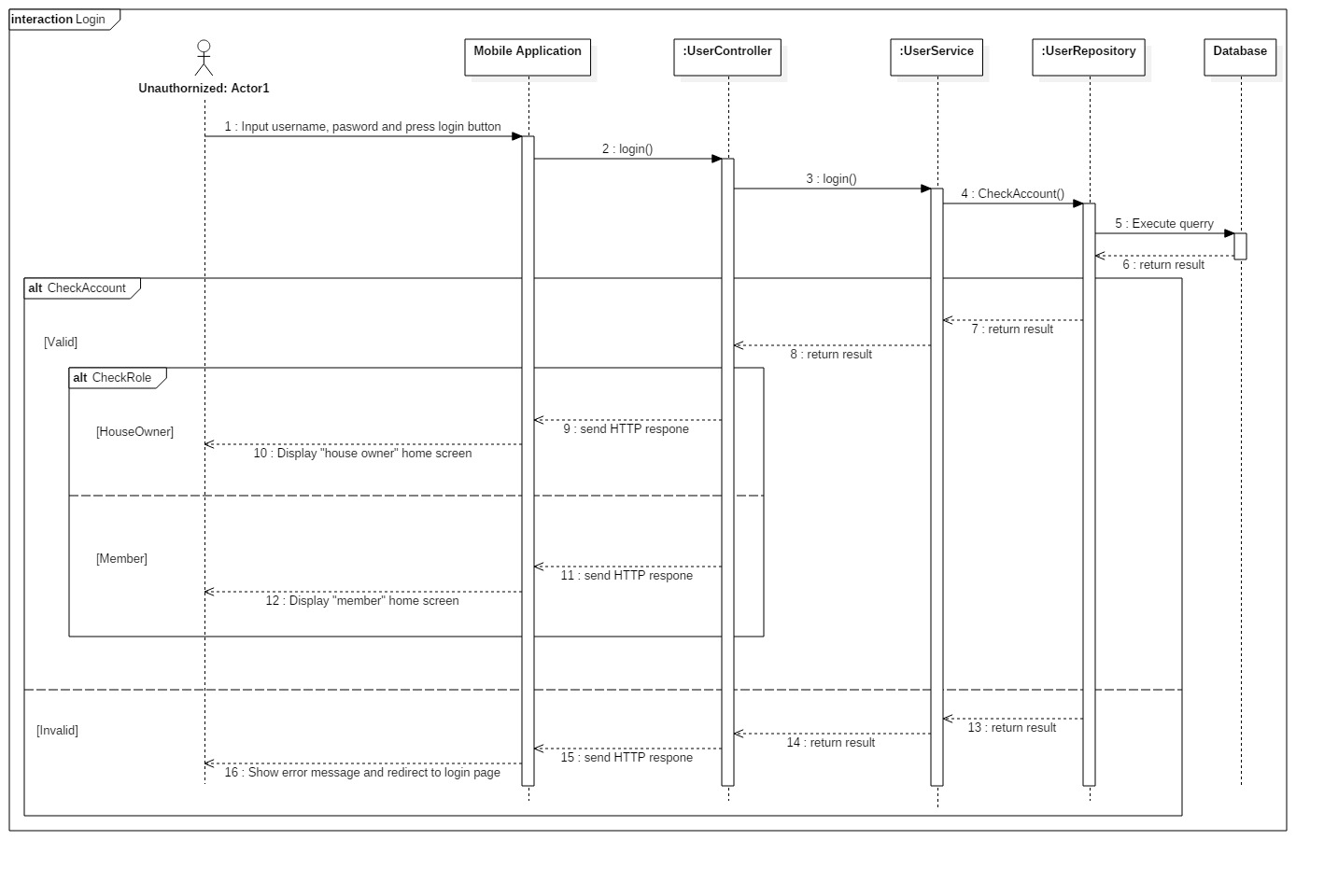


Figure 30: Sequence Diagram - <Unauthorized> Login

##### Create Room

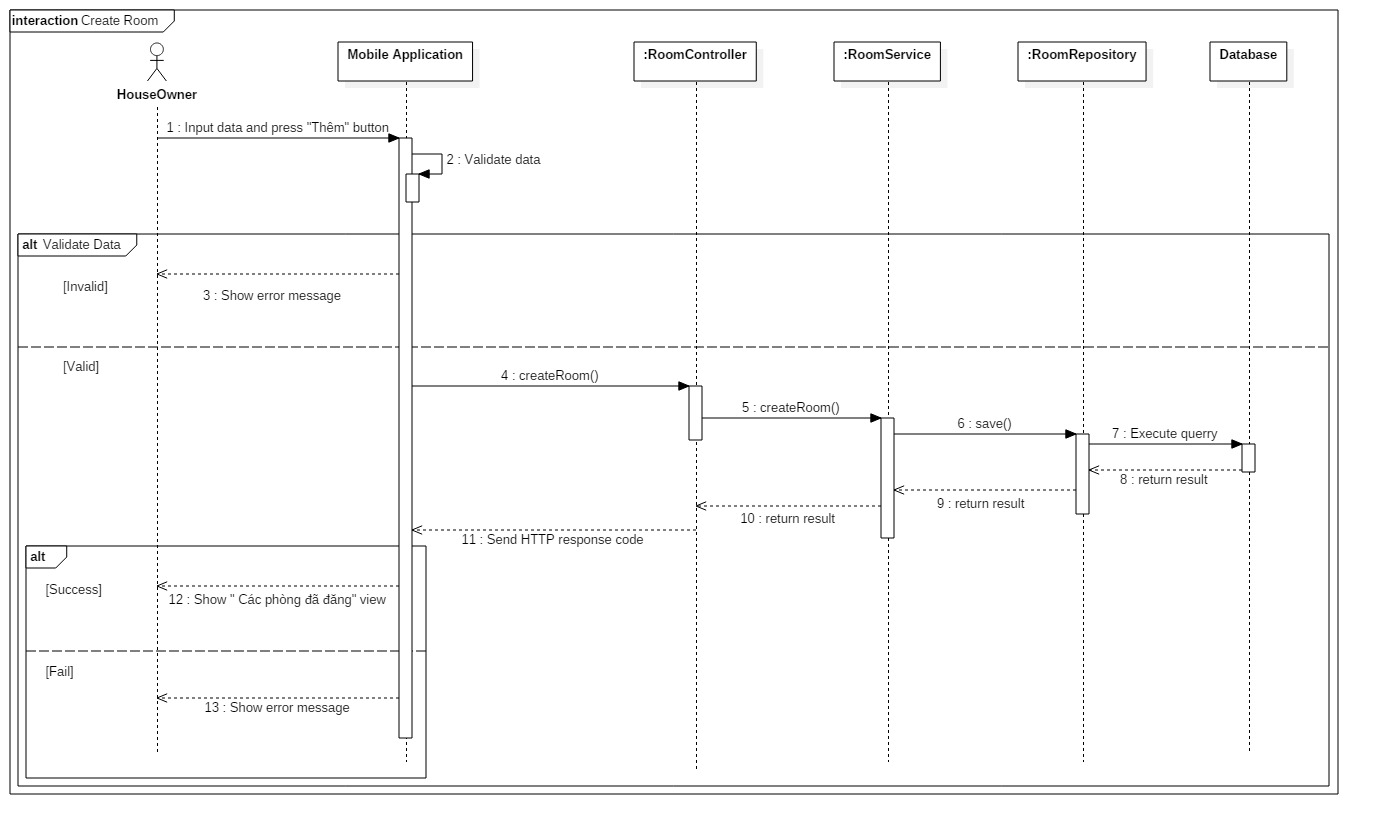


Figure 31: Sequence Diagram - <House owner> Create Room

##### Update Room

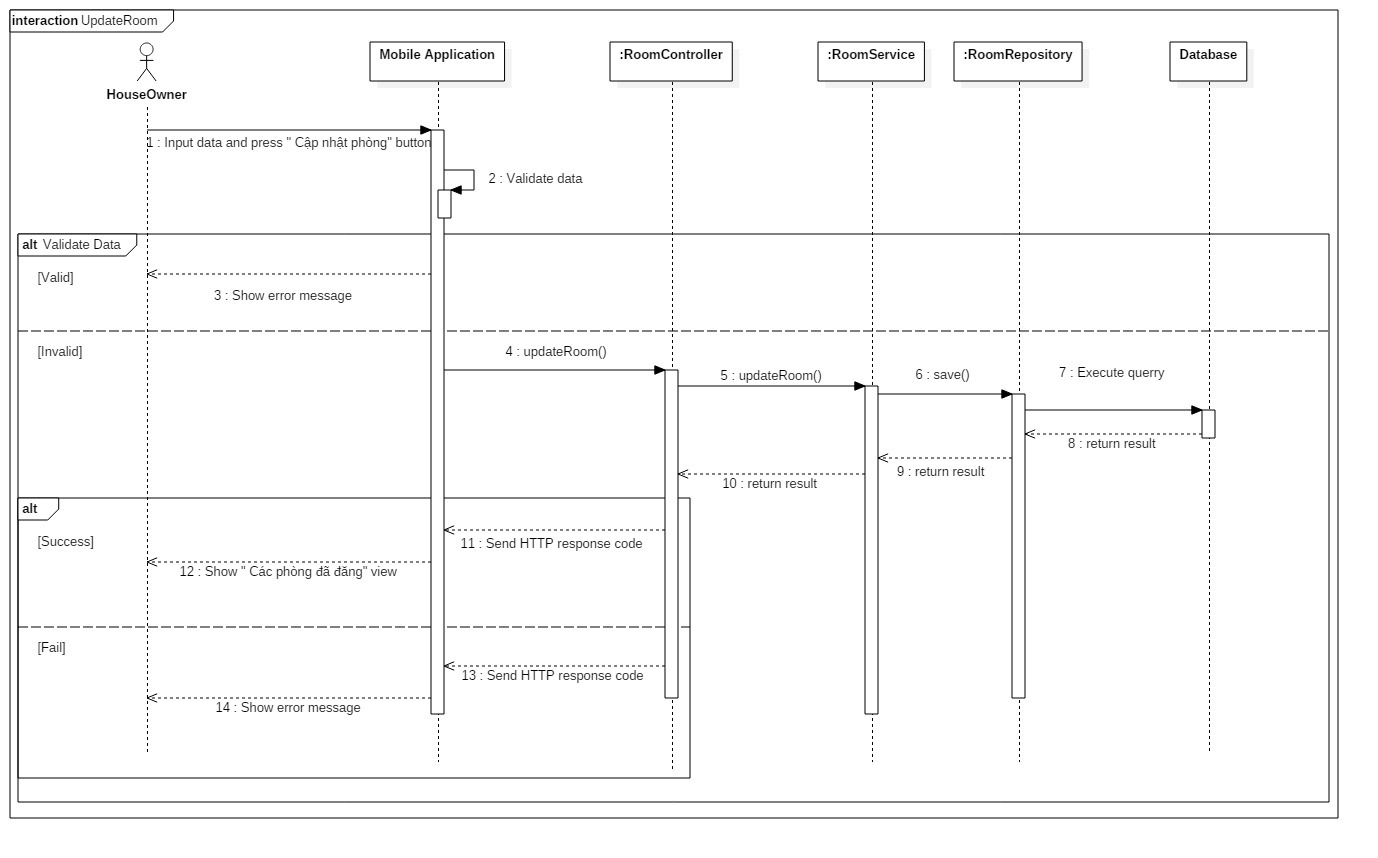


Figure 32: Sequence Diagram - <House owner> Update Room

##### Delete Room

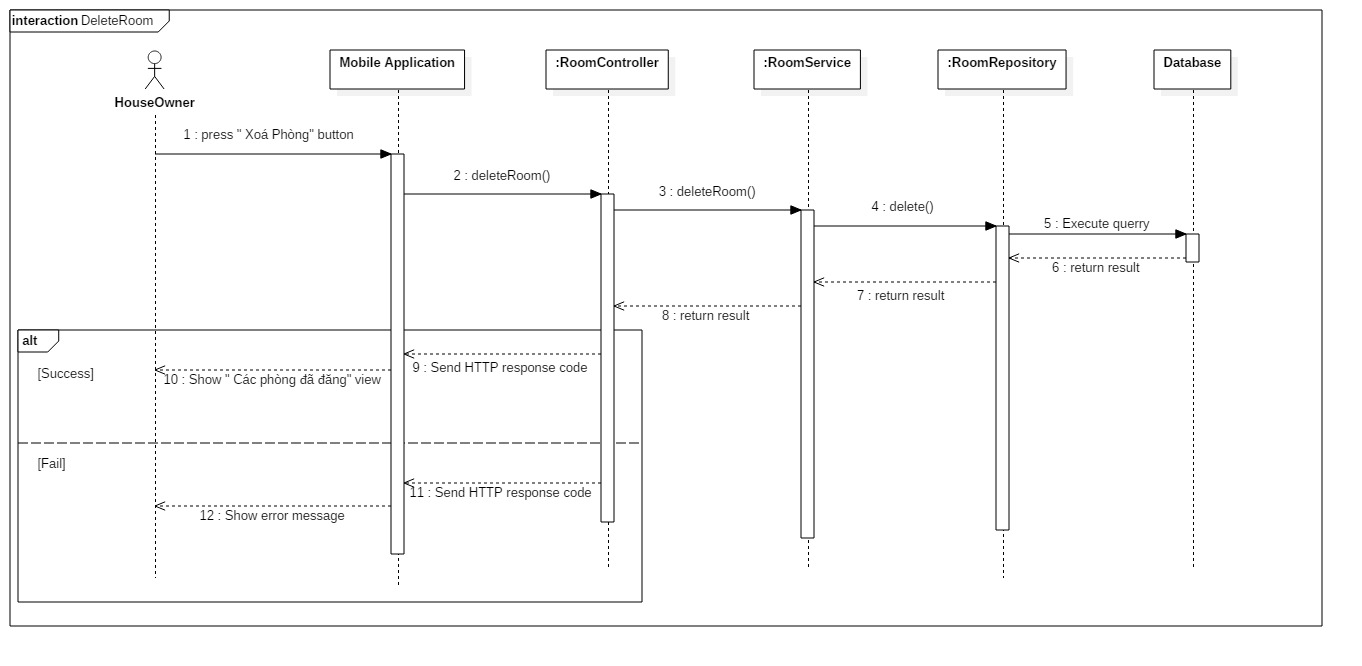


Figure 33: Sequence Diagram - <House owner> Delete Room

