

**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](#_Toc528612283)

[List of Table 6](#_Toc528612284)

[List of Figure 8](#_Toc528612285)

[Definitions, Acronyms, and Abbreviations 9](#_Toc528612286)

[A. Introduction 9](#_Toc528612287)

[1. Project Information 10](#_Toc528612288)

[2. Introduction 10](#_Toc528612289)

[3. Current Situation 10](#_Toc528612290)

[4. Problem Definition 10](#_Toc528612291)

[5. Proposed Solution 10](#_Toc528612292)

[5.1 Feature functions 11](#_Toc528612293)

[5.2 Advantages and Disadvantages 11](#_Toc528612294)

[6. Functional Requirements 11](#_Toc528612295)

[7. Role and Responsibility 12](#_Toc528612296)

[B. Software Project Management Plan 12](#_Toc528612297)

[1. Problem Definition 12](#_Toc528612298)

[1.1 Name of this Capstone Project 12](#_Toc528612299)

[1.2 Problem Abstract 12](#_Toc528612300)

[1.3 Project Overview 12](#_Toc528612301)

[1.3.1 Current Situation 12](#_Toc528612302)

[1.3.2 The Proposed System 13](#_Toc528612303)

[1.3.2.1 Mobile Application 13](#_Toc528612304)

[1.3.2.2 Web Application 13](#_Toc528612305)

[1.3.3 Boundaries of the System 13](#_Toc528612306)

[1.3.4 Future Plan 14](#_Toc528612307)

[1.3.5 Development Environment 14](#_Toc528612308)

[1.3.5.1 Hardware requirements 14](#_Toc528612309)

[1.3.5.2 Software requirements 14](#_Toc528612310)

[2. Project Organization 15](#_Toc528612311)

[2.1 Software Process Model 15](#_Toc528612312)

[2.2 Roles and Responsibilities 16](#_Toc528612313)

[2.3 Tools and Techniques 16](#_Toc528612314)

[3. Project Management Plan 17](#_Toc528612315)

[3.1 Product Backlog 17](#_Toc528612316)

[3.2 Sprint Backlog 18](#_Toc528612317)

[3.3 Meeting Minutes 18](#_Toc528612318)

[4. Coding Convention 18](#_Toc528612319)

[C. Software Requirement Specification 19](#_Toc528612320)

[1. User Requirement Specification 19](#_Toc528612321)

[1.1 Unauthorized User Requirement 19](#_Toc528612322)

[1.2 Authorized User Requirement 19](#_Toc528612323)

[1.3 House Owner Requirement 19](#_Toc528612324)

[1.4 Room Master Requirement 19](#_Toc528612325)

[1.5 Member Requirement 20](#_Toc528612326)

[1.6 Admin Requirement 20](#_Toc528612327)

[2. System Requirement Specification 20](#_Toc528612328)

[2.1 External Interface Requirement 20](#_Toc528612329)

[2.1.1 User Interface 20](#_Toc528612330)

[2.1.2 Hardware Interface. 20](#_Toc528612331)

[2.1.3 Software Interface 20](#_Toc528612332)

[2.1.4 Communication Protocol 20](#_Toc528612333)

[2.2 System Overview Use Case 21](#_Toc528612334)

[2.3 List of Use Case 22](#_Toc528612335)

[2.3.1 <Unauthorized User> Overview Use Case 22](#_Toc528612336)

[2.3.1.1 <Unauthorized User> Register 22](#_Toc528612337)

[2.3.1.2 <Unauthorized User> Login 24](#_Toc528612338)

[2.3.2 <Authorized User> Overview Use 25](#_Toc528612339)

[2.3.2.1 <Authorized User> Logout 25](#_Toc528612340)

[2.3.2.2 <Authorized User> View Notification 26](#_Toc528612341)

[2.3.2.3 <Authorized User> Update Info 27](#_Toc528612342)

[2.3.3 <House Owner> Overview Use Case 30](#_Toc528612343)

[2.3.3.1 <House Owner> Create Room 30](#_Toc528612344)

[2.3.3.2 <House owner> Update Room 32](#_Toc528612345)

[2.3.3.3 <House owner> Delete Room 34](#_Toc528612346)

[2.3.3.4 <House owner> Add Member into Room 36](#_Toc528612347)

[2.3.3.5 <House owner> Delete Member from Room 38](#_Toc528612348)

[2.3.4 <Room Master> Overview Use Case 40](#_Toc528612349)

[2.3.4.1 <Room Master> View Suggested List 41](#_Toc528612350)

[2.3.4.2 <Room Master> Create Post 42](#_Toc528612351)

[2.3.4.3 <Room Master> Update Post 44](#_Toc528612352)

[2.3.4.4 <Room Master> Delete Post 47](#_Toc528612353)

[2.3.4.5 <Room Master> View Partner Detail 48](#_Toc528612354)

[2.3.5 <Member> Overview Use Case 50](#_Toc528612355)

[2.3.5.1 <Member> View Suggested Room Post list 50](#_Toc528612356)

[2.3.5.2 <Member> Create Partner Post 51](#_Toc528612357)

[2.3.5.3 <Member> Update Partner Post 53](#_Toc528612358)

[2.3.5.4 <Member> Delete Partner Post 55](#_Toc528612359)

[2.3.5.5 <Member> Negotiate 57](#_Toc528612360)

[2.3.6 <Admin> Overview Use Case 58](#_Toc528612361)

[2.3.6.1 <Admin> Manage Rooms 59](#_Toc528612362)

[2.3.6.2 <Admin> Manage Post 60](#_Toc528612363)

[2.3.6.3 <Admin> Approve Room 61](#_Toc528612364)

[2.3.6.4 <Admin> Manage Users 63](#_Toc528612365)

[2.3.6.5 <Admin> View Users 64](#_Toc528612366)

[3. Software System Attribute 65](#_Toc528612367)

[3.1 Usability 65](#_Toc528612368)

[3.1.1 Graphic User Interface 65](#_Toc528612369)

[3.1.2 Usability 65](#_Toc528612370)

[3.2 Reliability 66](#_Toc528612371)

[3.3 Availability 66](#_Toc528612372)

[3.4 Security 66](#_Toc528612373)

[3.5 Maintainability 66](#_Toc528612374)

[3.6 Portability 66](#_Toc528612375)

[3.7 Performance(gia tri thuc te) 66](#_Toc528612376)

[4. Conceptual diagram 66](#_Toc528612377)

[D. Software Design Description 67](#_Toc528612378)

[1. Design Overview 67](#_Toc528612379)

[2. System Architecture Design 68](#_Toc528612380)

[2.1 System Architecture Design 68](#_Toc528612381)

[3. Component Diagram 69](#_Toc528612382)

[4. Detail Description 70](#_Toc528612383)

[4.1 Class Diagram 70](#_Toc528612384)

[4.2 Class Diagram Explanation 71](#_Toc528612385)

[4.2.1 User 71](#_Toc528612386)

[4.2.2 Room 72](#_Toc528612387)

[4.2.3 Post 72](#_Toc528612388)

[4.2.4 Role 73](#_Toc528612389)

[4.2.5 Favourite 73](#_Toc528612390)

[4.2.6 Utilities 73](#_Toc528612391)

[4.2.7 Reference 74](#_Toc528612392)

[4.2.8 District 74](#_Toc528612393)

[4.2.9 City 74](#_Toc528612394)

[4.2.10 Image 74](#_Toc528612395)

[4.2.11 Type 74](#_Toc528612396)

[4.2.12 Status 75](#_Toc528612397)

[4.2.13 Room\_Has\_Utilities 75](#_Toc528612398)

[4.2.14 Utilities\_Reference 75](#_Toc528612399)

[4.2.15 District\_Reference 75](#_Toc528612400)

[4.2.16 Room\_Has\_User 76](#_Toc528612401)

[4.3 Interactive diagram 76](#_Toc528612402)

[4.3.1 Sequence Diagram 76](#_Toc528612403)

[4.3.1.1 Register 76](#_Toc528612404)

[4.3.1.2 Login 76](#_Toc528612405)

[4.3.1.3 Create Room 77](#_Toc528612406)

[4.3.1.4 Update Room 77](#_Toc528612407)

[4.3.1.5 Delete Room 78](#_Toc528612408)

[4.3.1.6 View Single Room 78](#_Toc528612409)

[4.3.1.7 Add Member into Room 79](#_Toc528612410)

[4.3.1.8 Delete Member of Room 79](#_Toc528612411)

[4.3.1.9 Create Room Post 80](#_Toc528612412)

[4.3.1.10 Delete Room Post 80](#_Toc528612413)

[4.3.1.11 Update Room Post 81](#_Toc528612432)

[4.3.1.12 Create Partner Post 82](#_Toc528612433)

[4.3.1.13 Update Partner Post 82](#_Toc528612434)

[4.3.1.14 Delete Partner Post 82](#_Toc528612435)

[4.3.1.15 Search Post 83](#_Toc528612436)

[4.3.1.16 View Suggest List 83](#_Toc528612460)

[5. Interface 83](#_Toc528612461)

[5.1 Admin Interface Design 83](#_Toc528612462)

[5.1.1 Login 83](#_Toc528612463)

[5.1.2 Manage User 83](#_Toc528612464)

[5.1.3 Manage Room 83](#_Toc528612465)

[5.1.4 Manage Post 83](#_Toc528612466)

[5.1.5 View User List 84](#_Toc528612467)

[5.1.6 View Room List 84](#_Toc528612468)

[5.1.7 View Post List 84](#_Toc528612469)

[5.2 Unauthorized User Interface Design 84](#_Toc528612470)

[5.2.1 Register 84](#_Toc528612471)

[5.2.2 Login 84](#_Toc528612472)

[5.3 Authorized User Interface Design 84](#_Toc528612473)

[5.3.1 View Setting 84](#_Toc528612474)

[5.3.2 Update User Information 84](#_Toc528612475)

[5.3.3 View Profile 84](#_Toc528612476)

[5.3.4 View Notification 85](#_Toc528612477)

[5.4 House owner Interface Design 85](#_Toc528612478)

[5.4.1 Create Room 85](#_Toc528612479)

[5.4.2 Update Room 85](#_Toc528612480)

[5.4.3 View Room 85](#_Toc528612481)

[5.4.4 Delete Room 85](#_Toc528612482)

[5.4.5 Add member into Room 85](#_Toc528612483)

[5.4.6 Delete member of Room 85](#_Toc528612484)

[5.5 Room Master 86](#_Toc528612485)

[5.5.1 View suggested partner list 86](#_Toc528612486)

[5.5.2 View suggested room master list 86](#_Toc528612487)

[5.5.3 Create Post 86](#_Toc528612488)

[5.5.4 Update post 86](#_Toc528612489)

[5.5.5 View Post 86](#_Toc528612490)

[5.5.6 Delete Post 86](#_Toc528612491)

[5.6 Member 86](#_Toc528612492)

[5.6.1 View suggested room master list 86](#_Toc528612493)

[5.6.2 Create Post 86](#_Toc528612494)

[5.6.3 Update Post 86](#_Toc528612495)

[5.6.4 Delete Post 86](#_Toc528612496)

[5.6.5 View Post 86](#_Toc528612497)

[6. Database Design 86](#_Toc528612498)

[6.1 Entity relationship diagram (ERD) 86](#_Toc528612499)

[6.2 Entity dictionary 86](#_Toc528612500)

[7. Algorithms 87](#_Toc528612501)

[7.1 Definition 87](#_Toc528612502)

[7.2 Define Problem 87](#_Toc528612503)

[7.3 Solution 87](#_Toc528612504)

[7.4 Flowchart 87](#_Toc528612505)

# List of Table

[Table 1: Roles and Responsibilities 12](#_Toc528612506)

[Table 2: Hardware Requirements for Server 14](#_Toc528612507)

[Table 3: Hardware Requirement for Mobile 14](#_Toc528612508)

[Table 4: Software Requirements 15](#_Toc528612509)

[Table 5: Roles and Responsibilities 16](#_Toc528612510)

[Table 6: Tools and Techniques 17](#_Toc528612511)

[Table 7: Product Backlog 18](#_Toc528612512)

[Table 8: <Unauthorized User> Register specification 24](#_Toc528612513)

[Table 9: <Unauthorized User> Login specification 25](#_Toc528612514)

[Table 10: <Authorized User> Logout specification 26](#_Toc528612515)

[Table 11: <Authorized User> View Notification specification 27](#_Toc528612516)

[Table 12: <Authorized User> Update Information specification 30](#_Toc528612517)

[Table 13: <House Owner> Create Room specification 32](#_Toc528612518)

[Table 14: <House owner> Update Room specification 34](#_Toc528612519)

[Table 15: <House owner> Delete Room specification 36](#_Toc528612520)

[Table 16: <House owner> Add Member into Room specification 38](#_Toc528612521)

[Table 17: <House owner> Delete Member from Room specification 40](#_Toc528612522)

[Table 18: <Room Master> View Suggested Partners List Specification 42](#_Toc528612523)

[Table 19: <Room Master> Create Post specification 44](#_Toc528612524)

[Table 20: <Room Master> Update Post specification 47](#_Toc528612525)

[Table 21: <Room Master> Delete Post specification 48](#_Toc528612526)

[Table 22: <Room Master> View Partner Detail Specification 50](#_Toc528612527)

[Table 23: <Member> View Suggested Room Posts List Specification 51](#_Toc528612528)

[Table 24: <Member> Create partner Post specification 53](#_Toc528612529)

[Table 25: <Member> Update partner Post specification 55](#_Toc528612530)

[Table 26: <Member> Delete Partner Post specification 57](#_Toc528612531)

[Table 27: <Member> Negotiate specification 58](#_Toc528612532)

[Table 28: <Admin> Manage room specification 60](#_Toc528612533)

[Table 29: <Admin> Manage post specification 61](#_Toc528612534)

[Table 30: <Admin> Approve room specification 63](#_Toc528612535)

[Table 31: <Admin> Manage User specification 64](#_Toc528612536)

[Table 32: <Admin> View User specification 65](#_Toc528612537)

[Table 33: Conceptual Entity dictionary 67](#_Toc528612538)

[Table 34: Class Diagram Explanation 71](#_Toc528612539)

[Table 35: User Attributes 72](#_Toc528612540)

[Table 36: Room Attributes 72](#_Toc528612541)

[Table 37: Post Attributes 73](#_Toc528612542)

[Table 38: Role Attributes 73](#_Toc528612543)

[Table 39: Favourite Attributes 73](#_Toc528612544)

[Table 40: Utility Attributes 73](#_Toc528612545)

[Table 41: Reference Attributes 74](#_Toc528612546)

[Table 42: District Attributes 74](#_Toc528612547)

[Table 43: City Attributes 74](#_Toc528612548)

[Table 44: Image Attributes 74](#_Toc528612549)

[Table 45: Type Attributes 75](#_Toc528612550)

[Table 46: Status Attributes 75](#_Toc528612551)

[Table 47: Room Has Utilities Attributes 75](#_Toc528612552)

[Table 48: Utilities Reference Attributes 75](#_Toc528612553)

[Table 49: District Reference Attributes 75](#_Toc528612554)

[Table 50: Room Has User Attributes 76](#_Toc528612555)

[Table 51: Admin login - Field, Buttons 83](#_Toc528612556)

[Table 52: Admin Manage User - Field, Buttons, Hyperlinks 83](#_Toc528612557)

[Table 53: Admin Manage Room - Field, Buttons, Hyperlinks 83](#_Toc528612558)

[Table 54: Admin Manage Post - Field, Buttons, Hyperlinks 84](#_Toc528612559)

[Table 55: Admin View User List - Field, Buttons, Hyperlinks 84](#_Toc528612560)

[Table 56: Admin View Room List - Field, Buttons, Hyperlinks 84](#_Toc528612561)

[Table 57: Admin View Post List - Field, Buttons, Hyperlinks 84](#_Toc528612562)

[Table 58: Register - Field, Buttons 84](#_Toc528612563)

[Table 59: Register - Field, Buttons 84](#_Toc528612564)

[Table 60: View Setting - Field, Buttons, Hyperlinks 84](#_Toc528612565)

[Table 61: Update User Information - Field, Buttons, Hyperlinks 84](#_Toc528612566)

[Table 62: View Profile - Field, Buttons, Hyperlinks 85](#_Toc528612567)

[Table 63: View Notification - Field, Buttons, Hyperlinks 85](#_Toc528612568)

[Table 64: Create Room - Field, Buttons, Hyperlinks 85](#_Toc528612569)

[Table 65: Update Room - Field, Buttons, Hyperlinks 85](#_Toc528612570)

[Table 66: View Room - Field, Buttons, Hyperlinks 85](#_Toc528612571)

[Table 67: Delete Room - Field, Buttons, Hyperlinks 85](#_Toc528612572)

[Table 68: Add member into Room - Field, Buttons, Hyperlinks 85](#_Toc528612573)

[Table 69: Delete member of Room - Field, Buttons, Hyperlinks 85](#_Toc528612574)

[Table 70: Entity Data dictionary 86](#_Toc528612575)

# List of Figure

[Figure 1: Scrum Methodology 15](#_Toc528612576)

[Figure 2: System Overview Use Case (1) 21](#_Toc528612577)

[Figure 3: System Overview Use Case (2) 22](#_Toc528612578)

[Figure 4: <Unauthorized User> Overview Use Case 22](#_Toc528612579)

[Figure 5: <Unauthorized User> Register Use Case 23](#_Toc528612580)

[Figure 6: <Unauthorized User> Login Use Case 24](#_Toc528612581)

[Figure 7: <Authorized User> Overview Use Case 25](#_Toc528612582)

[Figure 8: <Authorized User> Logout Use Case 26](#_Toc528612583)

[Figure 9: <Authorized User> View Notification Use Case 27](#_Toc528612584)

[Figure 10: <Authorized User> Update Information Use Case 28](#_Toc528612585)

[Figure 11: <House Owner> Overview Use Case 30](#_Toc528612586)

[Figure 12: <House Owner> Create Room Use Case 30](#_Toc528612587)

[Figure 13: <House owner> Update Room Use Case 32](#_Toc528612588)

[Figure 14: <House owner> Delete Room Use Case 34](#_Toc528612589)

[Figure 15: <House owner> Add Member into Room Use Case 36](#_Toc528612590)

[Figure 16: <House owner> Delete Member from Room Use Case 39](#_Toc528612591)

[Figure 17: <Room Master> Overview Use Case 41](#_Toc528612592)

[Figure 18: <Room Master> View Suggested Partner List Use Case 41](#_Toc528612593)

[Figure 19: <Room Master> Create Post Use Case 43](#_Toc528612594)

[Figure 20: <Room Master> Update Post Use Case 44](#_Toc528612595)

[Figure 21: <Room Master> Delete Post Use Case 47](#_Toc528612596)

[Figure 22: <Room Master> View Partner Detail Use Case 48](#_Toc528612597)

[Figure 23: <Member> Overview Use Case 50](#_Toc528612598)

[Figure 24: <Member> View Suggested Room Posts List Use Case 51](#_Toc528612599)

[Figure 25:< Member> Create partner Post Use Case 52](#_Toc528612600)

