

**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](#_Toc530748577)

[List of Table 7](#_Toc530748578)

[List of Figure 9](#_Toc530748579)

[Definitions, Acronyms, and Abbreviations 10](#_Toc530748580)

[A. Introduction 10](#_Toc530748581)

[1. Project Information 10](#_Toc530748582)

[2. Introduction 11](#_Toc530748583)

[3. Current Situation 11](#_Toc530748584)

[4. Problem Definition 11](#_Toc530748585)

[5. Proposed Solution 11](#_Toc530748586)

[5.1 Feature functions 11](#_Toc530748587)

[5.2 Advantages and Disadvantages 12](#_Toc530748588)

[6. Functional Requirements 12](#_Toc530748589)

[7. Role and Responsibility 13](#_Toc530748590)

[B. Software Project Management Plan 13](#_Toc530748591)

[1. Problem Definition 13](#_Toc530748592)

[1.1 Name of this Capstone Project 13](#_Toc530748593)

[1.2 Problem Abstract 13](#_Toc530748594)

[1.3 Project Overview 13](#_Toc530748595)

[1.3.1 Current Situation 13](#_Toc530748596)

[1.3.2 The Proposed System 14](#_Toc530748597)

[1.3.2.1 Mobile Application 14](#_Toc530748598)

[1.3.2.2 Web Application 14](#_Toc530748599)

[1.3.3 Boundaries of the System 14](#_Toc530748600)

[1.3.4 Future Plan 15](#_Toc530748601)

[1.3.5 Development Environment 15](#_Toc530748602)

[1.3.5.1 Hardware requirements 15](#_Toc530748603)

[1.3.5.2 Software requirements 15](#_Toc530748604)

[2. Project Organization 16](#_Toc530748605)

[2.1 Software Process Model 16](#_Toc530748606)

[2.2 Roles and Responsibilities 16](#_Toc530748607)

[2.3 Tools and Techniques 17](#_Toc530748608)

[3. Project Management Plan 18](#_Toc530748609)

[3.1 Product Backlog 18](#_Toc530748610)

[3.2 Sprint Backlog 19](#_Toc530748611)

[3.3 Meeting Minutes 19](#_Toc530748612)

[4. Coding Convention 19](#_Toc530748613)

[C. Software Requirement Specification 19](#_Toc530748614)

[1. User Requirement Specification 19](#_Toc530748615)

[1.1 Unauthorized User Requirement 19](#_Toc530748616)

[1.2 Authorized User Requirement 20](#_Toc530748617)

[1.3 House Owner Requirement 20](#_Toc530748618)

[1.4 Room Master Requirement 20](#_Toc530748619)

[1.5 Member Requirement 20](#_Toc530748620)

[1.6 Admin Requirement 21](#_Toc530748621)

[2. System Requirement Specification 21](#_Toc530748622)

[2.1 External Interface Requirement 21](#_Toc530748623)

[2.1.1 User Interface 21](#_Toc530748624)

[2.1.2 Hardware Interface. 21](#_Toc530748625)

[2.1.3 Software Interface 21](#_Toc530748626)

[2.1.4 Communication Protocol 21](#_Toc530748627)

[2.2 System Overview Use Case 21](#_Toc530748628)

[2.3 List of Use Case 23](#_Toc530748629)

[2.3.1 <Unauthorized User> Overview Use Case 23](#_Toc530748630)

[2.3.1.1 <Unauthorized User> Register 23](#_Toc530748631)

[2.3.1.2 <Unauthorized User> Login 25](#_Toc530748632)

[2.3.2 <Authorized User> Overview Use 26](#_Toc530748633)

[2.3.2.1 <Authorized User> Logout 27](#_Toc530748634)

[2.3.2.2 <Authorized User> View Notification 28](#_Toc530748635)

[2.3.2.3 <Authorized User> Update Info 29](#_Toc530748636)

[2.3.3 <House Owner> Overview Use Case 31](#_Toc530748637)

[2.3.3.1 <House Owner> Create Room 32](#_Toc530748638)

[2.3.3.2 <House owner> Update Room 33](#_Toc530748639)

[2.3.3.3 <House owner> Delete Room 36](#_Toc530748640)

[2.3.3.4 <House owner> Add Member into Room 37](#_Toc530748641)

[2.3.3.5 <House owner> Delete Member from Room 40](#_Toc530748642)

[2.3.4 <Room Master> Overview Use Case 42](#_Toc530748643)

[2.3.4.1 <Room Master> View Suggested Nearby Room Master 42](#_Toc530748644)

[2.3.4.2 <Room Master> Create Finding Roommate Post 44](#_Toc530748645)

[2.3.4.3 <Room Master> Update Finding Roommate Post 47](#_Toc530748646)

[2.3.4.4 <Room Master> Delete Finding Roommate Post 49](#_Toc530748647)

[2.3.4.5 <Room Master> View Post Detail 51](#_Toc530748648)

[2.3.5 <Member> Overview Use Case 53](#_Toc530748649)

[2.3.5.1 <Member> View Suggested Finding Roommate Post list 54](#_Toc530748650)

[2.3.5.2 <Member> Create Finding Room Post 56](#_Toc530748651)

[2.3.5.3 <Member> Update Finding Room Post 58](#_Toc530748652)

[2.3.5.4 <Member> Delete Finding Room Post 60](#_Toc530748653)

[2.3.5.5 <Member> Negotiate 62](#_Toc530748654)

[2.3.6 <Admin> Overview Use Case 64](#_Toc530748655)

[2.3.6.1 <Admin> Manage Rooms 64](#_Toc530748656)

[2.3.6.2 <Admin> Approve Room 65](#_Toc530748657)

[2.3.6.3 <Admin> Manage Users 67](#_Toc530748658)

[2.3.6.4 <Admin> View Users 68](#_Toc530748659)

[3. Software System Attribute 69](#_Toc530748660)

[3.1 Usability 69](#_Toc530748661)

[3.1.1 Graphic User Interface 69](#_Toc530748662)

[3.1.2 Usability 69](#_Toc530748663)

[3.2 Reliability 70](#_Toc530748664)

[3.3 Availability 70](#_Toc530748665)

[3.4 Security 70](#_Toc530748666)

[3.5 Maintainability 70](#_Toc530748667)

[3.6 Portability 70](#_Toc530748668)

[3.7 Performance(gia tri thuc te) 70](#_Toc530748669)

[4. Conceptual diagram 70](#_Toc530748670)

[D. Software Design Description 71](#_Toc530748671)

[1. Design Overview 71](#_Toc530748672)

[2. System Architecture Design 72](#_Toc530748673)

[2.1 System Architecture Design 72](#_Toc530748674)

[3. Component Diagram 73](#_Toc530748675)

[4. Detail Description 74](#_Toc530748676)

[4.1.1 Class Diagram 74](#_Toc530748677)

[4.2 Class Diagram Explanation 75](#_Toc530748678)

[4.2.1 User 75](#_Toc530748679)

[4.2.2 Room 75](#_Toc530748680)

[4.2.3 Post 76](#_Toc530748681)

[4.2.4 Role 76](#_Toc530748682)

[4.2.5 Favourite 76](#_Toc530748683)

[4.2.6 Utilities 76](#_Toc530748684)

[4.2.7 Reference 77](#_Toc530748685)

[4.2.8 District 77](#_Toc530748686)

[4.2.9 City 77](#_Toc530748687)

[4.2.10 Image 77](#_Toc530748688)

[4.2.11 Type 77](#_Toc530748689)

[4.2.12 Status 77](#_Toc530748690)

[4.2.13 Room\_Has\_Utilities 77](#_Toc530748691)

[4.2.14 Utilities\_Reference 78](#_Toc530748692)

[4.2.15 District\_Reference 78](#_Toc530748693)

[4.2.16 Room\_Has\_User 78](#_Toc530748694)

[4.3 Interactive diagram 78](#_Toc530748695)

[4.3.1 Sequence Diagram 78](#_Toc530748696)

[4.3.1.1 Register 78](#_Toc530748697)

[4.3.1.2 Login 79](#_Toc530748698)

[4.3.1.3 Create Room 80](#_Toc530748699)

[4.3.1.4 Update Room 80](#_Toc530748700)

[4.3.1.5 Delete Room 80](#_Toc530748701)

[4.3.1.6 View Single Room 81](#_Toc530748702)

[4.3.1.7 Add Member into Room 81](#_Toc530748703)

[4.3.1.8 Delete Member of Room 82](#_Toc530748704)

[4.3.1.9 Create Finding Roommate Post 82](#_Toc530748705)

[4.3.1.10 Delete Finding Roommate Post 83](#_Toc530748706)

[4.3.1.11 Update Finding Roommate Post 84](#_Toc530748725)

[4.3.1.12 View Suggest List 84](#_Toc530748745)

[4.3.1.13 Create Finding Room Post 84](#_Toc530748746)

[4.3.1.14 Update Finding Room Post 85](#_Toc530748747)

[4.3.1.15 Delete Finding Room Post 85](#_Toc530748748)

[5. Interface 85](#_Toc530748749)

[5.1 Admin Interface Design 85](#_Toc530748750)

[5.1.1 Login 85](#_Toc530748751)

[5.1.2 Manage User 85](#_Toc530748752)

[5.1.3 Manage Room 86](#_Toc530748753)

[5.1.4 View User List 86](#_Toc530748754)

[5.1.5 View Room List 86](#_Toc530748755)

[5.1.6 Approve Room 87](#_Toc530748756)

[5.2 Unauthorized User Interface Design 87](#_Toc530748757)

[5.2.1 Launch Screen 87](#_Toc530748758)

[5.2.2 Sign Up 88](#_Toc530748759)

[5.2.3 Sign In 88](#_Toc530748760)

[5.3 Authorized User Interface Design 89](#_Toc530748761)

[5.3.1 View Setting 89](#_Toc530748762)

[5.3.2 Update User Information 90](#_Toc530748763)

[5.3.3 View Profile 90](#_Toc530748764)

[5.3.4 View Notification 90](#_Toc530748765)

[5.3.5 View Bookmark 91](#_Toc530748766)

[5.3.6 Request Location Popup Screen 92](#_Toc530748767)

[5.4 House owner Interface Design 93](#_Toc530748768)

[5.4.1 Create Room 93](#_Toc530748769)

[5.4.2 Update Room 94](#_Toc530748770)

[5.4.3 View Room 94](#_Toc530748771)

[5.4.4 Delete Room 95](#_Toc530748772)

[5.4.5 Add member into Room 95](#_Toc530748773)

[5.4.6 Delete member of Room 95](#_Toc530748774)

[5.5 Room Master 95](#_Toc530748775)

[5.5.1 View suggested partner list 95](#_Toc530748776)

[5.5.2 View suggested room master list 95](#_Toc530748777)

[5.5.3 Create Post 95](#_Toc530748778)

[5.5.4 Update post 95](#_Toc530748779)

[5.5.5 View Post 95](#_Toc530748780)

[5.5.6 Delete Post 96](#_Toc530748781)

[5.6 Member 96](#_Toc530748782)

[5.6.1 View suggested room master list 96](#_Toc530748783)

[5.6.2 Create Post 96](#_Toc530748784)

[5.6.3 Update Post 96](#_Toc530748785)

[5.6.4 Delete Post 96](#_Toc530748786)

[5.6.5 View Post 96](#_Toc530748787)

[6. Database Design 97](#_Toc530748788)

[6.1 Entity relationship diagram (ERD) 97](#_Toc530748789)

[6.2 Entity dictionary 98](#_Toc530748790)

[7. Algorithms 98](#_Toc530748791)

[7.1 Definition 98](#_Toc530748792)

[7.2 Define Problem 98](#_Toc530748793)

[7.3 Solution 99](#_Toc530748794)

[E. System Implementation & Test 99](#_Toc530748795)

[1. Introduction 99](#_Toc530748796)

[1.1 Overview 99](#_Toc530748797)

[1.2 Test approach 99](#_Toc530748798)

[2. Database Relationship Diagram 100](#_Toc530748799)

[2.1 Physical Diagram 100](#_Toc530748800)

[2.2 Data dictionary 100](#_Toc530748801)

[3. Performance Measure 103](#_Toc530748802)

[3.1 Show list result of finding room post performance 103](#_Toc530748803)

[3.2 Show list result of finding roommate post performance 103](#_Toc530748804)

[3.3 Show list result of all post performance 104](#_Toc530748805)

[3.4 Show list result of suggested post performance 104](#_Toc530748806)

[3.5 Add new room performance 104](#_Toc530748807)

[3.6 Update room performance 104](#_Toc530748808)

[3.7 Add new post performance 104](#_Toc530748809)

[3.8 Update post performance 104](#_Toc530748810)

[4. Test Plan 104](#_Toc530748811)

[4.1 Features to be tested 104](#_Toc530748812)

[4.2 Features not to be tested 104](#_Toc530748813)

[5. System Testing Test Case 104](#_Toc530748814)

[5.1 Test case 104](#_Toc530748815)

[F. Software User’s Manual 104](#_Toc530748816)

[1. Installation Guide 105](#_Toc530748817)

[1.1 Setting up environment at server side 105](#_Toc530748818)

[1.1.1 Hardware Requirement 105](#_Toc530748819)

[1.1.2 Software Requirement 105](#_Toc530748820)

[1.2 Deployment at server side 105](#_Toc530748821)

[1.2.1 Check Environment 105](#_Toc530748822)

[1.2.2 Import Database 105](#_Toc530748823)

[1.2.3 Build Project 105](#_Toc530748824)

[1.2.4 Deploy Project to IIS 105](#_Toc530748825)

[2. User Manual 105](#_Toc530748826)

[2.1 Unauthorized User 105](#_Toc530748827)

[2.1.1 Login 105](#_Toc530748828)

[2.2 Authorized User 105](#_Toc530748829)

[2.3 Admin 105](#_Toc530748830)

[2.4 House Owner 105](#_Toc530748831)

[2.5 Room Master 105](#_Toc530748832)

[2.3 Member 105](#_Toc530748833)

[G. Appendix 105](#_Toc530748834)

# List of Table

[Table 1: Roles and Responsibilities 13](#_Toc530748835)

[Table 2: Hardware Requirements for Server 15](#_Toc530748836)

[Table 3: Hardware Requirement for Mobile 15](#_Toc530748837)

[Table 4: Software Requirements 16](#_Toc530748838)

[Table 5: Roles and Responsibilities 17](#_Toc530748839)

[Table 6: Tools and Techniques 18](#_Toc530748840)

[Table 7: Product Backlog 19](#_Toc530748841)

[Table 8: <Unauthorized User> Register specification 25](#_Toc530748842)

[Table 9: <Unauthorized User> Login specification 26](#_Toc530748843)

[Table 10: <Authorized User> Logout specification 28](#_Toc530748844)

[Table 11: <Authorized User> View Notification specification 29](#_Toc530748845)

[Table 12: <Authorized User> Update Information specification 31](#_Toc530748846)

[Table 13: <House Owner> Create Room specification 33](#_Toc530748847)

[Table 14: <House owner> Update Room specification 36](#_Toc530748848)

[Table 15: <House owner> Delete Room specification 37](#_Toc530748849)

[Table 16: <House owner> Add Member into Room specification 40](#_Toc530748850)

[Table 17: <House owner> Delete Member from Room specification 42](#_Toc530748851)

[Table 18: <Room Master> View Suggested Nearby Room Master Post Specification 44](#_Toc530748852)

[Table 19: <Room Master> Create Finding Roommate Post specification 47](#_Toc530748853)

[Table 20: <Room Master> Update Finding Roommate Post specification 49](#_Toc530748854)

[Table 21: <Room Master> Delete Finding Roommate Post specification 51](#_Toc530748855)

[Table 22: <Room Master> View Post Detail Specification 53](#_Toc530748856)

[Table 23: <Member> View Suggested Finding Roommate Posts List Specification 56](#_Toc530748857)

[Table 24: <Member> Create Finding Room Post specification 58](#_Toc530748858)

[Table 25: <Member> Update Finding Room Post specification 60](#_Toc530748859)

[Table 26: <Member> Delete Finding Room Post specification 62](#_Toc530748860)

[Table 27: <Member> Negotiate specification 64](#_Toc530748861)

[Table 28: <Admin> Manage room specification 65](#_Toc530748862)

[Table 29: <Admin> Approve room specification 67](#_Toc530748863)

[Table 30: <Admin> Manage User specification 68](#_Toc530748864)

[Table 31: <Admin> View User specification 69](#_Toc530748865)

[Table 32: Conceptual Entity dictionary 71](#_Toc530748866)

[Table 33: Component Diagram Explanation 74](#_Toc530748867)

[Table 34: Class Diagram Explanation 75](#_Toc530748868)

[Table 35: User Attributes 75](#_Toc530748869)

[Table 36: Room Attributes 76](#_Toc530748870)

[Table 37: Post Attributes 76](#_Toc530748871)

[Table 38: Role Attributes 76](#_Toc530748872)

[Table 39: Favourite Attributes 76](#_Toc530748873)

[Table 40: Utility Attributes 77](#_Toc530748874)

[Table 41: Reference Attributes 77](#_Toc530748875)

[Table 42: District Attributes 77](#_Toc530748876)

[Table 43: City Attributes 77](#_Toc530748877)

[Table 44: Image Attributes 77](#_Toc530748878)

[Table 45: Type Attributes 77](#_Toc530748879)

[Table 46: Status Attributes 77](#_Toc530748880)

[Table 47: Room Has Utilities Attributes 78](#_Toc530748881)

[Table 48: Utilities Reference Attributes 78](#_Toc530748882)

[Table 49: District Reference Attributes 78](#_Toc530748883)

[Table 50: Room Has User Attributes 78](#_Toc530748884)

[Table 51: Admin login - Field, Buttons 85](#_Toc530748885)

[Table 52: Admin Manage User - Field, Buttons, Hyperlinks 86](#_Toc530748886)

[Table 53: Admin Manage Room - Field, Buttons, Hyperlinks 86](#_Toc530748887)

[Table 54: Admin View User List - Field, Buttons, Hyperlinks 86](#_Toc530748888)

[Table 55: Admin View Room List - Field, Buttons, Hyperlinks 87](#_Toc530748889)

[Table 56: Admin Approve Room - Field, Buttons, Hyperlinks 87](#_Toc530748890)

[Table 57: Launch Screen - Buttons 88](#_Toc530748891)

[Table 58: Sign Up- Field, Buttons 88](#_Toc530748892)

[Table 59: Sign In - Field, Buttons 89](#_Toc530748893)

[Table 60: View Setting - Field, Buttons, Hyperlinks 90](#_Toc530748894)

[Table 61: Update User Information - Field, Buttons, Hyperlinks 90](#_Toc530748895)

[Table 62: View Profile - Field, Buttons, Hyperlinks 90](#_Toc530748896)

[Table 63: View Notification - Field 91](#_Toc530748897)

[Table 64: View Bookmark - Field, Buttons, Hyperlinks 92](#_Toc530748898)

[Table 65: Request Location - Buttons 93](#_Toc530748899)

[Table 66: Create Room - Field, Buttons, Hyperlinks 94](#_Toc530748900)

[Table 67: Update Room - Field, Buttons, Hyperlinks 94](#_Toc530748901)

[Table 68: View Room - Field, Buttons, Hyperlinks 95](#_Toc530748902)

[Table 69: Delete Room - Field, Buttons, Hyperlinks 95](#_Toc530748903)

[Table 70: Add member into Room - Field, Buttons, Hyperlinks 95](#_Toc530748904)

[Table 71: Delete member of Room - Field, Buttons, Hyperlinks 95](#_Toc530748905)

[Table 72: Entity Data dictionary 98](#_Toc530748906)

[Table 73: Data dictionary 101](#_Toc530748907)

[Table 74: Data Table Description 103](#_Toc530748908)

# List of Figure

[Figure 1: Scrum Methodology 16](#_Toc530748909)

[Figure 2: System Overview Use Case (1) 22](#_Toc530748910)

[Figure 3: System Overview Use Case (2) 23](#_Toc530748911)

[Figure 4: <Unauthorized User> Overview Use Case 23](#_Toc530748912)

[Figure 5: <Unauthorized User> Register Use Case 24](#_Toc530748913)

[Figure 6: <Unauthorized User> Login Use Case 25](#_Toc530748914)

[Figure 7: <Authorized User> Overview Use Case 27](#_Toc530748915)

[Figure 8: <Authorized User> Logout Use Case 27](#_Toc530748916)

[Figure 9: <Authorized User> View Notification Use Case 28](#_Toc530748917)

[Figure 10: <Authorized User> Update Information Use Case 29](#_Toc530748918)

[Figure 11: <House Owner> Overview Use Case 31](#_Toc530748919)

[Figure 12: <House Owner> Create Room Use Case 32](#_Toc530748920)

[Figure 13: <House owner> Update Room Use Case 34](#_Toc530748921)

[Figure 14: <House owner> Delete Room Use Case 36](#_Toc530748922)

[Figure 15: <House owner> Add Member into Room Use Case 37](#_Toc530748923)

[Figure 16: <House owner> Delete Member from Room Use Case 40](#_Toc530748924)

[Figure 17: <Room Master> Overview Use Case 42](#_Toc530748925)

[Figure 18: <Room Master> View Suggested Nearby Room Master Post Use Case 43](#_Toc530748926)

[Figure 19: <Room Master> Create Finding Roommate Post Use Case 45](#_Toc530748927)

[Figure 20: <Room Master> Update Finding Roommate Post Use Case 47](#_Toc530748928)

[Figure 21: <Room Master> Delete Finding Roommate Post Use Case 50](#_Toc530748929)

[Figure 22: <Room Master> View Post Detail Use Case 52](#_Toc530748930)

[Figure 23: <Member> Overview Use Case 54](#_Toc530748931)

[Figure 24: <Member> View Suggested Finding Roommate Posts List Use Case 55](#_Toc530748932)

[Figure 25:< Member> Create Finding Room Post Use Case 56](#_Toc530748933)

[Figure 26:< Member> Update Finding Room Post Use Case 58](#_Toc530748934)

[Figure 27: <Member> Delete Finding Room Post Use Case 61](#_Toc530748935)

[Figure 28: <Member> Negotiate Use Case 62](#_Toc530748936)

[Figure 29: <Admin> Overview Use Case 64](#_Toc530748937)

[Figure 30: <Admin> Manage Room Use Case 64](#_Toc530748938)

[Figure 31: <Admin> Approve Room Use Case 66](#_Toc530748939)

[Figure 32: <Admin> Manage User Use Case 67](#_Toc530748940)

[Figure 33: <Admin> View User Use Case 68](#_Toc530748941)

[Figure 34: Conceptual Diagram 71](#_Toc530748942)

[Figure 35: System Architecture Design 73](#_Toc530748943)

[Figure 36: Component Diagram 73](#_Toc530748944)