##### View Single Room

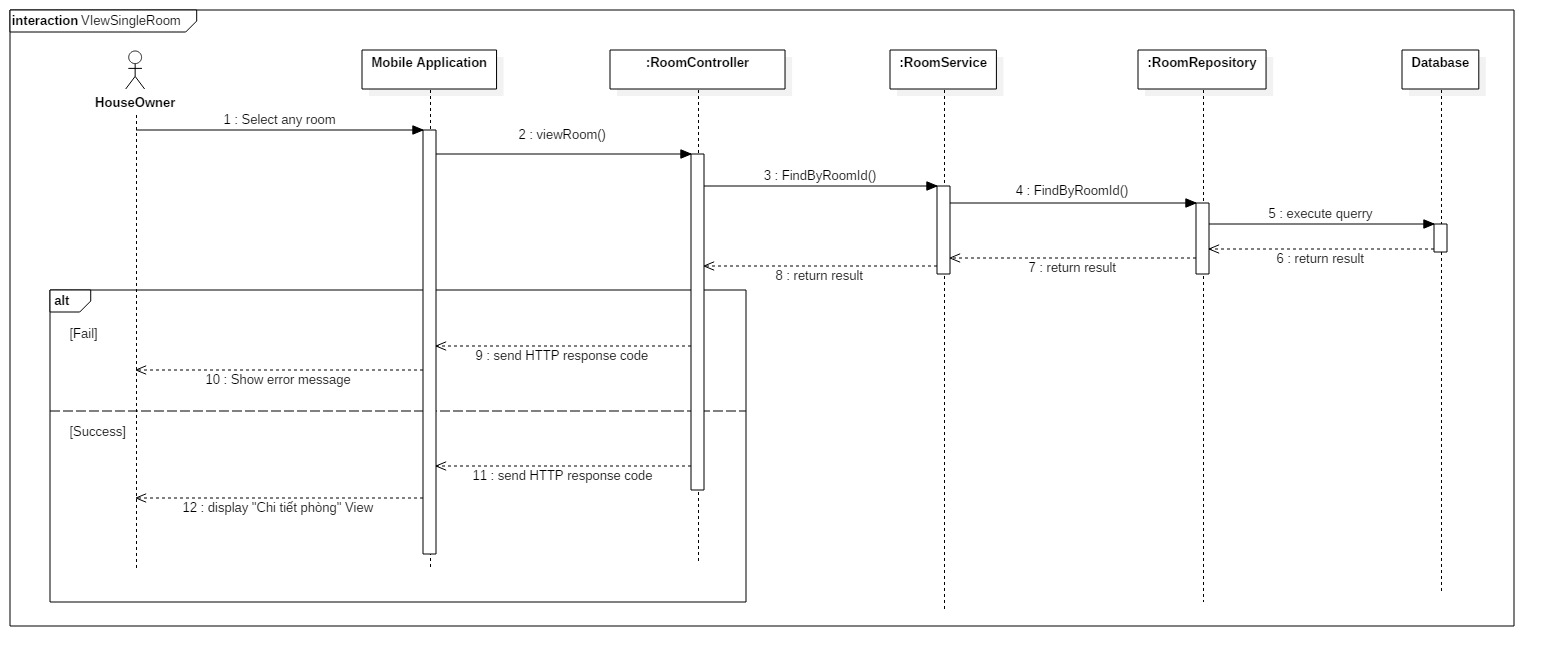


Figure 34: Sequence Diagram - <House owner> View Single Room

##### Add Member into Room

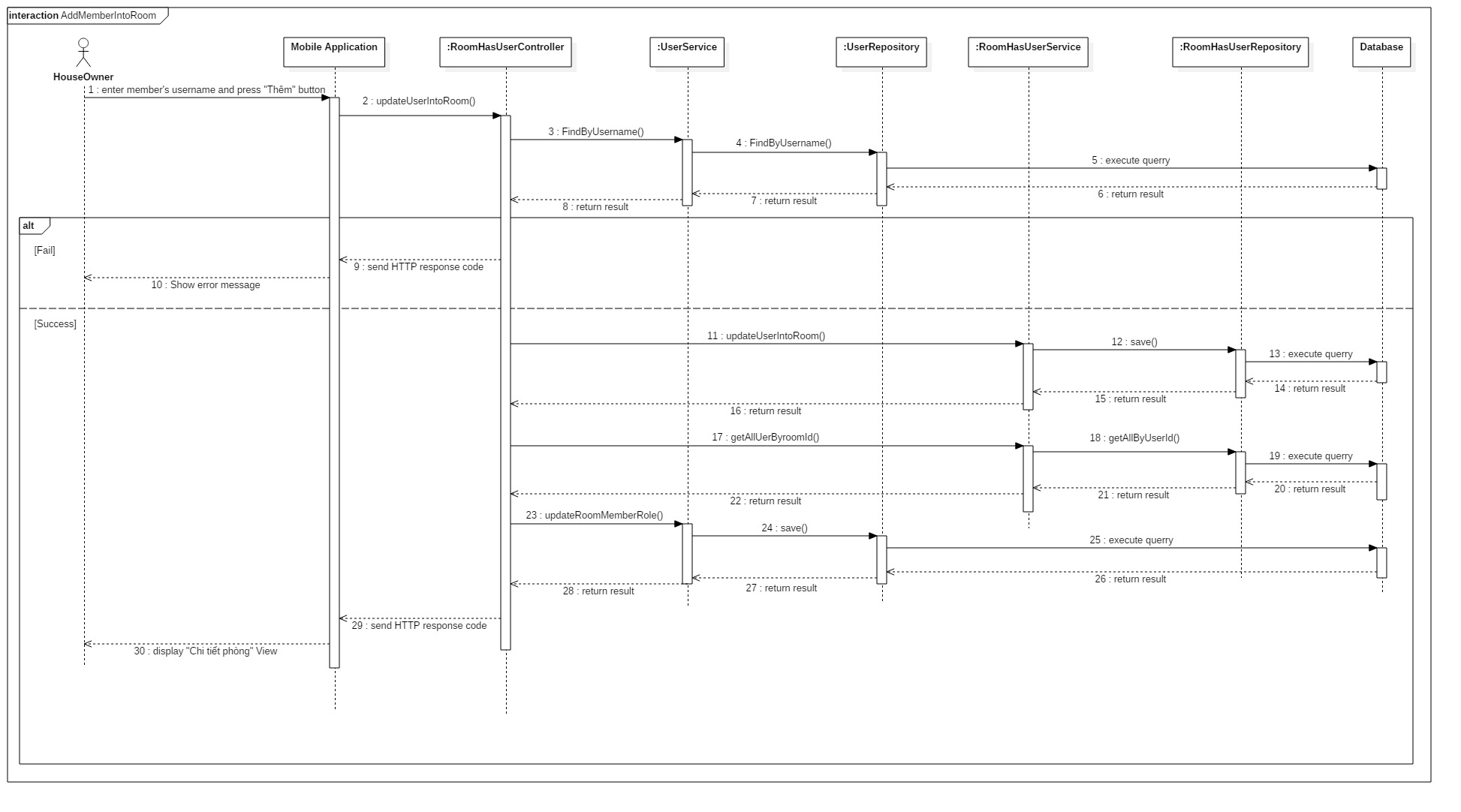


Figure 35: Sequence Diagram - <House owner> Add Member into Room

##### Delete Member of Room

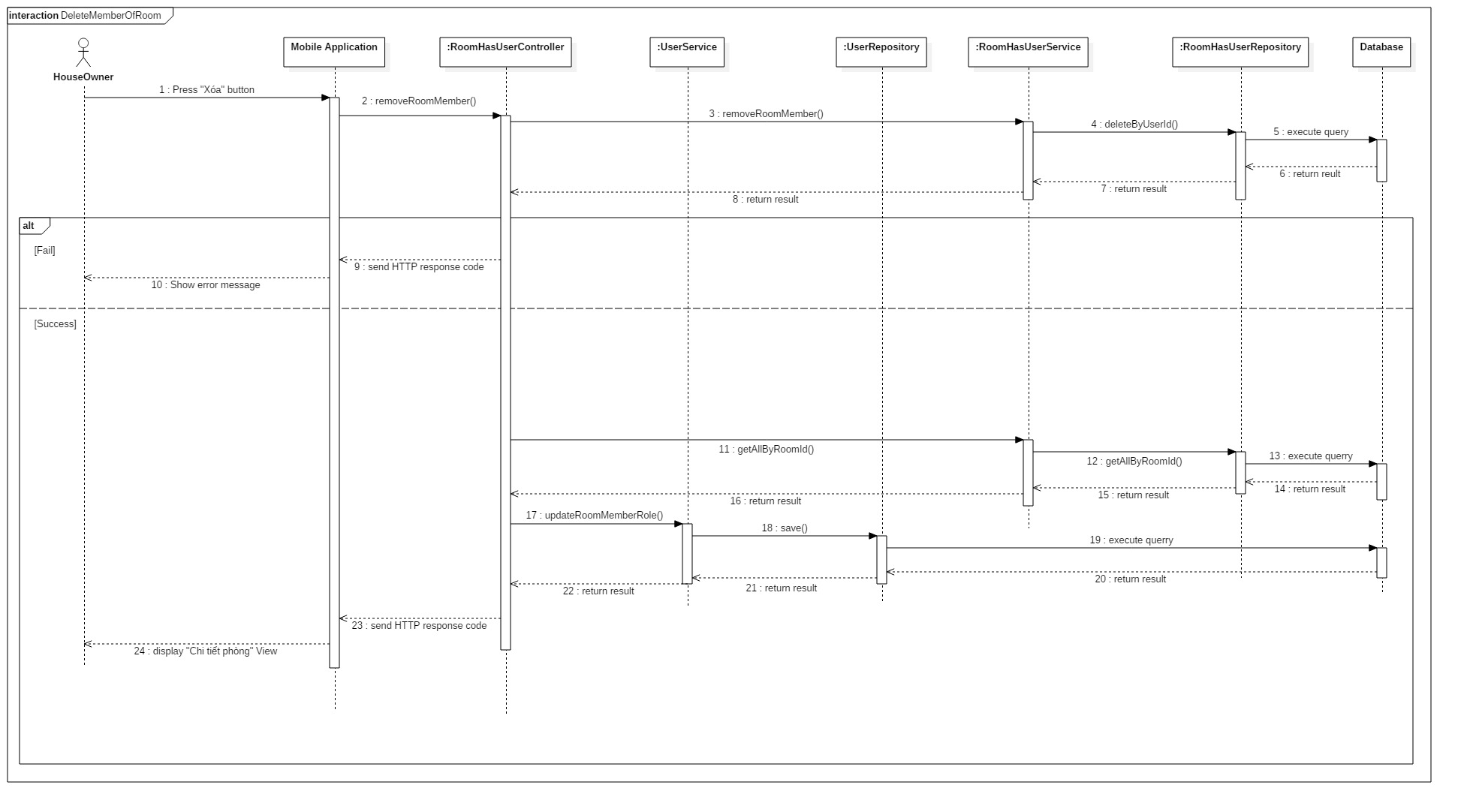


Figure 36: Sequence Diagram - <House owner> Delete Members of Room

##### Create Room Post

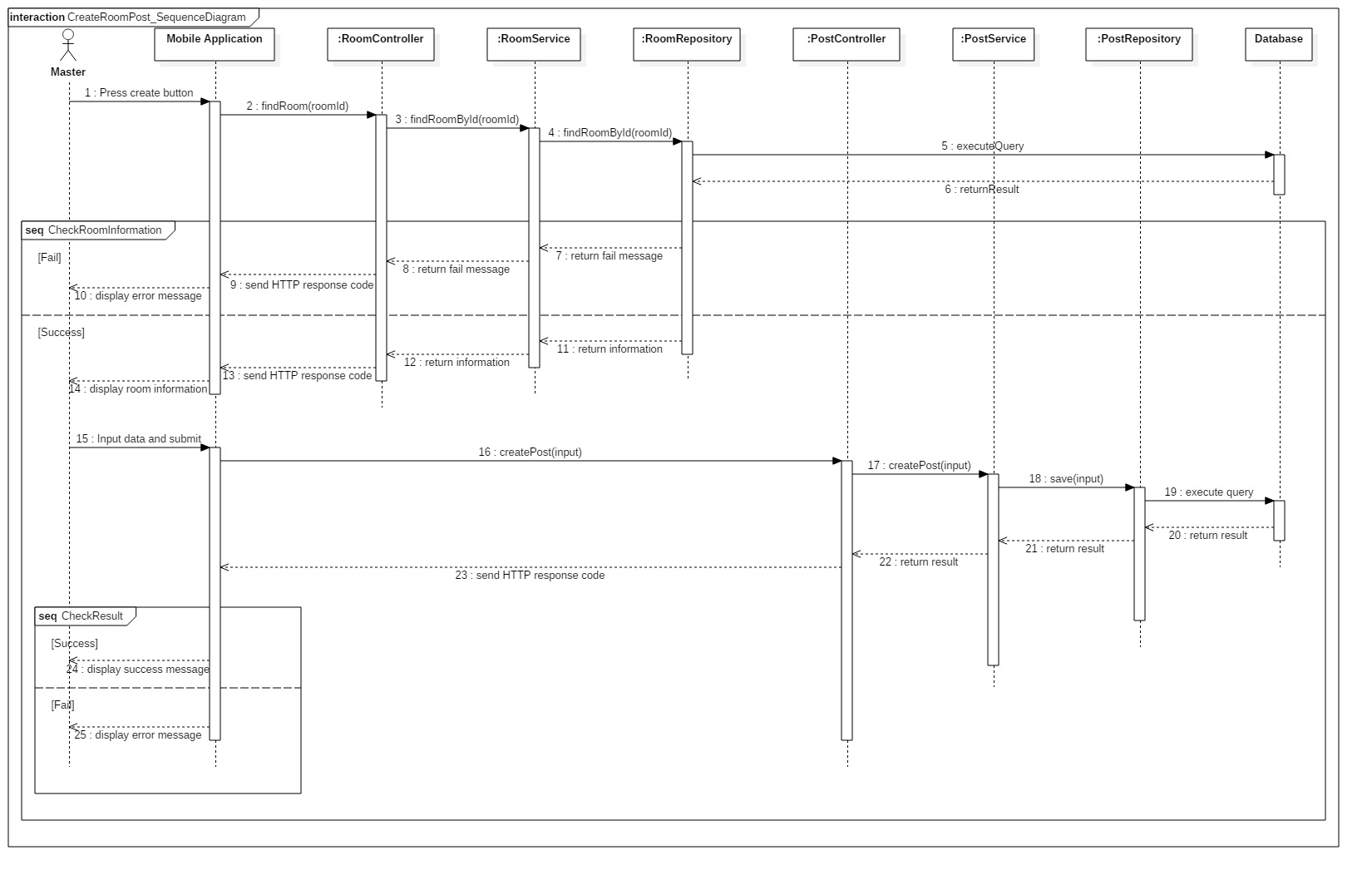