[Figure 26:< Member> Update partner Post Use Case 54](#_Toc528612601)

[Figure 27: <Member> Delete Partner Post Use Case 56](#_Toc528612602)

[Figure 28: <Member> Negotiate Use Case 57](#_Toc528612603)

[Figure 29: <Admin> Overview Use Case 59](#_Toc528612604)

[Figure 30: <Admin> Manage Room Use Case 59](#_Toc528612605)

[Figure 31: <Admin> Manage Post Use Case 60](#_Toc528612606)

[Figure 32: <Admin> Approve Room Use Case 62](#_Toc528612607)

[Figure 33: <Admin> Manage User Use Case 63](#_Toc528612608)

[Figure 34: <Admin> View User Use Case 64](#_Toc528612609)

[Figure 35: Conceptual Diagram 67](#_Toc528612610)

[Figure 36: System Architecture Design 69](#_Toc528612611)

[Figure 37: Component Diagram 69](#_Toc528612612)

[Figure 38: Class Diagram 71](#_Toc528612613)

[Figure 39: Sequence Diagram - <Unauthorized> Register 76](#_Toc528612614)

[Figure 40: Sequence Diagram - <Unauthorized> Login 77](#_Toc528612615)

[Figure 41: Sequence Diagram - <House owner> Create Room 77](#_Toc528612616)

[Figure 42: Sequence Diagram - <House owner> Update Room 78](#_Toc528612617)

[Figure 43: Sequence Diagram - <House owner> Delete Room 78](#_Toc528612618)

[Figure 44: Sequence Diagram - <House owner> View Single Room 79](#_Toc528612619)

[Figure 45: Sequence Diagram - <House owner> Add Member into Room 79](#_Toc528612620)

[Figure 46: Sequence Diagram - <House owner> Delete Members of Room 80](#_Toc528612621)

[Figure 47: Sequence Diagram - <Room Master> Create Room Post 80](#_Toc528612622)

[Figure 48: Sequence Diagram - < Room Master > Delete Room 81](#_Toc528612623)

[Figure 49: Sequence Diagram - <Room Master> Update Room Post 81](#_Toc528612624)

[Figure 50: Sequence Diagram - <Member> Create Partner Post 82](#_Toc528612625)

[Figure 51: Sequence Diagram - <Member> Update Partner Post 82](#_Toc528612626)

[Figure 52: Sequence Diagram - < Member > Delete Partner Post 83](#_Toc528612627)

[Figure 53: Sequence Diagram - <Authorized> Search Post 83](#_Toc528612628)

[Figure 54: Sequence Diagram - <Authorized> View Suggest List 83](#_Toc528612629)

[Figure 55: <Admin> Login Interface 83](#_Toc528612630)

[Figure 56: <Admin> Manage User Interface 83](#_Toc528612631)

[Figure 57: <Admin> Manage Room Interface 83](#_Toc528612632)

[Figure 58: <Admin> Manage Post Interface 83](#_Toc528612633)

[Figure 59: <Admin> View User List Interface 84](#_Toc528612634)

[Figure 60: <Admin> View Room List Interface 84](#_Toc528612635)

[Figure 61: <Admin> View Post List Interface 84](#_Toc528612636)

[Figure 62: <Unauthorized User> Register Interface 84](#_Toc528612637)

[Figure 63: <Unauthorized User> Register Interface 84](#_Toc528612638)

[Figure 64: <Authorized User> View Setting Interface 84](#_Toc528612639)

[Figure 65: <Authorized User> Update Information Interface 84](#_Toc528612640)

[Figure 66: <Authorized User> View Profile Interface 85](#_Toc528612641)

[Figure 67: <Authorized User> View Notification Interface 85](#_Toc528612642)

[Figure 68: <House owner> Create Room Interface 85](#_Toc528612643)

[Figure 69: < House owner > Update Room Interface 85](#_Toc528612644)

[Figure 70: < House owner > View Room Interface 85](#_Toc528612645)

[Figure 71: < House owner > Delete Room Interface 85](#_Toc528612646)

[Figure 72: < House owner > Add member into Room Interface 85](#_Toc528612647)

[Figure 73: < House owner > Delete member of Room Interface 85](#_Toc528612648)

[Figure 74: Entity Relationship diagram 86](#_Toc528612649)

[Figure 75: Flowchart 87](#_Toc528612650)

# 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 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 Guest 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 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.
* Room Master can create and manage their own post.
* System matching Room Master with utilities and suggest appropriate room for guest.
* Guest can view information of room of lodger.
* Room Master 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 Room Master)
* View suggested partner list.
* Manage post.
* View partner list.
* View post.
* Mobile application (for guest)
* View suggest Room Master 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 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 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 rom master, guest, 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 guest.

**- Member:**

* Member can create utilities and system can match their utilities with location room master go to make a suggestion.
* Member 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 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.

#### 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

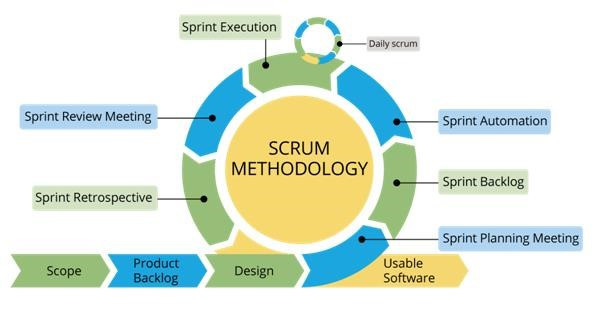


Figure 1: Scrum Methodology

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 tasks  - 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 | Swift 3 |
| Back-end | Intellij |  |
| Web server |  |  |
| 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

|  |  |  |
| --- | --- | --- |
| **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 7: Product Backlog

### Sprint Backlog

Refer to “Sprint Backlog” folder.

### Meeting Minutes

Refer to “Meeting Minutes” folder.

## 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 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
* 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 member
* Remove room member

### 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 partner list
* View partner list

### Member Requirement

Member is an authorized user who is looking for a room

Member can do the following functions:

* Search
* Create partner post
* Update partner post
* View partner post
* Delete partner post
* View post list
* View suggested post 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.

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

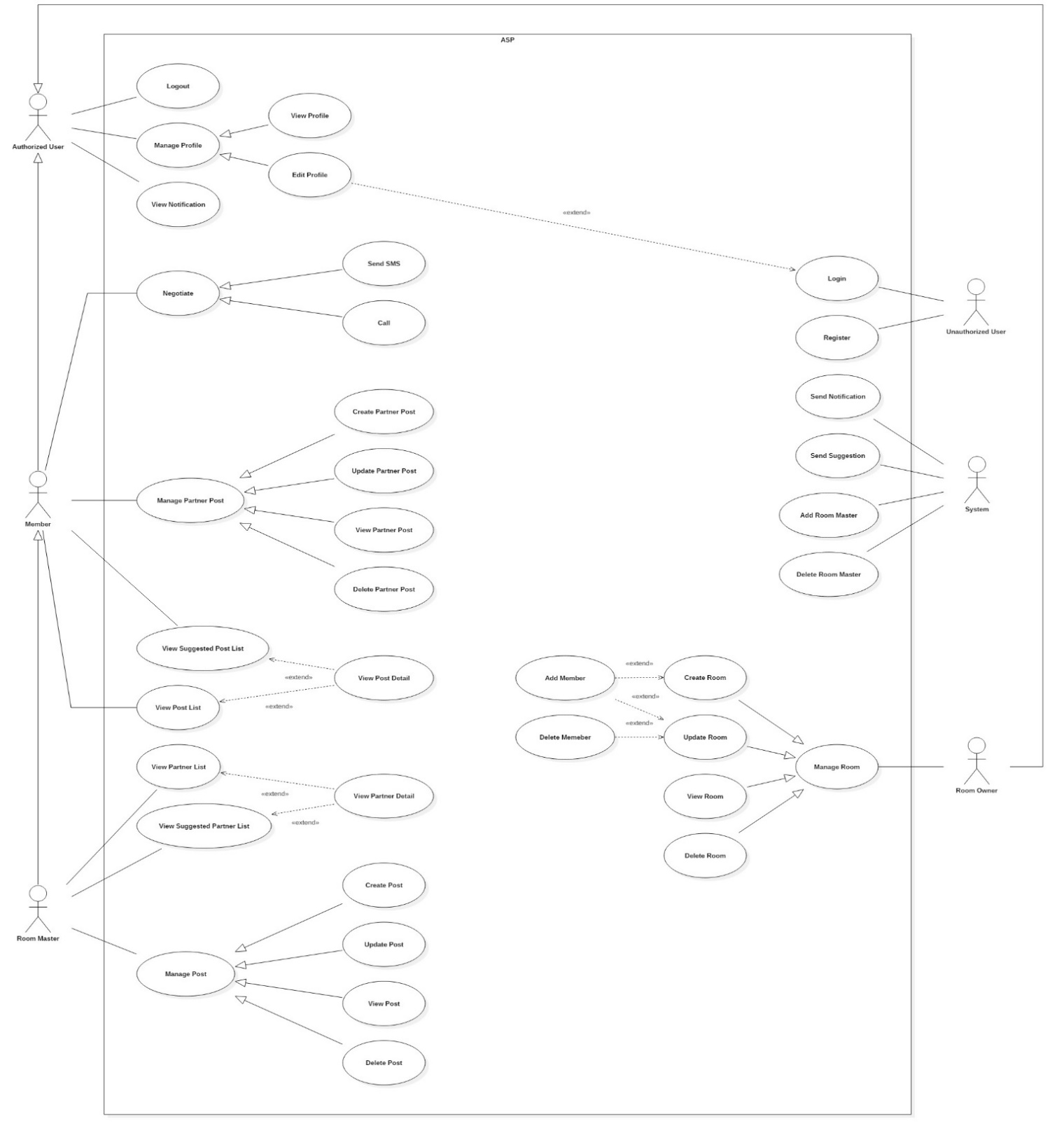


Figure 2: System Overview Use Case (1)

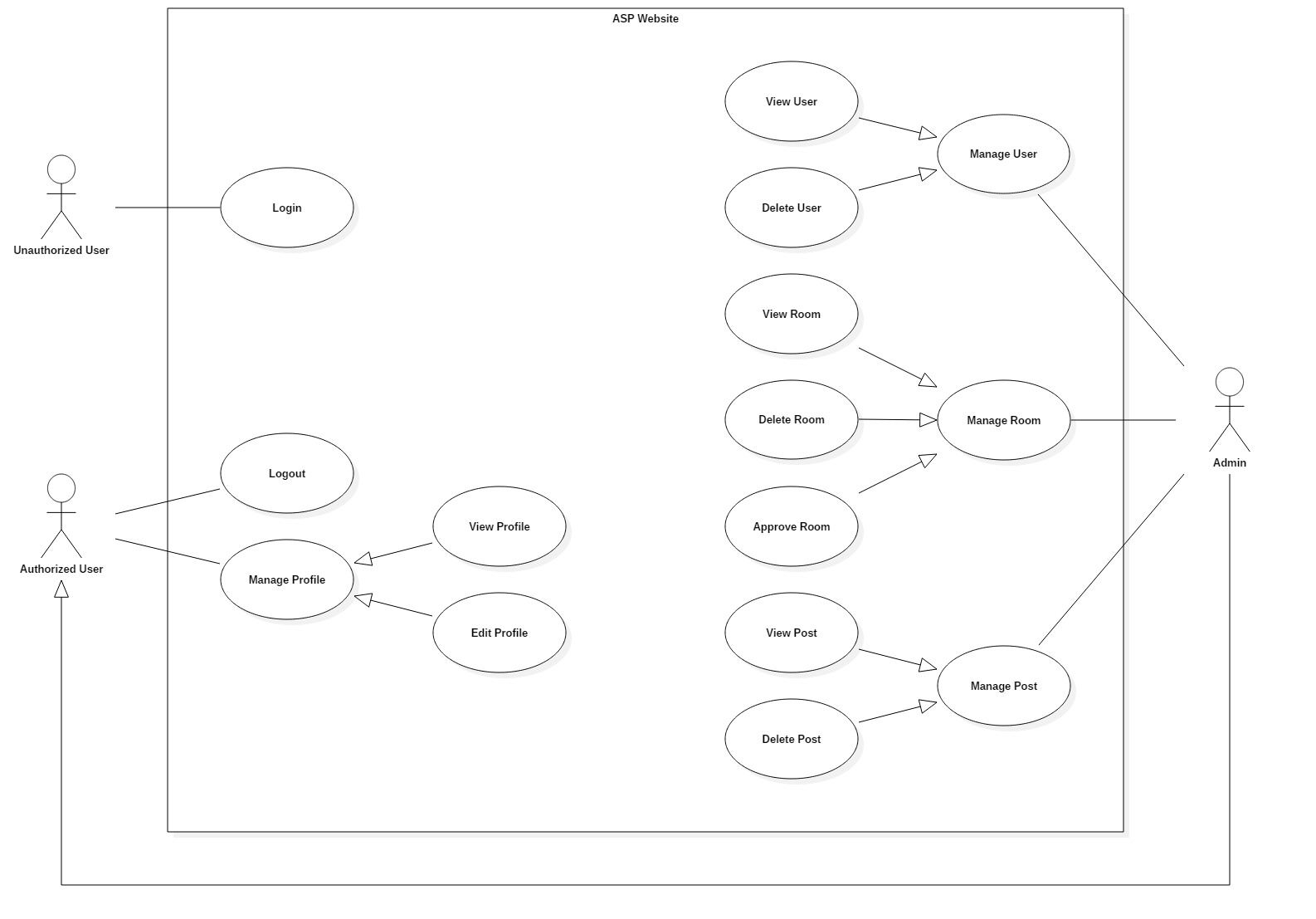


Figure 3: System Overview Use Case (2)

### List of Use Case

#### <Unauthorized User> Overview Use Case

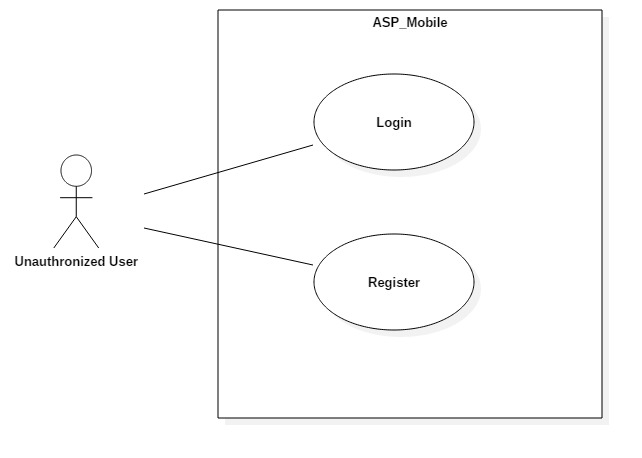


Figure 4: <Unauthorized User> Overview Use Case

##### <Unauthorized User> Register

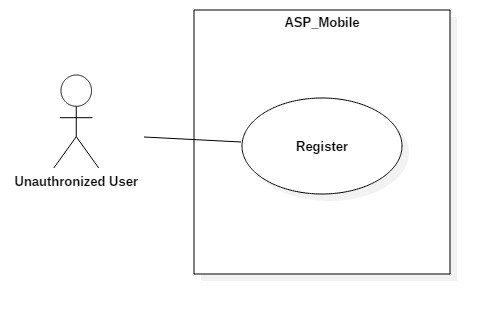


Figure 5: <Unauthorized User> Register Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP1** | | | |
| **Use Case No.** | UC\_ ASP1 | **Use Case Version** | 0.2 |
| **Use Case Name** | Register | | |
| **Author** | Nguyễn Đỗ Minh Đức | | |
| **Date** | 17/10/2018 | **Priority** | High |
| **Actor:**   * Unauthorized User   **Summary:**   * This use case allows unauthorized user to create account.   **Goal:**   * Create account then login to the system.   **Triggers:**   * Unauthorized click on “Đăng ký” button in first launch screen.   **Preconditions:**   * User have not logged in to system.   **Post conditions:**   * Success: Actor creates an account successfully. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Actor goes to register screen | System requires information:   * “Tài khoản”: Text input * “Mật khẩu”: Text input * “Email”: Text input * “Giới tính”: Radio button * “Số điện thoại”: Text input * “Ngày sinh”: Date time picker * “Ảnh đại diện”: Image upload * “Đăng ký”button. | | 2 | Actor inputs informations | [Exception no.1] | | 3 | Actor sends command to login  (Press “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 input 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:**   * Username must be in range of 6-35 characters * Password must be in range of 4-30 characters. * Password must be encrypted before being sent to server. * User can choose the role (House owner/member). * Each role have different register form. * User have not logged in to system. | | | |

Table 8: <Unauthorized User> Register specification

##### <Unauthorized User> Login

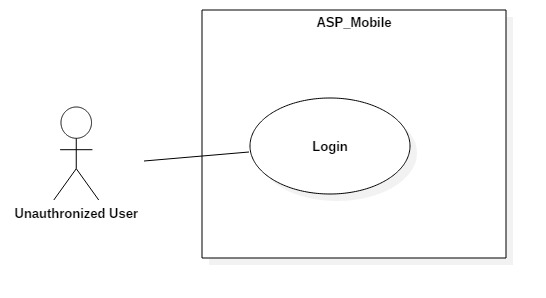


Figure 6: <Unauthorized User> Login Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP2** | | | |
| **Use Case No.** | UC\_ ASP2 | **Use Case Version** | 0.2 |
| **Use Case Name** | Login | | |
| **Author** | Nguyễn Đỗ Minh Đức | | |
| **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 press on “Đăng nhập” button in first launch screen.   **Preconditions:**   * User have not logged in to system.   **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:   * “Tài khoản”: Text input * “Mật khẩu”: Text input * “Đăng nhập” button. | | 2 | Actor inputs information  (username and password) | [Exception no.1] | | 3 | Actor sends command to login  (Press “Login” button after input login form). | [Exception no.2]  System navigates to login screen. |   **Alternative Scenario:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | 1 | Actorinput 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:**   * Username must be in range of 5-35 characters * Password must be in range of 4-30 characters. * User have not logged in to system. | | | |