[Figure 37: Class Diagram 74](#_Toc530748945)

[Figure 38: Sequence Diagram - <Unauthorized> Register 79](#_Toc530748946)

[Figure 39: Sequence Diagram - <Unauthorized> Login 79](#_Toc530748947)

[Figure 40: Sequence Diagram - <House owner> Create Room 80](#_Toc530748948)

[Figure 41: Sequence Diagram - <House owner> Update Room 80](#_Toc530748949)

[Figure 42: Sequence Diagram - <House owner> Delete Room 81](#_Toc530748950)

[Figure 43: Sequence Diagram - <House owner> View Single Room 81](#_Toc530748951)

[Figure 44: Sequence Diagram - <House owner> Add Member into Room 82](#_Toc530748952)

[Figure 45: Sequence Diagram - <House owner> Delete Members of Room 82](#_Toc530748953)

[Figure 46: Sequence Diagram - <Room Master> Create Finding Roommate Post 83](#_Toc530748954)

[Figure 47: Sequence Diagram - <Room Master> Delete Finding Roommate Post 83](#_Toc530748955)

[Figure 48: Sequence Diagram - <Room Master> Update Finding Roommate Post 84](#_Toc530748956)

[Figure 49: Sequence Diagram - <Room Master> View Suggest List 84](#_Toc530748957)

[Figure 50: Sequence Diagram - <Member> Create Finding Room Post 85](#_Toc530748958)

[Figure 51: Sequence Diagram - <Member> Update Finding Room Post 85](#_Toc530748959)

[Figure 52: Sequence Diagram - <Member> Delete Finding Room Post 85](#_Toc530748960)

[Figure 53: <Admin> Login Interface 85](#_Toc530748961)

[Figure 54: <Admin> Manage User Interface 86](#_Toc530748962)

[Figure 55: <Admin> Manage Room Interface 86](#_Toc530748963)

[Figure 56: <Admin> View User List Interface 86](#_Toc530748964)

[Figure 57: <Admin> View Room List Interface 87](#_Toc530748965)

[Figure 58: <Admin> Approve Room Interface 87](#_Toc530748966)

[Figure 59: <Unauthorized User> Launch Screen Interface 88](#_Toc530748967)

[Figure 60: <Unauthorized User> Sign Up Interface 88](#_Toc530748968)

[Figure 61: <Unauthorized User> Sign In Interface 89](#_Toc530748969)

[Figure 62: <Authorized User> View Setting Interface 90](#_Toc530748970)

[Figure 63: <Authorized User> Update Information Interface 90](#_Toc530748971)

[Figure 64: <Authorized User> View Profile Interface 90](#_Toc530748972)

[Figure 65: <Authorized User> View Notification Interface 91](#_Toc530748973)

[Figure 66: <Authorized User> View Bookmark 92](#_Toc530748974)

[Figure 67: <Authorized User> Request Location Interface 93](#_Toc530748975)

[Figure 68: <House owner> Create Room Interface 94](#_Toc530748976)

[Figure 69: < House owner > Update Room Interface 94](#_Toc530748977)

[Figure 70: < House owner > View Room Interface 95](#_Toc530748978)

[Figure 71: < House owner > Delete Room Interface 95](#_Toc530748979)

[Figure 72: < House owner > Add member into Room Interface 95](#_Toc530748980)

[Figure 73: < House owner > Delete member of Room Interface 95](#_Toc530748981)

[Figure 74: Entity Relationship diagram 98](#_Toc530748982)

[Figure 75: Physical diagram 100](#_Toc530748983)

# 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

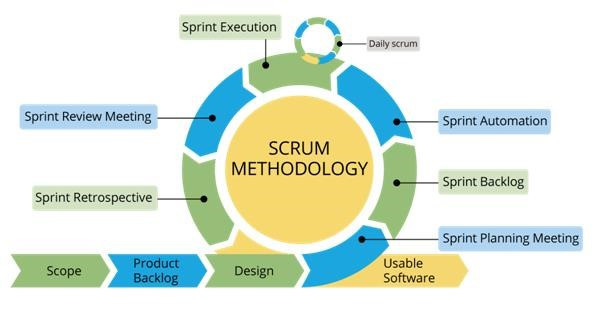


Figure 1: Scrum Methodology

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 nearby room master posts.

### Member Requirement

Member is an authorized user who is looking for a room

Member can do the following functions:

* Search
* Create finding room post
* Update finding room post
* View finding room post
* Delete finding room 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 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

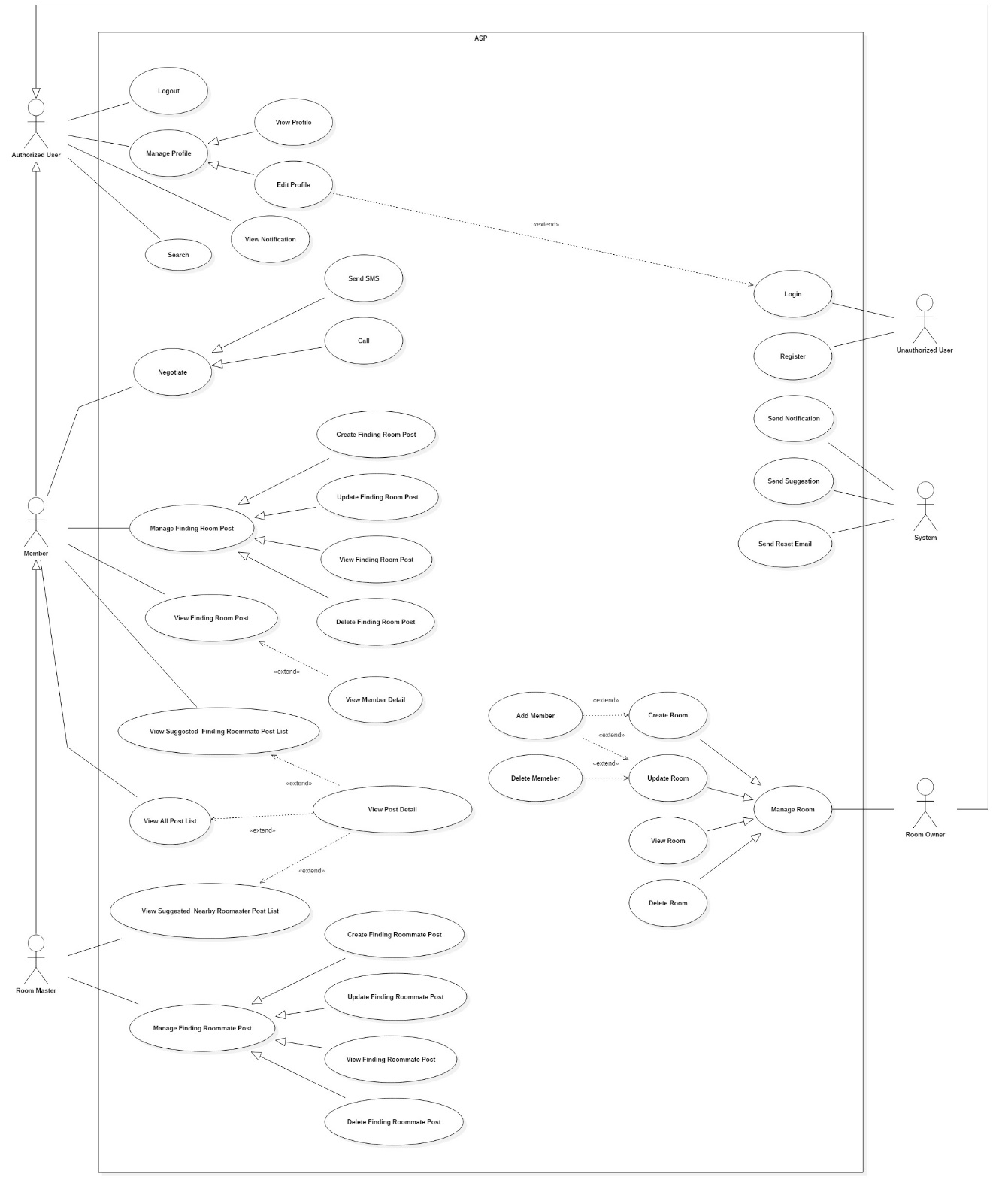


Figure 2: System Overview Use Case (1)



Figure 3: System Overview Use Case (2)

### List of Use Case

#### <Unauthorized User> Overview Use Case

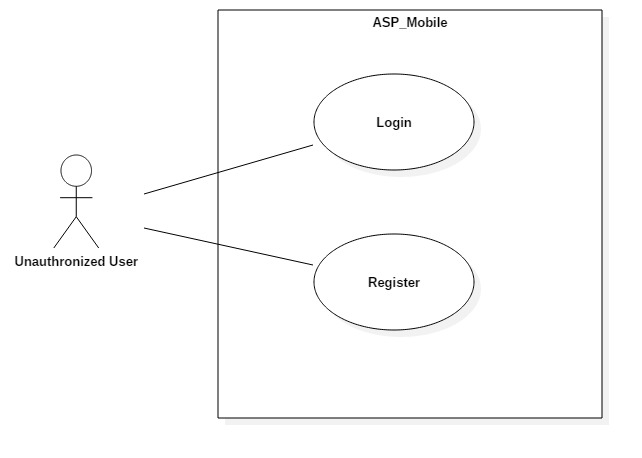


Figure 4: <Unauthorized User> Overview Use Case

##### <Unauthorized User> Register

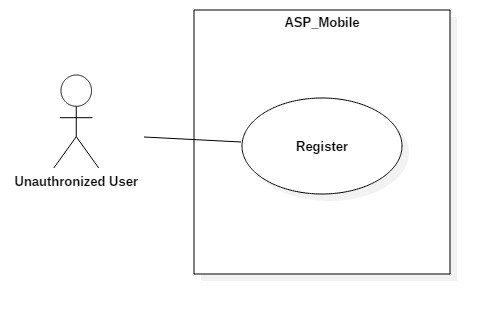


Figure 5: <Unauthorized User> Register Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP\_1** | | | |
| **Use Case No.** | UC\_ ASP\_1 | **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 accesses 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 information | [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 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

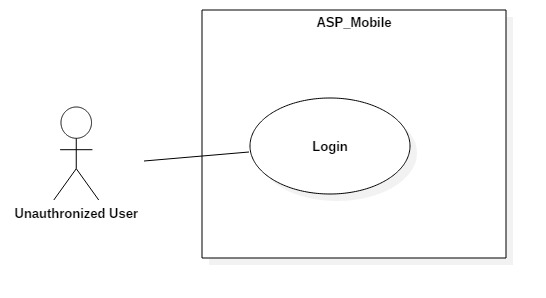


Figure 6: <Unauthorized User> Login Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP\_2** | | | |
| **Use Case No.** | UC\_ ASP\_2 | **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 accesses 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

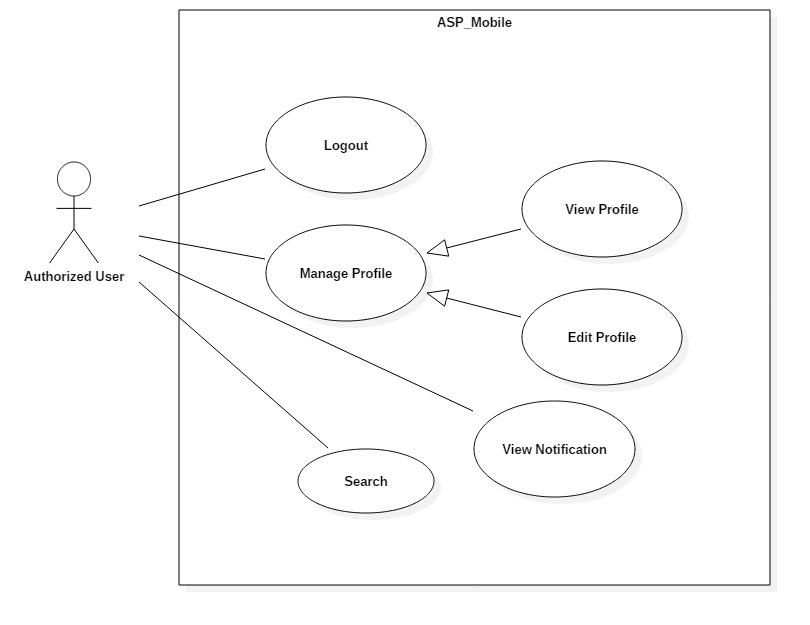


Figure 7: <Authorized User> Overview Use Case

##### <Authorized User> Logout

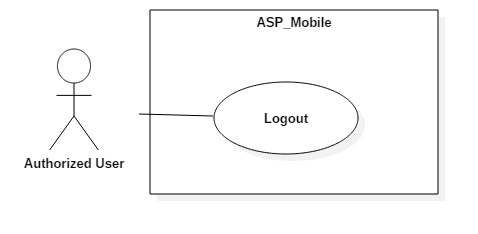


Figure 8: <Authorized User> Logout Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP\_3** | | | |
| **Use Case No.** | UC\_ ASP\_3 | **Use Case Version** | 0.2 |
| **Use Case Name** | Logout | | |
| **Author** | Vương Minh Thông | | |
| **Date** | 17/10/2018 | **Priority** | High |
| **Actor:**   * Authorized User   **Summary:**   * This use case allows authorized user signs out from the system.   **Goal:**   * Authenticated user stops accessing to the system. * User’s session is removed from the system.   **Triggers:**   * 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 presses “Tài khoản” button. | User navigated to setting screen | | 2 | Actor presses “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

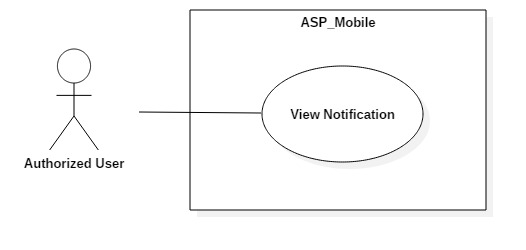


Figure 9: <Authorized User> View Notification Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP\_4** | | | |
| **Use Case No.** | UC\_ ASP\_4 | **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 presses on “Thông báo” button. | System navigates to notification screen with list of notifications. |   **Alternative Scenario:**N/A  **Exceptions:** N/A  **Relationships:** UC\_ ASP\_24, UC\_ ASP\_9, UC\_ ASP\_10  **Business Rules:**   * User has already accessed to the system. * User can be any role to get notified. * If there is no notification, display “Hiện tại bạn chưa có thông báo mới”. | | | |

Table 11: <Authorized User> View Notification specification

##### <Authorized User> Update Info

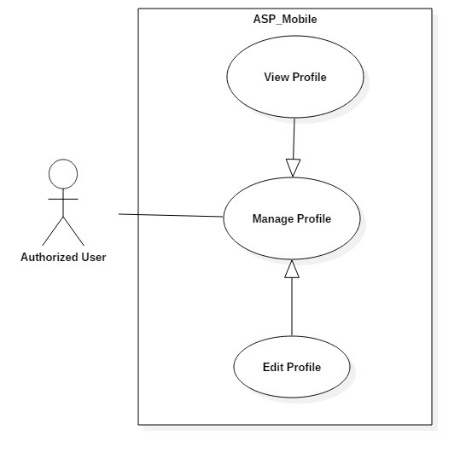


Figure 10: <Authorized User> Update Information Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP\_5** | | | |
| **Use Case No.** | UC\_ ASP\_5 | **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 presses on account field in setting screen. * Actor presses 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 presses on “Tài khoản” button. | System navigates to setting screen. | | 2 | Actor presses 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 information in some fields. | 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. * Information are updated must be not be in wrong format. * Email should not be empty. | | | |

Table 12: <Authorized User> Update Information specification

#### <House Owner> Overview Use Case

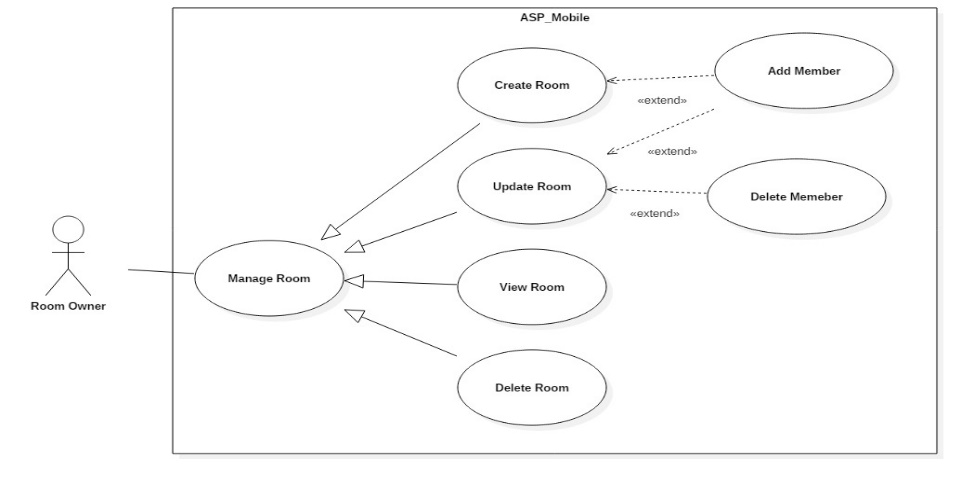


Figure 11: <House Owner> Overview Use Case

##### <House Owner> Create Room

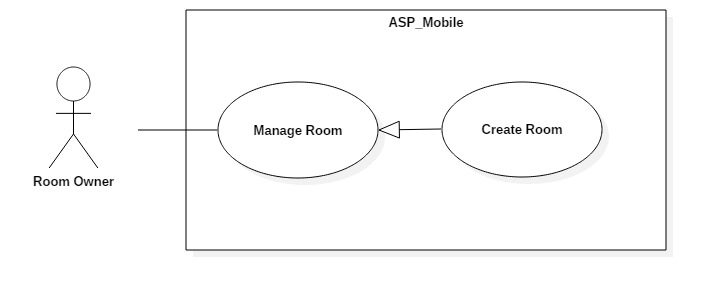


Figure 12: <House Owner> Create Room Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP\_6** | | | |
| **Use Case No.** | UC\_ ASP\_6 | **Use Case Version** | 0.2 |
| **Use Case Name** | Create Room | | |
| **Author** | Vương Minh Thông | | |
| **Date** | 17/10/2018 | **Priority** | High |
| **Actor:**   * House Owner   **Summary:**   * This use case allows house owner to create new room.   **Goal:**   * To create new room.   **Triggers:**   * Press “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 presses 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 | | 2 | Actor fills form and presses 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 must not be empty. * Room price must not be empty. * Address must not be empty. * User must be house owner role to create room. | | | |

Table 13: <House Owner> Create Room specification

##### <House owner> Update Room

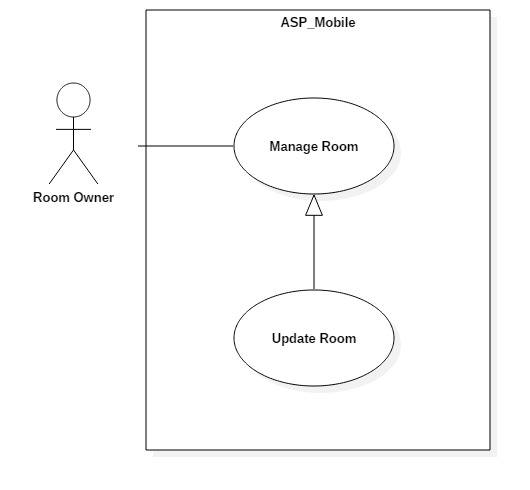


Figure 13: <House owner> Update Room Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP\_7** | | | |
| **Use Case No.** | UC\_ ASP\_7 | **Use Case Version** | 0.2 |
| **Use Case Name** | Update Room | | |
| **Author** | Vương Minh Thông | | |
| **Date** | 17/10/2018 | **Priority** | High |
| **Actor:**   * House owner   **Summary:**   * This use case allows House owner update room.   **Goal:**   * To update room.   **Triggers:**   * Actor press on 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 presses “Tài khoản” button. | System navigates to “Tải khoản” screen.  System displays “Tài khoản” screen with:  - “Đã đăng” tab. | | 2 | Actor presses on certain room. | System will navigate to this room screen with common detail.  - “Hình ảnh”  - “Tên phòng”  - “Giá phòng”  - “Diện tích”  - “Địa chỉ”  - “Mô tả”  - “Ngày đăng bài”  - “Tiện ích”  - “Thành viên trong phòng” table  - “Chỉnh sửa” button  - “Xóa” button | | 3 | Actor presses 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 | | 4 | Actor fills form presses 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. * User must have role as House Owner. * Room’s information is updated must be matched the defined room’s information structure. * Room name must not be empty. * Room price must not be empty. * Room address must not be empty. | | | |

Table 14: <House owner> Update Room specification

##### <House owner> Delete Room

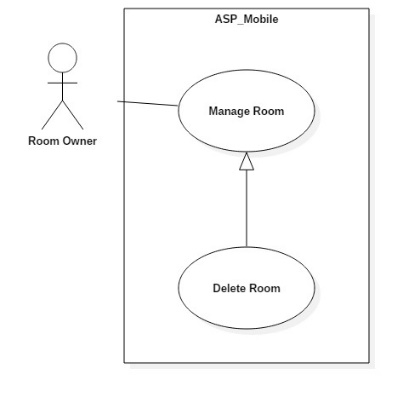


Figure 14: <House owner> Delete Room Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP\_8** | | | |
| **Use Case No.** | UC\_ ASP\_8 | **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 presses “Tài khoản” button. | System navigates to “Tải khoản” screen.  System displays “Tài khoản” screen with:  - “Đã đăng” tab. | | 2 | Actor presses on certain room. | System will navigate to this room screen with common detail.  - “Hình ảnh”  - “Tên phòng”  - “Giá phòng”  - “Diện tích”  - “Địa chỉ”  - “Mô tả”  - “Ngày đăng bài”  - “Tiện ích”  - “Thành viên trong phòng” table  - “Chỉnh sửa” button  - “Xóa” button | | 3 | Actor presses on “Xoá” button. | Room 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. | | | |

