PROJECT

Room4U

Prepare by: **SC Team**

Class: Fat2 F2-13-1402-t0

Date: 04 October 2015

Version: 1.0

|  |  |  |  |
| --- | --- | --- | --- |
| **Faculty:** |  | **Mr. Hoang Duc Quang** |  |
| **Batch No:** |  | **F2131402T0** |  |
| **Group No.:** |  | **1** |  |
| **Students:** |  | **Fullname** | **Roll No.** |
|  | 1. | Ho Chau Can | Student797559 |
|  | 2. | Trinh Quang Khai | Student797694 |
|  | 3. | Nguyen Viet Hung | Student797581 |
|  | 4. | Ho Dac Huy | Student797574 |
|  | 5. | Tran Minh Cuong | Student734425 |

TASK SHEET

* + - 1. Document Control

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Project Ref.No: 1** | | **Project Title:**  **Room4U** | **Date of Preparation of Activity Plan** | | | |
| **Sr. No.** | **Task** | | **Actual Start Date** | **Actual Days** | **Team Member Names** | **Status** |
| 01 | **INTRODUCTION** | | 1 Oct 2015 | 18 | Ho Chau Can | Completed |
| 02 | **EXISTING SCENARIO** | | Ho Chau Can | Completed |
| 03 | **PROPOSED SOLUTION** | | Ho Chau Can | Completed |
| 04 | **FUNCTION OF APPLICATION** | | Ho Chau Can | Completed |
| 05 | **ROLES OF APPLICATION** | | Ho Chau Can | Completed |
| 06 | **FUNCTION REQUIREMENT SPECIFICATION** | | Ho Dac Huy | Completed |
| 07 | **APPENDIX** | | Ho Dac Huy | Completed |
| 08 | **USE CASE AND SEQUENCE DIAGRAM** | | Nguyen Viet Hung  Ho Dac Huy | Completed |
| 09 | **ENTITY RELATIONSHIP DIAGRAM** | | Nguyen Viet Hung | Completed |
| 10 | **ENTITY PROPERTIES** | | Nguyen Viet Hung | Completed |
| 11 | **DATABASE RELATIONSHIP DIAGRAM** | | Trinh Quang Khai | Completed |
| 12 | **DATABASE STRUCTURE** | | Trinh Quang Khai | Completed |
| 13 | **SITE MAP** | | Trinh Quang Khai  Tran Minh Cuong | Completed |
| 14 | **GUI DESIGN** | | Tran Minh Cuong | Completed |

* + - 1. Function Control

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Project Ref.No: 1** | | **Project Title:**  **Room4U** | **Date of Preparation of Activity Plan** | | | |
| **Sr. No.** | **Task** | | **Actual Start Date** | **Actual Days** | **Team Member Names** | **Status** |
| 01 | * Searching room with criteria: Customer can search the accommodation on demand which base on their current location or specified place on the Map with radius. * Booking calendar: Customer can identify the room they like is available or not in specific time, so they can book the room they like in advance. * Managing Room Booking List * Report of Room Booking | | 1 Oct 2015 | 18 | Ho Chau Can | Completed |
| 02 | * Account registration and management: Landlord and customer can register account to the application with the respective authorization * Update profile and change password in Profile Page * Delete user and reset password * Report of Customer | | Trinh Quang Khai | Completed |
| 03 | * Posting room: post room with information in detail to the web application by Landlord. * Update and Delete posted room. * Managing Room Detail which having room location info info in detail * Manging Landlord profile. * Report of posted room. | | Nguyen Viet Hung | Completed |
| 04 | • Ratings and comments: allowing customer to optionally submit a rating along with their comment. Any comment will be sent to room owner (Landlord) who will answer that comment accordingly.   * Report of Rating and Comments | | Ho Dac Huy | Completed |
| 05 | • Notification: Customer will be notified by Email about available room posted by Landlord, which adapt to Customer’s requirement as they registered room before.   * FAQ: Important information about a business is shared to clarify questions and uncertainties on the part of customer and landlord. * Report of Notification | | Tran Minh Cuong | Completed |

Table of Contents

Page

REVIEW 1

1. **INTRODUCTION 8**
2. **EXISTING SCENARIO 8**
3. **PROPOSED SOLUTION 8**
4. **FUNCTION OF APPLICATION 8**
5. **ROLES OF APPLICATION 10**

1. Customer 10

2. Landlord 10

3. Admin 10

1. **FUNCTION REQUIREMENT SPECIFICATION 11**
2. **APPENDIX 14**

1. Hardware requirement 14

2. Software requirement 14

REVIEW 2

1. **USE CASE AND SEQUENCE DIAGRAM 17**

REVIEW 3

1. **ENTITY RELATIONSHIP DIAGRAM 75**
2. **ENTITY PROPERTIES 76**
3. **DATABASE RELATIONSHIP DIAGRAM 83**
4. **DATABASE STRUCTURE 84**
5. **SITE MAP 87**
6. **GUI DESIGN 89**

Review 1

Prepare by: **SC Team**

Class: Fat2 F2-13-1402-t0

# INTRODUCTION

‘Room4U’ is a trusted community marketplace for people who is listing, finding and booking accommodation in anywhere. They can looking for the places to stay around their location rely on Google Map immediately or able to register room beforehand. The application will send notification to customer as the room is available.