Table 9: <Unauthorized User> Login specification

#### <Authorized User> Overview Use

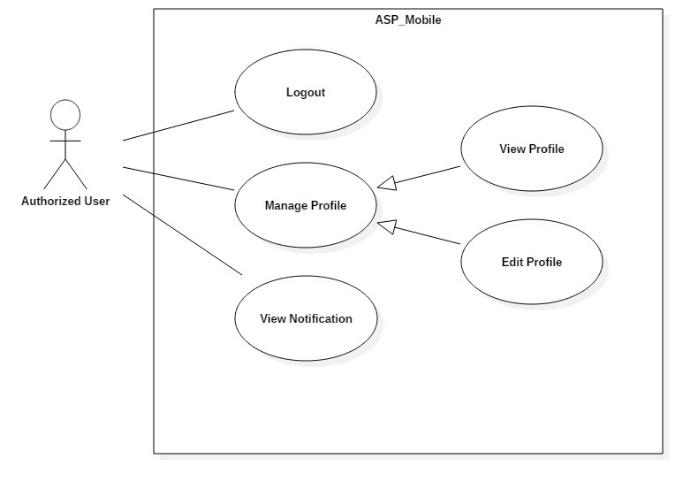


Figure 7: <Authorized User> Overview Use Case

##### <Authorized User> Logout

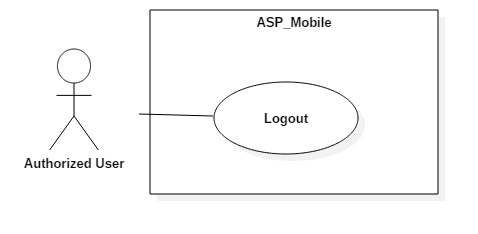


Figure 8: <Authorized User> Logout Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP3** | | | |
| **Use Case No.** | UC\_ ASP3 | **Use Case Version** | 0.2 |
| **Use Case Name** | Nguyễn Đỗ Minh Đức | | |
| **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:**   * Press “Tài khoản” button in home screen. * Press “Đăng Xuất” button.   **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 “Tài khoản” button. | User navigated to setting screen | | 2 | Actor press “Đăng xuất” button. | User session is removed,  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 10: <Authorized User> Logout specification

##### <Authorized User> View Notification

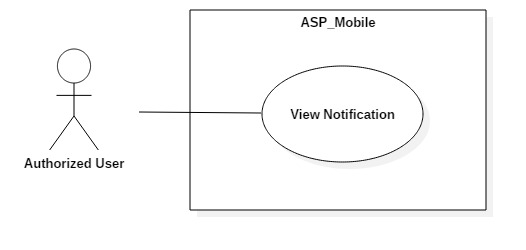


Figure 9: <Authorized User> View Notification Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **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:**   * 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 press on “Thông báo” button. | System navigates to notification screen with list of notifications. |   **Alternative Scenario:**N/A  **Exceptions:** N/A  **Relationships:** UC\_ ASP24, UC\_ ASP9, UC\_ ASP10  **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”. | | | |

Table 11: <Authorized User> View Notification specification

##### <Authorized User> Update Info

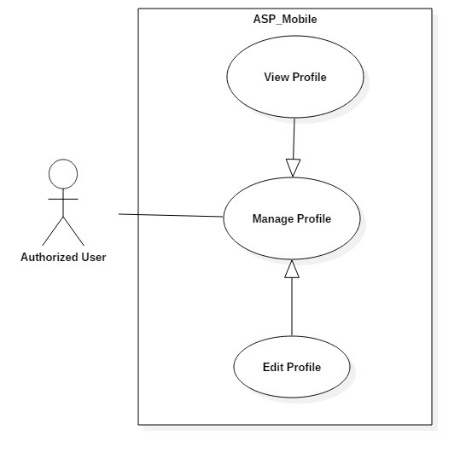


Figure 10: <Authorized User> Update Information Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP5** | | | |
| **Use Case No.** | UC\_ ASP5 | **Use Case Version** | 0.2 |
| **Use Case Name** | Update User Info | | |
| **Author** | Nguyễn Đỗ Minh Đức | | |
| **Date** | 17/10/2018 | **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 press on account field in setting screen. * Actor press on “Chỉnh sửa” button.   **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 “Tài khoản” 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”: Checkmark * “Lưu” button | | 3 | Fill form and 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 user 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. * Informations are updated must be matched the defined informations structure in system. * Email is not null. | | | |

Table 12: <Authorized User> Update Information specification

#### <House Owner> Overview Use Case

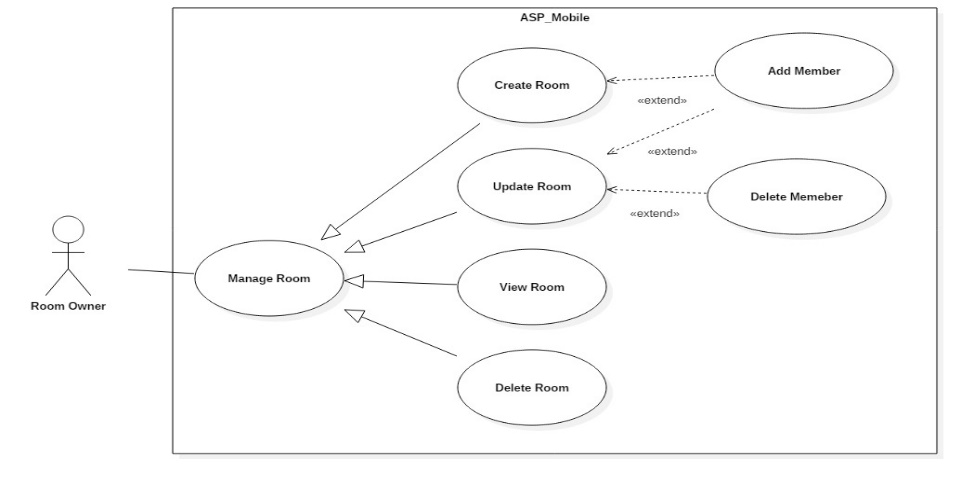


Figure 11: <House Owner> Overview Use Case

##### <House Owner> Create Room

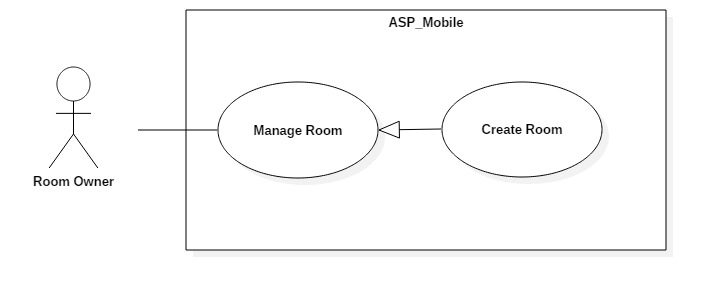


Figure 12: <House Owner> Create Room Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **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:**   * Press “Tạo phòng” button in home screen.   **Preconditions:**   * User logged in the system. * User is approved to be house owner by admin.   **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.  - “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 | | 3 | Fill form and 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. * User must be house owner role to create room. | | | |

Table 13: <House Owner> Create Room specification

##### <House owner> Update Room

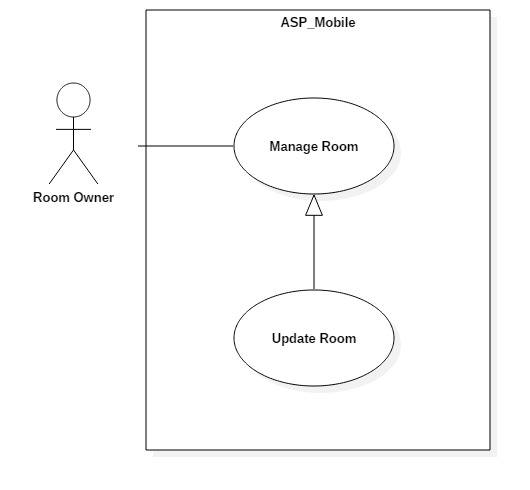


Figure 13: <House owner> Update Room Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **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 press on certain room in account screen. * Press “Chỉnh sửa” 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 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”  - “Mô tả”  - “Ngày đăng bài”  - “Tiện ích”  - “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:  - “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 | | 6 | Fill form 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’s information is updated must be matched the defined room’s information structure. * Room name is not null. * Room price is not null. * Room address is not null. | | | |

Table 14: <House owner> Update Room specification

##### <House owner> Delete Room

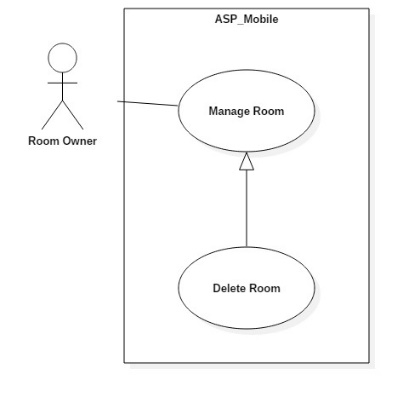


Figure 14: <House owner> Delete Room Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **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 press on certain room in account screen. * Press “Xoá phòng” button.   **Preconditions:**   * User logged in the system. * Rooms have been 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”  - “Mô tả”  - “Ngày đăng bài”  - “Tiện ích”  - “Thành viên trong phòng” table  - “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. |   **Alternative Scenario:**N/A  **Relationships:** N/A  **Business Rules:**   * User already logged in the system. * Room already uploaded to the system. * User must be house owner role. | | | |

Table 15: <House owner> Delete Room specification

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

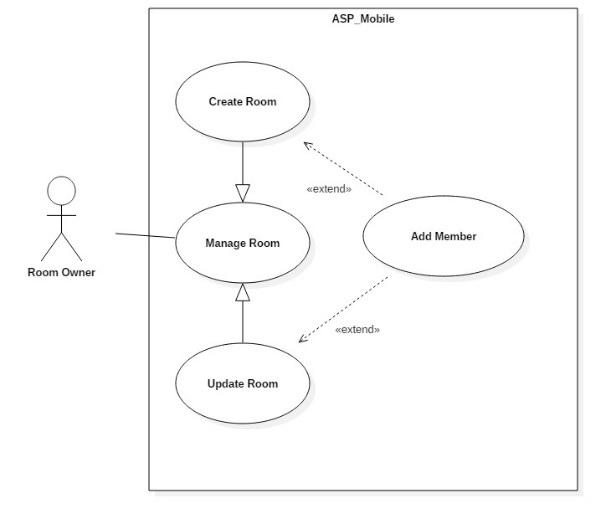


Figure 15: <House owner> Add Member into Room Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **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 press on certain room in account screen. * Actor press “Thêm thành viên” 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 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”  - “Diện tích”  - “Địa chỉ”  - “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. * User must be house owner role. * Username must have been in system. * Date in must be later than date create room. * First member is added in room automatically become room master. | | | |

Table 16: <House owner> Add Member into Room specification

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

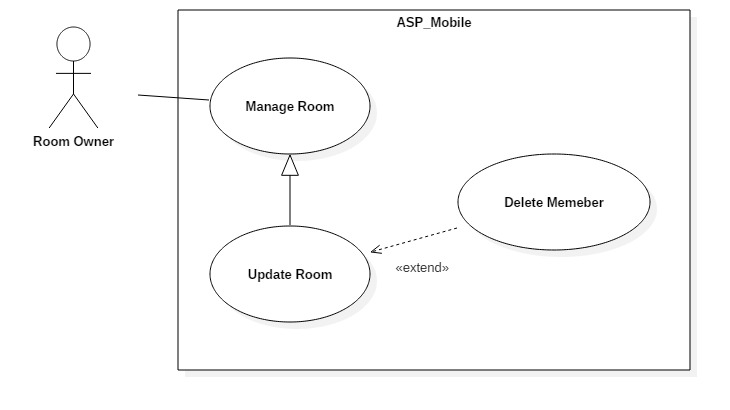


Figure 16: <House owner> Delete Member from Room Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **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 press on certain room in account screen. * Actor Press “Xoá thành viên” 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 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”  - “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”  - “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. * Member already added into room. * User must be house owner role. * If house owner remove the room master. System automatically promote the next member to room master. | | | |

Table 17: <House owner> Delete Member from Room specification

#### <Room Master> Overview Use Case

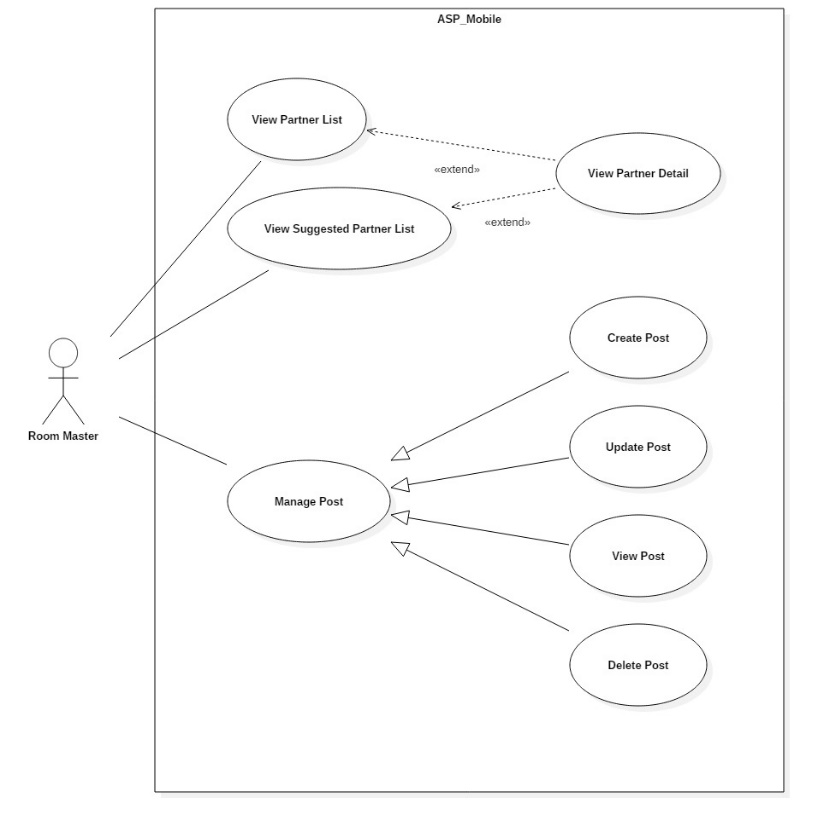


Figure 17: <Room Master> Overview Use Case

##### <Room Master> View Suggested List

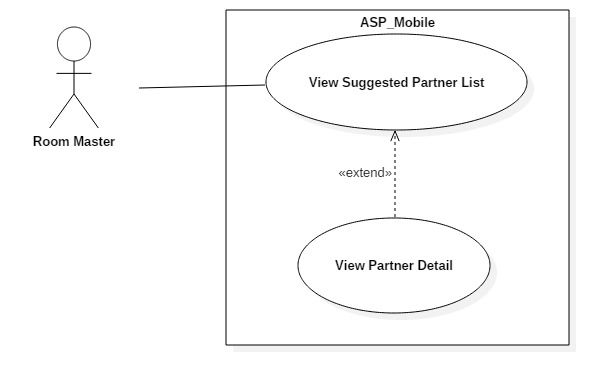


Figure 18: <Room Master> View Suggested Partner List Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP11** | | | |
| **Use Case No.** | UC\_ ASP11 | **Use Case Version** | 0.2 |
| **Use Case Name** | View Suggested List | | |
| **Author** | Nguyễn Quang Nhật | | |
| **Date** | 17/10/2018 | **Priority** | High |
| **Actor:**   * Room Master   **Summary:**   * This use case allows room master to view suggested partners list.   **Goal:**   * To view suggested partners list.   **Triggers:**   * Actor logged into system and goes to home screen.   **Preconditions:**   * User logged in the system.   **Post conditions:**   * Success: User can view suggested partner list. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Actor press “Trang chủ” button. | System displays home screen with:  - “Những bài gần bạn”: Horizontal scroll view.  - “Người dùng chung quận”: Vertical 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 to home screen, system automatically calculate the distance based on the current location. * If they do not accept location access request from the application, display all newest accommodations in home screen. | | | |

Table 18: <Room Master> View Suggested Partners List Specification

##### <Room Master> Create Post

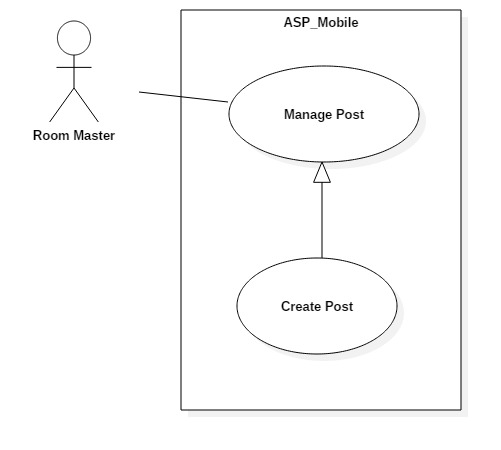


Figure 19: <Room Master> Create Post Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP12** | | | |
| **Use Case No.** | UC\_ ASP12 | **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 post.   **Goal:**   * To create new post.   **Triggers:**   * Actor press on “Tạo bài tìm bạn” 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 “Đăng bài tìm bạn” 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 “Đăng bài” 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 some fields with requirement. | 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. * Post number of partner is not null. * Post price is not null. * User must be room master role. * User must be added into room. | | | |

Table 19: <Room Master> Create Post specification

##### <Room Master> Update Post

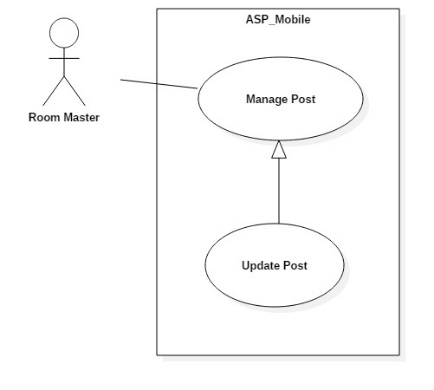


Figure 20: <Room Master> Update Post Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP13** | | | |
| **Use Case No.** | UC\_ ASP13 | **Use Case Version** | 0.2 |
| **Use Case Name** | Update Post | | |
| **Author** | Nguyễn Quang Nhật | | |
| **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 press on “Chỉnh sửa” button in post view.   **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 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”  - “Số điện thoại”  - “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” ”: Text input  - “Số điện thoại”: Text input  - “Giá phòng” ”: Text input  - “Diện tích”  - “Địa chỉ”  - “Ngày đăng bài”  - “Mô tả” ”: Text input  - “Lưu” button | | 6 | Fill form and 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 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. * Informations are updated must be matched the defined informations structure in system. | | | |

Table 20: <Room Master> Update Post specification

##### <Room Master> Delete Post

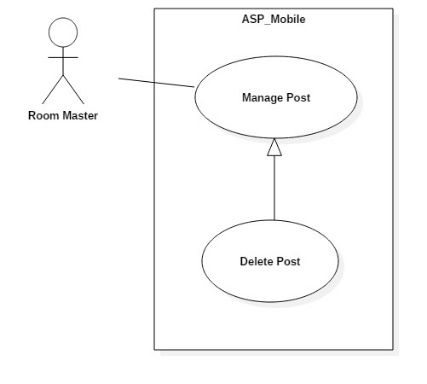


Figure 21: <Room Master> Delete Post Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP14** | | | |
| **Use Case No.** | UC\_ ASP14 | **Use Case Version** | 0.2 |
| **Use Case Name** | Delete Post | | |
| **Author** | Nguyễn Quang Nhật | | |
| **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 press on “Xoá” button in post view.   **Preconditions:**   * User logged in the system. * Post have been 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”  - “Số điện thoại”  - “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. * User must be room master role. | | | |