Table 15: <House owner> Delete Room specification

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

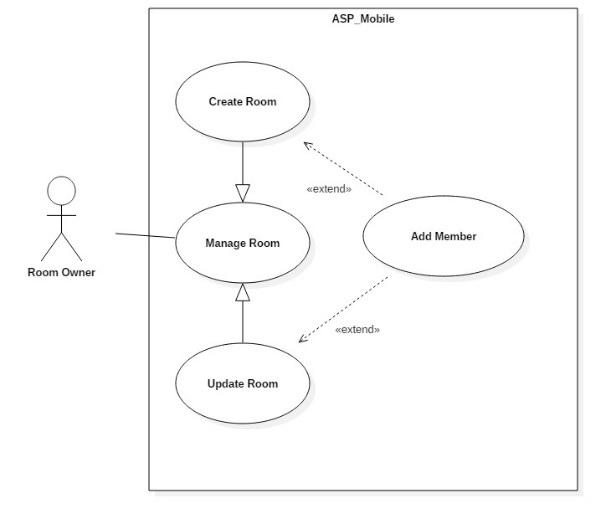


Figure 15: <House owner> Add Member into Room Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP\_9** | | | |
| **Use Case No.** | UC\_ ASP\_9 | **Use Case Version** | 0.2 |
| **Use Case Name** | Add Member into Room | | |
| **Author** | Hồ Công Trình | | |
| **Date** | 17/10/2018 | **Priority** | High |
| **Actor:**   * House owner   **Summary:**   * This use case allows house owner add member into Room.   **Goal:**   * To add member into room.   **Triggers:**   * Actor press on certain room in account screen. * Actor press “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 presses “Tài khoản” button. | System navigates to “Tải khoản” screen.  System displays “Tài khoản” screen with:  - “Đã đăng” tab. | | 2 | Actor presses on certain room. | System will navigate to this room screen with common detail.  - “Hình ảnh”  - “Tên phòng”  - “Giá phòng”  - “Diện tích”  - “Địa chỉ”  - “Ngày đăng bài”  - “Mô tả”  - “Chỉnh sửa” button  - “Xóa” button | | 5 | Actor presses 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 presses 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 presses 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 inputs wrong information in some fields. | System notices that user need to re-input all these fields:  - “Tài khoản”: System displays warning message: “Tài khoản không tồn tại hoặc bị trùng”.  - “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

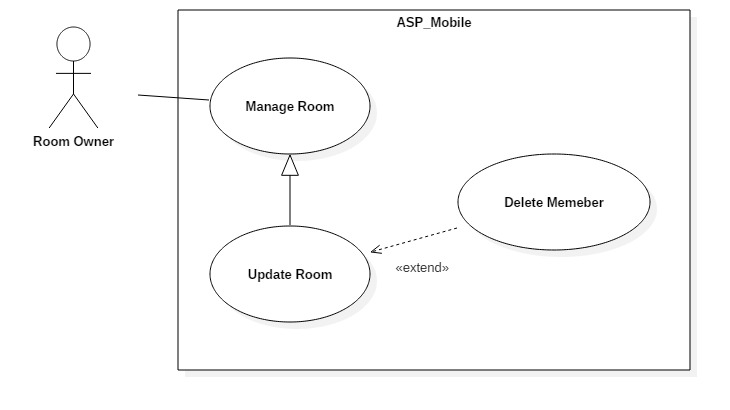


Figure 16: <House owner> Delete Member from Room Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP\_10** | | | |
| **Use Case No.** | UC\_ ASP\_10 | **Use Case Version** | 0.2 |
| **Use Case Name** | Delete Member from Room | | |
| **Author** | Hồ Công Trình | | |
| **Date** | 17/10/2018 | **Priority** | High |
| **Actor:**   * House owner   **Summary:**   * This use case allows house owner delete member from room.   **Goal:**   * To delete member from room.   **Triggers:**   * Actor presses on certain room in account screen. * Actor presses “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 presses “Tài khoản” button. | System navigates to “Tải khoản” screen.  System displays “Tài khoản” screen with:  - “Đã đăng” tab. | | 2 | Actor presses on certain room. | System will navigate to this room screen with common detail.  - “Hình ảnh”  - “Tên phòng”  - “Giá”  - “Diện tích”  - “Địa chỉ”  - “Ngày đăng bài”  - “Mô tả”  - “Chỉnh sửa” button  - “Xóa” button | | 5 | Actor presses 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 chooses 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 presses 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 Room Master was removed, system will promote the next member to room master automatically. | | | |

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

#### <Room Master> Overview Use Case

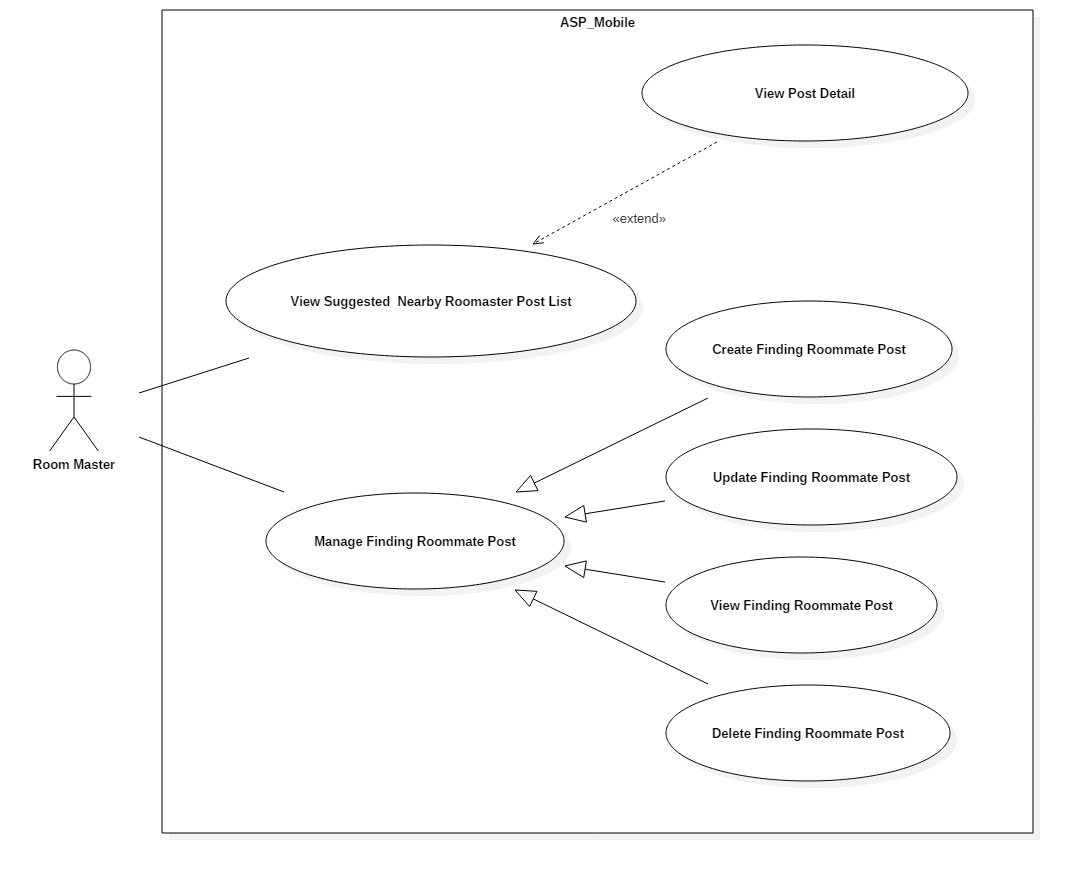


Figure 17: <Room Master> Overview Use Case

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

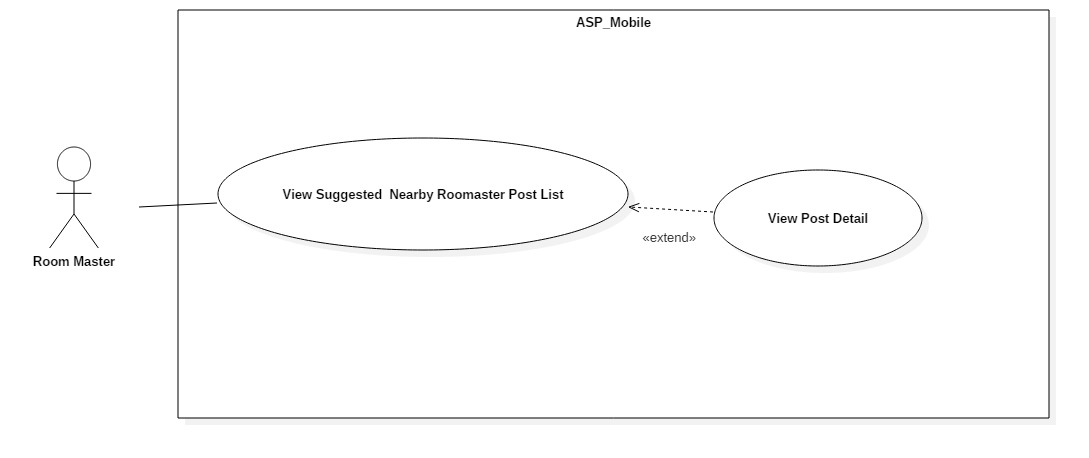


Figure 18: <Room Master> View Suggested Nearby Room Master Post Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP\_11** | | | |
| **Use Case No.** | UC\_ ASP\_11 | **Use Case Version** | 0.2 |
| **Use Case Name** | View Suggested Nearby Room master Post | | |
| **Author** | Nguyễn Đỗ Minh Đức | | |
| **Date** | 17/10/2018 | **Priority** | High |
| **Actor:**   * Room Master   **Summary:**   * This use case allows room master to view suggested nearby room master posts.   **Goal:**   * To view suggested nearby room master posts.   **Triggers:**   * Actor logged into system and goes to home screen.   **Preconditions:**   * User logged in the system.   **Post conditions:**   * Success: User can view suggested finding nearby room master posts. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Actor 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. * 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 Nearby Room Master Post Specification

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

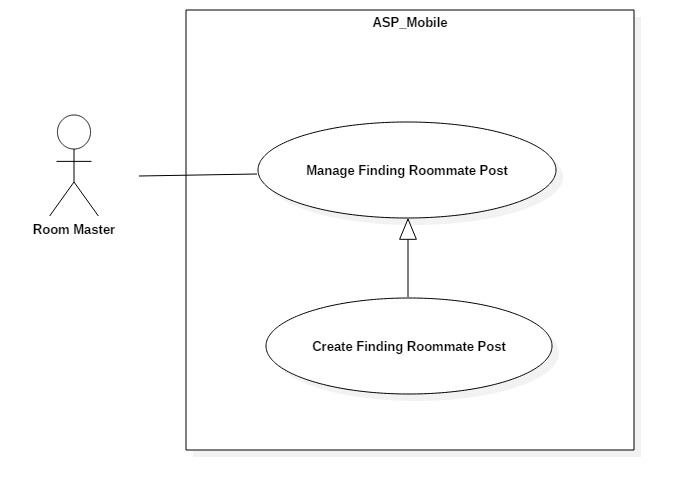


Figure 19: <Room Master> Create Finding Roommate Post Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP\_12** | | | |
| **Use Case No.** | UC\_ ASP\_12 | **Use Case Version** | 0.2 |
| **Use Case Name** | Create Finding Roommate Post | | |
| **Author** | Nguyễn Quang Nhật | | |
| **Date** | 17/10/2018 | **Priority** | High |
| **Actor:**   * Room Master   **Summary:**   * This use case allows room master to create new finding roommate post.   **Goal:**   * To create new post.   **Triggers:**   * Actor press on “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 information in some fields. | System notices that user need to re-input all these field:  - “Giá”: System displays warning message: “Không được sử dụng chữ hoặc ký tự đặc biệt !@#$%^&\*()-=\_+...”.  - “Số bạn ở ghép”: System displays warning message: “Không được sử dụng chữ hoặc ký tự đặc biệt !@#$%^&\*()-=\_+... Và phải ít hơn số thành viên tối đa của phòng”. |   **Relationships:** N/A  **Business Rules:**   * User already logged in the system. * Number of partner must not be empty. * Price must not be empty. * User must be room master. * User must be added into room. | | | |

Table 19: <Room Master> Create Finding Roommate Post specification

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

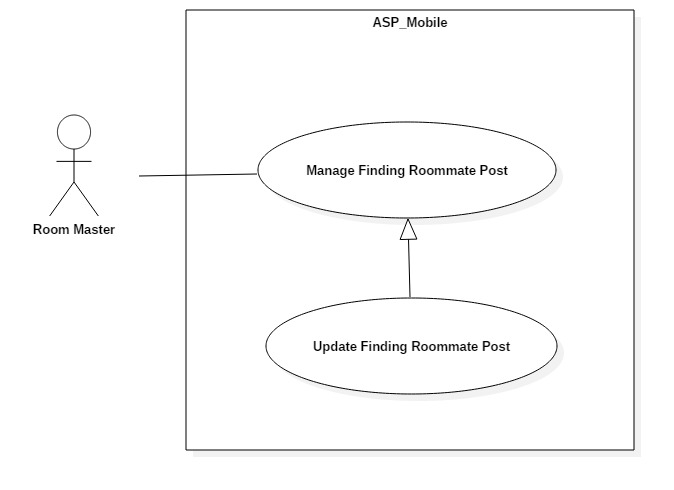


Figure 20: <Room Master> Update Finding Roommate Post Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP\_13** | | | |
| **Use Case No.** | UC\_ ASP\_13 | **Use Case Version** | 0.2 |
| **Use Case Name** | Update Finding Roommate Post | | |
| **Author** | Nguyễn Quang Nhật | | |
| **Date** | 17/10/2018 | **Priority** | High |
| **Actor:**   * Room Master   **Summary:**   * This use case allows room master to update finding roommate post.   **Goal:**   * To update post.   **Triggers:**   * Actor press on “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 presses “Tài khoản” button. | System navigates to “Tải khoản” screen.  System displays “Tài khoản” screen with:  - “Đã đăng” tab. | | 2 | Actor presses 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 presses 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 | Actor fills form and presses 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 information in some fields. | System notices that user need to re-input all these field:  - “Giá”: System displays warning message: “Không được sử dụng chữ hoặc ký tự đặc biệt !@#$%^&\*()-=\_+...”.  - “Số bạn ở ghép”: System displays warning message: “Không được sử dụng chữ hoặc ký tự đặc biệt !@#$%^&\*()-=\_+... Và phải ít hơn số thành viên phòng tối đa”. |   **Relationships:** N/A  **Business Rules:**   * User already logged in the system. * Post number of partner is not null. * Post price is not null. * User must be room master role. * Information are updated must not be in wrong format. | | | |

Table 20: <Room Master> Update Finding Roommate Post specification

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

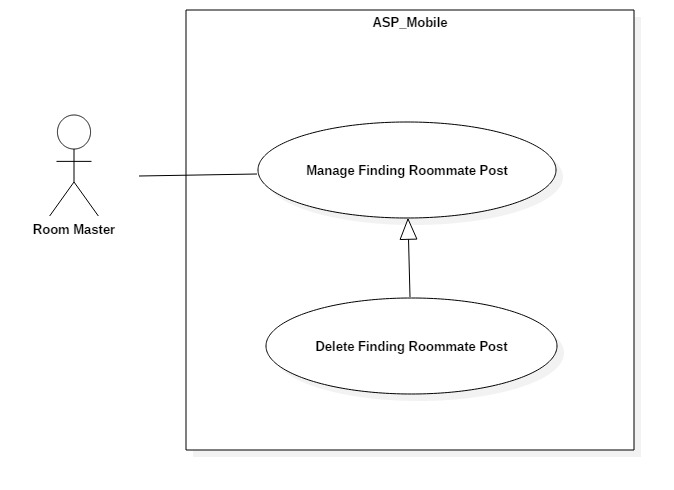


Figure 21: <Room Master> Delete Finding Roommate Post Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP\_14** | | | |
| **Use Case No.** | UC\_ ASP\_14 | **Use Case Version** | 0.2 |
| **Use Case Name** | Delete Finding Roommate Post | | |
| **Author** | Nguyễn Quang Nhật | | |
| **Date** | 17/10/2018 | **Priority** | High |
| **Actor:**   * Room Master   **Summary:**   * This use case allows room master to delete finding roommate 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 presses “Tài khoản” button. | System navigates to “Tải khoản” screen.  System displays “Tài khoản” screen with:  - “Đã đăng” tab. | | 2 | Actor presses 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 presses 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 Finding Roommate Post specification

##### <Room Master> View Post Detail

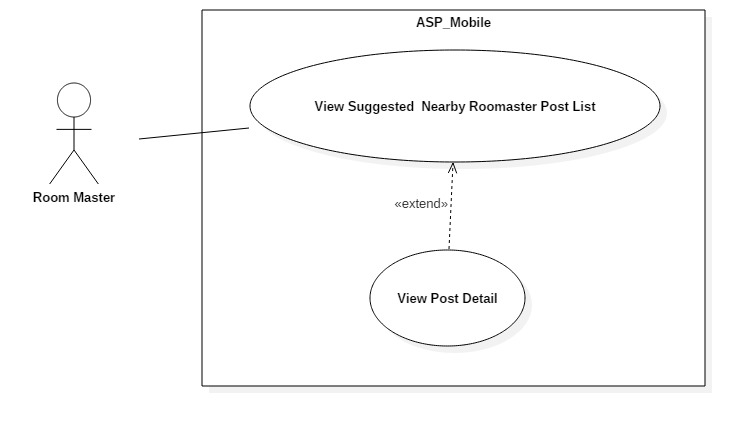


Figure 22: <Room Master> View Post Detail Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP\_15** | | | |
| **Use Case No.** | UC\_ ASP\_15 | **Use Case Version** | 0.2 |
| **Use Case Name** | View Post Detail | | |
| **Author** | Nguyễn Quang Nhật | | |
| **Date** | 17/10/2018 | **Priority** | High |
| **Actor:**   * Room Master   **Summary:**   * This use case allows room master to view post detail.   **Goal:**   * To view post’s detail.   **Triggers:**   * Actor logged into system and stand in home screen. * Press on any finding room post.   **Preconditions:**   * User logged in the system.   **Post conditions:**   * Success: User can view post’s details. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Actor presses “Trang chủ” button. | System displays home screen with:  - “Những bài gần bạn”: Horizontal scroll view.  - “Người dùng chung quận”: Vertical scroll view. | | 2 | Actor presses 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” |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:** extend **UC\_ ASP\_11**  **Business Rules:**   * User already logged in the system. * User must be room master role. | | | |

Table 22: <Room Master> View Post Detail Specification

#### <Member> Overview Use Case



Figure 23: <Member> Overview Use Case

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

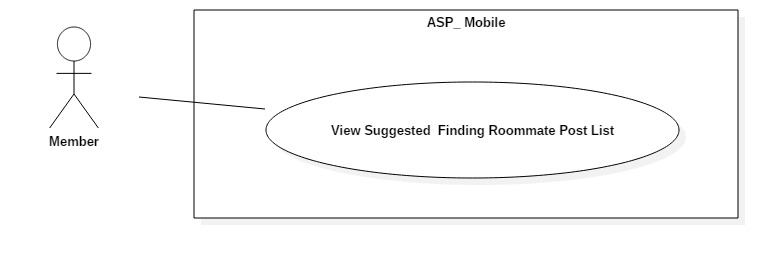


Figure 24: <Member> View Suggested Finding Roommate Posts List Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP\_16** | | | |
| **Use Case No.** | UC\_ ASP\_16 | **Use Case Version** | 0.2 |
| **Use Case Name** | View Suggested Finding Roommate Posts | | |
| **Author** | Nguyễn Quang Nhật | | |
| **Date** | 17/10/2018 | **Priority** | High |
| **Actor:**   * Member   **Summary:**   * This use case allows member to view suggested finding roommate posts.   **Goal:**   * To 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 finding roommate posts. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Actor presses “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 Finding Roommate Posts List Specification

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

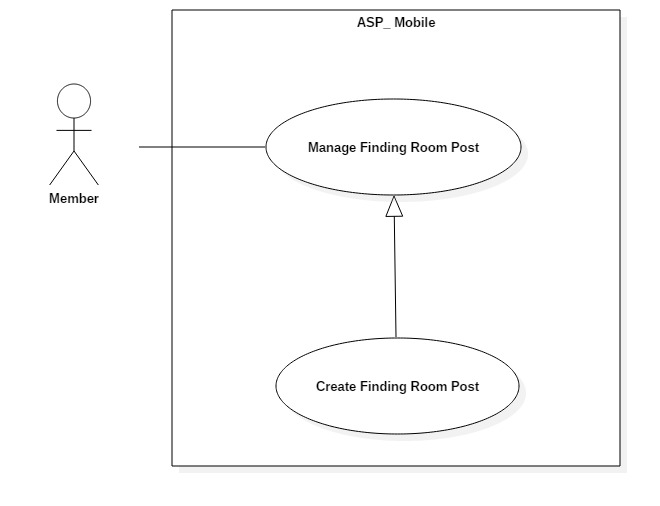


Figure 25:< Member> Create Finding Room Post Use Case

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

Table 24: <Member> Create Finding Room Post specification

##### <Member> Update Finding Room Post

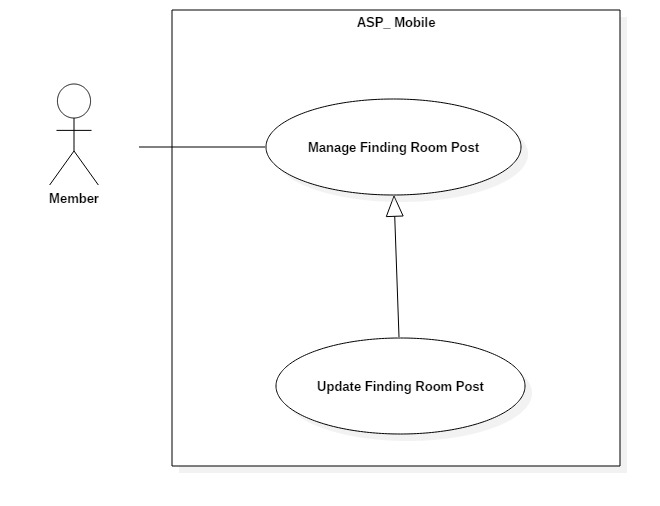


Figure 26:< Member> Update Finding Room Post Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP\_18** | | | |
| **Use Case No.** | UC\_ ASP\_18 | **Use Case Version** | 0.2 |
| **Use Case Name** | Update Finding Room 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 finding room post.   **Triggers:**   * Actor sends update finding room post command.   **Preconditions:**   * User logged in the system. * Post already uploaded.   **Post conditions:**   * Success: User can update finding room post. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Actor presses “Tài khoản” button. | System navigates to “Tải khoản” screen.  System displays “Tài khoản” screen with:  - “Đã đăng” tab. | | 2 | Actor presses 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 presses on “Chỉnh sửa” button. | Systems navigate to update finding room post screen.  System displays update screen with:  - “Tên bài đăng”  - “Khoảng giá”  - “Diện tích”  - “Mô tả”  - “Số điện thoại”  - “Quận”  - “Lưu” button | | 6 | Actor fills form and presses 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 inputs wrong information in some fields. | 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. * Information are updated must be matched the defined information structure in system. * User must be member role. * Finding room post already uploaded to the system. | | | |