Figure 37: Sequence Diagram - <Room Master> Create Room Post

##### Delete Room Post

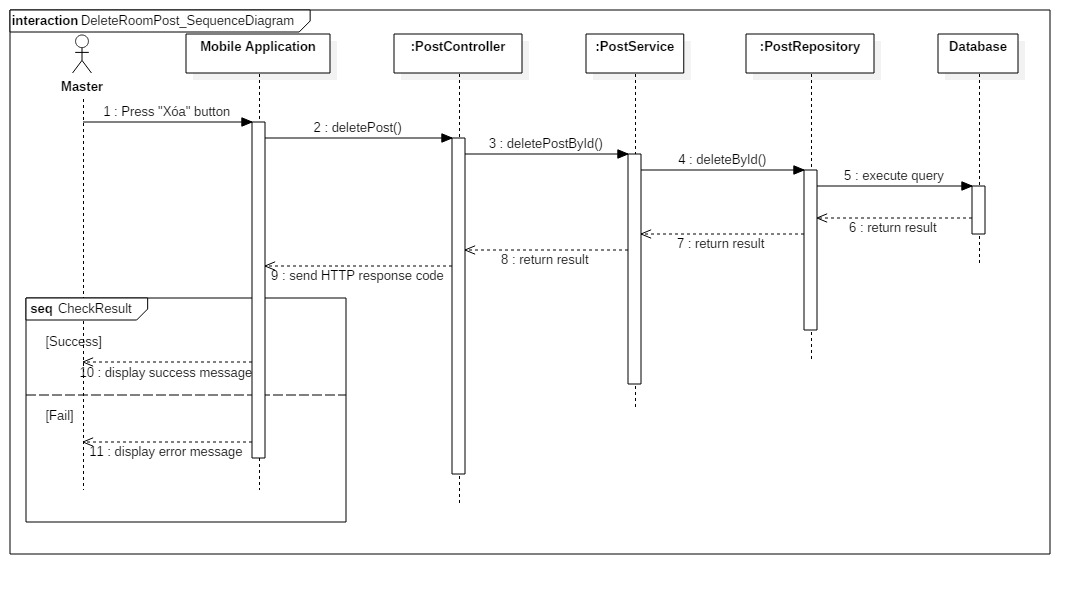


Figure 38: Sequence Diagram - < Room Master > Delete Room



##### Update Room Post

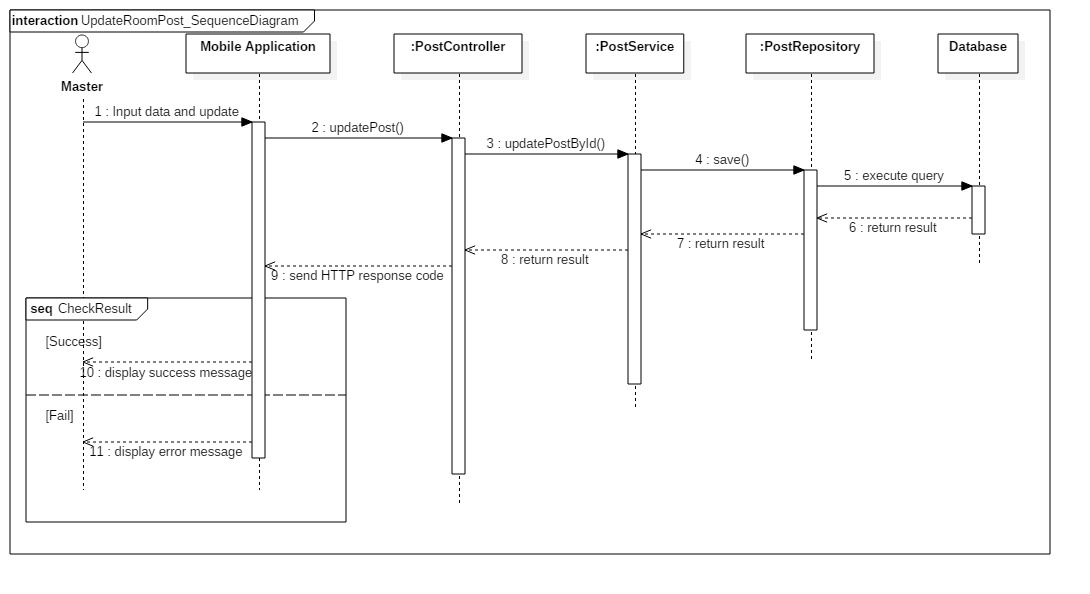


Figure 39: Sequence Diagram - <Room Master> Update Room Post

##### Create Partner Post

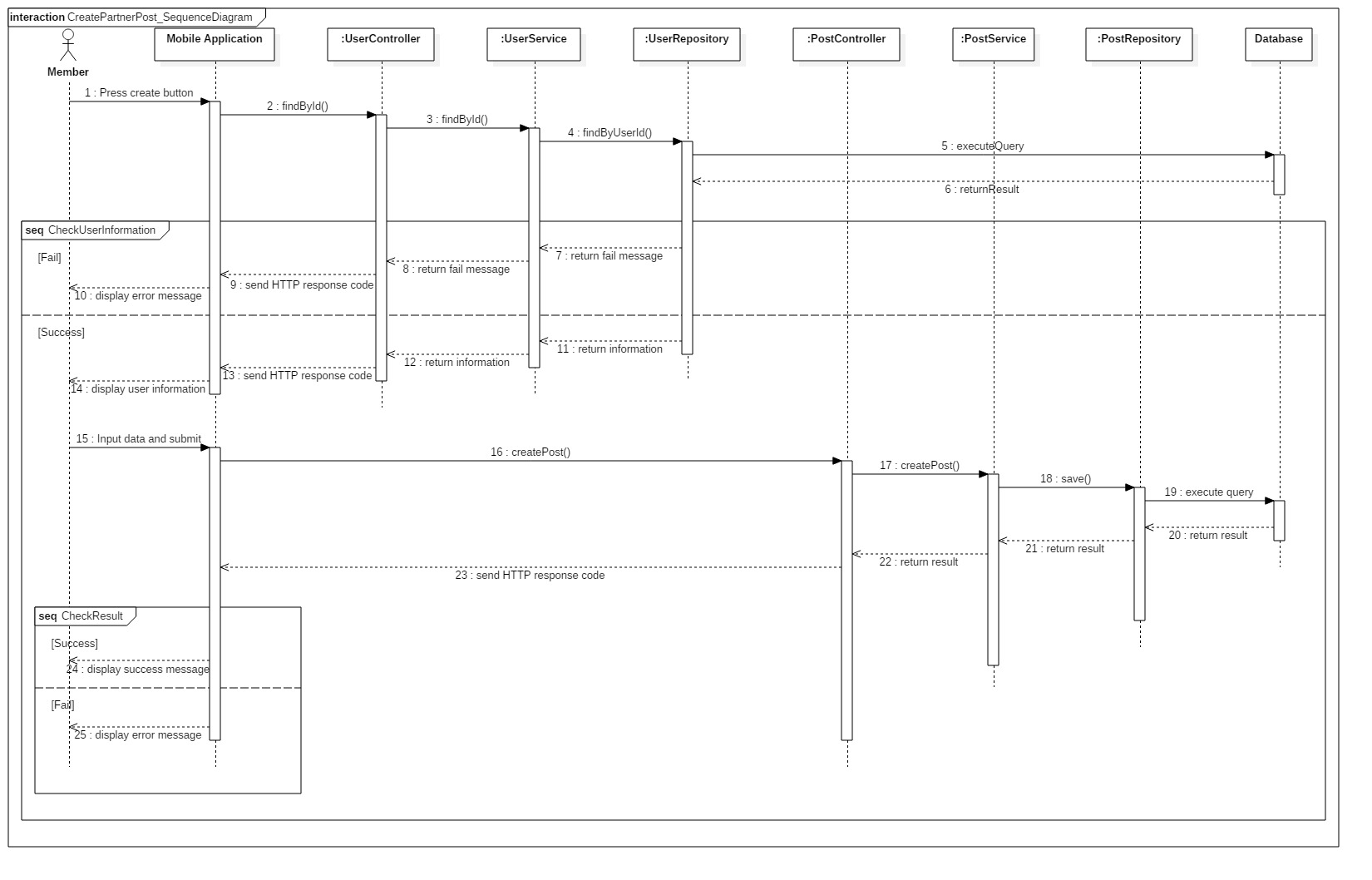


Figure 40: Sequence Diagram - <Member> Create Partner Post

##### Update Partner Post

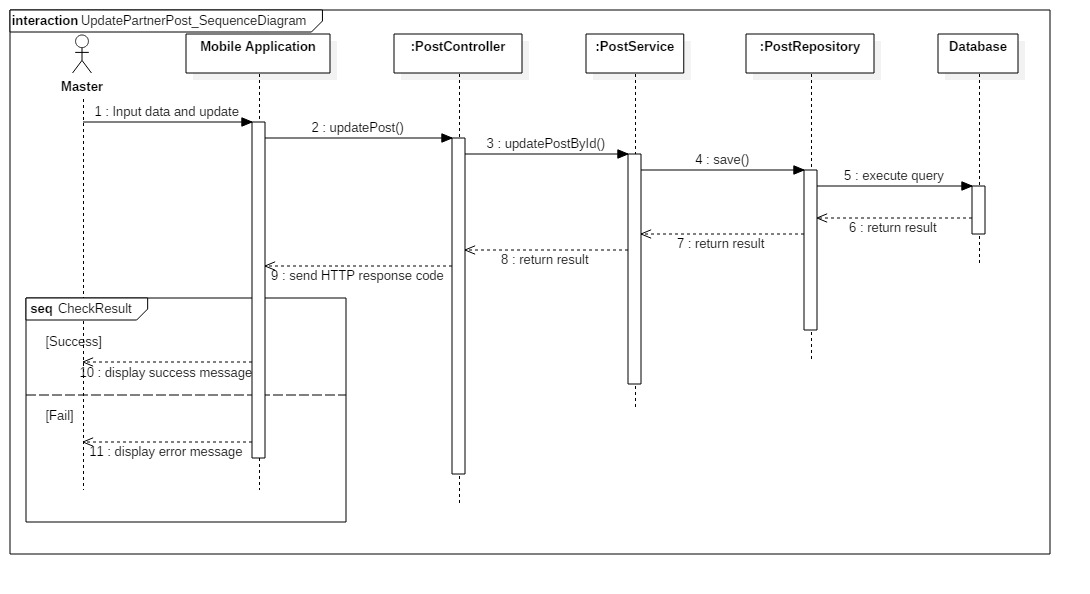