Table 21: <Room Master> Delete Post specification

##### <Room Master> View Partner Detail

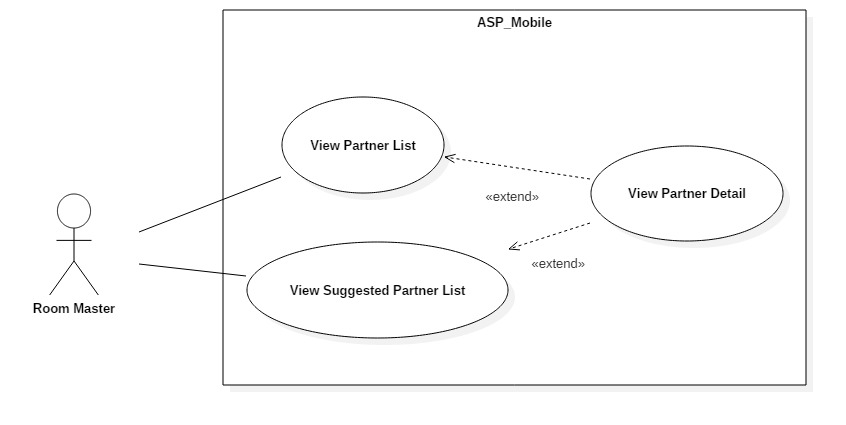


Figure 22: <Room Master> View Partner Detail Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP15** | | | |
| **Use Case No.** | UC\_ ASP15 | **Use Case Version** | 0.2 |
| **Use Case Name** | View Partner Detail | | |
| **Author** | Nguyễn Quang Nhật | | |
| **Date** | 17/10/2018 | **Priority** | High |
| **Actor:**   * Room Master   **Summary:**   * This use case allows room master to view partner detail.   **Goal:**   * To view partner detail.   **Triggers:**   * Actor logged into system and stand in home screen. * Press on any finding rom post.   **Preconditions:**   * User logged in the system.   **Post conditions:**   * Success: User can view suggested partner list. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Actor press “Trang chủ” button. | System displays home screen with:  - “Những bài gần bạn”: Horizontal scroll view.  - “Người dùng chung quận”: Vertical scroll view. | | 2 | Actor press on any finding rom post. | - System will navigate to this post screen with common detail.  - “Tài khoản”  - “Tên bài đăng”  - “Khoảng giá”  - “Số điện thoại”  - “Diện tích”  - “Mô tả”  - “Quận” | | 2 | Actor press on “Tài khoản” field | - System will navigate to partner detail screen with.  - “Ảnh đại diện”  - “Họ và tên”  - “Email”  - “Số điện thoại”  - “Tiện ích”  - “Quận” |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships: USE CASE – UC\_ ASP11**  **Business Rules:**   * User already logged in the system. * User must be room master role. | | | |

Table 22: <Room Master> View Partner Detail Specification

#### <Member> Overview Use Case



Figure 23: <Member> Overview Use Case

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

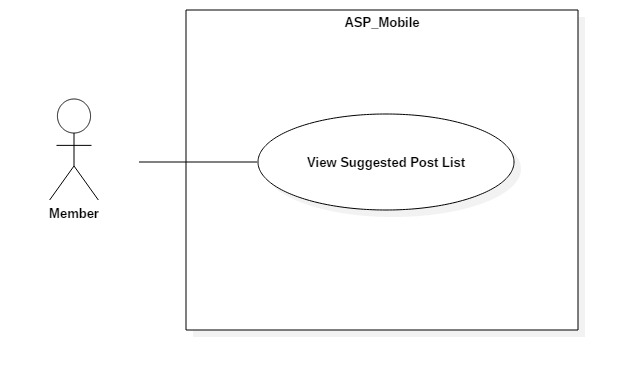


Figure 24: <Member> View Suggested Room Posts List Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP16** | | | |
| **Use Case No.** | UC\_ ASP16 | **Use Case Version** | 0.2 |
| **Use Case Name** | View Suggested Room Posts List | | |
| **Author** | Nguyễn Quang Nhật | | |
| **Date** | 17/10/2018 | **Priority** | High |
| **Actor:**   * Member   **Summary:**   * This use case allows member to view suggested partners list.   **Goal:**   * To view 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 partner list. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Actor press “Trang chủ” button. | System displays home screen with:  - “Những bài gần bạn”: Horizontal scroll view.  - “Người dùng chung quận”: Vertical 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 to home screen, system automatically calculate the distance based on the current location. * If they do not accept location access request from the application, display all newest accommodations in home screen. | | | |

Table 23: <Member> View Suggested Room Posts List Specification

##### <Member> Create Partner Post

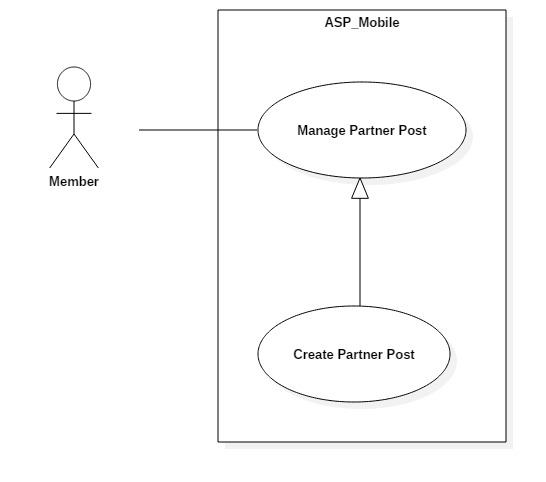


Figure 25:< Member> Create partner Post Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP17** | | | |
| **Use Case No.** | UC\_ ASP17 | **Use Case Version** | 0.2 |
| **Use Case Name** | Create partner Post | | |
| **Author** | Nguyễn Đỗ Minh Đức | | |
| **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”: Checkmark  - “Mô tả”: Text input.  - “Số điện thoại”: Text input  - “Giới tính” switch button  - “Đăng bài” button | | 3 | Fill form and 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 input wrong some fields with requirement. | 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 is not null. * Phone number is not null. * After member create finding rom post. System get user’s utilities and apply in account setting utilities. | | | |

Table 24: <Member> Create partner Post specification

##### <Member> Update Partner Post

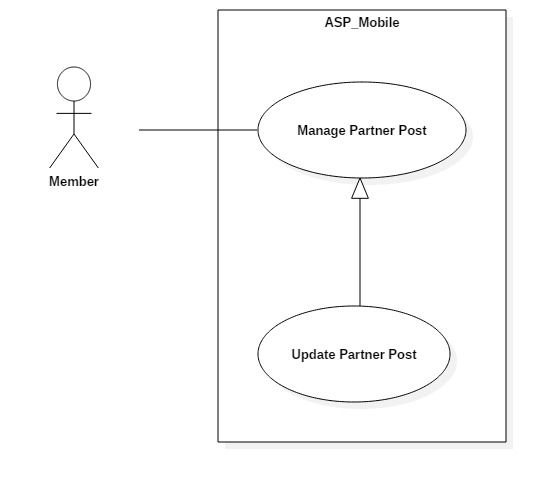


Figure 26:< Member> Update partner Post Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP18** | | | |
| **Use Case No.** | UC\_ ASP18 | **Use Case Version** | 0.2 |
| **Use Case Name** | Update partner Post | | |
| **Author** | Nguyễn Đỗ Minh Đức | | |
| **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”  - “Số điện thoại”  - “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”  - “Số điện thoại”  - “Quận”  - “Lưu” button | | 6 | Fill form and 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 user 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. * Informations are updated must be matched the defined informations structure in system. * User must be member role. * Partner post already uploaded to the system. | | | |

Table 25: <Member> Update partner Post specification

##### <Member> Delete Partner Post

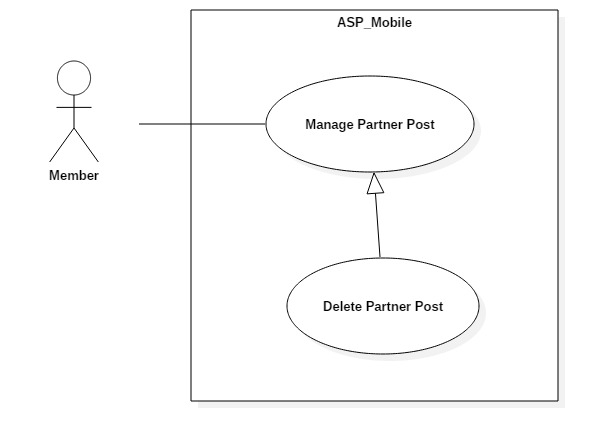


Figure 27: <Member> Delete Partner Post Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP19** | | | |
| **Use Case No.** | UC\_ ASP19 | **Use Case Version** | 0.2 |
| **Use Case Name** | Delete Partner Post | | |
| **Author** | Nguyễn Đỗ Minh Đức | | |
| **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 press on “Xoá” button in post view.   **Preconditions:**   * User logged in the system. * Post have been 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.  - “Tên bài đăng”  - “Khoảng giá”  - “Số điện thoại”  - “Diện tích”  - “Mô tả”  - “Quận”  - “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. * Partner post already uploaded to the system. * User must be member role. | | | |

Table 26: <Member> Delete Partner Post specification

##### <Member> Negotiate

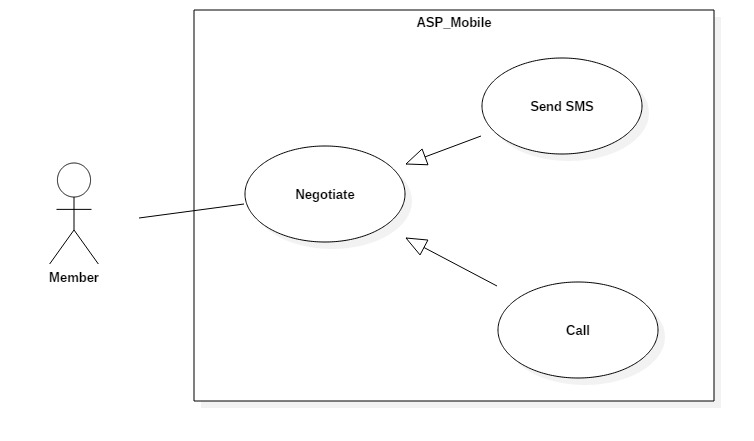


Figure 28: <Member> Negotiate Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP20** | | | |
| **Use Case No.** | UC\_ ASP20 | **Use Case Version** | 0.2 |
| **Use Case Name** | Negotiate | | |
| **Author** | Hồ Công Trình | | |
| **Date** | 17/10/2018 | **Priority** | High |
| **Actor:**   * Member   **Summary:**   * This use case allows member to contact with room master.   **Goal:**   * To make contact with other users.   **Triggers:**   * Actor press on SMS/Call button in single post.   **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 any post in system | System displays details post screen with:  - “Tên bài”  - “Khoảng Giá”  - “Diện tích”  - “Quận”  - “Mô tả”  - “Giới tính”  - “SMS” button  - “Gọi điện” button | | 3 | Actor press on “SMS” button | System open sms default application on mobile with filled phone number in send to box |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** |  |  |  |  | | --- | --- | --- | | 1 | Actor press on “Call” button | System open call default application on mobile with phone number in dialog. |   **Exceptions:** N/A  **Relationships:** N/A  **Business Rules:**   * User already logged in the system. * User must accept phone, sms request from the application. * System automatically display phone number in post to dialog. | | | |

Table 27: <Member> Negotiate specification

#### <Admin> Overview Use Case

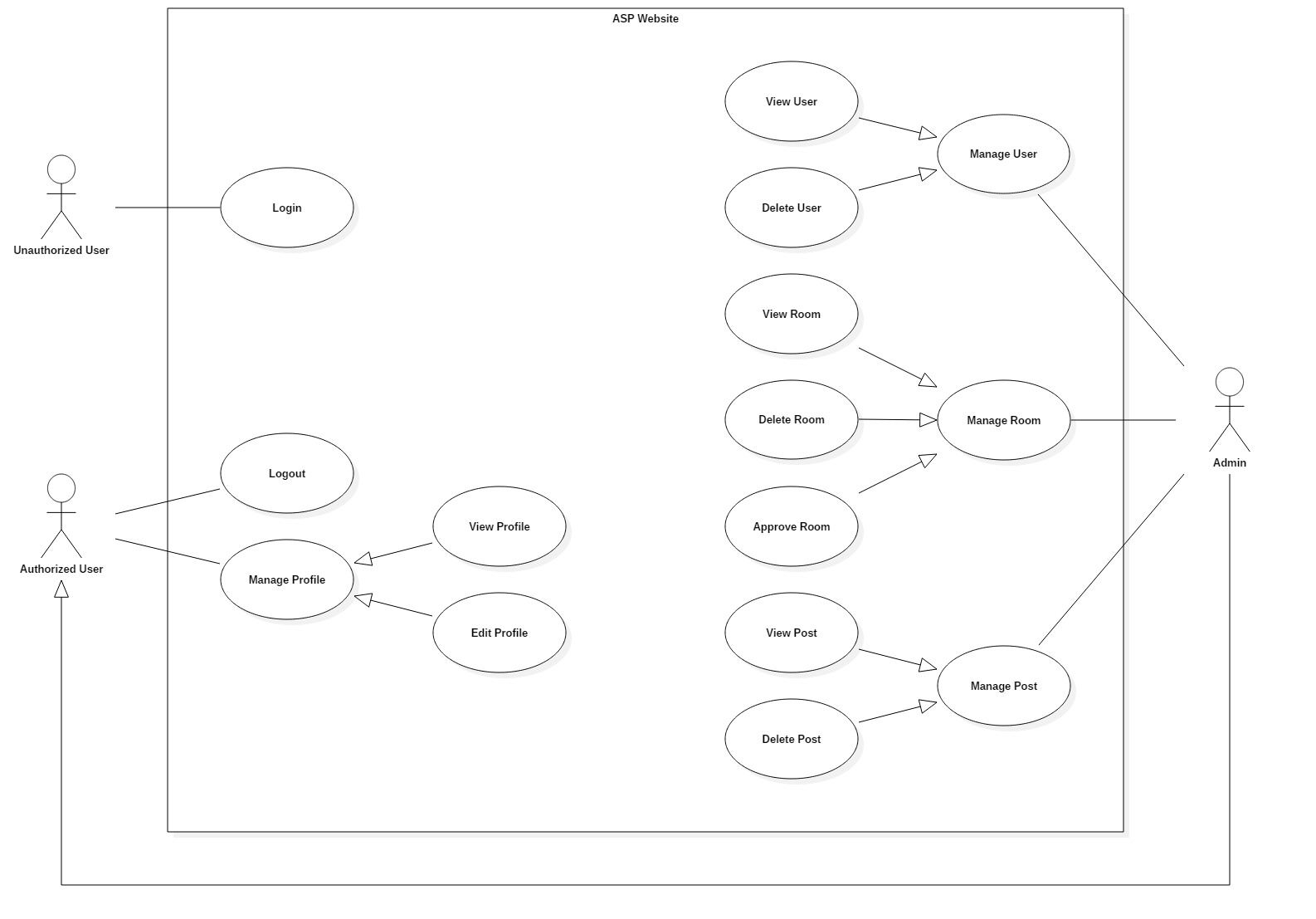


Figure 29: <Admin> Overview Use Case

##### <Admin> Manage Rooms

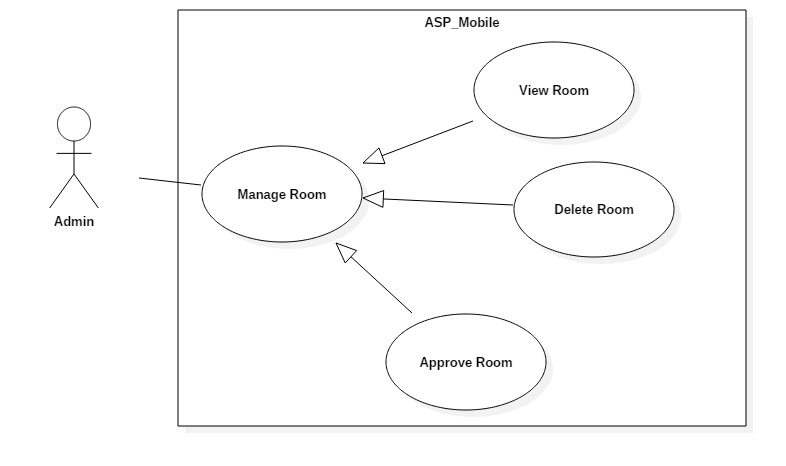


Figure 30: <Admin> Manage Room Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP21** | | | |
| **Use Case No.** | UC\_ ASP21 | **Use Case Version** | 0.2 |
| **Use Case Name** | Manage Room | | |
| **Author** | Hồ Công Trình | | |
| **Date** | 17/10/2018 | **Priority** | High |
| **Actor:**   * Admin   **Summary:**   * This use case allows admin to manage rooms.   **Goal:**   * To manage rooms.   **Triggers:**   * Actor access in web admin. * Actor click on “Phòng” tab in dashboard.   **Preconditions:**   * User logged in the system.   **Post conditions:**   * Success: User can manage all rooms. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Actor clicks “Phòng” tab. | System navigates to “Phòng” page.  System displays “Phòng” page with:  - List of rooms.  - “Quản lý” buttons. |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:** N/A  **Business Rules:**   * Admin already logged in the system. * Rooms are created. * Admin can create, update, view detail, delete room. | | | |

Table 28: <Admin> Manage room specification

##### <Admin> Manage Post

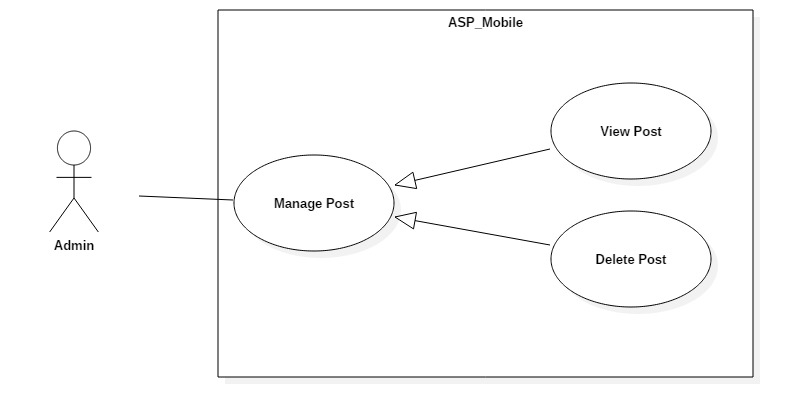


Figure 31: <Admin> Manage Post Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP22** | | | |
| **Use Case No.** | UC\_ ASP22 | **Use Case Version** | 0.2 |
| **Use Case Name** | Manage Post | | |
| **Author** | Hồ Công Trình | | |
| **Date** | 17/10/2018 | **Priority** | High |
| **Actor:**   * Admin   **Summary:**   * This use case allows admin to manage posts.   **Goal:**   * To manage posts.   **Triggers:**   * Actor access in web admin. * Actor click on “Bài viết” tab in dashboard.   **Preconditions:**   * User logged in the system.   **Post conditions:**   * Success: User can manage all posts. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Actor clicks “Bài viết” tab. | System navigates to “Bài viết” page.  System displays “Bài viết” page with:  - List of posts.  - “Quản lý” buttons. |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:** N/A  **Business Rules:**   * Admin already logged in the system. * Posts are created. * Admin can create, update, view detail, delete both type of post. | | | |