Table 25: <Member> Update Finding Room Post specification

##### <Member> Delete Finding Room Post

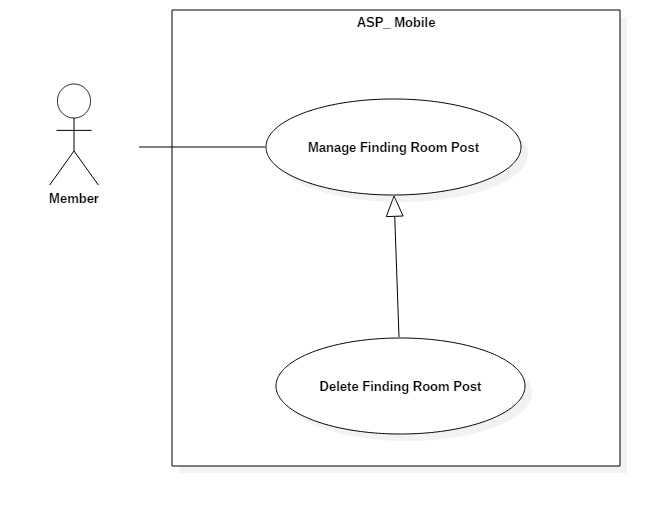


Figure 27: <Member> Delete Finding Room Post Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP\_19** | | | |
| **Use Case No.** | UC\_ ASP\_19 | **Use Case Version** | 0.2 |
| **Use Case Name** | Delete Finding Room Post | | |
| **Author** | Nguyễn Đỗ Minh Đức | | |
| **Date** | 17/10/2018 | **Priority** | High |
| **Actor:**   * Room Master   **Summary:**   * This use case allows room master to delete finding room 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 finding room post. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Actor presses “Tài khoản” button. | System navigates to “Tải khoản” screen.  System displays “Tài khoản” screen with:  - “Đã đăng” tab. | | 2 | Actor presses 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 presses 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. * Finding room post already uploaded to the system. * User must be member role. | | | |

Table 26: <Member> Delete Finding Room Post specification

##### <Member> Negotiate

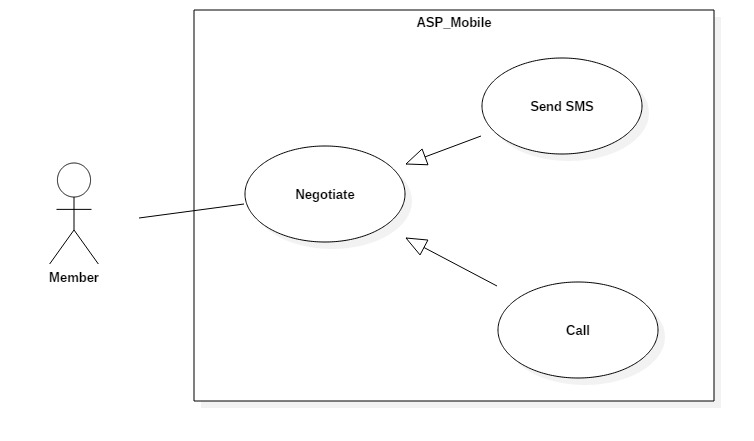


Figure 28: <Member> Negotiate Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP\_20** | | | |
| **Use Case No.** | UC\_ ASP\_20 | **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 presses 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 presses 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 presses 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

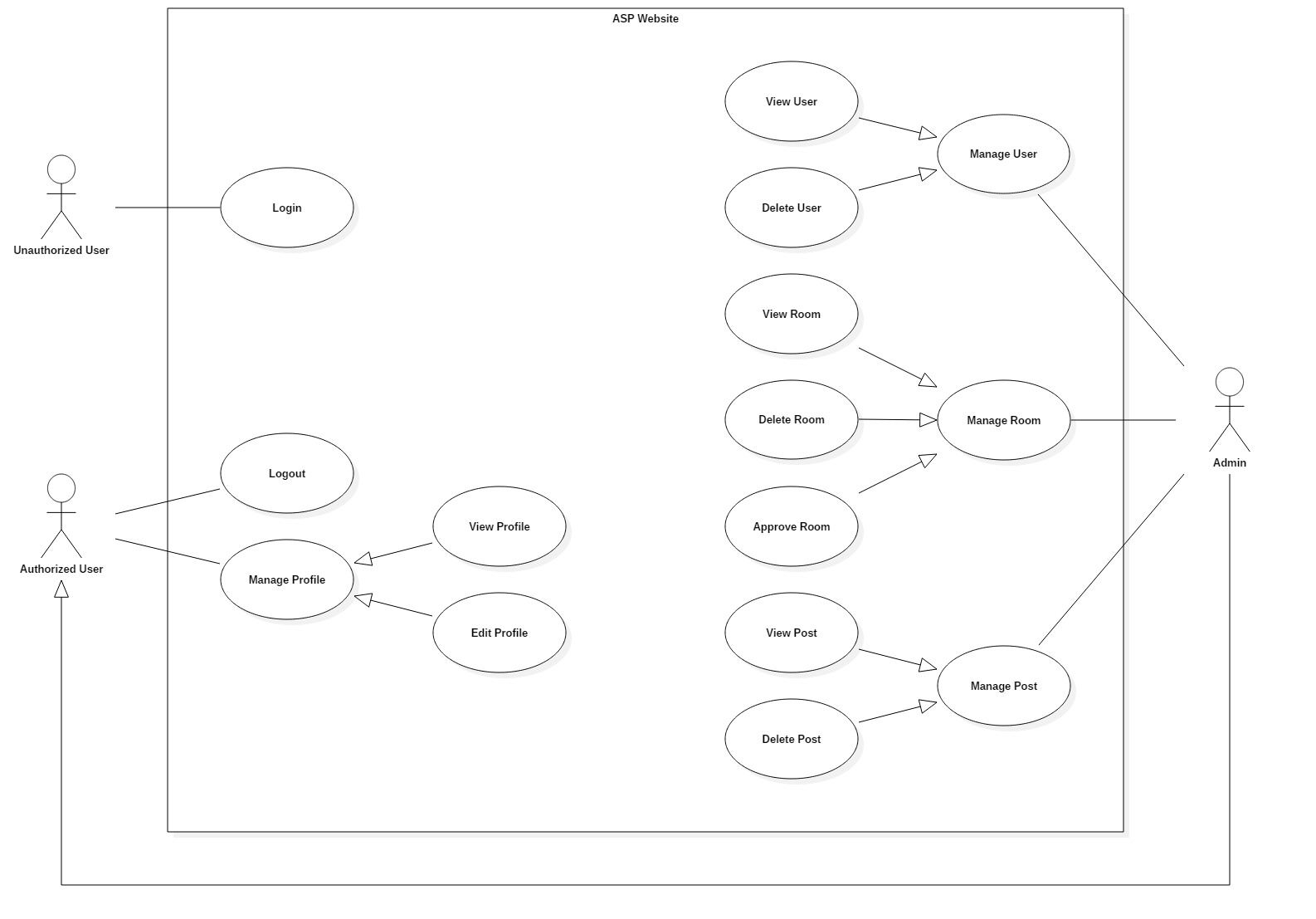


Figure 29: <Admin> Overview Use Case

##### <Admin> Manage Rooms

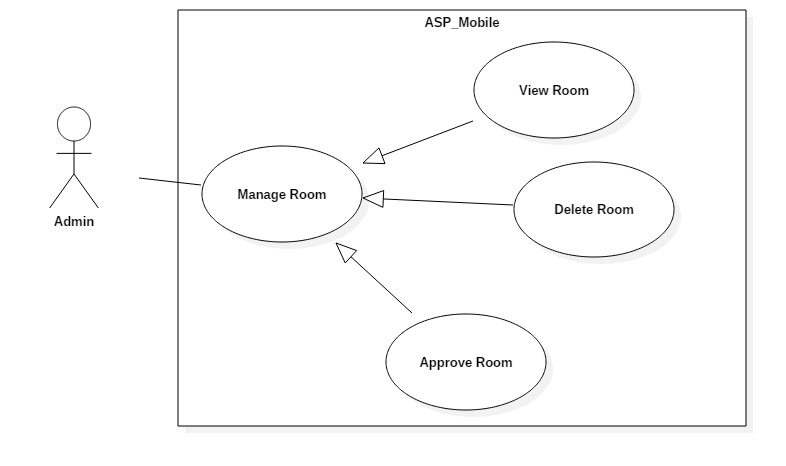


Figure 30: <Admin> Manage Room Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP\_21** | | | |
| **Use Case No.** | UC\_ ASP\_21 | **Use Case Version** | 0.2 |
| **Use Case Name** | Manage Room | | |
| **Author** | Vương Minh Thông | | |
| **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> Approve Room

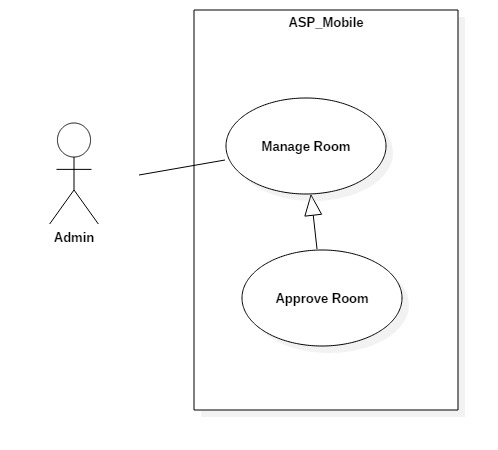


Figure 31: <Admin> Approve Room Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP\_23** | | | |
| **Use Case No.** | UC\_ ASP\_23 | **Use Case Version** | 0.2 |
| **Use Case Name** | Approve room | | |
| **Author** | Vương Minh Thông | | |
| **Date** | 17/10/2018 | **Priority** | High |
| **Actor:**   * Admin   **Summary:**   * This use case allows admin to approve room.   **Goal:**   * To approve room.   **Triggers:**   * Actor 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 29: <Admin> Approve room specification

##### <Admin> Manage Users

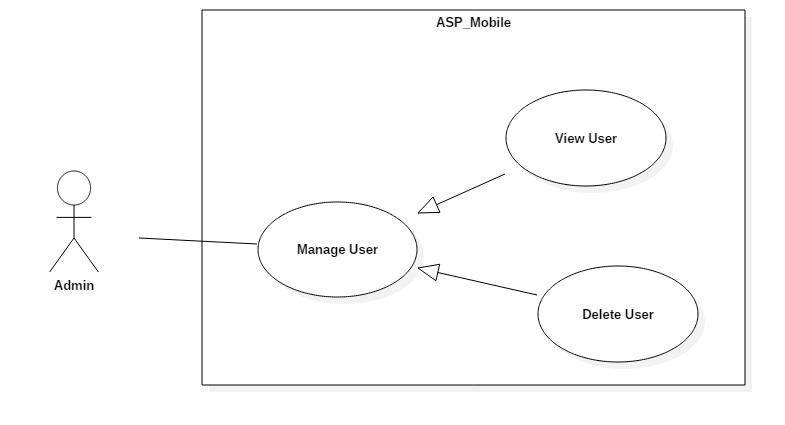


Figure 32: <Admin> Manage User Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP\_24** | | | |
| **Use Case No.** | UC\_ ASP\_24 | **Use Case Version** | 0.2 |
| **Use Case Name** | Manage User | | |
| **Author** | Nguyễn Quang Nhật | | |
| **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 30: <Admin> Manage User specification

##### <Admin> View Users

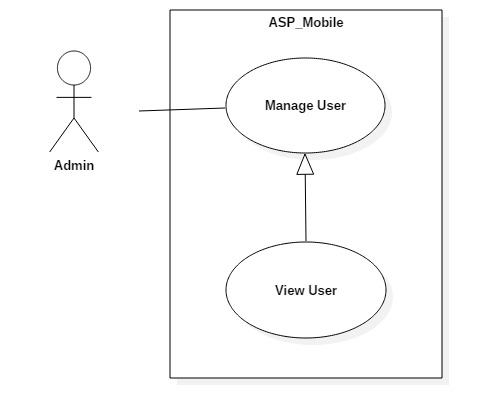


Figure 33: <Admin> View User Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ASP\_25** | | | |
| **Use Case No.** | UC\_ ASP\_25 | **Use Case Version** | 0.2 |
| **Use Case Name** | View User | | |
| **Author** | Nguyễn Quang Nhật | | |
| **Date** | 17/10/2018 | **Priority** | High |
| **Actor:**   * Admin   **Summary:**   * This use case allows admin to view user list.   **Goal:**   * To view user list.   **Triggers:**   * Actor click on “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 31: <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

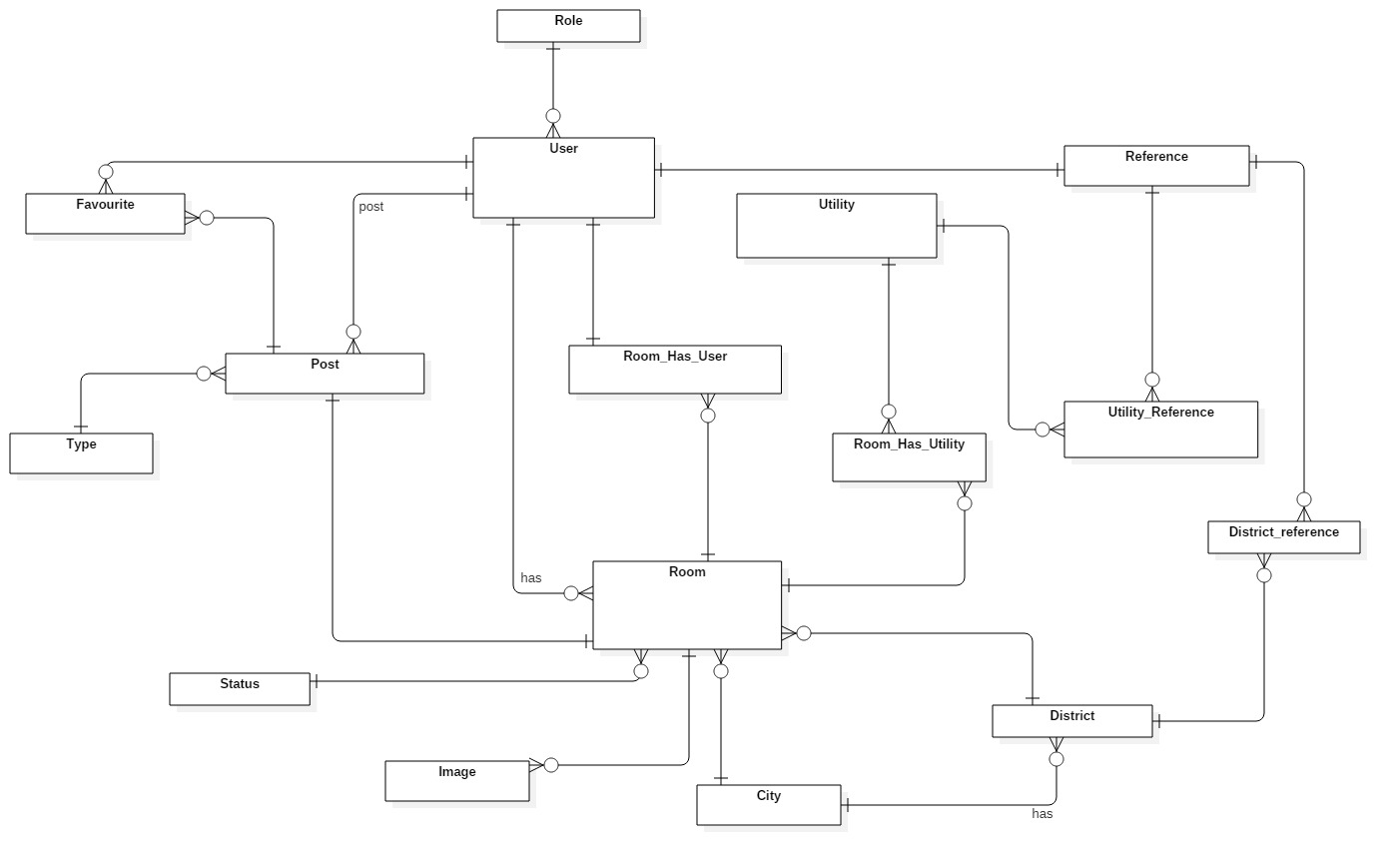


Figure 34: Conceptual Diagram

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

Table 32: 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.

## 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.

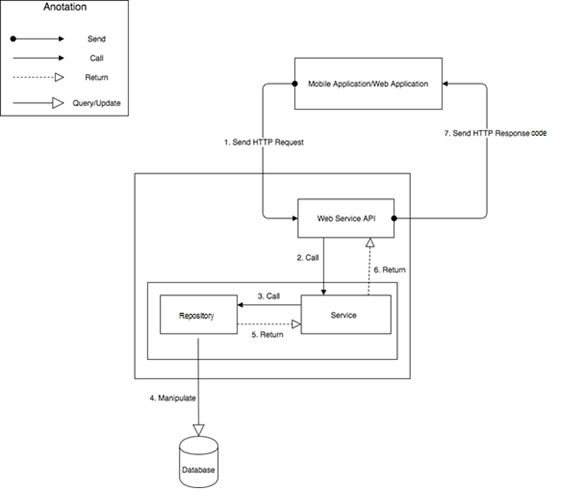


Figure 35: System Architecture Design

## Component Diagram

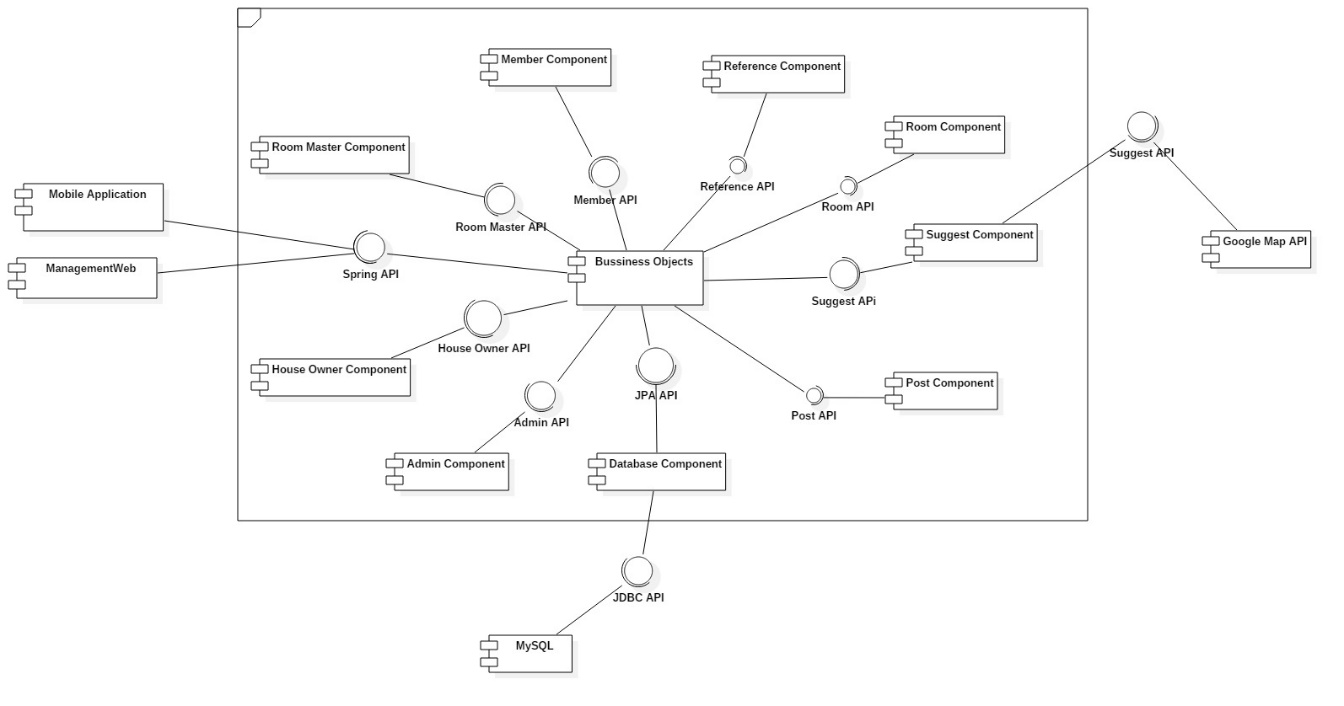


Figure 36: Component Diagram

|  |  |
| --- | --- |
| **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. |

Table 33: Component Diagram Explanation

## Detail Description

### Class Diagram

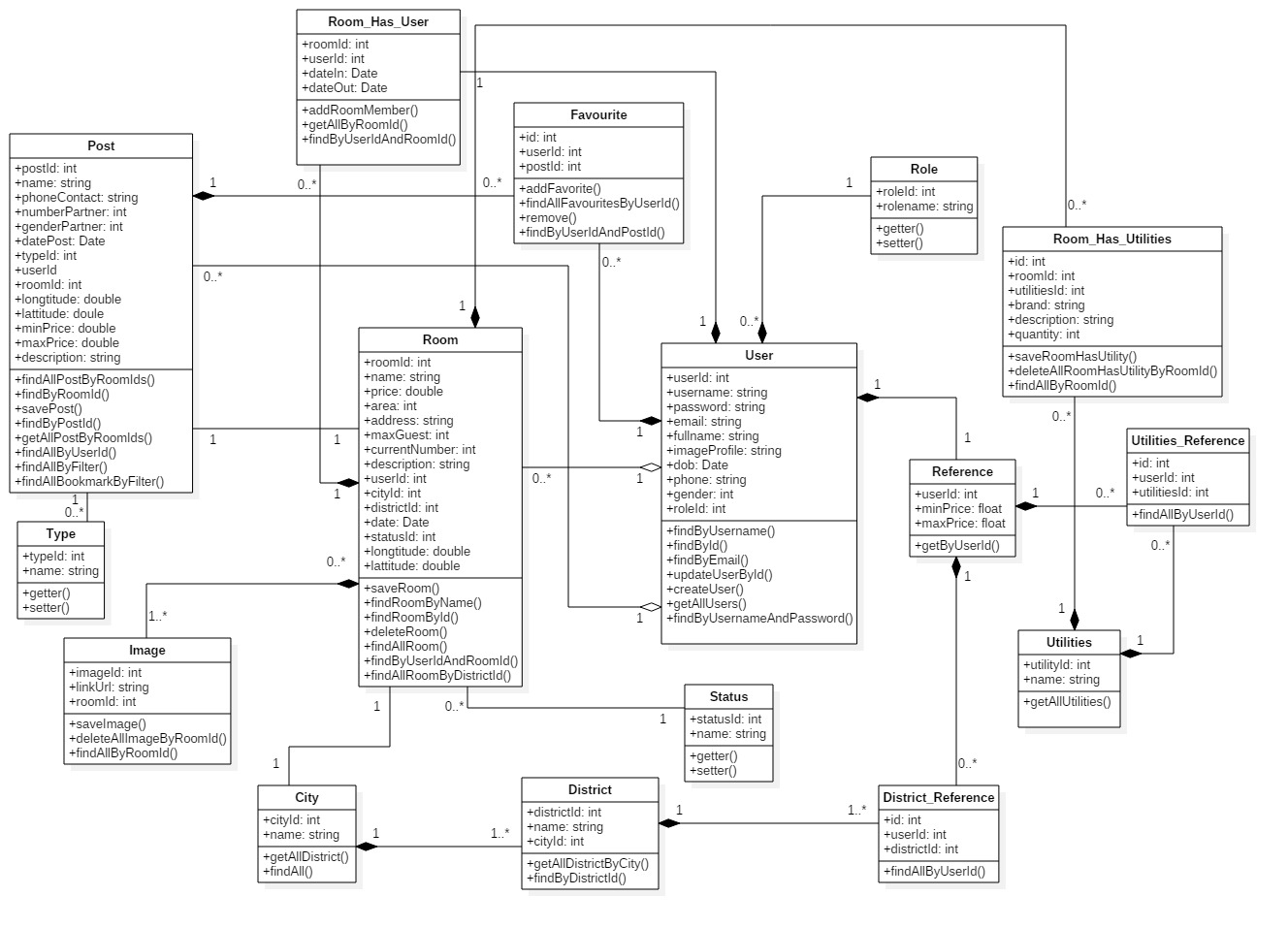


Figure 37: Class Diagram

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

Table 34: Class Diagram Explanation

### Class Diagram Explanation

#### User

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| userId | Integer | private | Unique identifier of user |
| username | String | private | Username of user |
| password | String | private | Password of user |
| email | String | private | Email of user |
| fullname | String | private | Full name of user |
| imageProfile | String | private | Image profile of user |
| dob | Date | private | Date of birth of user |
| phone | String | private | Phone number of user |
| gender | Integer | private | Gender of user |
| roleId | Integer | private | Role of user |