Figure 41: Sequence Diagram - < Member > Update Partner Post

##### Delete Partner Post

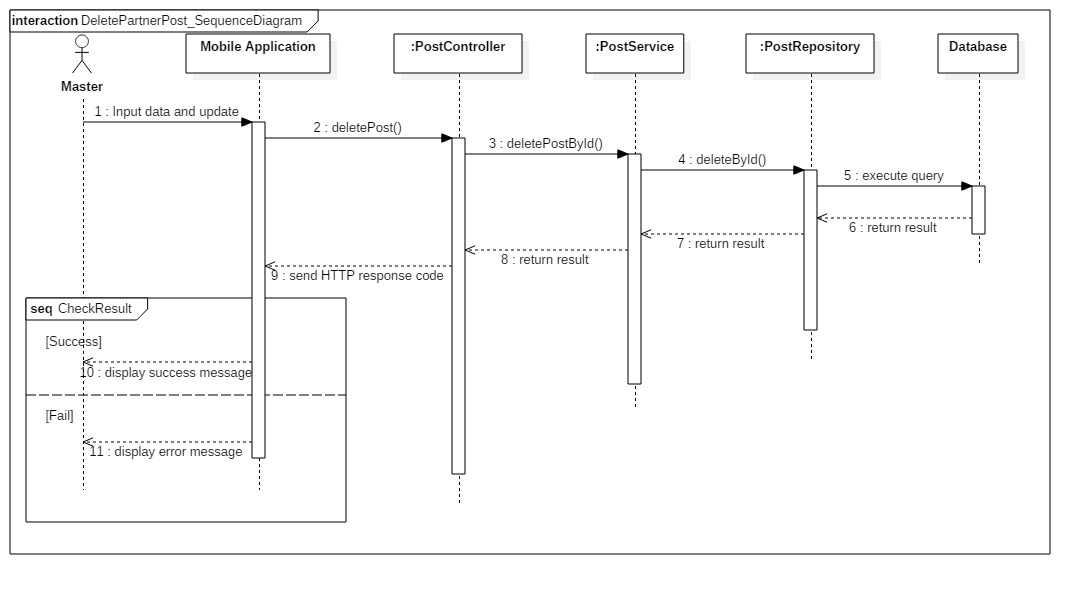


Figure 42: Sequence Diagram - < Member > Delete Partner Post

##### Search Post

Figure 43: Sequence Diagram - <Authorized> Search Post



##### View Suggest List

Figure 44: Sequence Diagram - <Authorized> View Suggest List

## Interface

### Admin Interface Design

#### Login

Figure 45: <Admin> Login Interface