Table 29: <Admin> Manage post specification

##### <Admin> Approve Room

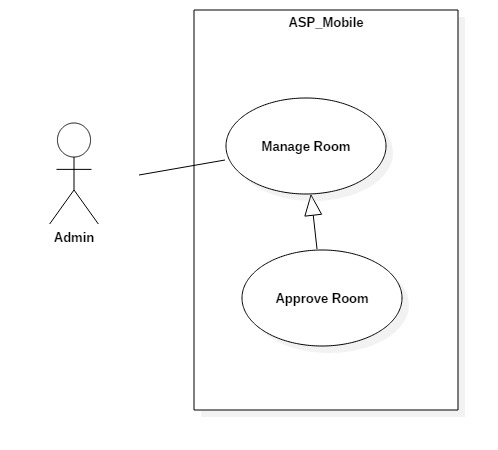


Figure 32: <Admin> Approve Room Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP23** | | | |
| **Use Case No.** | UC\_ ASP23 | **Use Case Version** | 0.2 |
| **Use Case Name** | Approve room | | |
| **Author** | Hồ Công Trình | | |
| **Date** | 17/10/2018 | **Priority** | High |
| **Actor:**   * Admin   **Summary:**   * This use case allows admin to approve room.   **Goal:**   * To approve room.   **Triggers:**   * Actor access in web admin. * Actor click on “Phòng” tab in dashboard. * Actor click on any room then click “Quản lý” button * 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 “Phòng” tab. | System navigates to “Phòng” page.  System displays “Phòng” page with:  - List of rooms.  - Manage buttons in each row of table room list. | | 2 | Actor clicks “Quản lý” button. | System displays room’s detail page with:  - Room’s informations.  - “Approve” button. | | 3 | Actor clicks “Approve” button. | System navigates to “Phòng” page.  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 30: <Admin> Approve room specification

##### <Admin> Manage Users

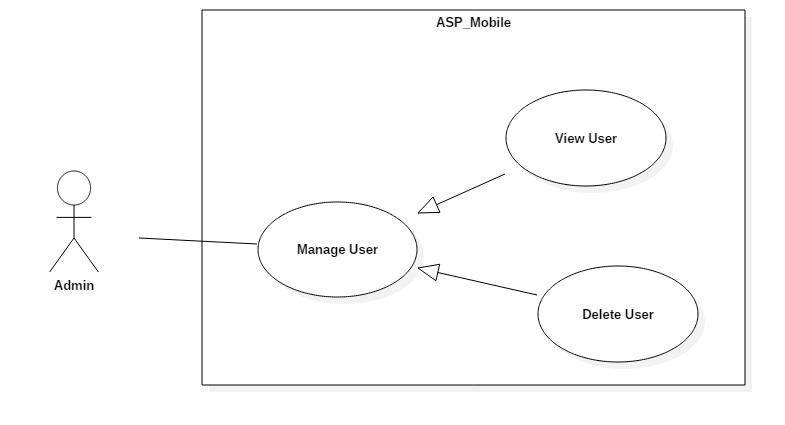


Figure 33: <Admin> Manage User Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP24** | | | |
| **Use Case No.** | UC\_ ASP24 | **Use Case Version** | 0.2 |
| **Use Case Name** | Manage User | | |
| **Author** | Hồ Công Trình | | |
| **Date** | 17/10/2018 | **Priority** | High |
| **Actor:**   * Admin   **Summary:**   * This use case allows admin to manage users.   **Goal:**   * To manage posts.   **Triggers:**   * Actor access in web admin. * Actor click on “Người dùng” tab in dashboard.   **Preconditions:**   * User logged in the system.   **Post conditions:**   * Success: User can manage all posts. * 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” buttons. |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:** N/A  **Business Rules:**   * Admin already logged in the system. * Admin can create, update, view detail, delete user. | | | |

Table 31: <Admin> Manage User specification

##### <Admin> View Users

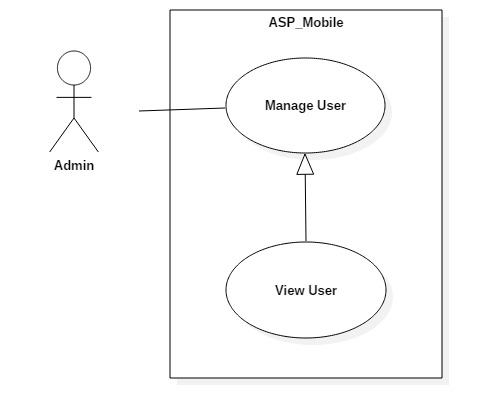


Figure 34: <Admin> View User Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP25** | | | |
| **Use Case No.** | UC\_ ASP25 | **Use Case Version** | 0.2 |
| **Use Case Name** | View User | | |
| **Author** | Hồ Công Trình | | |
| **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 “Người dùng” 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 32: <Admin> View User specification

## 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 do not 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(gia tri thuc te)

* 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

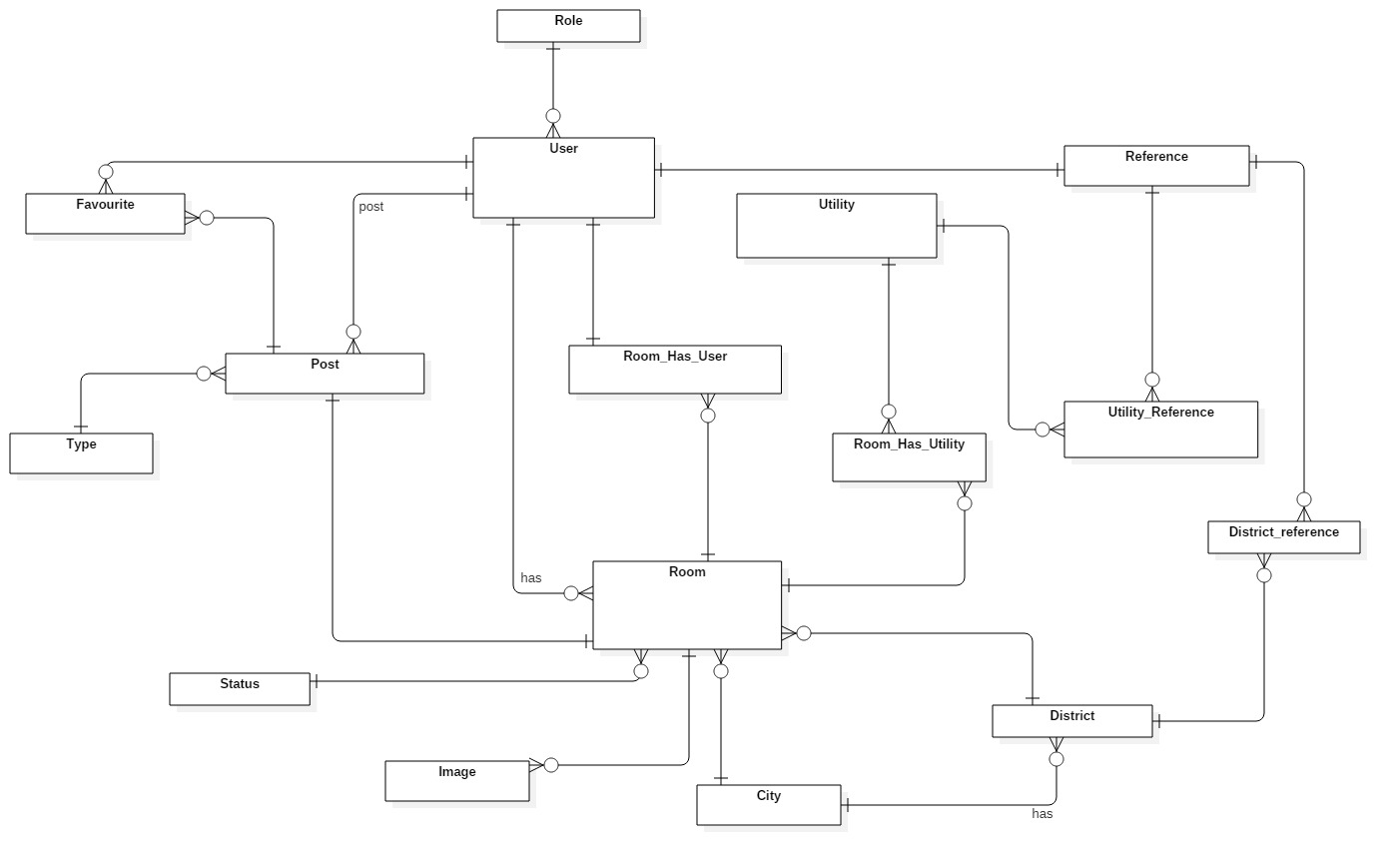


Figure 35: Conceptual Diagram

|  |  |
| --- | --- |
| 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. |
| Favourites | Contains favourite 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 33: Conceptual Entity dictionary

# 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

### System Architecture Design

We choose REST API model as main model because of following advantages:

* **Separation between the client and the server**: the REST protocol totally separates the user interface from the server and the data storage. This has some advantages when making development.
* **Visibility, reliability and scalability**. The separation between client and server has one evident advantage, and that is that each development team can scale the product without too much problem.
* **The REST API is always independent of the type of platform or languages**: the REST API always adapts to the type of syntax or platforms being used, which gives considerable freedom when changing or testing new environments within the development.

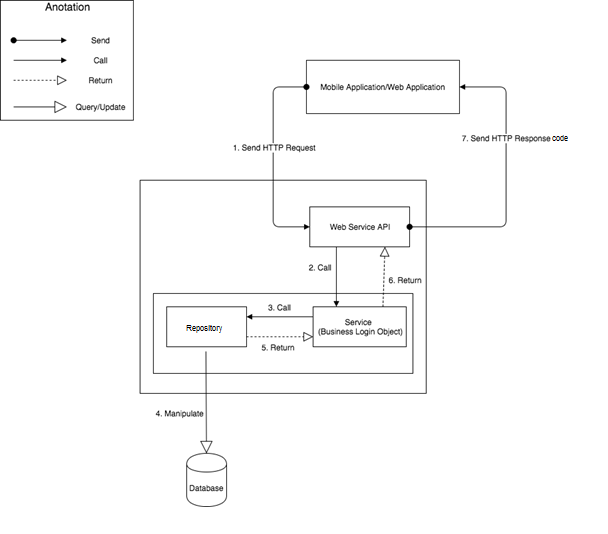


Figure 36: System Architecture Design

## Component Diagram

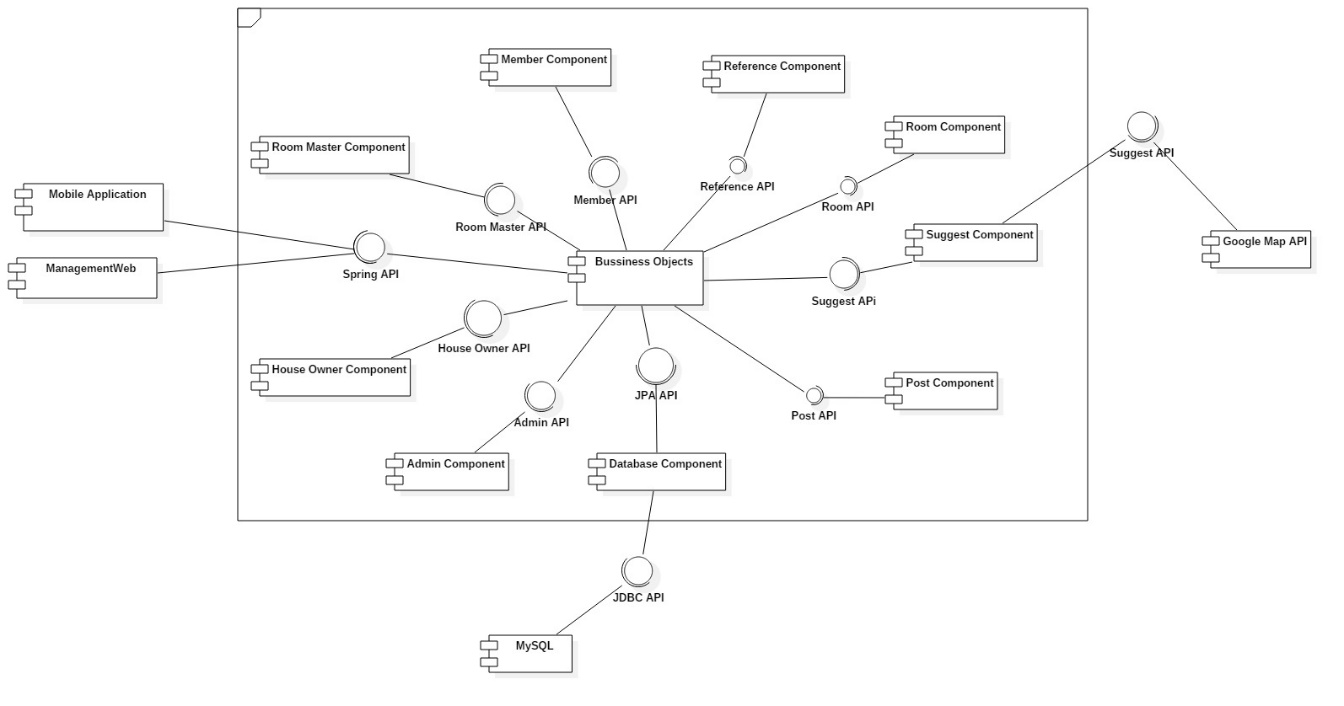


Figure 37: Component Diagram

|  |  |
| --- | --- |
| **COMPONENT DICTIONARY: DESCRIBES COMPONENT** | |
| **Component Name** | **Description** |
| Mobile Application | Web application package: View, Controller |
| Management Web | Web application package: View, Controller |
| House Owner Component | Component to handle house owner activities in the system. |
| Room Master Component | Component to handle room master activities in the system |
| Member Component | Component to handle member activities in the system |
| Admin Component | Component to handle admin activities in the system |
| District Component | Component to handle district activities in the system |
| Room Component | Component to handle room activities in the system |
| Post Component | Component to handle post activities in the system |
| Suggest Component | Component to handle suggest activities in the system |
| Business Object | Common objects to handle domain business operations for each components |
| Database Component | Component to handle interaction between the system and database |
| MySQL | Component to hosting database from a remote location |
| Google Map API | Component to get longitude, latitude for suggest component. |

## Detail Description

### Class Diagram

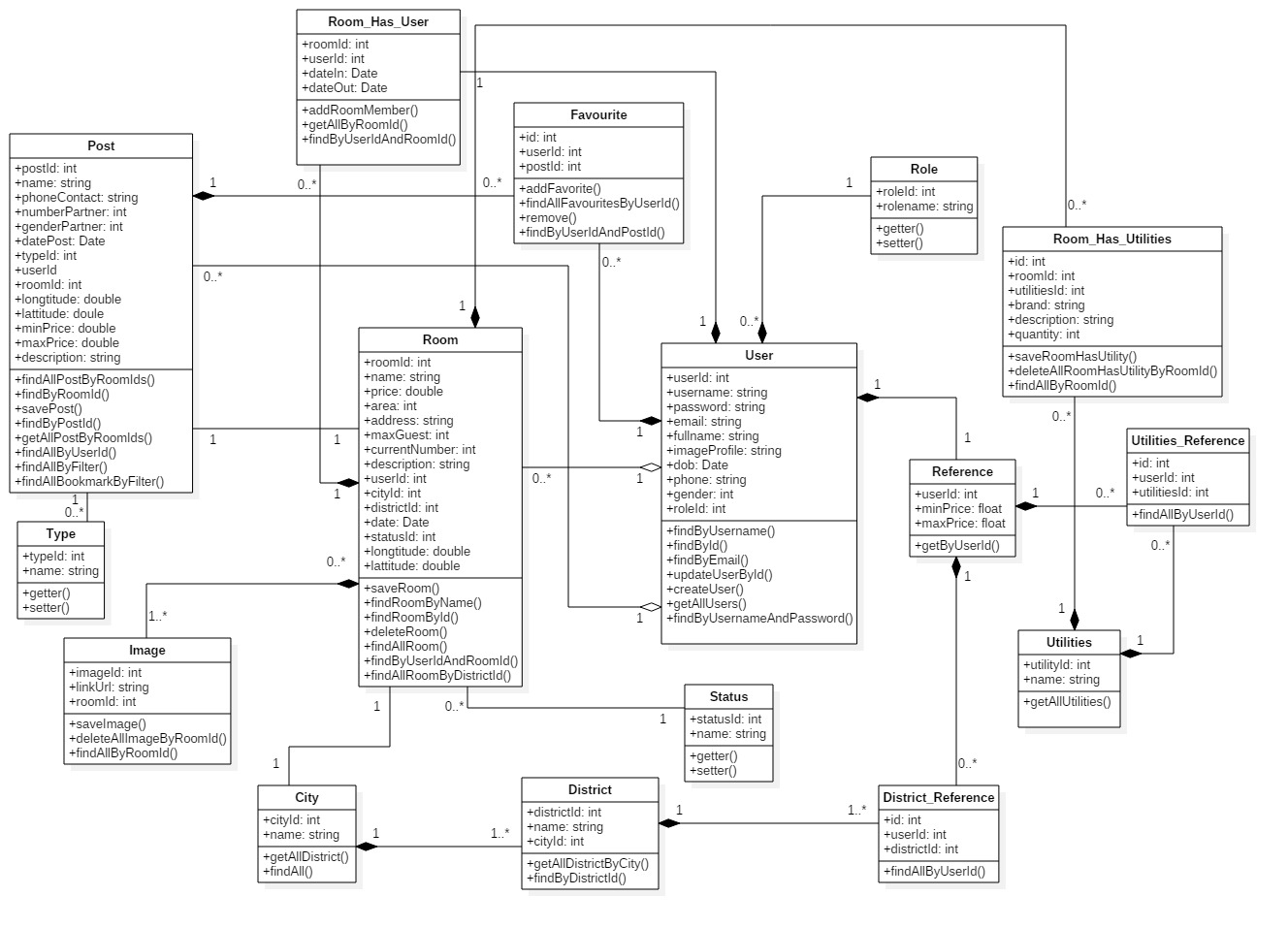


Figure 38: Class Diagram

|  |  |  |
| --- | --- | --- |
| **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 favourite 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 |

Table 34: Class Diagram Explanation

### Class Diagram Explanation

#### User

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| userId | Integer | private |  |
| username | String | private |  |
| password | String | private |  |
| email | String | private |  |
| fullname | String | private |  |
| imageProfile | String | private |  |
| dob | Date | private |  |
| phone | String | private |  |
| gender | Integer | private |  |
| roleId | Integer | private |  |
| **Method** | **Return Type** | **Visibility** | **Description** |
| findByUsername | Attribute type | Public |  |
| findById | Attribute type | Public |  |
| findByEmail |  |  |  |
| updateUserById | void | Public |  |
| createUser | void | Public |  |
| getAllUsers | Attribute type | Public |  |
| findByUsernameAndPassword | Attribute type | Public |  |