Table 35: User Attributes

#### Room

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| roomId | Integer | private | Unique identifier of room |
| name | String | private | Name of room |
| price | Double | private | Price of room |
| area | Integer | private | Area of room |
| address | String | private | Address of room |
| maxGuest | Integer | private | Maximum number of guest in room |
| currentNumber | Integer | private | Current number of guest in room |
| description | String | private | Description of room |
| userId | Integer | private | Identifier of user in room |
| cityId | Integer | private | City’s identify of room |
| districtId | Integer | private | District’s identify of room |
| date | Date | private | Created date of room |
| statusId | Integer | private | Status of room that be approved or declined |
| longtitude | Double | private | Longitude of room |
| lattitude | Double | private | Latitude of room |

Table 36: Room Attributes

#### Post

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| postId | Integer | private | Unique identifier of post |
| name | String | private | Name of post |
| phoneContact | String | private | Phone number of room master |
| numberPartner | Integer | private | Number of partner wanted in post |
| genderPartner | Integer | private | Gender of partner wanted in post |
| datePost | Date | private | Created date of post |
| typeId | Integer | private | Type of post, define finding room post or finding roommate post. |
| userId | Integer | private | Identifier of user in room |
| roomId | Integer | private | Room’s identifier of post |
| longtitude | Double | private | Longitude of post |
| lattitude | Double | private | Latitude of post |
| minPrice | Double | private | Min price of post |
| maxPrice | Double | private | Max price of post |
| description | String | private | Description of post |

Table 37: Post Attributes

#### Role

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| roleId | Integer | private | Unique identifier of role |
| rolename | String | private | Name of role |

Table 38: Role Attributes

#### Favourite

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | Integer | private | Unique identifier of favourite |
| userId | Integer | private | User’s identifier of favourite |
| postId | Integer | private | Post’s identifier of favourite |

Table 39: Favourite Attributes

#### Utilities

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| utilitiesId | Integer | private | Unique identifier of utilities |
| name | String | private | Name of Utilities |

Table 40: Utility Attributes

#### Reference

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| userId | Integer | private | User’s identifier of reference |
| minPrice | Double | private | Min price is chosen by user |
| maxPrice | Double | private | Max price is chosen by user |

Table 41: Reference Attributes

#### District

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| districtId | Integer | private | Unique identifier of district |
| name | String | private | Name of district |
| cityId | Integer | private | City’s identifier of district |

Table 42: District Attributes

#### City

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| cityId | Integer | private | Unique identifier of city |
| name | String | private | Name of city |

Table 43: City Attributes

#### Image

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| imageId | Integer | private | Unique identifier of image |
| linkUrl | String | private | Url of image |
| roomid | Integer | private | Room’s identifier of image |

Table 44: Image Attributes

#### Type

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| typeId | Integer | private | Unique identifier of type. |
| name | String | private | Name of type |

Table 45: Type Attributes

#### Status

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| statusId | Integer | private | Unique identifier of status |
| name | String | private | Name of status |

Table 46: Status Attributes

#### Room\_Has\_Utilities

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| Id | Integer | private | Unique identifier of room’s utilities |
| roomId | Integer | private | Room’s identifier in room has utilities field |
| utilitiesId | Integer | private | Utility’s identifier relate to room |
| brand | String | private | Brand of utilities are selected |
| description | String | private | Description of utilities are selected |
| quantity | Integer | private | Describe how many utilities |

Table 47: Room Has Utilities Attributes

#### Utilities\_Reference

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | Integer | private | Unique identifier of reference |
| userId | Integer | private | User’s identifier of reference |
| utilitiesId | Integer | private | Utility’s identifier of reference |

Table 48: Utilities Reference Attributes

#### District\_Reference

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | Integer | private | Unique identifier of district’s reference |
| userId | Integer | private | User’s identifier of reference |
| districtId | Integer | private | District’s identifier of reference |

Table 49: District Reference Attributes

#### Room\_Has\_User

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| roomId | Integer | private | Room’s identifier has user |
| userId | Integer | private | User’s identifier in room |
| dateIn | Date | private | Date user move in |
| dateOut | Date | private | Date user move out |