Table 23: Admin login - Field, Buttons, Hyperlinks

#### Manage User

Figure 46: <Admin> Manage User Interface

Table 24: Admin Manage User - Field, Buttons, Hyperlinks

#### View Room List

Figure 47: <Admin> View Room List Interface

Table 25: Admin View Order List - Field, Buttons, Hyperlinks

### Unauthorized User Interface Design

#### Login

Figure 48: <Unauthorized User> Login Interface

Table 26: Login - Field, Buttons, Hyperlinks

### Authorized User Interface Design

#### View Setting

Figure 49: <Authorized User> View Setting Interface

Table 27: View Setting - Field, Buttons, Hyperlinks

#### Update User Information

Figure 50: <Authorized User> Update Information Interface (1)

C:\Users\VuTTSE60983\Desktop\w.PNG

Figure 51: <Authorized User> Update Information Interface (2)

Table 28: Update User Information - Field, Buttons, Hyperlinks

### House owner Interface Design

#### Create Room

Figure 52: <House owner> Create Room Interface

Table 29: Create Room - Field, Buttons, Hyperlinks

#### Update Room

Figure 53: < House owner > Update Room Interface

Table 30: Update Room - Field, Buttons, Hyperlinks

#### View Room

Figure 54: < House owner > View Room Interface

Table 31: View Room - Field, Buttons, Hyperlinks

### Room Master

#### View suggested partner list

#### View suggested room master list

#### Create Post

#### Update post

### Member

#### View suggested room master list