Table 35: User Attributes

#### Room

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| roomId | Integer | private |  |
| name | String | private |  |
| price | Double | private |  |
| area | Integer | private |  |
| address | String | private |  |
| maxGuest | Integer | private |  |
| currentNumber | Integer | private |  |
| description | String | private |  |
| userId | Integer | private |  |
| cityId | Integer | private |  |
| districtId | Integer | private |  |
| date | Date | private |  |
| statusId | Integer | private |  |
| longtitude | Double | private |  |
| lattitude | Double | private |  |
| **Method** | **Return Type** | **Visibility** | **Description** |
| saveRoom |  | Public |  |
| findRoomByName |  | Public |  |
| findRoomById |  |  |  |
| deleteRoom |  | Public |  |
| findAllRoom |  | Public |  |
| findByUserIdAndRoomId |  | Public |  |
| findAllRoomByDistrictId |  | Public |  |

Table 36: Room Attributes

#### Post

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| postId | Integer | private |  |
| name | String | private |  |
| phoneContact | String | private |  |
| numberPartner | Integer | private |  |
| genderPartner | Integer | private |  |
| datePost | Date | private |  |
| typeId | Integer | private |  |
| userId | Integer | private |  |
| roomId | Integer | private |  |
| longtitude | Double | private |  |
| lattitude | Double | private |  |
| minPrice | Double | private |  |
| maxPrice | Double | private |  |
| description | String | private |  |
| **Method** | **Return Type** | **Visibility** | **Description** |
| findAllPostByRoomIds |  | Public |  |
| findByRoomId |  | Public |  |
| savePost() |  |  |  |
| findByPostId() |  | Public |  |
| getAllPostByRoomIds() |  | Public |  |
| findAllByUserId() |  | Public |  |
| findAllBookmarkByFilter() |  | Public |  |
| findAllByFilter() |  | Public |  |

Table 37: Post Attributes

#### Role

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| roleId | Integer | private |  |
| rolename | String | private |  |
| **Method** | **Return Type** | **Visibility** | **Description** |
| Getter() |  | Public |  |
| Setter() |  | Public |  |

Table 38: Role Attributes

#### Favourite

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | Integer | private |  |
| userId | Integer | private |  |
| postId | Integer | private |  |
| **Method** | **Return Type** | **Visibility** | **Description** |
| addFavorite() |  | Public |  |
| findAllFavouritesByUserId() |  | Public |  |
| remove() |  | Public |  |
| findByUserIdAndPostId() |  | Public |  |

Table 39: Favourite Attributes

#### Utilities

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| utilitiesId | Integer | private |  |
| name | String | private |  |
| **Method** | **Return Type** | **Visibility** | **Description** |
| getAllUtilities() |  | Public |  |

Table 40: Utility Attributes

#### Reference

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| userId | Integer | private |  |
| minPrice | Double | private |  |
| maxPrice | Double | private |  |
| **Method** | **Return Type** | **Visibility** | **Description** |
| getByUserId() |  | Public |  |

Table 41: Reference Attributes

#### District

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| districtId | Integer | private |  |
| name | String | private |  |
| cityId | Integer | private |  |
| **Method** | **Return Type** | **Visibility** | **Description** |
| getAllDistrictByCity() |  | Public |  |
| findByDistrictId() |  | Public |  |

Table 42: District Attributes

#### City

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| cityId | Integer | private |  |
| name | String | private |  |
| **Method** | **Return Type** | **Visibility** | **Description** |
| getAllDistrict() |  | Public |  |
| findAll() |  | Public |  |

Table 43: City Attributes

#### Image

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| imageId | Integer | private |  |
| linkUrl | String | private |  |
| roomid | Integer | private |  |
| **Method** | **Return Type** | **Visibility** | **Description** |
| saveImage() |  | Public |  |
| deleteAllImageByRoomId() |  | Public |  |
| findAllByRoomId() |  | Public |  |

Table 44: Image Attributes

#### Type

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| typeId | Integer | private |  |
| name | String | private |  |
| **Method** | **Return Type** | **Visibility** | **Description** |
| getter() |  | Public |  |
| setter() |  | Public |  |

Table 45: Type Attributes

#### Status

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| statusId | Integer | private |  |
| name | String | private |  |
| **Method** | **Return Type** | **Visibility** | **Description** |
| getter() |  | Public |  |
| setter() |  | Public |  |

Table 46: Status Attributes

#### Room\_Has\_Utilities

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| Id | Integer | private |  |
| roomId | Integer | private |  |
| utilitiesId | Integer | private |  |
| brand | String | private |  |
| description | String | private |  |
| quantity | Integer | private |  |
| **Method** | **Return Type** | **Visibility** | **Description** |
| saveRoomHasUtility() |  | Public |  |
| deleteAllRoomHasUtilityByRoomId() |  | Public |  |
| findAllByRoomId() |  | Public |  |

Table 47: Room Has Utilities Attributes

#### Utilities\_Reference

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | Integer | private |  |
| userId | Integer | private |  |
| utilitiesId | Integer | private |  |
| **Method** | **Return Type** | **Visibility** | **Description** |
| findAllByUserId() |  | Public |  |

Table 48: Utilities Reference Attributes

#### District\_Reference

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | Integer | private |  |
| userId | Integer | private |  |
| districtId | Integer | private |  |
| **Method** | **Return Type** | **Visibility** | **Description** |
| findAllByUserId() |  | Public |  |

Table 49: District Reference Attributes

#### Room\_Has\_User

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| roomId | Integer | private |  |
| userId | Integer | private |  |
| dateIn | Date | private |  |
| dateOut | Date | private |  |
| **Method** | **Return Type** | **Visibility** | **Description** |
| addRoomMember() |  | Public |  |
| getAllByRoomId() |  | Public |  |
| findByUserIdAndRoomId() |  | Public |  |