Table 50: Room Has User Attributes

### Interactive diagram

#### Sequence Diagram

##### Register

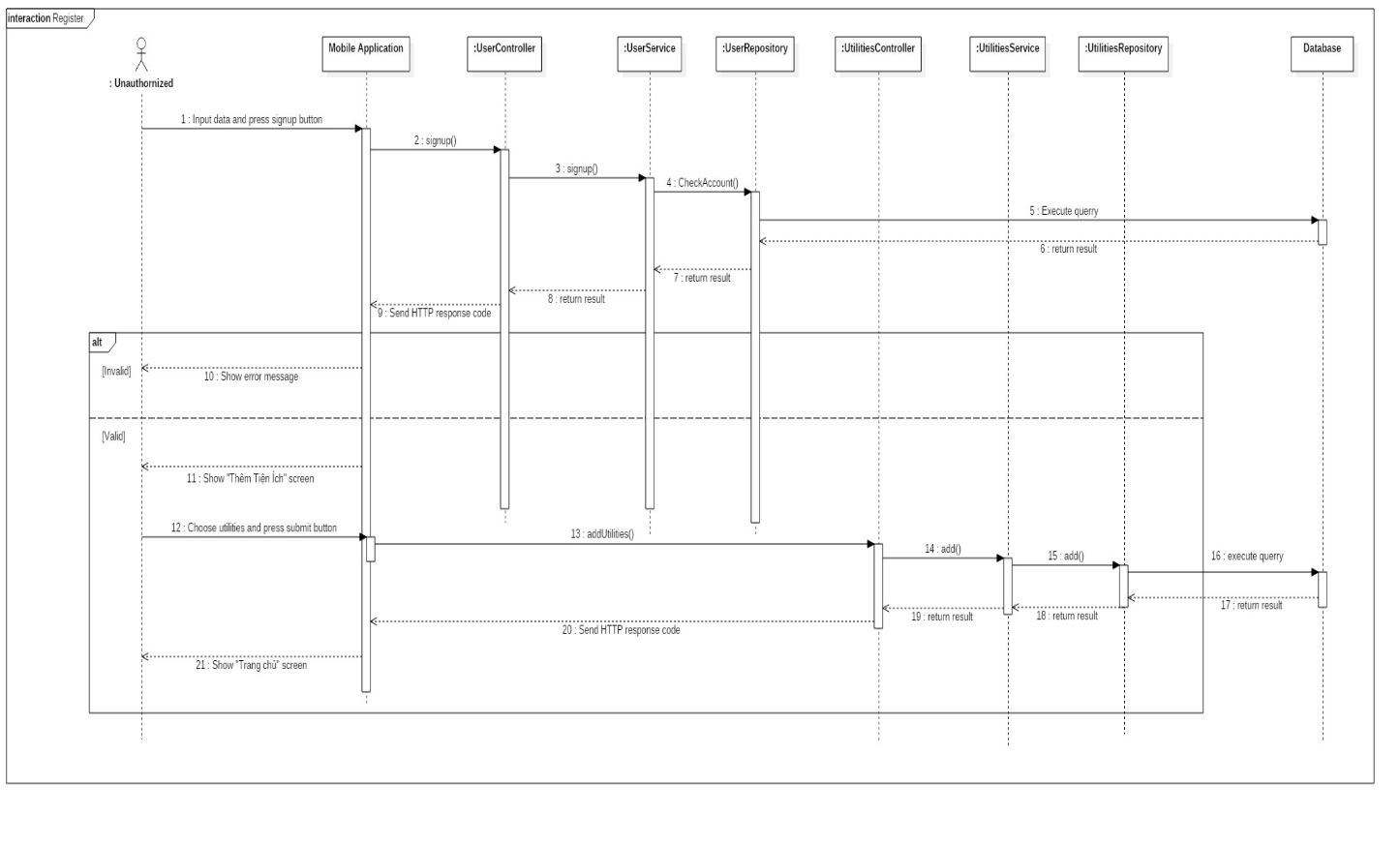


Figure 38: Sequence Diagram - <Unauthorized> Register

##### Login

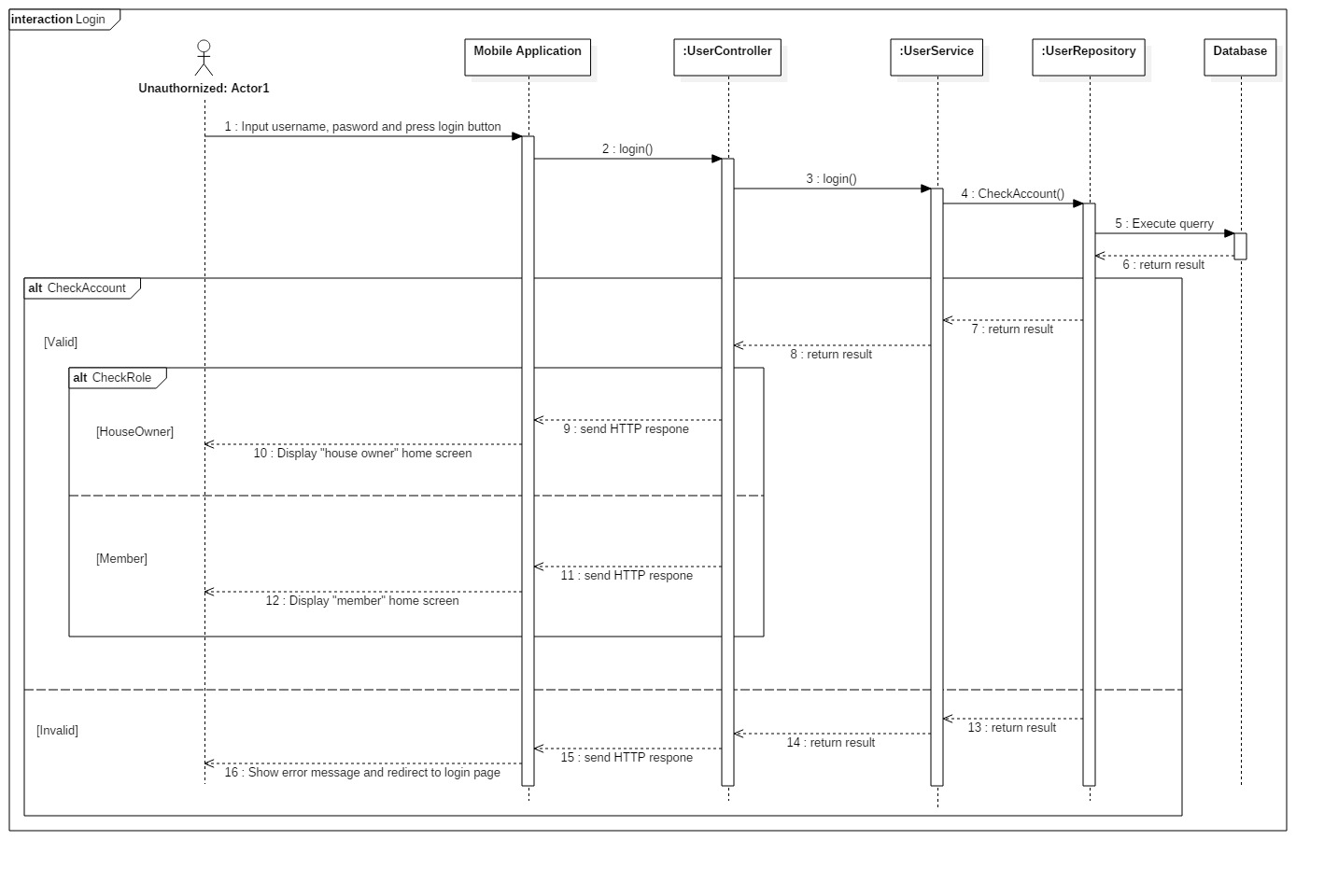


Figure 39: Sequence Diagram - <Unauthorized> Login

##### Create Room

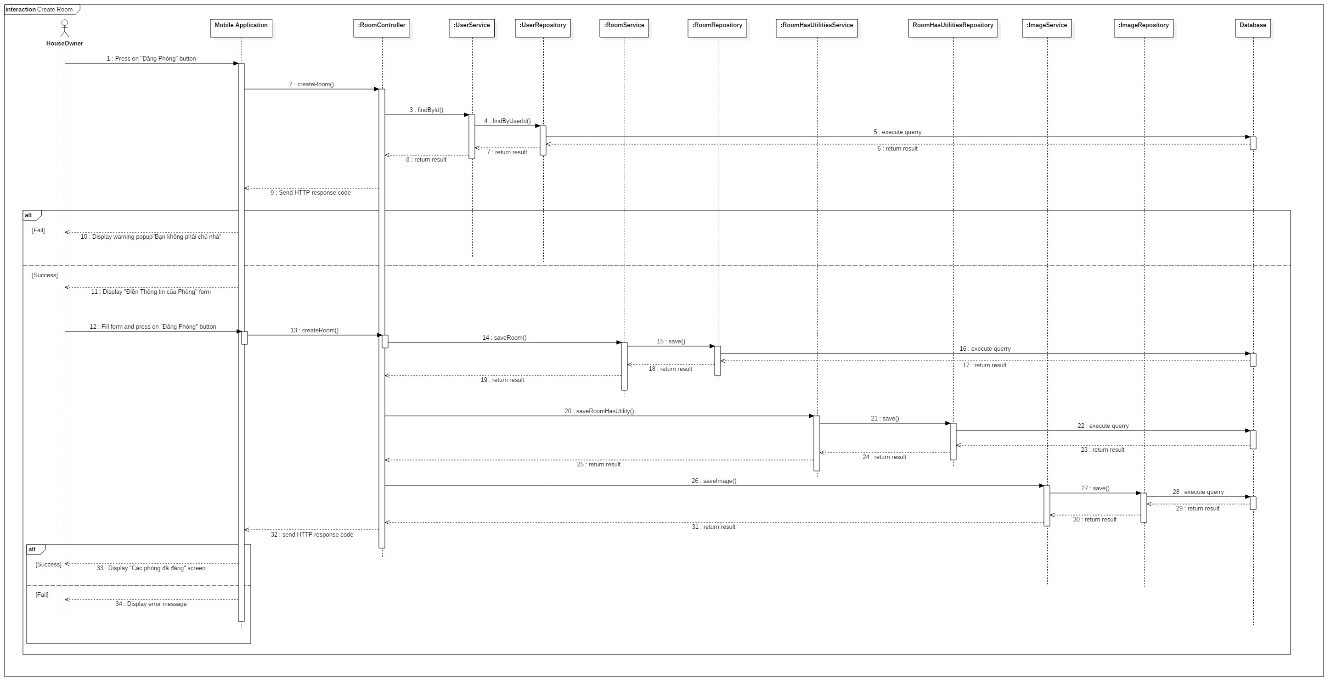


Figure 40: Sequence Diagram - <House owner> Create Room

##### Update Room

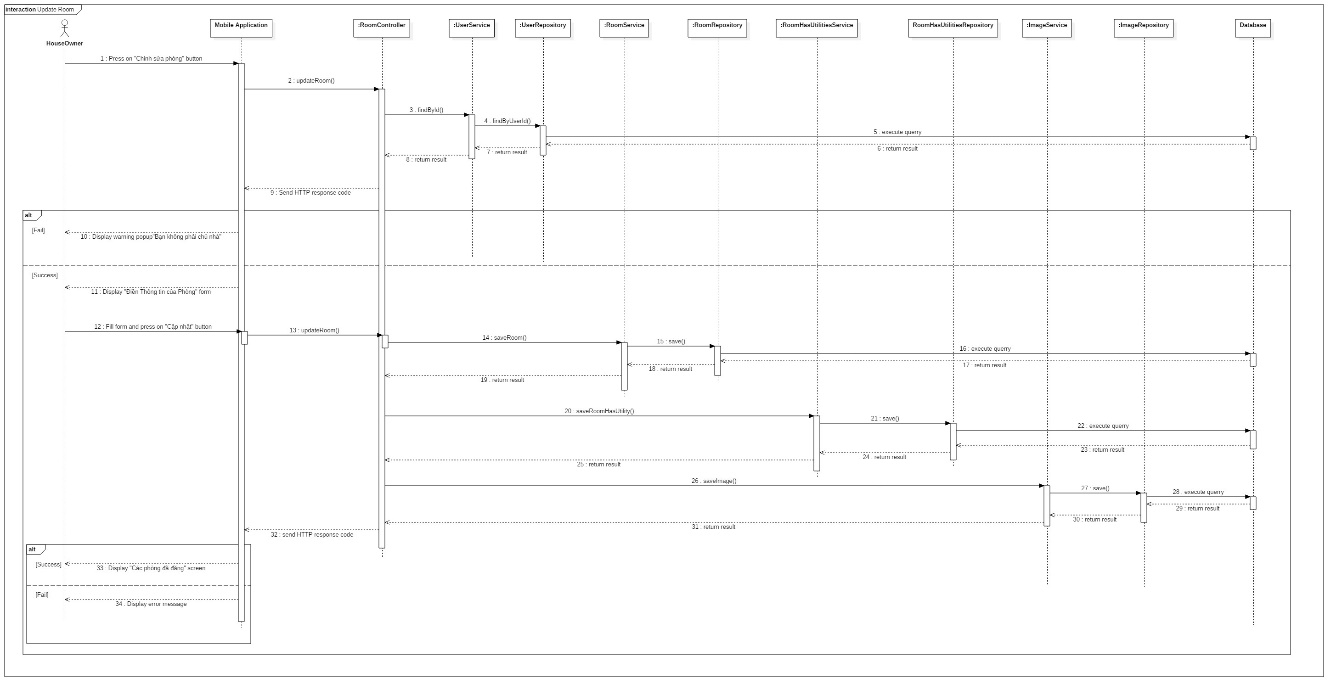


Figure 41: Sequence Diagram - <House owner> Update Room

##### Delete Room

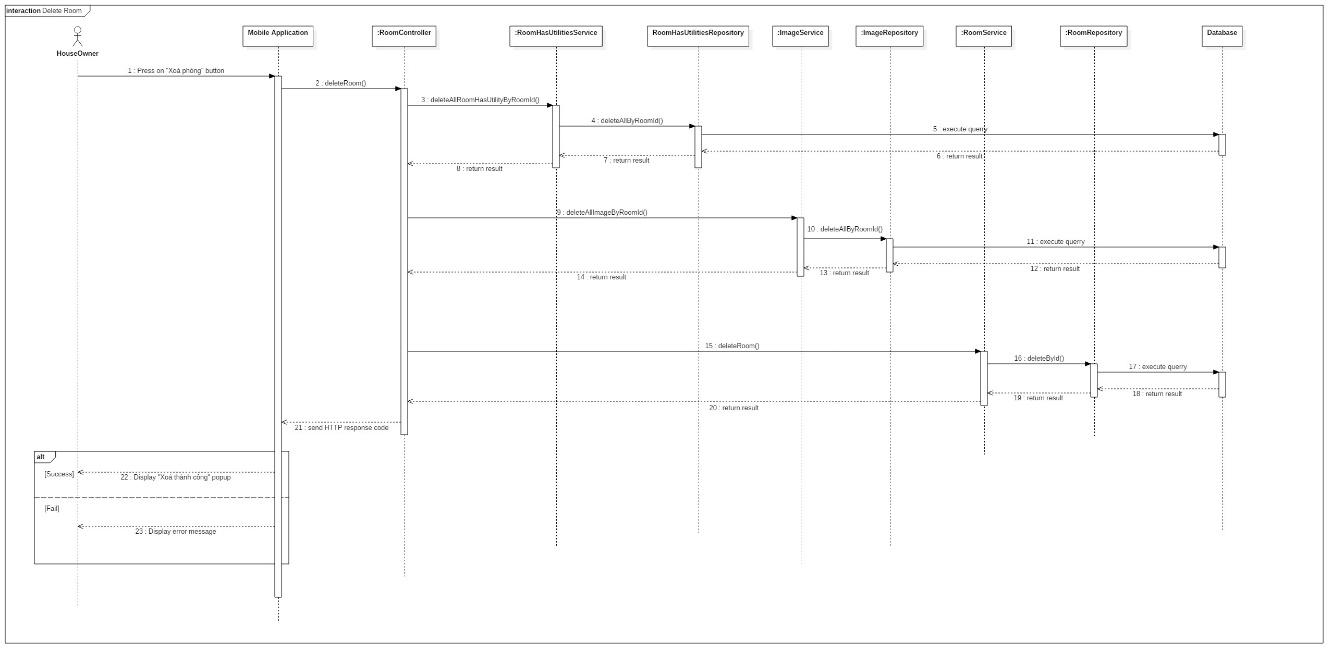


Figure 42: Sequence Diagram - <House owner> Delete Room

##### View Single Room

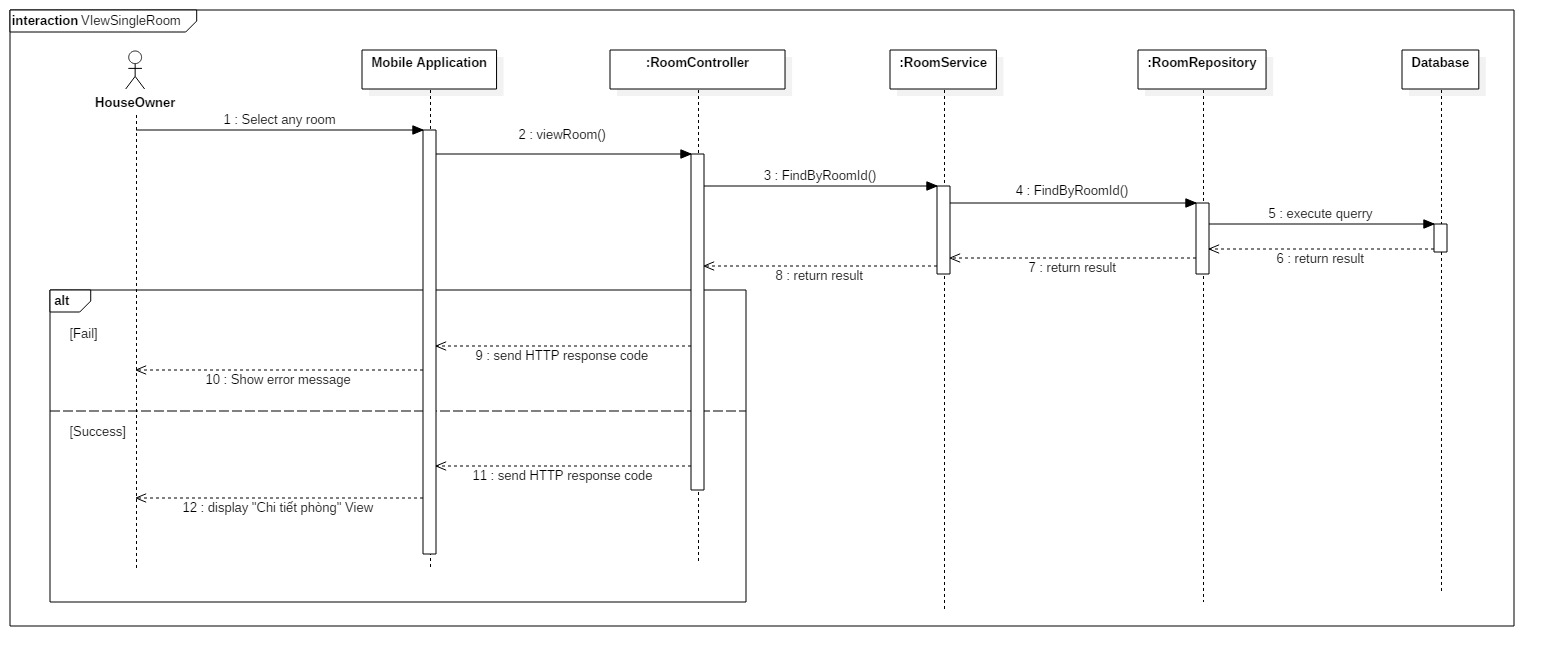


Figure 43: Sequence Diagram - <House owner> View Single Room

##### Add Member into Room

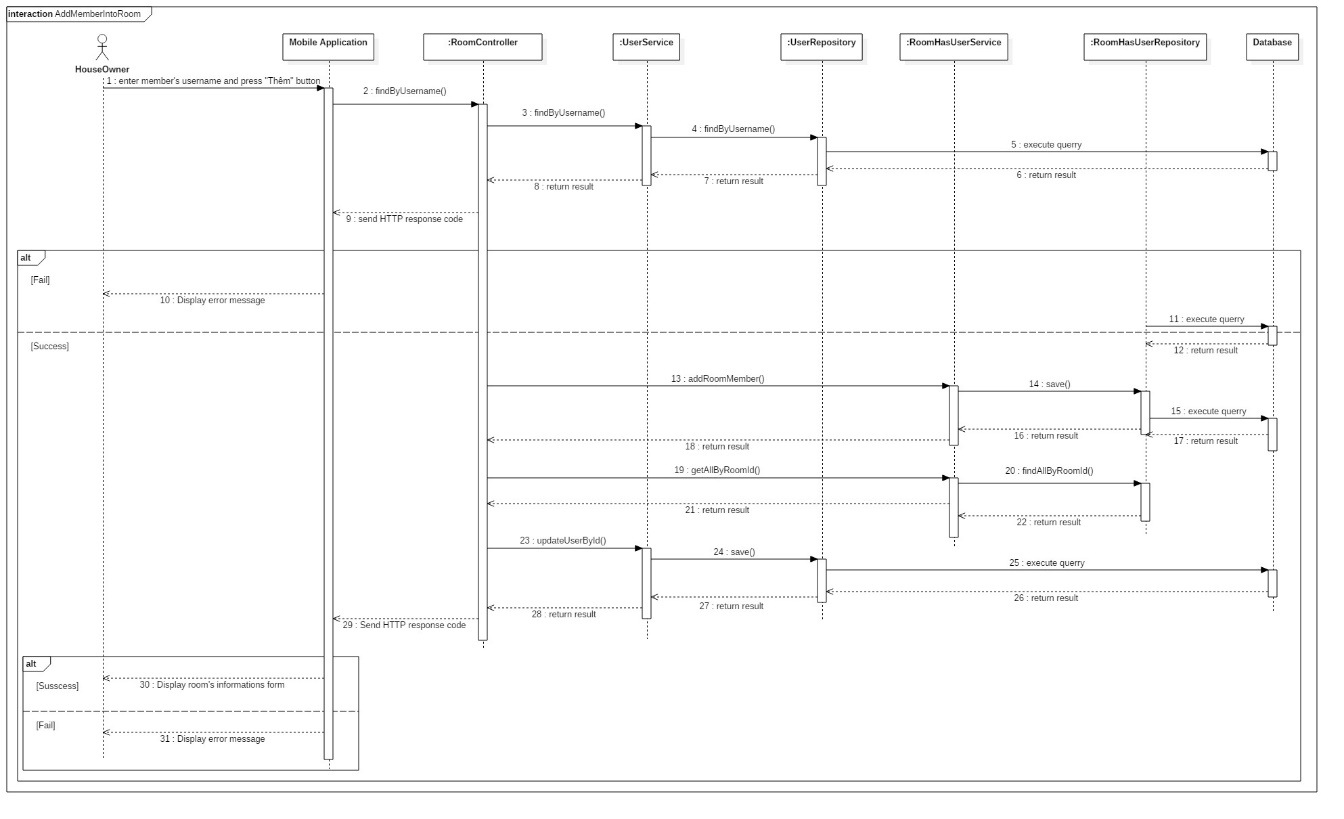


Figure 44: Sequence Diagram - <House owner> Add Member into Room

##### Delete Member of Room

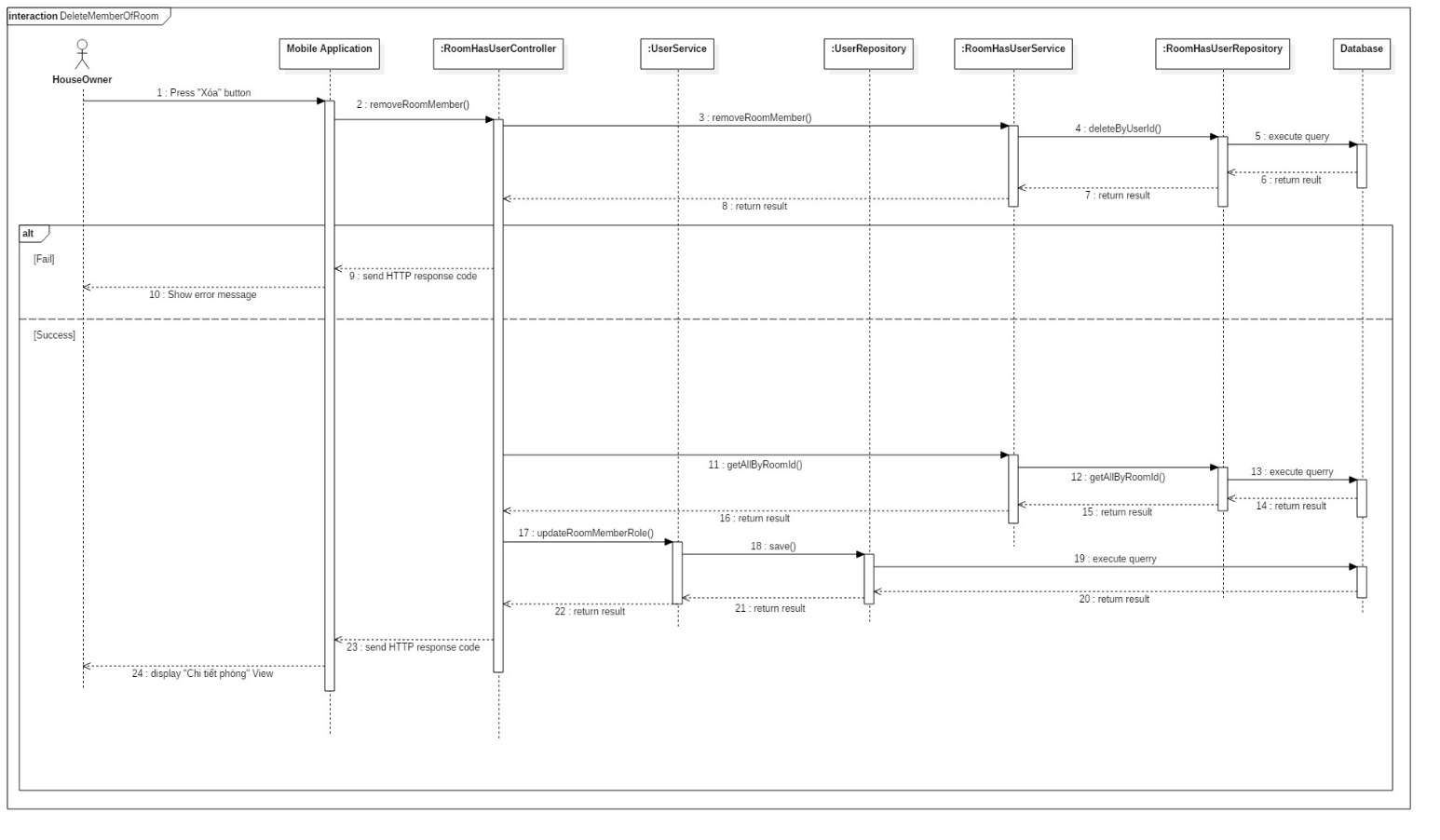


Figure 45: Sequence Diagram - <House owner> Delete Members of Room

##### Create Finding Roommate Post

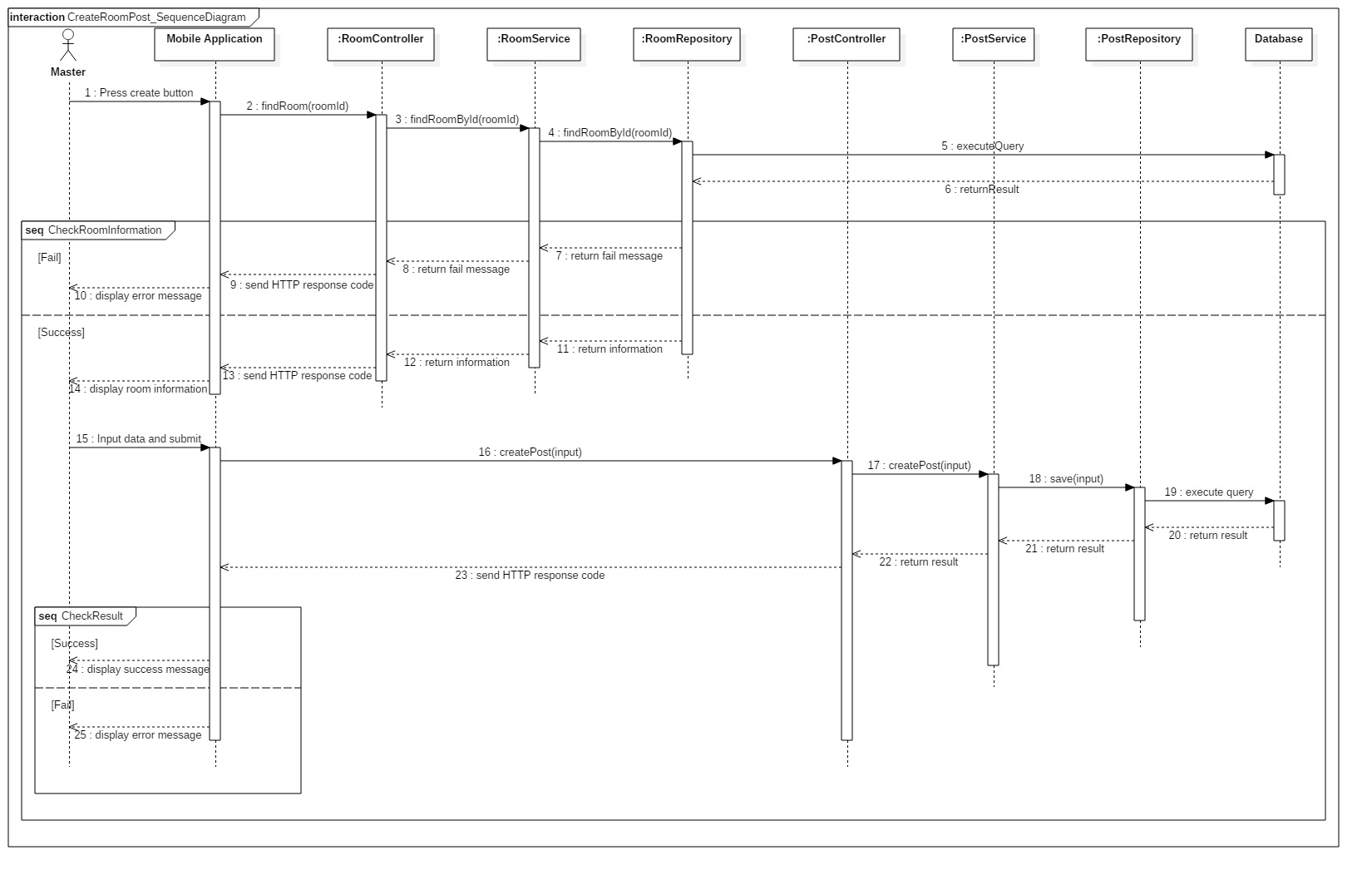


Figure 46: Sequence Diagram - <Room Master> Create Finding Roommate Post

##### Delete Finding Roommate Post

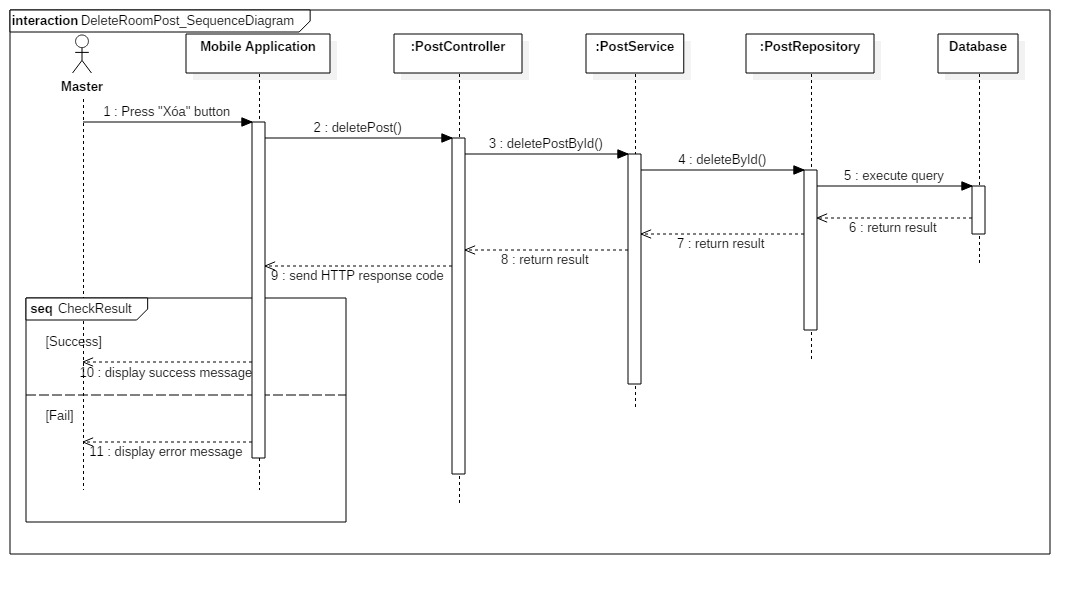


Figure 47: Sequence Diagram - <Room Master> Delete Finding Roommate Post



##### Update Finding Roommate Post

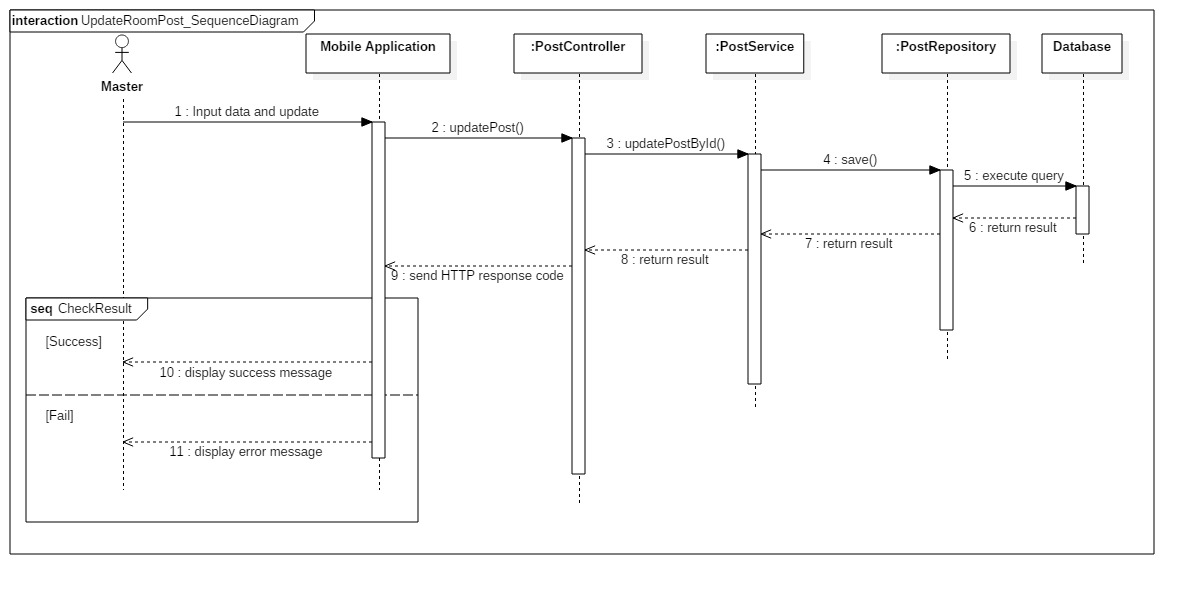


Figure 48: Sequence Diagram - <Room Master> Update Finding Roommate Post



##### View Suggest List



Figure 49: Sequence Diagram - <Room Master> View Suggest List

##### Create Finding Room Post

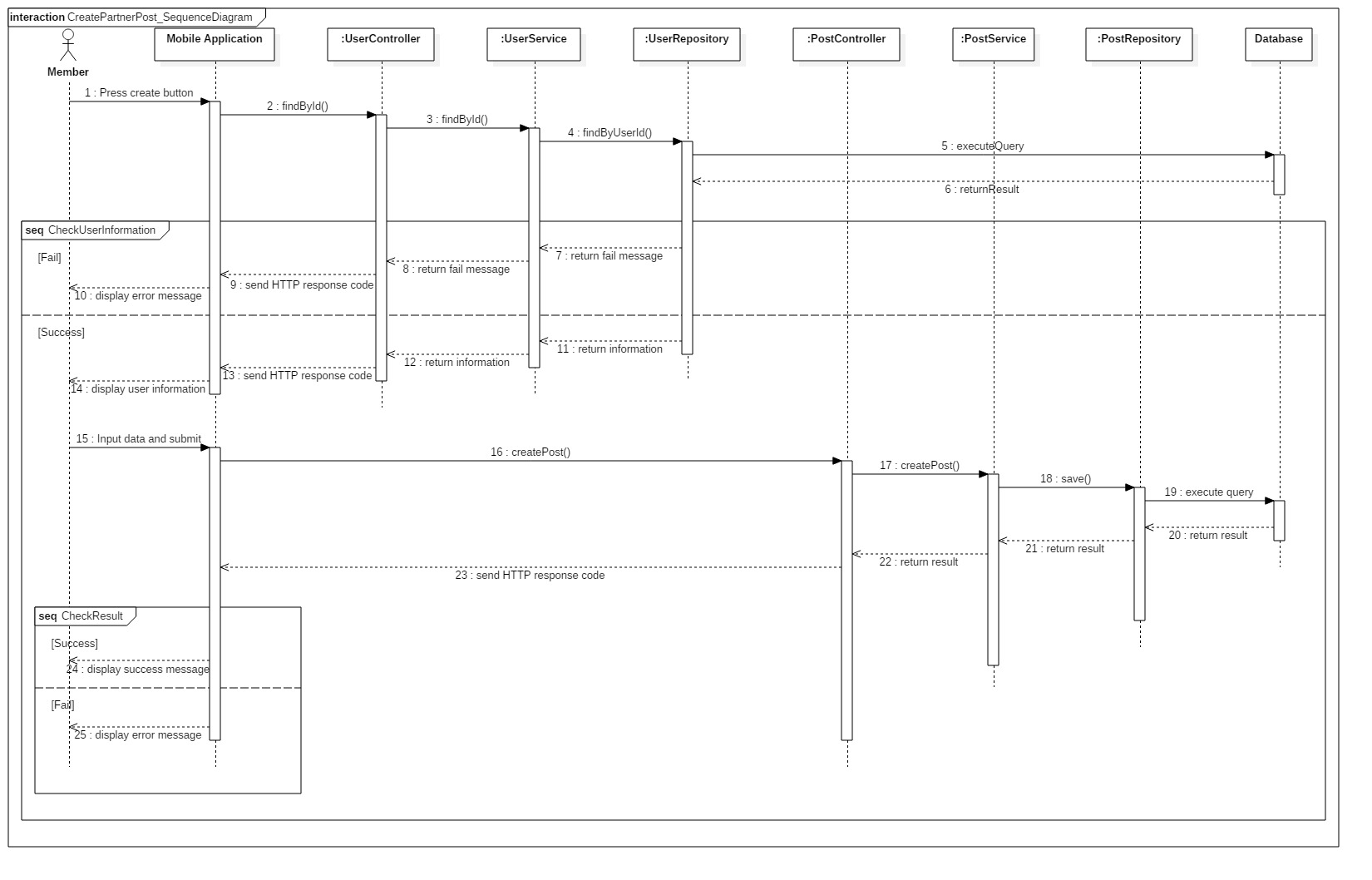


Figure 50: Sequence Diagram - <Member> Create Finding Room Post

##### Update Finding Room Post

Figure 51: Sequence Diagram - <Member> Update Finding Room Post

##### Delete Finding Room Post

Figure 52: Sequence Diagram - <Member> Delete Finding Room Post

## Interface

### Admin Interface Design

#### Login

Figure 53: <Admin> Login Interface

Table 51: Admin login - Field, Buttons

#### Manage User

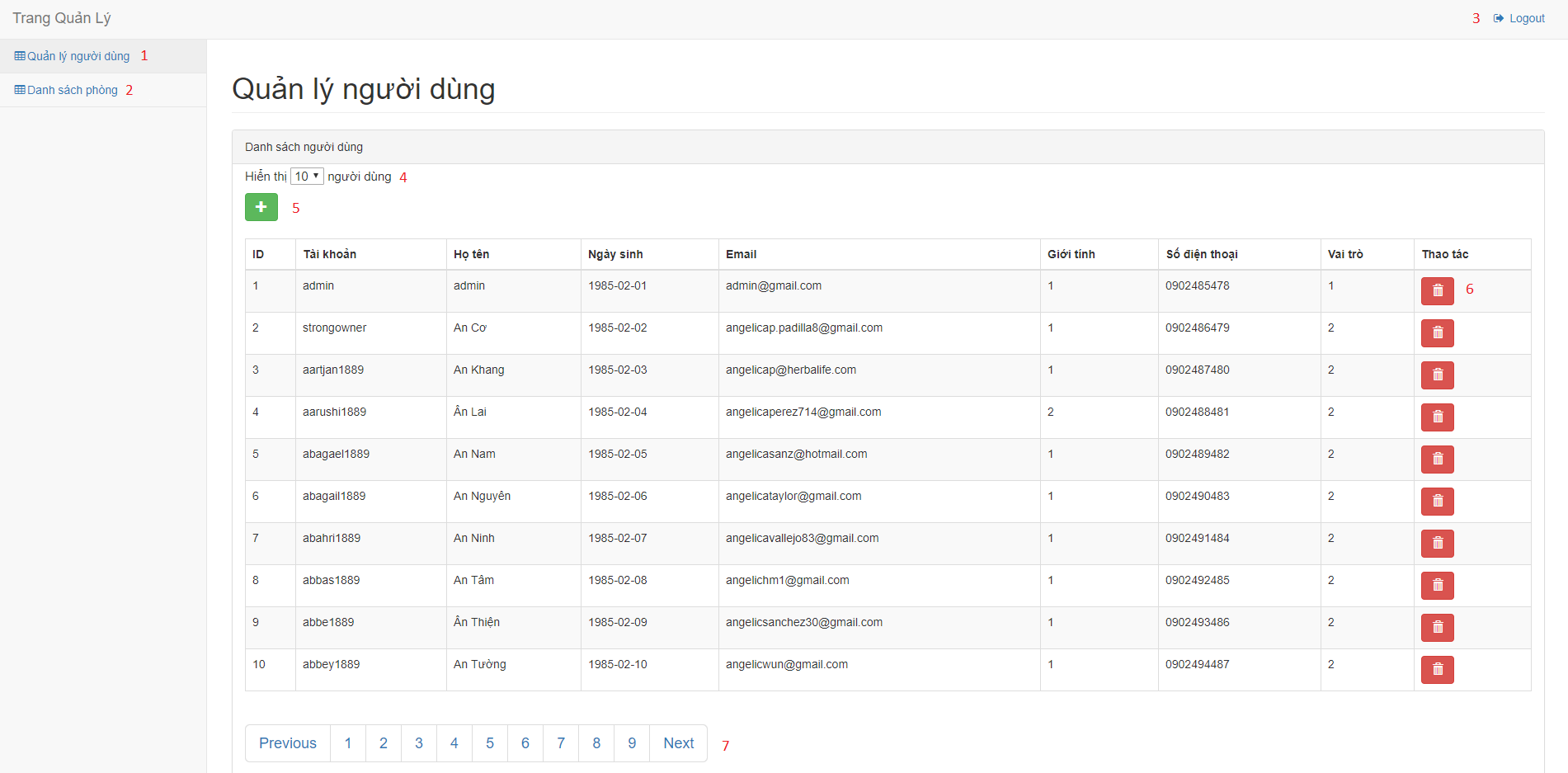


Figure 54: <Admin> Manage User Interface

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

#### Manage Room

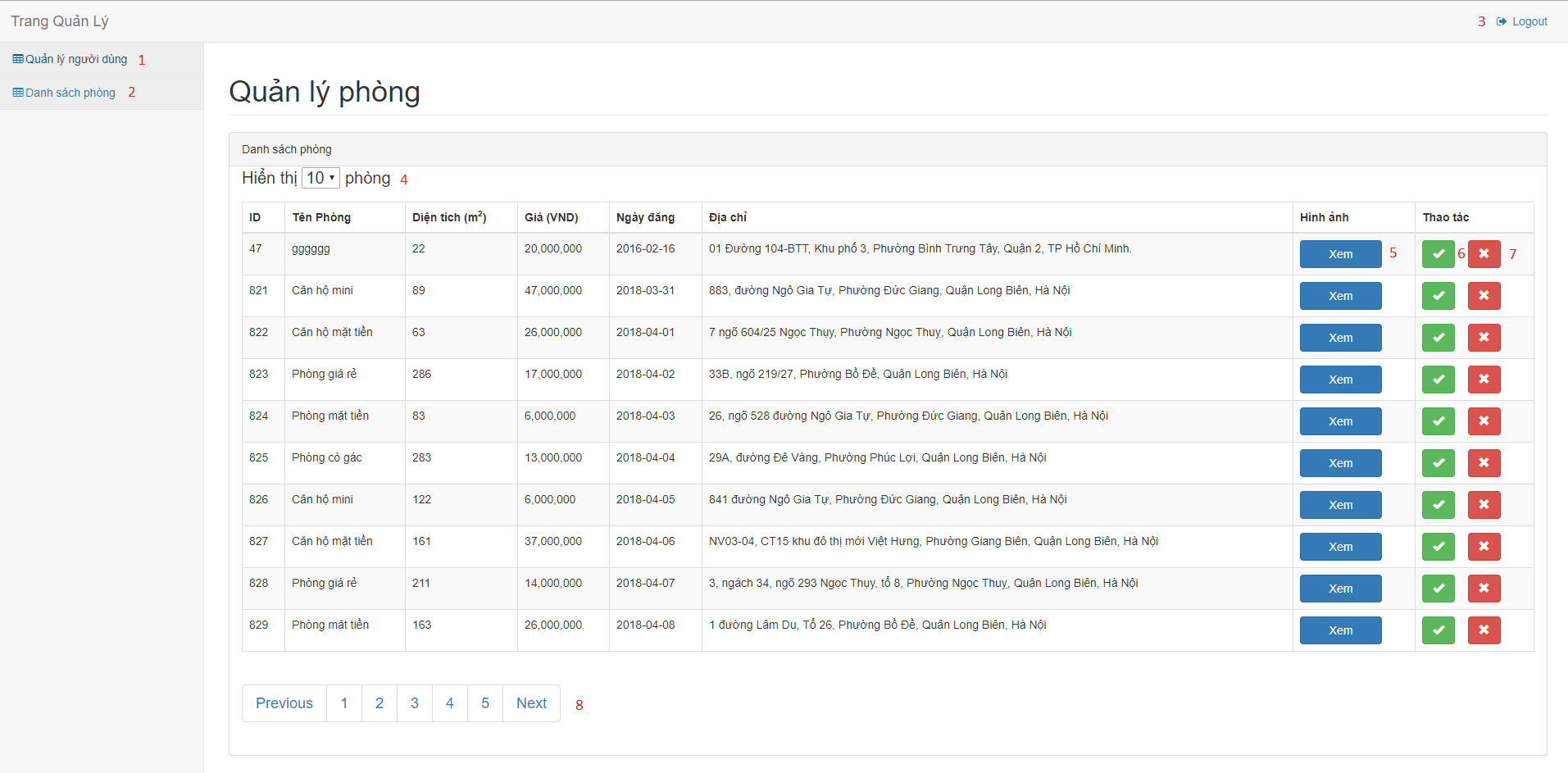


Figure 55: <Admin> Manage Room Interface

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

#### View User List

Figure 56: <Admin> View User List Interface