#### Create Post

#### Update Post

## Database Design

### Entity relationship diagram (ERD)

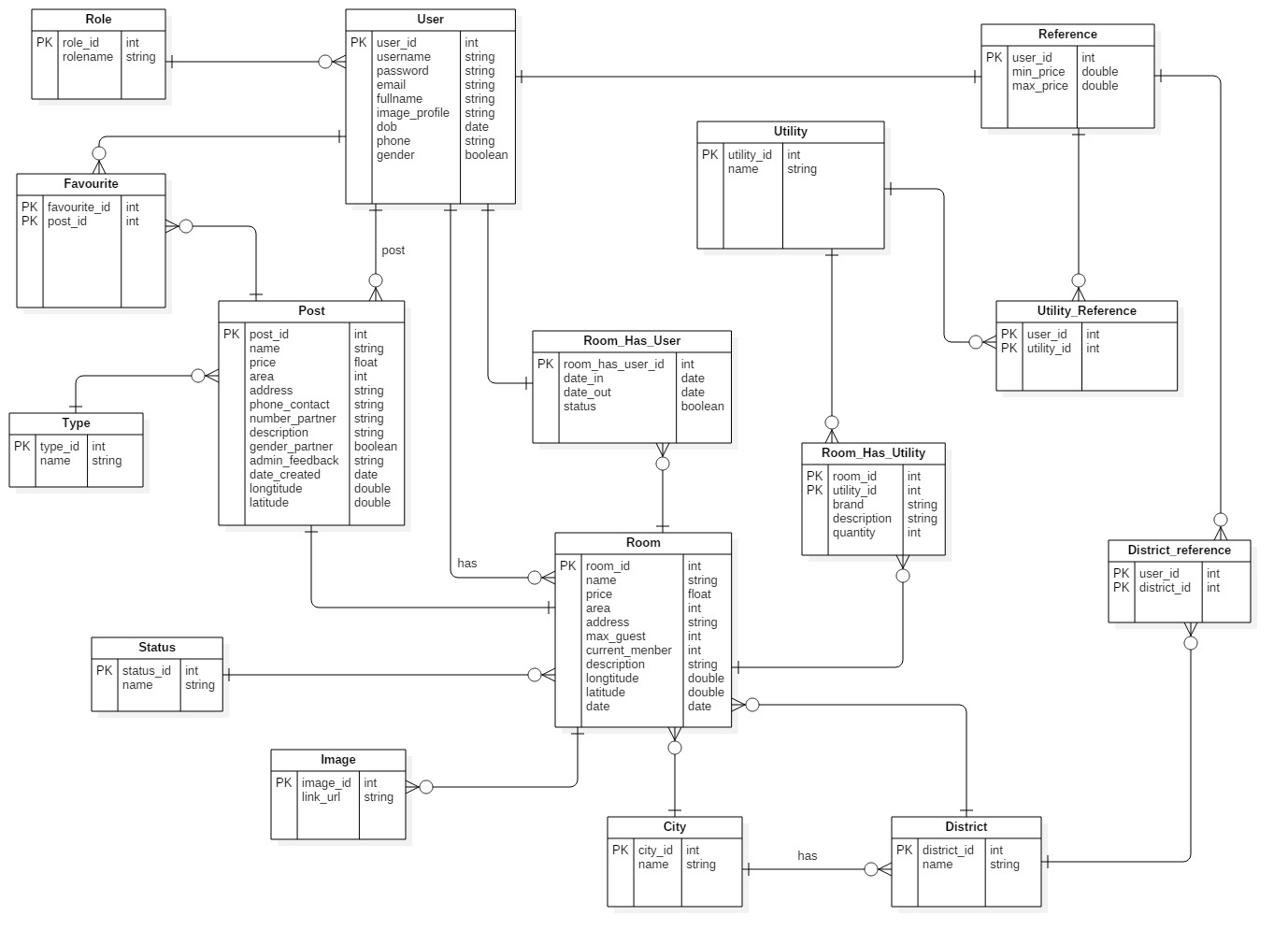


Figure 55: Entity Relationship diagram

### Entity dictionary

|  |  |
| --- | --- |
| **ENTITY DATA DICTIONARY: DESCRIBE CONTENT OF ALL ENTITIES** | |
| **Entity Name** | **Description** |
|  |  |

Table 32: Entity Data dictionary

## Algorithms

### Definition

### Define Problem

### Solution

### Flowchart

Figure 56: Flowchart