Table 50: Room Has User Attributes

### Interactive diagram

#### Sequence Diagram

##### Register

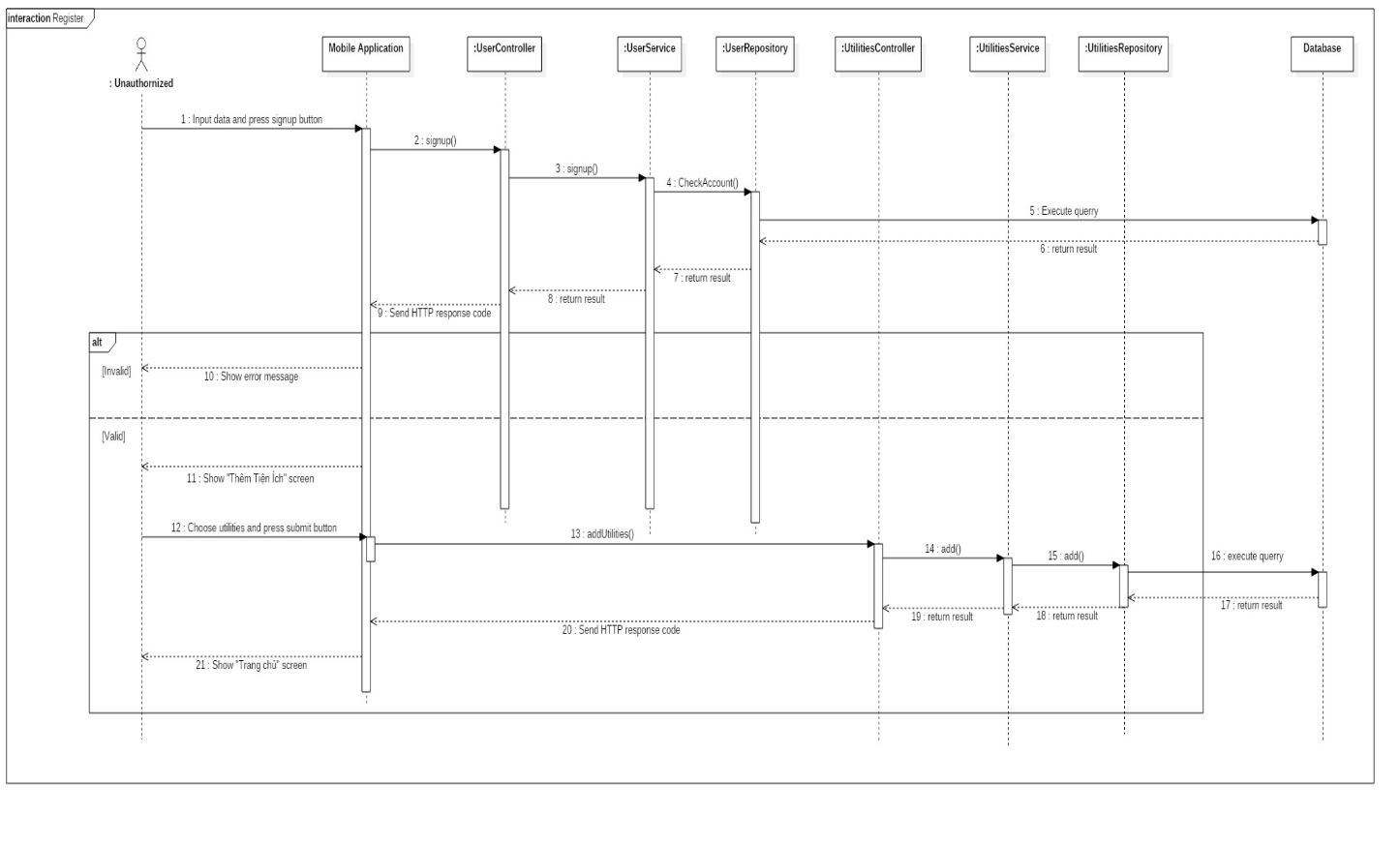


Figure 39: Sequence Diagram - <Unauthorized> Register

##### Login

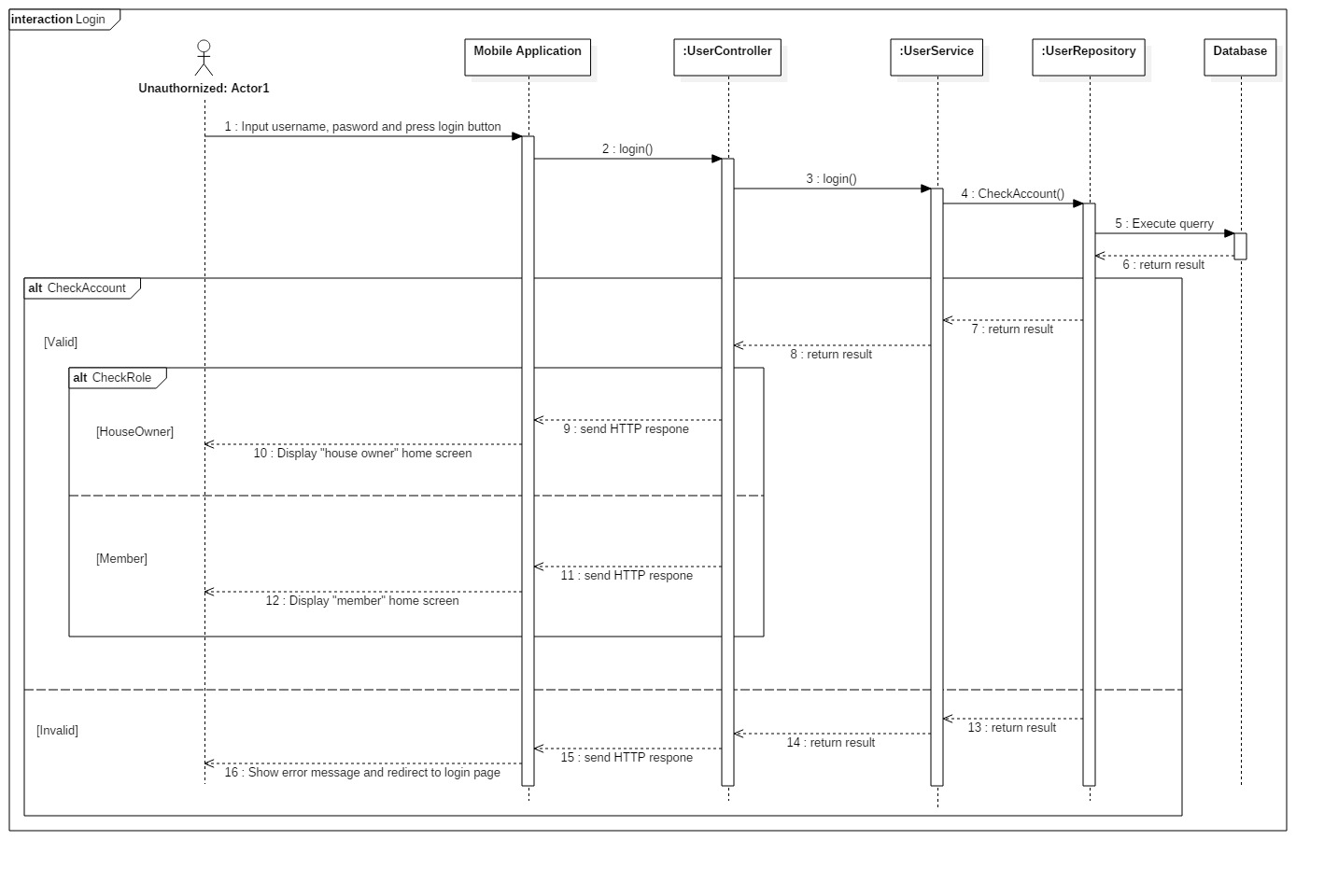


Figure 40: Sequence Diagram - <Unauthorized> Login

##### Create Room

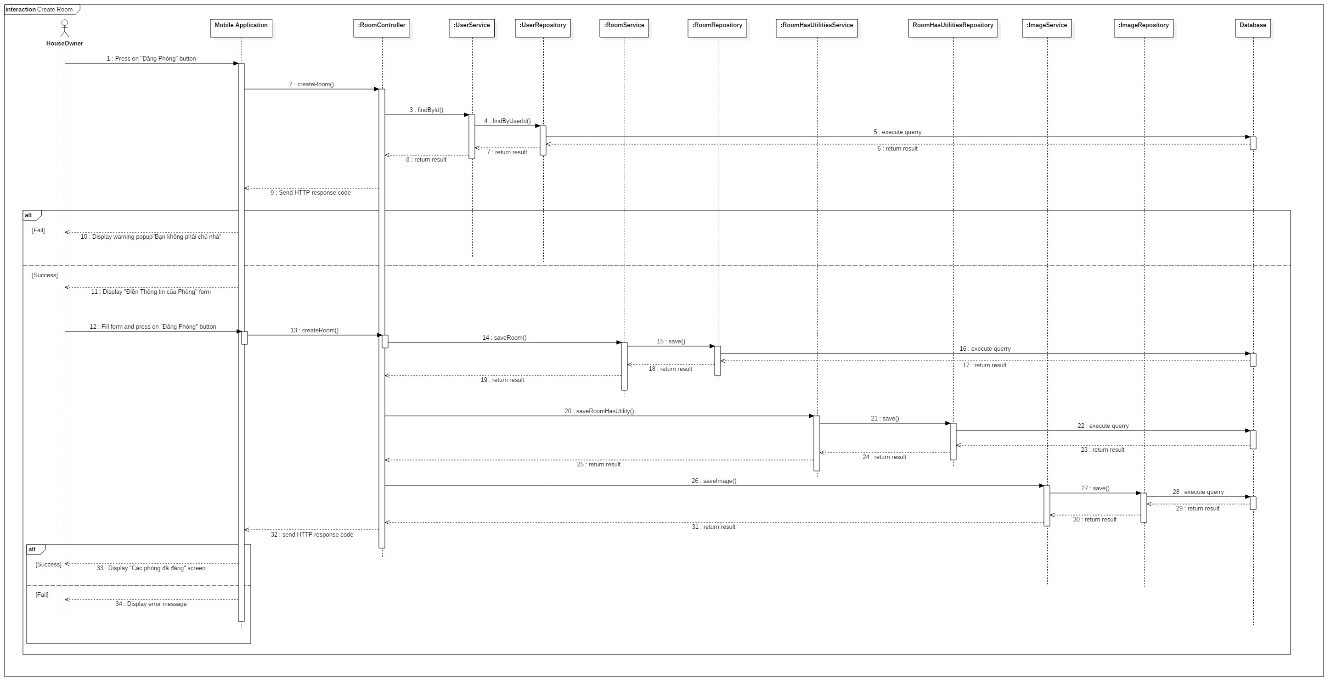


Figure 41: Sequence Diagram - <House owner> Create Room

##### Update Room

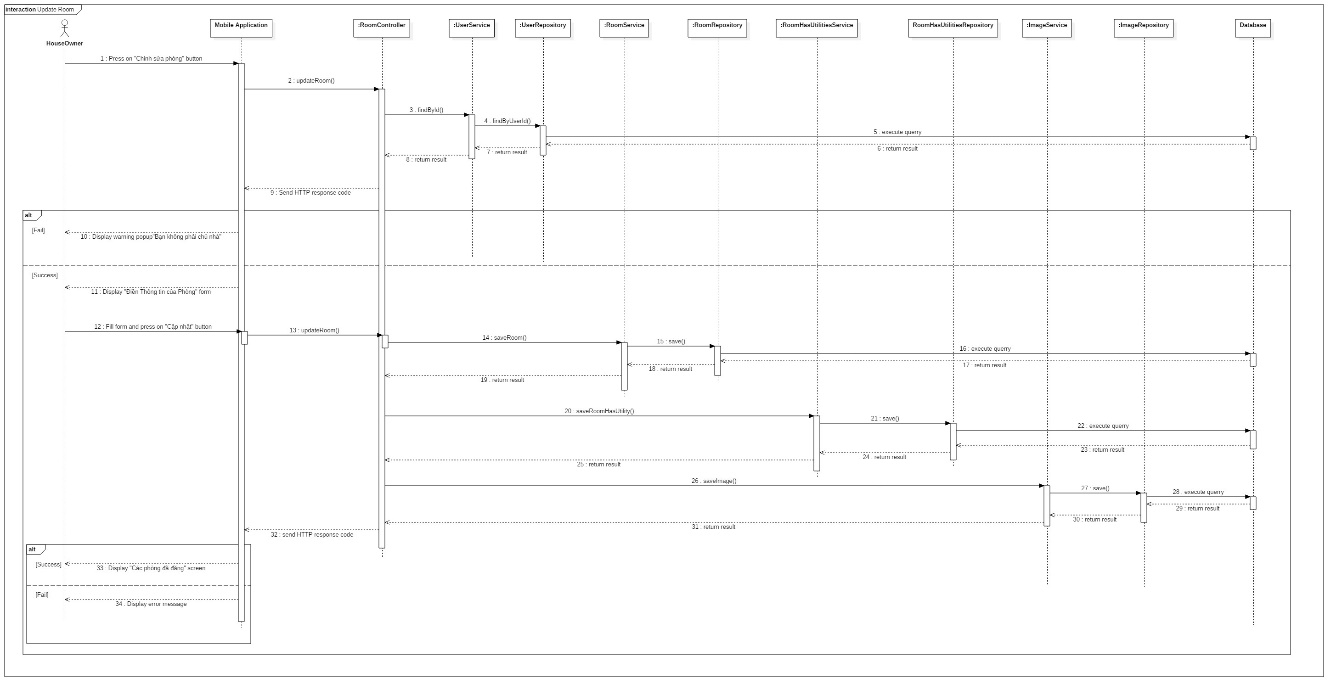


Figure 42: Sequence Diagram - <House owner> Update Room

##### Delete Room

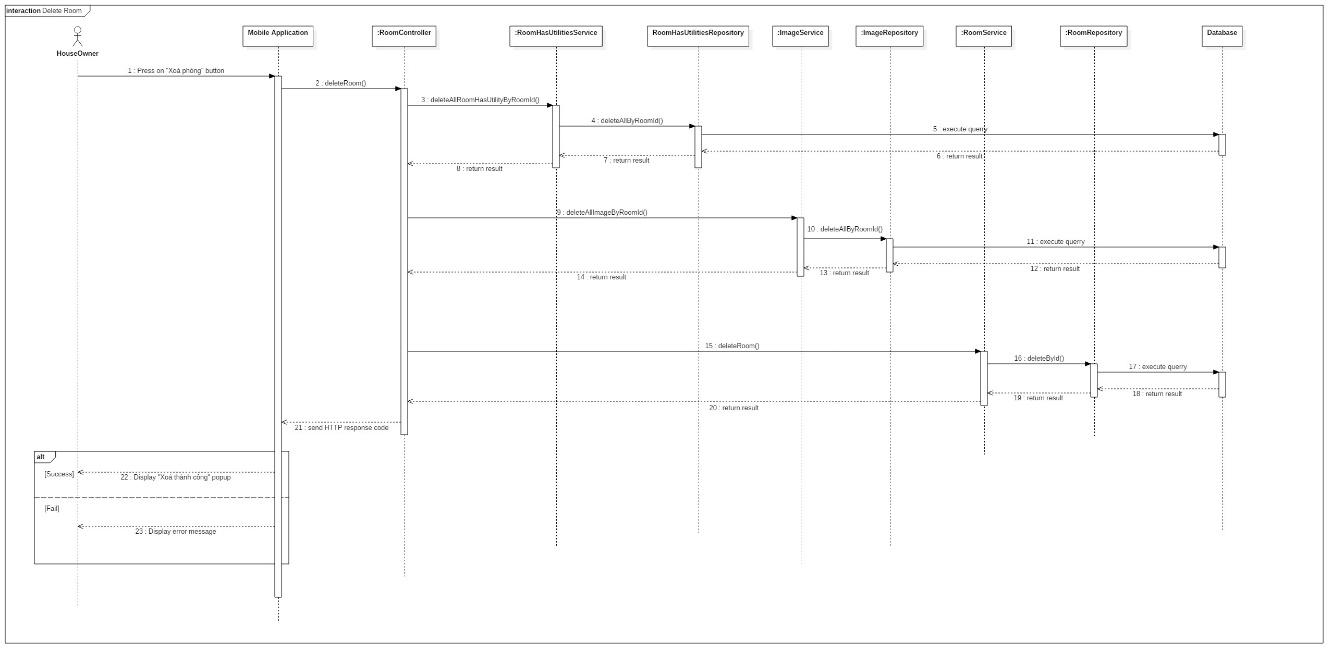


Figure 43: Sequence Diagram - <House owner> Delete Room

##### View Single Room

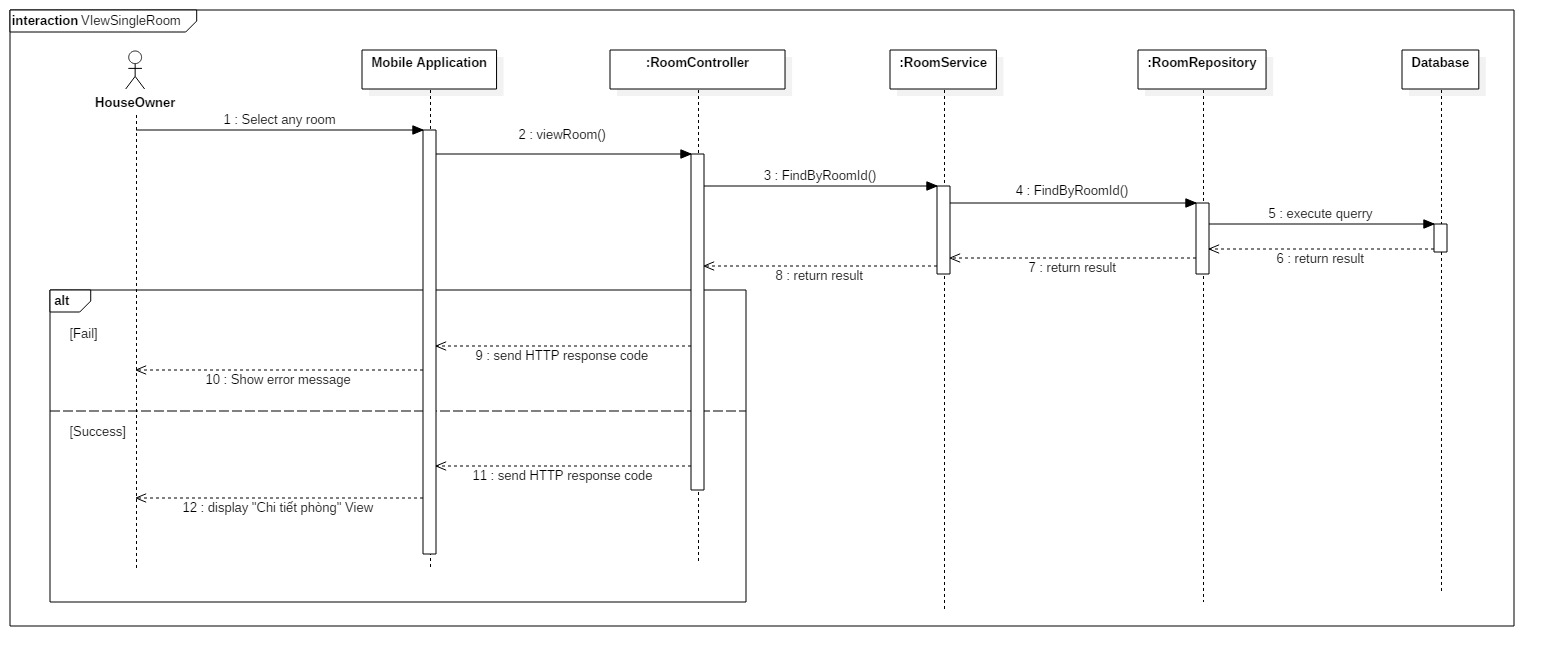


Figure 44: Sequence Diagram - <House owner> View Single Room

##### Add Member into Room

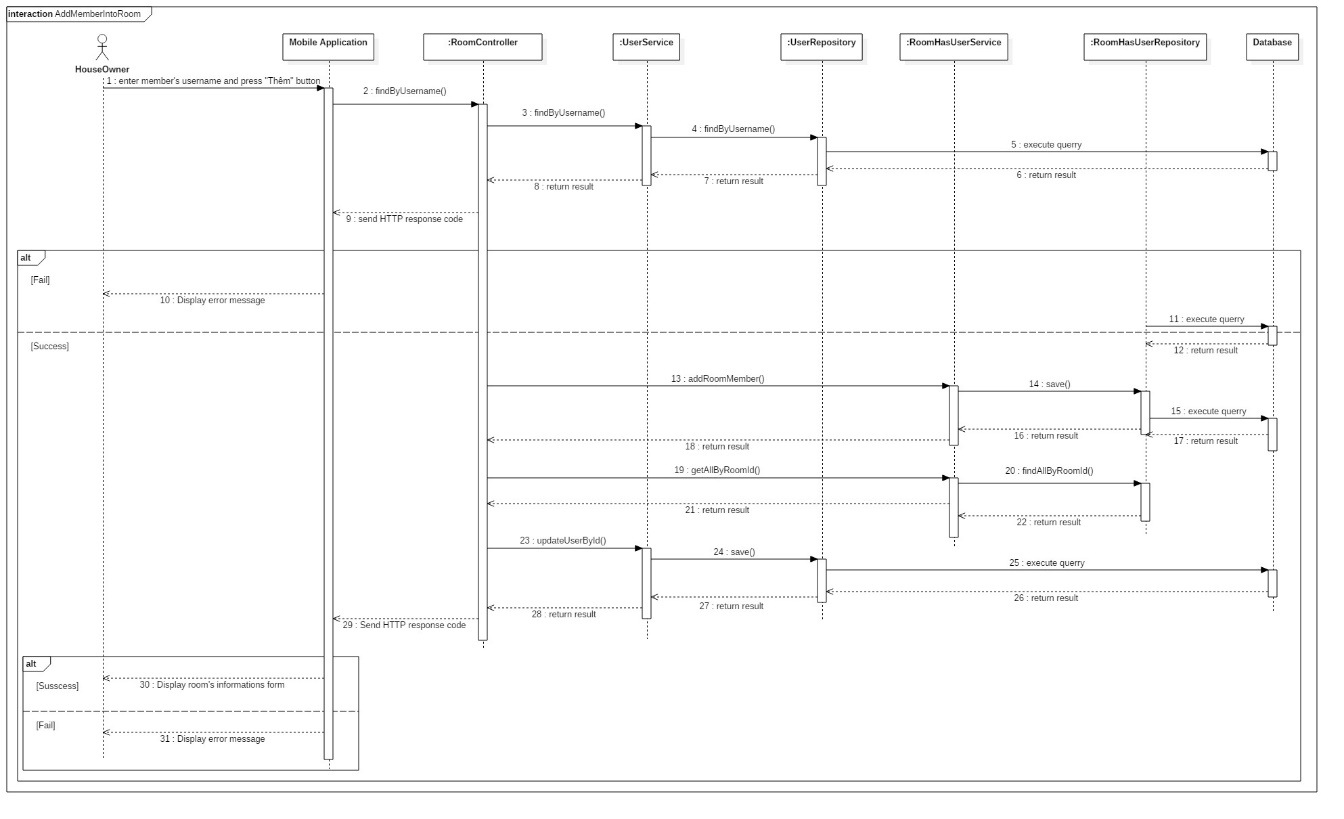


Figure 45: Sequence Diagram - <House owner> Add Member into Room

##### Delete Member of Room

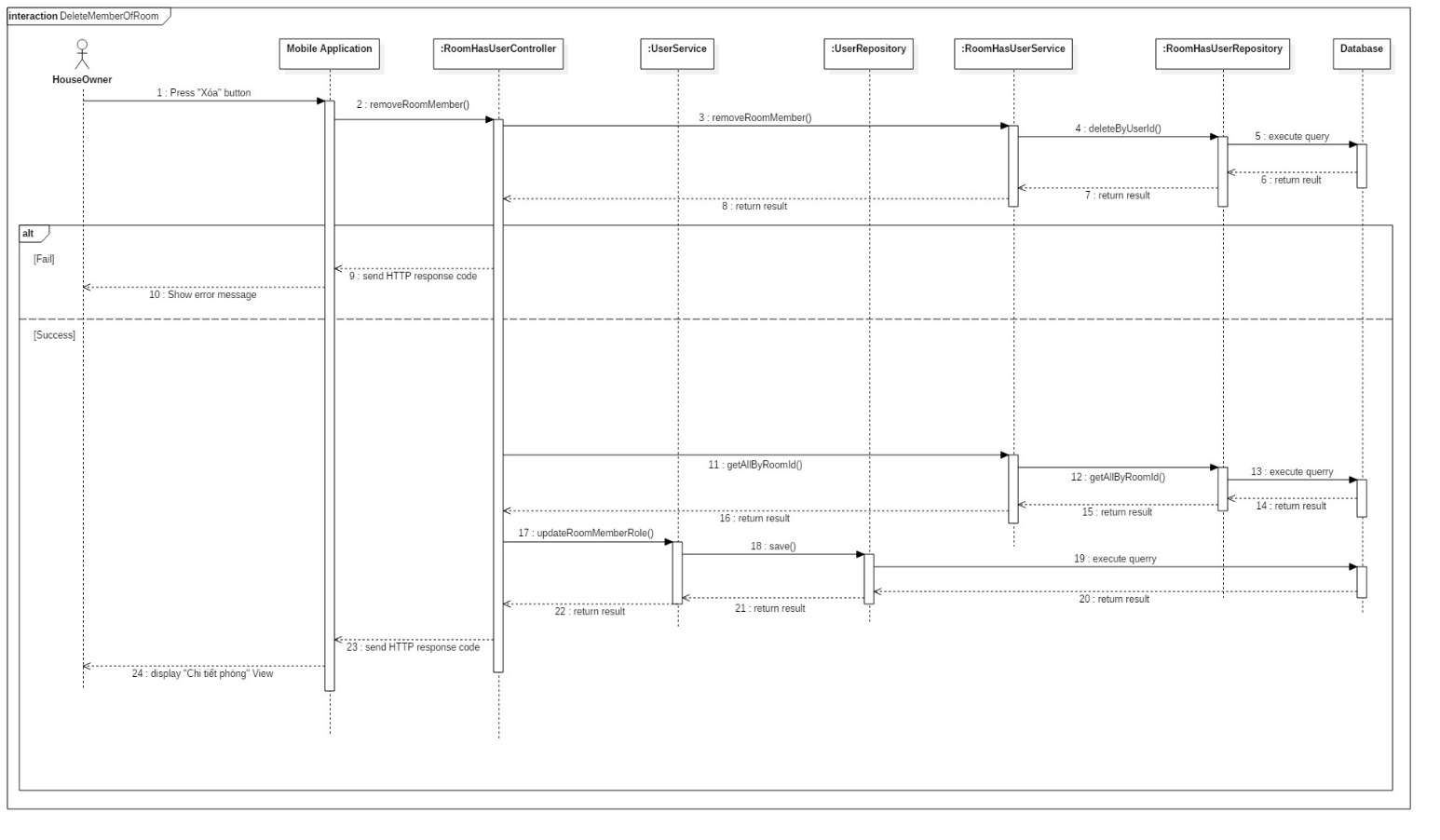


Figure 46: Sequence Diagram - <House owner> Delete Members of Room

##### Create Room Post

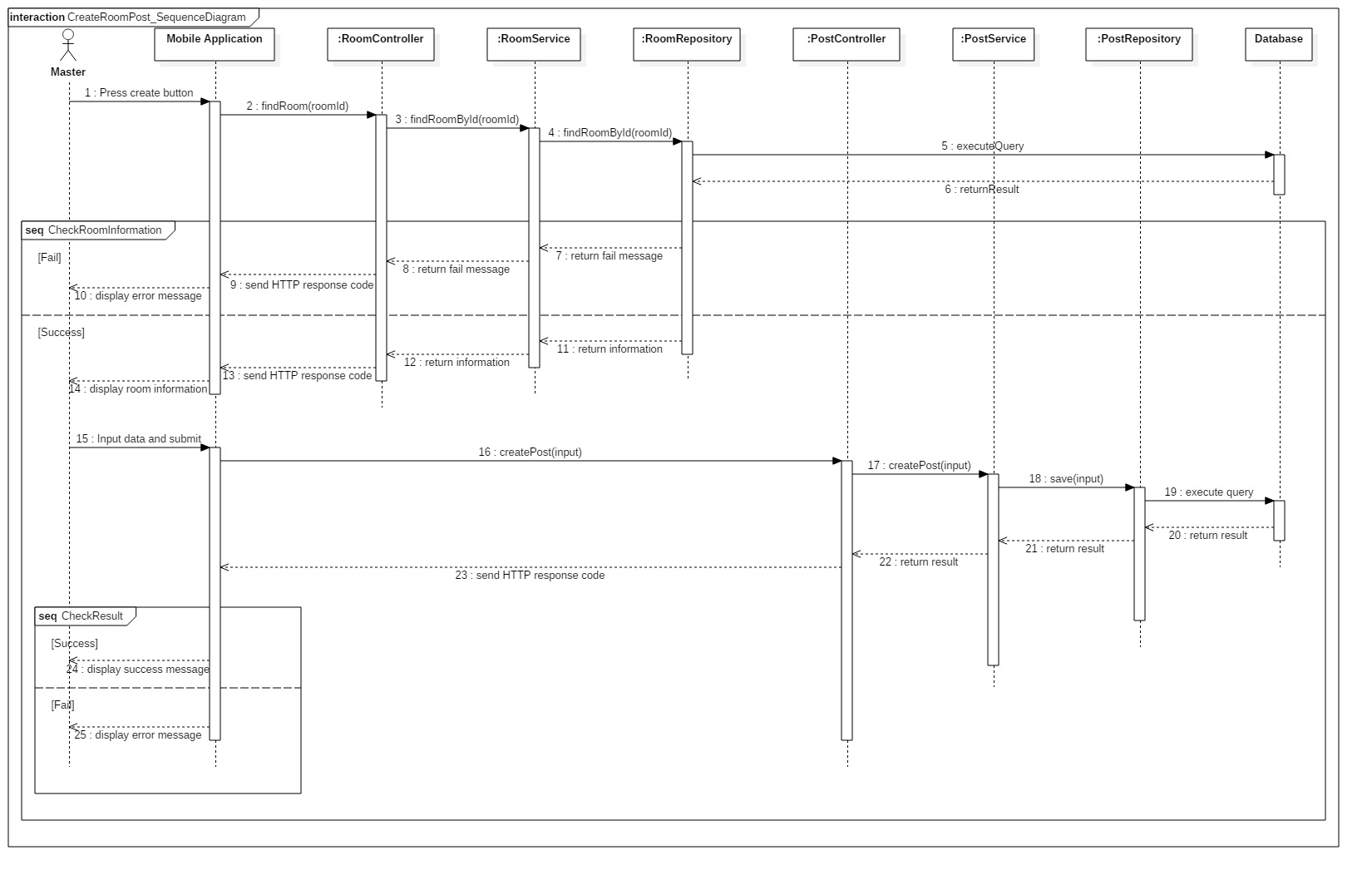


Figure 47: Sequence Diagram - <Room Master> Create Room Post

##### Delete Room Post

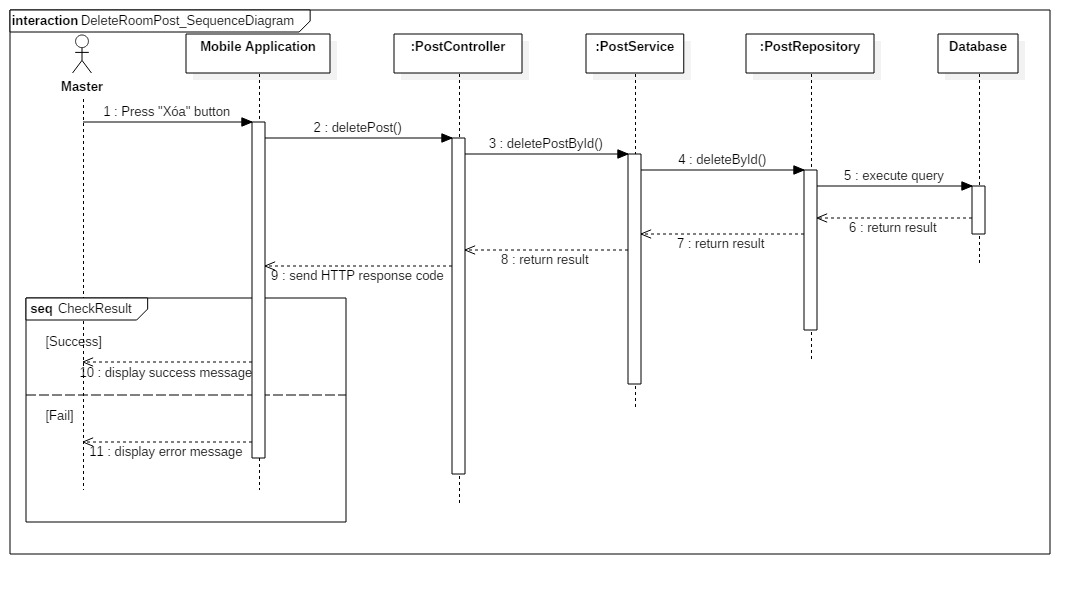


Figure 48: Sequence Diagram - < Room Master > Delete Room



##### Update Room Post

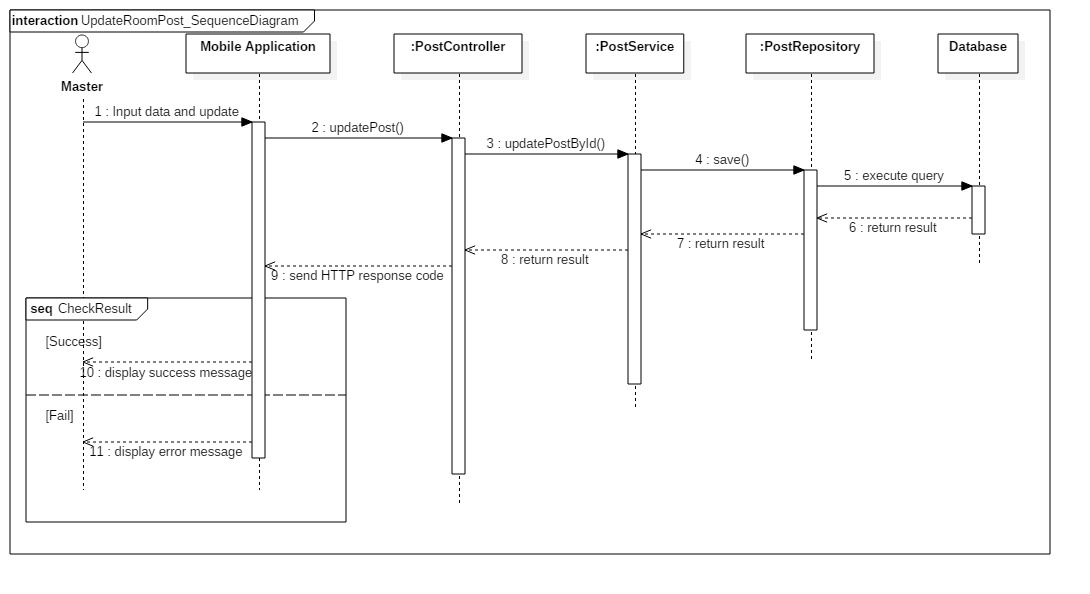


Figure 49: Sequence Diagram - <Room Master> Update Room Post

##### Create Partner Post

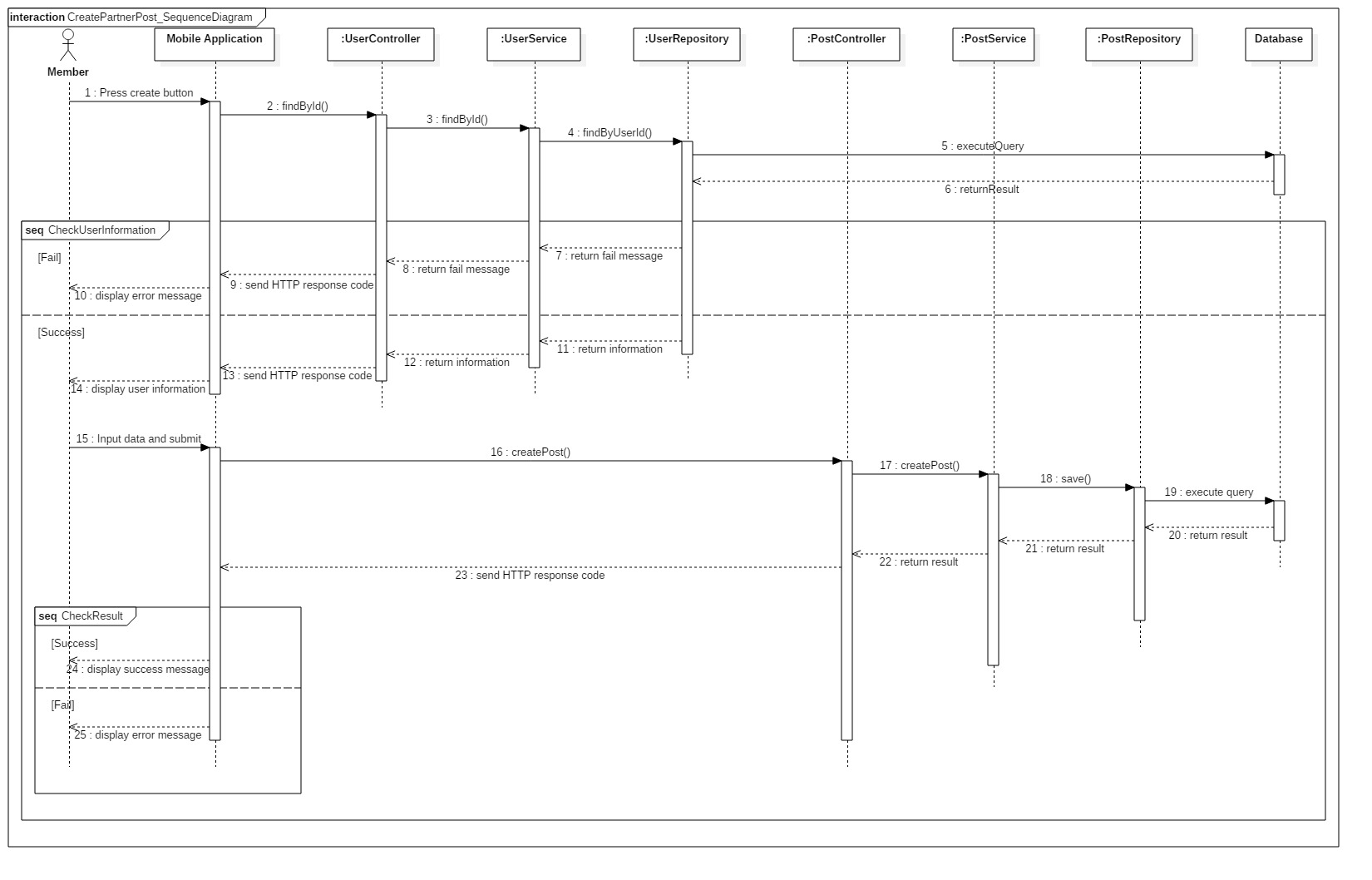


Figure 50: Sequence Diagram - <Member> Create Partner Post

##### Update Partner Post

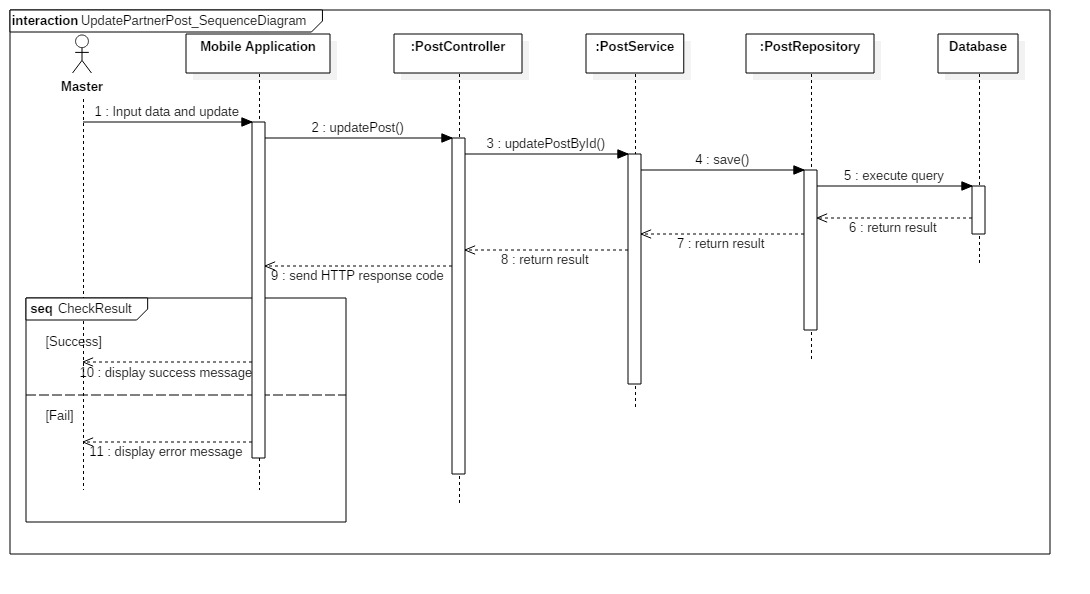


Figure 51: Sequence Diagram - <Member> Update Partner Post

##### Delete Partner Post

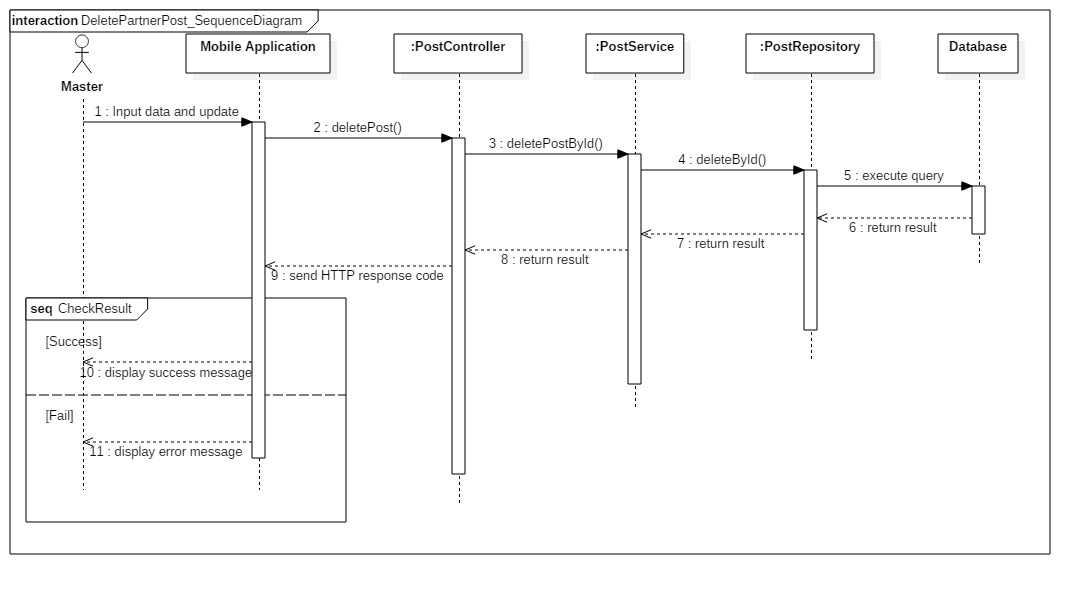


Figure 52: Sequence Diagram - < Member > Delete Partner Post

##### Search Post

Figure 53: Sequence Diagram - <Authorized> Search Post



##### View Suggest List

Figure 54: Sequence Diagram - <Authorized> View Suggest List

## Interface

### Admin Interface Design

#### Login

Figure 55: <Admin> Login Interface

Table 51: Admin login - Field, Buttons

#### Manage User

Figure 56: <Admin> Manage User Interface

Table 52: Admin Manage User - Field, Buttons, Hyperlinks

#### Manage Room

Figure 57: <Admin> Manage Room Interface

Table 53: Admin Manage Room - Field, Buttons, Hyperlinks

#### Manage Post

Figure 58: <Admin> Manage Post Interface

Table 54: Admin Manage Post - Field, Buttons, Hyperlinks

#### View User List

Figure 59: <Admin> View User List Interface