Table 54: Admin View User List - Field, Buttons, Hyperlinks

#### View Room List

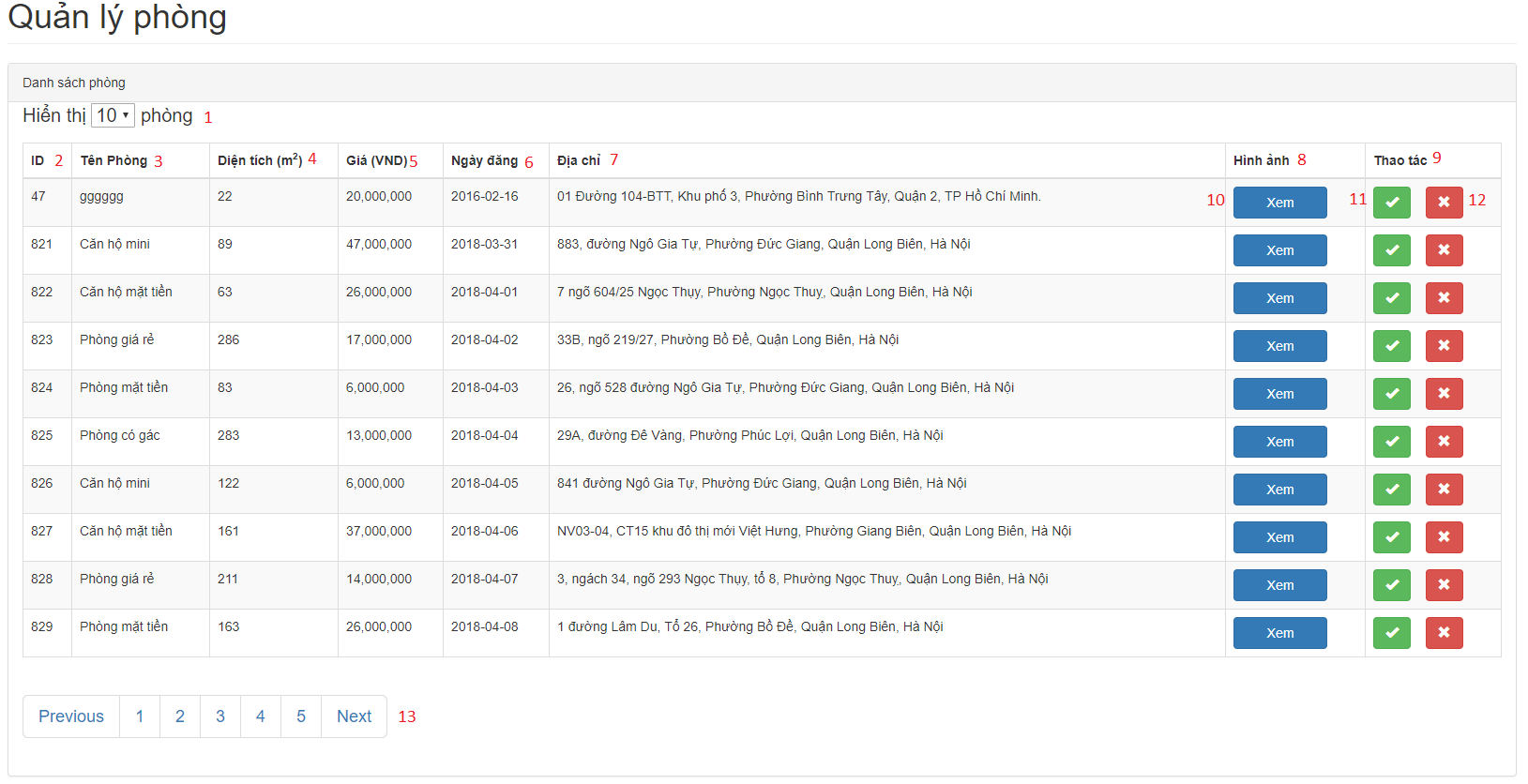


Figure 57: <Admin> View Room List Interface

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

#### Approve Room



Figure 58: <Admin> Approve Room Interface

Table 56: Admin Approve Room - Field, Buttons, Hyperlinks

### Unauthorized User Interface Design

#### Launch Screen

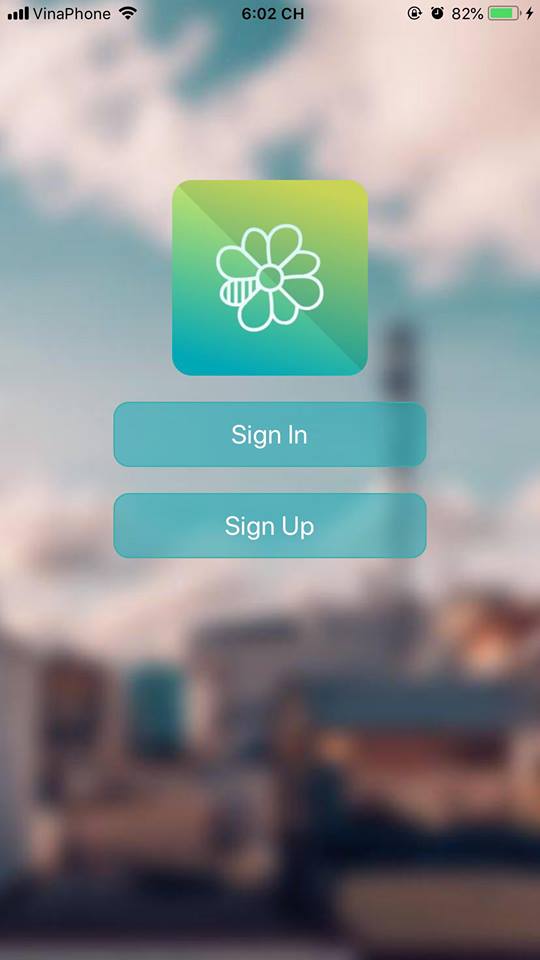


Figure 59: <Unauthorized User> Launch Screen Interface

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **BUTTONS/HYPERLINKS** | | | | |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| **1** | Sign in | Sign in to system | N/A | Go to home screen |
| **2** | Sign up | Create new account | N/A | Go to Sign Up screen |

Table 57: Launch Screen - Buttons

#### Sign Up

Figure 60: <Unauthorized User> Sign Up Interface

Table 58: Sign Up- Field, Buttons

#### Sign In

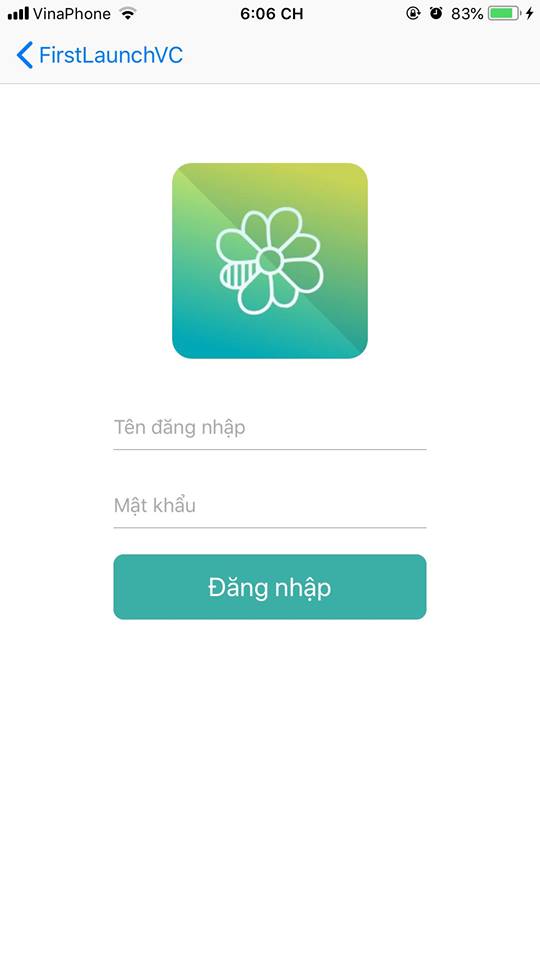


Figure 61: <Unauthorized User> Sign In Interface

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **FIELDS** | | | | | | | |
| **No** | **Field Name** | **Description** | **Read Only** | **Mandatory** | **Control Type** | **Data Type** | **Length** |
| **1** | Username | Fill username | No | Yes | Textbox | String | N/A |
| **2** | Password | Fill password | No | Yes | Textbox | String | N/A |
| **BUTTONS/HYPERLINKS** | | | | | | | |
| **No** | **Function** | **Description** | | **Validation** | **Outcome** | | |
| **3** | Sign In | Login to system | | N/A | Go to home screen | | |

Table 59: Sign In - Field, Buttons

### Authorized User Interface Design

#### View Setting

Figure 62: <Authorized User> View Setting Interface

Table 60: View Setting - Field, Buttons, Hyperlinks

#### Update User Information

Figure 63: <Authorized User> Update Information Interface

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

#### View Profile

Figure 64: <Authorized User> View Profile Interface

Table 62: View Profile - Field, Buttons, Hyperlinks

#### View Notification

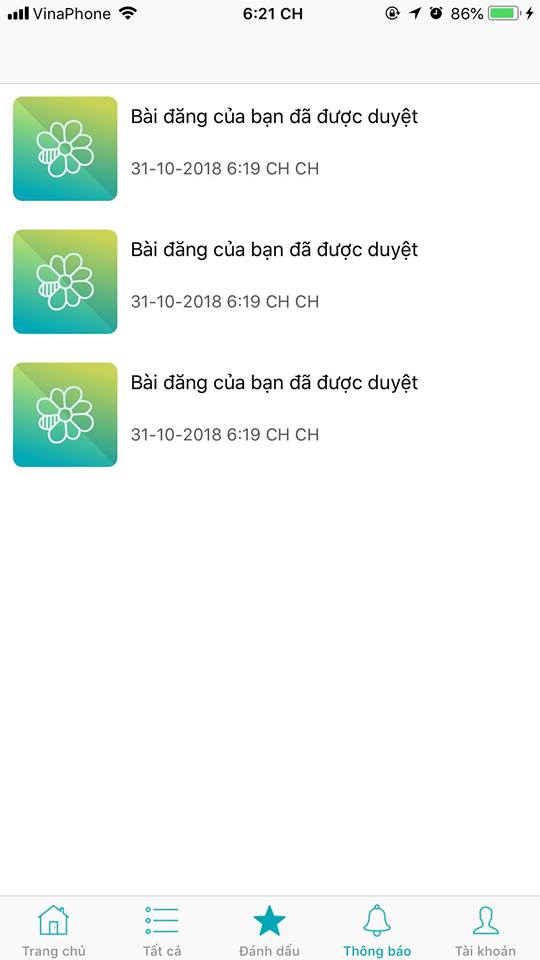


Figure 65: <Authorized User> View Notification Interface

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **FIELDS** | | | | | | | |
| **No** | **Field Name** | **Description** | **Read Only** | **Mandatory** | **Control Type** | **Data Type** | **Length** |
| **1** | Notification list | Display content of notification | Yes | Yes | Label | String | N/A |