Table 55: Admin View User List - Field, Buttons, Hyperlinks

#### View Room List

Figure 60: <Admin> View Room List Interface

Table 56: Admin View Room List - Field, Buttons, Hyperlinks

#### View Post List

Figure 61: <Admin> View Post List Interface

Table 57: Admin View Post List - Field, Buttons, Hyperlinks

### Unauthorized User Interface Design

#### Register

Figure 62: <Unauthorized User> Register Interface

Table 58: Register - Field, Buttons

#### Login

Figure 63: <Unauthorized User> Register Interface

Table 59: Register - Field, Buttons

### Authorized User Interface Design

#### View Setting

Figure 64: <Authorized User> View Setting Interface

Table 60: View Setting - Field, Buttons, Hyperlinks

#### Update User Information

Figure 65: <Authorized User> Update Information Interface

Table 61: Update User Information - Field, Buttons, Hyperlinks

#### View Profile

Figure 66: <Authorized User> View Profile Interface

Table 62: View Profile - Field, Buttons, Hyperlinks

#### View Notification

Figure 67: <Authorized User> View Notification Interface

Table 63: View Notification - Field, Buttons, Hyperlinks

### House owner Interface Design

#### Create Room

Figure 68: <House owner> Create Room Interface

Table 64: Create Room - Field, Buttons, Hyperlinks

#### Update Room

Figure 69: < House owner > Update Room Interface

Table 65: Update Room - Field, Buttons, Hyperlinks

#### View Room

Figure 70: < House owner > View Room Interface

Table 66: View Room - Field, Buttons, Hyperlinks

#### Delete Room

Figure 71: < House owner > Delete Room Interface

Table 67: Delete Room - Field, Buttons, Hyperlinks

#### Add member into Room

Figure 72: < House owner > Add member into Room Interface

Table 68: Add member into Room - Field, Buttons, Hyperlinks

#### Delete member of Room

Figure 73: < House owner > Delete member of Room Interface

Table 69: Delete member of Room - Field, Buttons, Hyperlinks

### Room Master

#### View suggested partner list

#### View suggested room master list

#### Create Post

#### Update post

#### View Post

#### Delete Post

### Member

#### View suggested room master list

#### Create Post

#### Update Post

#### Delete Post

#### View Post

## Database Design

### Entity relationship diagram (ERD)

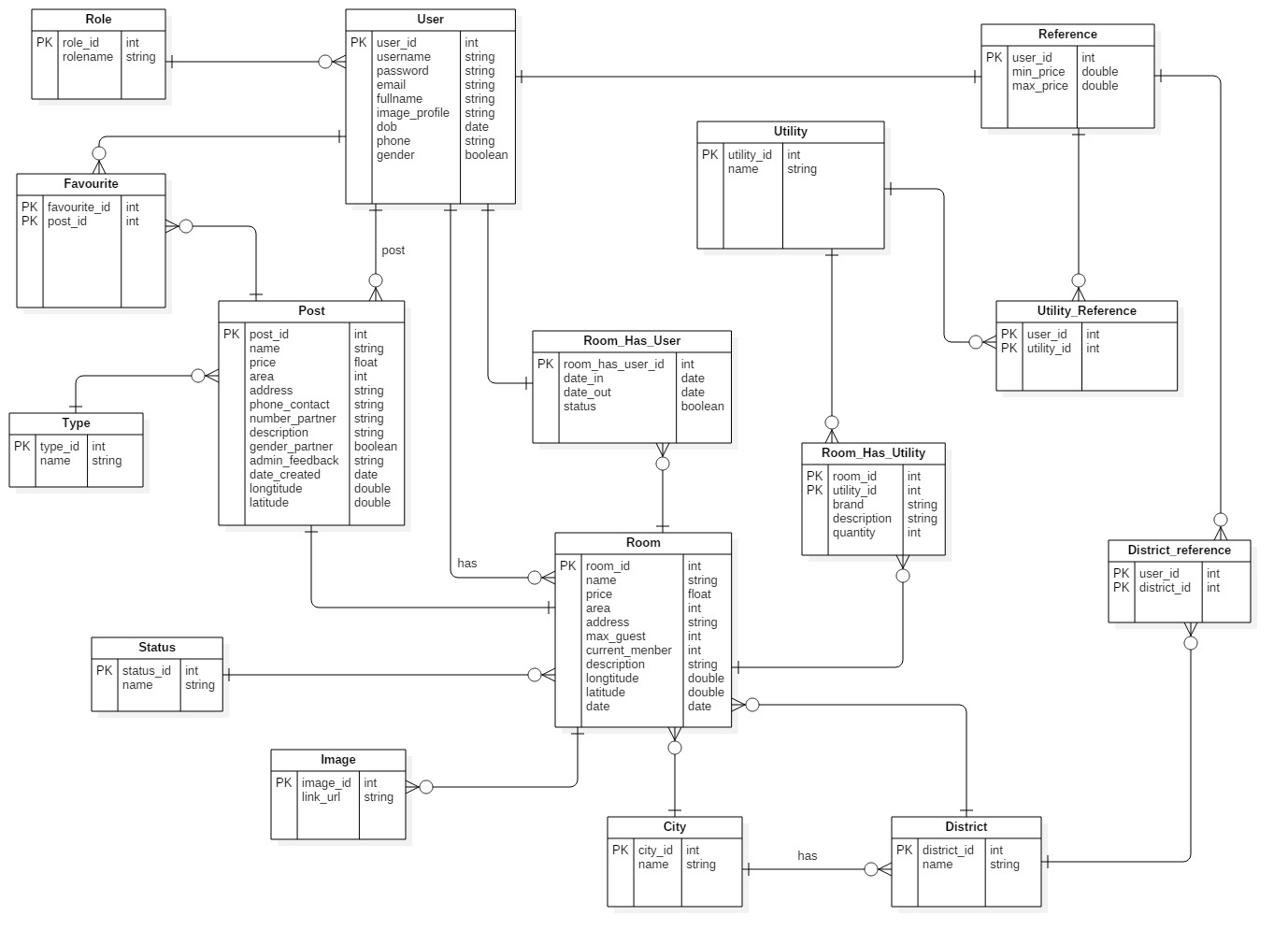


Figure 74: Entity Relationship diagram

### Entity dictionary

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

Table 70: Entity Data dictionary

## Algorithms

### Definition

### Define Problem

### Solution

### Flowchart

Figure 75: Flowchart