Table 63: View Notification - Field

#### View Bookmark

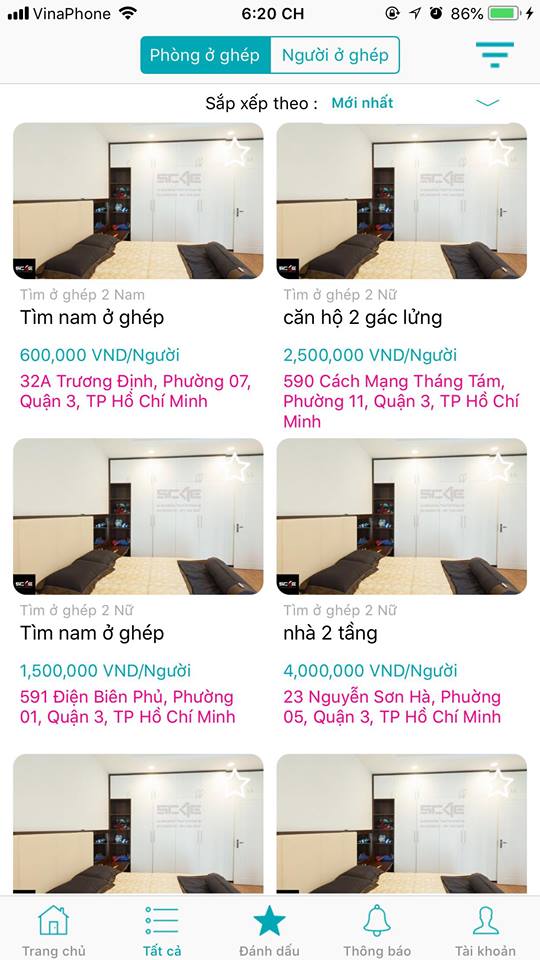


Figure 66: <Authorized User> View Bookmark

Table 64: View Bookmark - Field, Buttons, Hyperlinks

#### Request Location Popup Screen

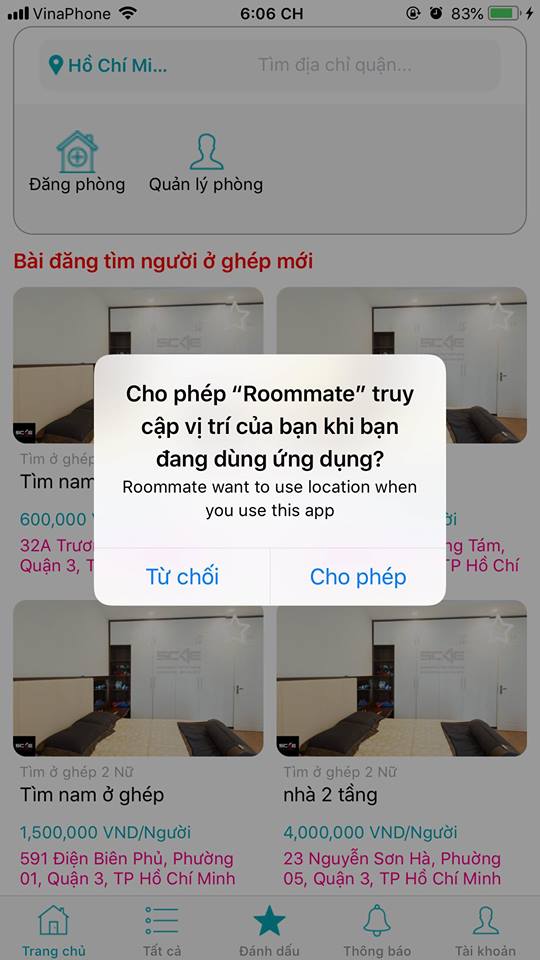


Figure 67: <Authorized User> Request Location Interface

Table 65: Request Location - Buttons

### House owner Interface Design

#### Create Room

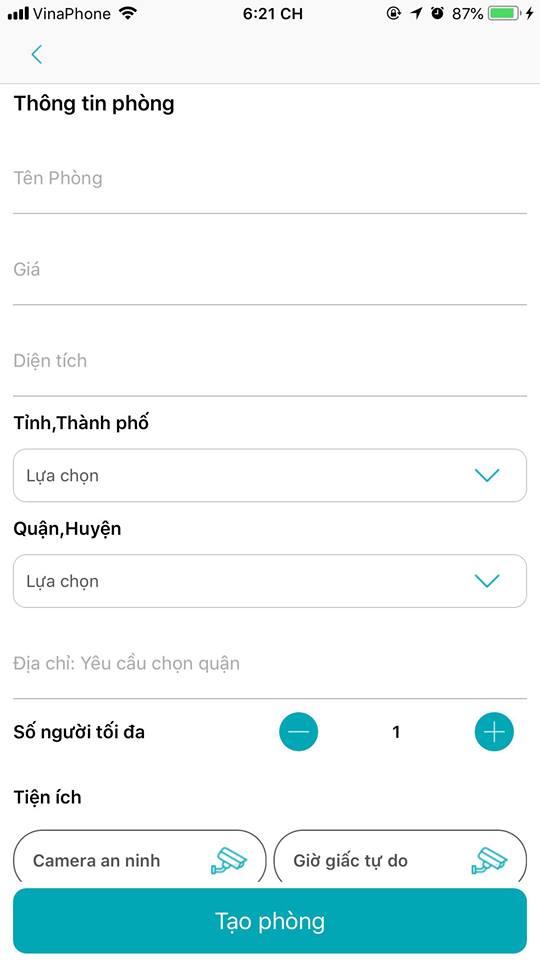


Figure 68: <House owner> Create Room Interface

Table 66: Create Room - Field, Buttons, Hyperlinks

#### Update Room

Figure 69: < House owner > Update Room Interface

Table 67: Update Room - Field, Buttons, Hyperlinks

#### View Room

Figure 70: < House owner > View Room Interface

Table 68: View Room - Field, Buttons, Hyperlinks

#### Delete Room

Figure 71: < House owner > Delete Room Interface

Table 69: Delete Room - Field, Buttons, Hyperlinks

#### Add member into Room

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

Table 70: Add member into Room - Field, Buttons, Hyperlinks

#### Delete member of Room

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

Table 71: Delete member of Room - Field, Buttons, Hyperlinks

### Room Master

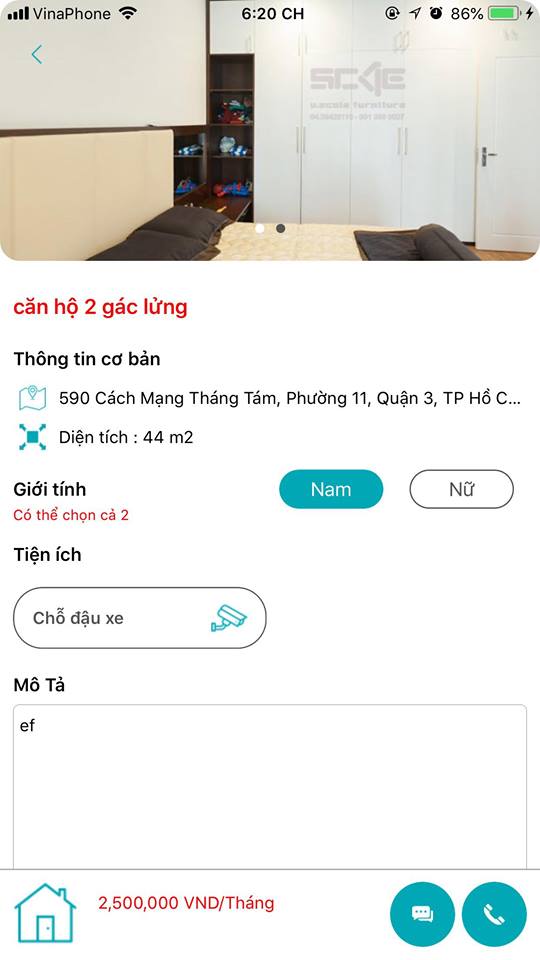
#### View suggested partner list

#### View suggested room master list

#### Create Post

#### Update post

#### View Post



#### Delete Post

### Member

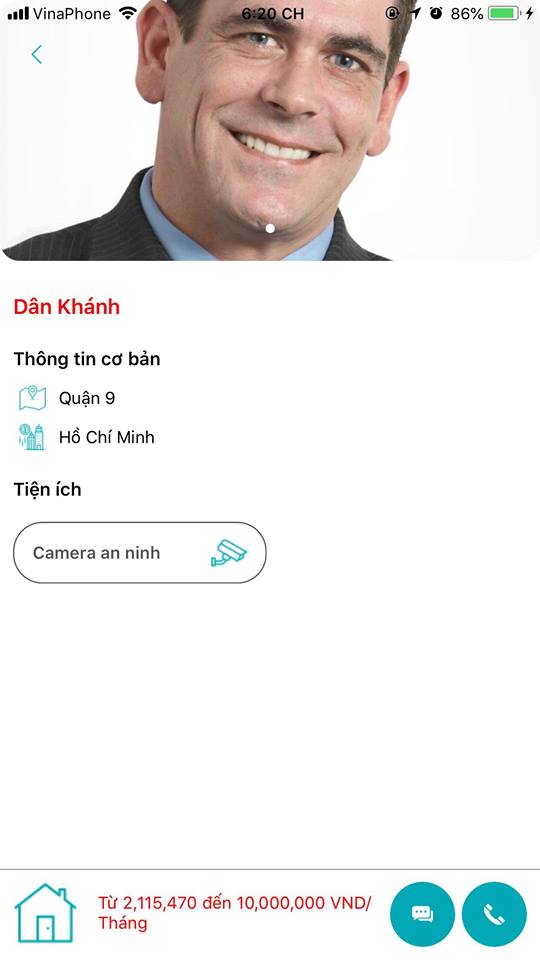
#### View suggested room master list

#### Create Post

#### Update Post

#### Delete Post

#### View Post



## Database Design

### Entity relationship diagram (ERD)

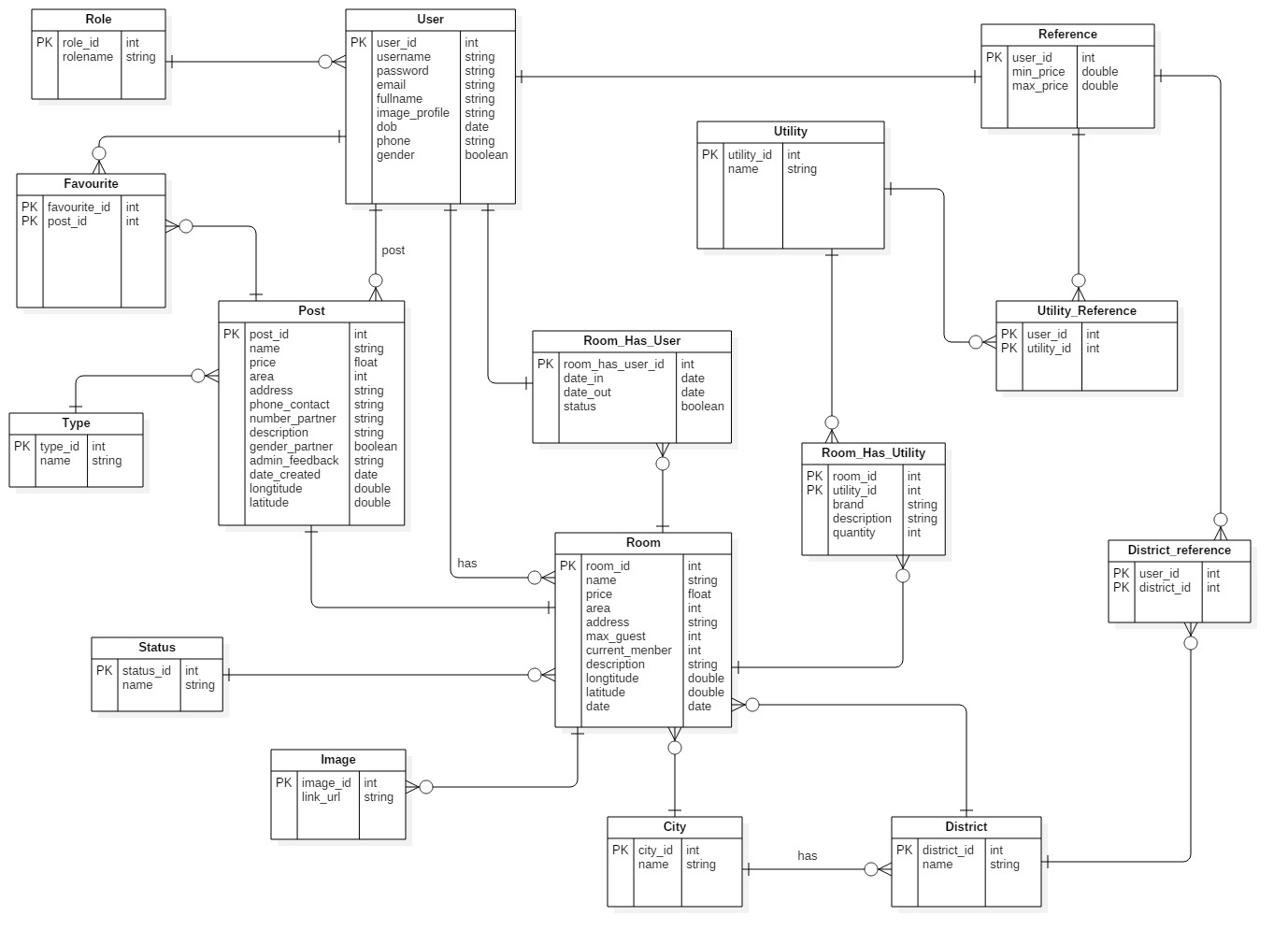


Figure 74: Entity Relationship diagram

### Entity dictionary

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

Table 72: Entity Data dictionary

## Algorithms

### Definition

### Define Problem

### Solution

# E. System Implementation & Test

## Introduction

### Overview

This section describes the approach and methodologies used by group to plan, organize and manage the testing of ASP system. It provides in the detail all necessary information about the implementation and testing procedure of the system included test plans, test cases, test result, test environments, pass/fail criteria and risks estimations as well as a checklist to cover all possible cases.

### Test approach

We use system test approach to estimate our keeping track of children system quality, tester to aware of what ASP should do but does not have the knowledge of how because it does it. If tester had knowledge about the technical inside it, testing process won’t be objectivity.

## Database Relationship Diagram

### Physical Diagram

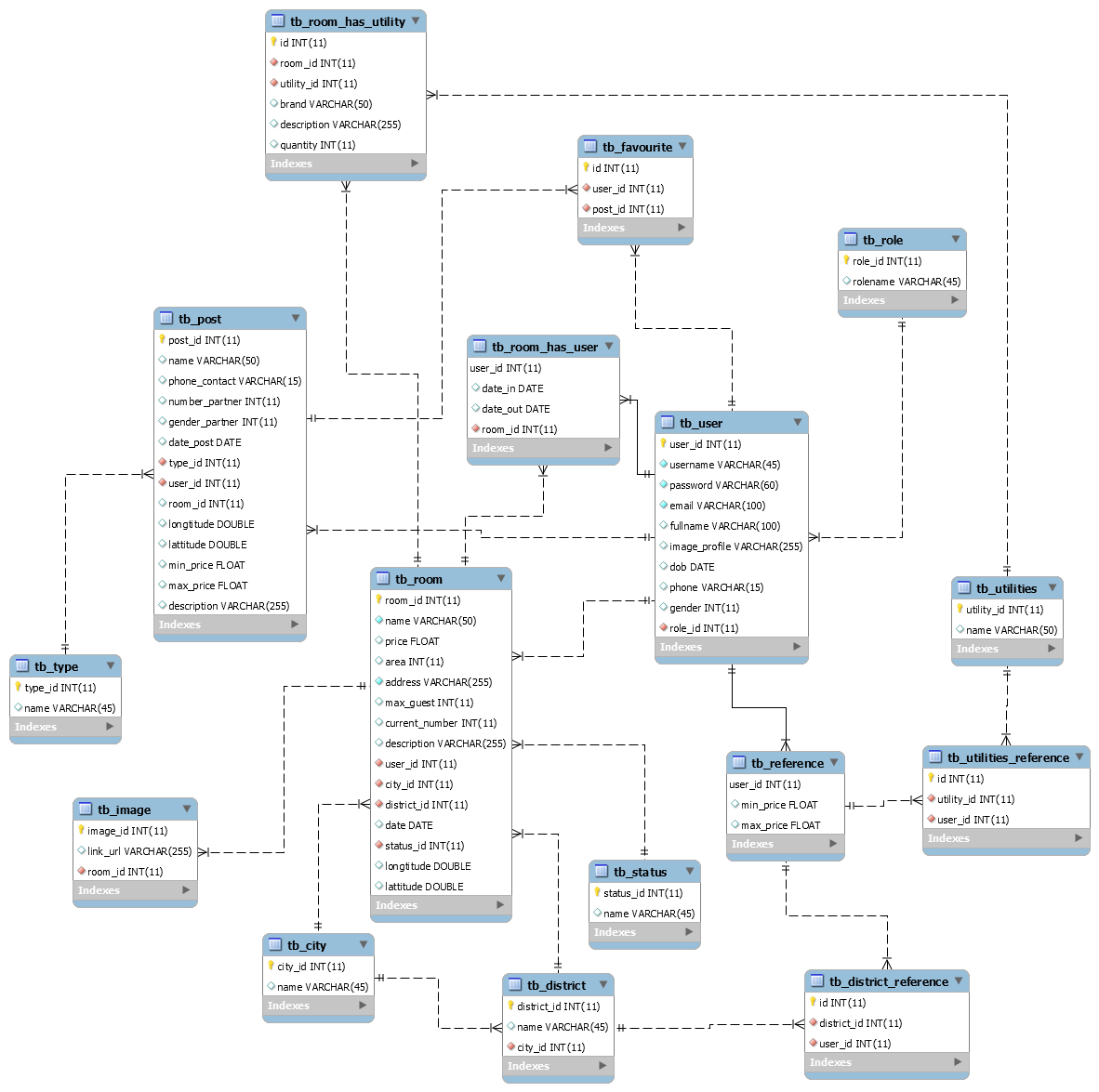


Figure 75: Physical diagram

### Data dictionary

|  |  |
| --- | --- |
| **DATA DICTIONARY: DESCRIBE CONTENT OF ALL TABLES** | |
| **Table name** | **Description** |
| **User** | Contains users information |
| **Room** | Contains rooms information |
| **Post** | Contains posts information |
| **Role** | Describe roles in system. |
| **Favorites** | Contains favorite information |
| **Utilities** | Contains utilities information |
| **Reference** | Contains references information |
| **District** | Contains districts information |
| **City** | Contains city information |
| **Image** | Contains images information |
| **Type** | Describe type of post in system. |
| **Status** | Describe status of room in system. |
| **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 73: Data dictionary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table Name** | **Attributes** | **Description** | **Domain** | **YES** |
| **User** | user\_id | Unique identifier of user | INT(11) | NO |
| username | Username of user | VARCHAR(45) | NO |
| password | Password of user | VARCHAR(60) | NO |
| email | Email of user | VARCHAR(100) | NO |
| fullname | Full name of user | VARCHAR(100) | YES |
| image\_profile | Image profile of user | VARCHAR(255) | YES |
| dob | Date of birth of user | DATE | YES |
| phone | Phone number of user | VARCHAR(15) | YES |
| gender | Gender of user | INT(11) | YES |
| role\_id | Role of user | INT(11) | NO |
| **Room** | room\_id | Unique identifier of room | INT(11) | NO |
| name | Name of room | VARCHAR(50) | NO |
| price | Price of room | FLOAT | NO |
| area | Area of room | INT(11) | YES |
| address | Address of room | VARCHAR(255) | NO |
| max\_guest | Maximum number of guest in room | INT(11) | YES |
| current\_number | Current number of guest in room | INT(11) | YES |
| description | Description of room | VARCHAR(255) | YES |
| user\_id | Identifier of user in room | INT(11) | NO |
| city\_id | City’s identify of room | INT(11) | NO |
| district\_id | District’s identify of room | INT(11) | NO |
| date | Created date of room | DATE | YES |
| status\_id | Status of room that be approved or declined | INT(11) | NO |
| longtitude | Longitude of room | DOUBLE | YES |
| lattitude | Latitude of room | DOUBLE | YES |
| **Post** | post\_id | Unique identifier of post | INT(11) | NO |
| name | Name of post | VARCHAR(50) | YES |
| phone\_contact | Phone number of room master | VARCHAR(15) | YES |
| number\_partner | Number of partner wanted in post | INT(11) | YES |
| gener\_partner | Gender of partner wanted in post | INT(11) | YES |
| date\_post | Created date of post | DATE | YES |
| type\_id | Type of post, define finding room post or finding roommate post. | INT(11) | NO |
| user\_id | Identifier of user in room | INT(11) | NO |
| room\_id | Room’s identifier of post | INT(11) | YES |
| longtitude | Longitude of post | DOUBLE | YES |
| lattitude | Latitude of post | DOUBLE | YES |
| min\_price | Min price of post | FLOAT | YES |
| maxprice | Max price of post | FLOAT | YES |
| description | Description of post | VARCHAR(255) | YES |
| **Role** | role\_id | Unique identifier of role | INT(11) | NO |
| rolename | Name of role | VARCHAR(45) | YES |
| **Favourite** | id | Unique identifier of favourite | INT(11) | NO |
| user\_id | User’s identifier of favourite | INT(11) | NO |
| post\_id | Post’s identifier of favourite | INT(11) | NO |
| **Utilities** | utilitiy\_id | Unique identifier of utility | INT(11) | NO |
| name | Name of utility | VARCHAR(50) | YES |
| **Reference** | user\_id | User’s identifier of reference | INT(11) | NO |
| min\_price | Min price is chosen by user | FLOAT | YES |
| max\_price | Max price is chosen by user | FLOAT | YES |
| **District** | district\_id | Unique identifier of district | INT(11) | NO |
| name | Name of district | VARCHAR(45) | YES |
| city\_id | City’s identifier relate to district | INT(11) | NO |
| **City** | city\_id | Unique identifier of city | INT(11) | NO |
| name | Name of city | VARCHAR(45) | YES |
| **Image** | image\_id | Unique identifier of image | INT(11) | NO |
| link\_url | Url of image | VARCHAR(255) | YES |
| room\_id | Room’s identifier of image | INT(11) | NO |
| **Type** | type\_id | Unique identifier of type. | INT(11) | NO |
| name | Name of type | VARCHAR(45) | YES |
| **Status** | status\_id | Unique identifier of status. | INT(11) | NO |
| name | Name of status | VARCHAR(45) | YES |
| **Room\_Has\_Utilities** | id | Unique identifier of room’s utilities | INT(11) | NO |
| room\_id | Room’s identifier in room has utilities field | INT(11) | NO |
| utilitiy\_id | Utility’s identifier relate to room | INT(11) | NO |
| brand | Brand of utilities are selected | VARCHAR(50) | YES |
| description | Description of utilities are selected | VARCHAR(255) | YES |
| quantity | Describe how many utilities | INT(11) | YES |
| **Utilities\_Reference** | id | Unique identifier of reference | INT(11) | NO |
| user\_id | User’s identifier of reference | INT(11) | NO |
| utilitiy\_id | Utility’s identifier of reference | INT(11) | NO |
| **District\_Reference** | id | Unique identifier of district’s reference | INT(11) | NO |
| user\_id | User’s identifier of reference | INT(11) | NO |
| district\_id | District’s identifier of reference | INT(11) | NO |
| **Room\_Has\_User** | room\_id | Room’s identifier has user | INT(11) | NO |
| user\_id | User’s identifier in room | INT(11) | NO |
| date\_in | Date user move in | DATE | NO |
| date\_out | Date user move out | DATE | NO |

Table 74: Data Table Description

## Performance Measure

### Show list result of finding room post performance

In general, when user open application, show list take less than 3 second to perform.

### Show list result of finding roommate post performance

In general, when user open application, show list take less than 1 second to perform.

### Show list result of all post performance

In general, when user open application, show list take less than 1 second to perform.

### Show list result of suggested post performance

In general, when user open application, show list take less than 1 second to perform.

### Add new room performance

New room will be shown at room list after user added new room completely take less than 1 second.

### Update room performance

### Add new post performance

New post will be shown at post list after user added new post completely take less than 1 second.

### Update post performance

## Test Plan

The overall purpose of testing is to ensure ASP meets its entire technical, functional and business requirement. The purpose of this document is to describe the overall test plan and strategy for testing the ASP. The approach described in this document provides the framework for all testing related to this application. Individual test cases will be written for each version of the application that is released. This document will also be updated as required for each release

### Features to be tested

* + - Unauthorized User: Login, Register.
    - Authorized User: Update user info, view notification, book mark post.
    - House Owner: Create room, update room, add member into room, remove member of room.
    - Room Master: Create finding roommate post, update finding roommate post, view suggested nearby room master’s posts.
    - Member: Create finding room post, update finding room post, view suggested list.
    - Admin: Manage user, manage room.

### Features not to be tested

## System Testing Test Case

### Test case

# F. Software User’s Manual

## Installation Guide

### 1.1 Setting up environment at server side

#### Hardware Requirement

#### Software Requirement

### Deployment at server side

#### Check Environment

#### Import Database

#### Build Project

#### Deploy Project to IIS

## 2. User Manual

### 2.1 Unauthorized User

#### 2.1.1 Login

### 2.2 Authorized User

### 2.3 Admin

### 2.4 House Owner

### 2.5 Room Master

### 2.3 Member

# G. Appendix