# EXISTING SCENARIO

In the modern life, people is moving quickly from place to place for business trip, conference, study and so on, so they need to find accommodation in their destination as soon as possible, they are now taking much time to finding out the most favourite location such as near market place, school or office via internet, newspaper or the Map.

# PROPOSED SOLUTION

We propose the Application which store thousands of diversity accommodation from various place in order to cater for customer who easily searching accommodation on demand base on common Google Map. Those accommodation will be posted by landlord who registered to the Application.

This is web page application which is designed with responsive UI, so customer can access the App via desktop or smartphone in friendly design. The App utilize Google Map API to detect the current location of customer and offering them how many room are available around them as well as how far they can reach to that room.

# FUNCTION OF APPLICATION

* Account registration and management: Landlord and customer can register account to the application with the respective authorization. They can update profile and change password. Customer profile will store the history of accommodation they have been booking. Landlord can manage the list of room they have posted.
* Posting new room with information in detail to the web application by Landlord.
* Searching room with criteria: Customer can search the accommodation on demand which base on their current location or specified place on the Map with radius.
* Booking calendar: Customer can identify the room they like is available or not in specific time, so they can book the room they like in advance.
* Notification: Customer will be notified by Email about available room posted by Landlord, which adapt to Customer’s requirement when they registered before.
* Ratings and comments: allowing customer to optionally submit a rating along with their comment. Any comment will be sent to room owner (Landlord) who will answer that comment accordingly.
* FAQ: Important information about a business is shared to clarify questions and uncertainties on the part of customer and landlord.
* Administrator:
* Managing the Dashboard which having some report about Landlord/Customer and room booking.
* Managing the account of Landlord and Customer like Delete and Reset password.
* Manging Room posted by landlord which having delete right.
* Managing the list of room booking with delete right.
* Managing the list of registered room in advance by Customer with delete right.
* Managing the comments from Customer and Landlord with delete right.
* Managing FAQ with insert/update/delete right.

# ROLES OF APPLICATION

* **CUSTOMER**
* Customer Profile [View & Insert & Update]
* Booking Order Details [View & Insert & Update]
* Comments [View & Insert]
* Ratings [View & Insert]
* **LANDLORD**
* Landlord Profile [View & Insert & Update]
* Accommodation Posts [View & Insert & Update & Delete]
* Booking Order Details [View & Insert & Update]
* Comments [View & Insert & Update]
* Ratings [View & Insert]
* Report [View]
* **ADMIN**
* Management Of Customer, Landlord Profile [View & Delete] & Reset Password
* Management Of Accommodation Posts [View & Delete]
* Management Of Booking Order Details [View & Delete]
* Management Of Comments [View & Delete]
* Management of Rating [View & Delete]
* Management of Notification [View & Delete]
* FAQ Page [View & Insert & Update & Delete]
* Report [View ]

# FUNCTION REQUIREMENT SPECIFICATION

## User can register as Landlord or Customer

### 1.1 Description

The system must provide a page to allow guest create their accounts with the following information:

* User Name
* Password
* Name
* Profile image
* Phone
* Email
* Select to be Landlord or Customer (an account cannot be Landlord and Customer)

### 1.2 Functional Requirements

1. User must provide Username, Password, Name, Profile image, Phone, Email to create an account.
2. User must select Landlord or Customer

## Users can login as Landlord or Customer

### 2.1 Description

The system must provide a page to allow users login as Landlord or Customer. The system must distinguish Landlord, Customer accounts after Login.

### 2.2 Functional Requirements

1. Users must provide Username and Password to login into the website.
2. System must distinguish Users type to display pages which customers can view.
3. Customers can only access to pages they have permission.

## Landlord Management

### 3.1 Description

System must provide for Landlord a management page.

### 3.2 Functional Requirements

1. Landlord can modify their account, managing posted room and report.

## Landlord Functions: Search, book, submit property, room for renting. Comment and rating

### 4.1 Description

Customer and Landlord can search a property, room to rent. Landlord can submit property, room for renting by Customer. Customer and Landlord can comment and rating. System must provide pages to allow Customer and Landlord access these functions.

* Information will be provided when search property, room:
* Landlord choose a location on Map, then select radius.
* Information will be provided when book property, room:
* Book date
* Information Landlord provide when submit property, room:
* Room name.
* Price.
* Address.
* Description.
* Number of bed.
* Number of people.
* Number of toilets.
* Pictures.
* Information will be provided when comment, rating property, room:
* Comment content
* Score rating.

**4.2 Function Requirements**

1. Landlord can search to rent property, room.
2. Landlord can book property, room.
3. Landlord can submit property, room for renting
4. Landlord can comment and rating.

## Customer Management

### Description

System must provide for Customer a management page.

### Functional Requirements

1. Customer can modify their account.

## Customer Functions: search, book property, room. Comment and rating

### Description

Customer can search a property, room to rent. Customer can submit property, room for renting. Customer can comment and rating posts. System must provide pages to allow Landlord access these functions.

* Information Customer provide when search property, room:
  + Customer choose a location on Map, select radius.
* Information Customer provide when book property, room:
  + Book date
* Information Customer provide when comment, rating property, room:
  + Comment content
  + Score rating.

### Functional Requirements

1. Customer can search to rent property, room.
2. Customer can book property, room.
3. Customer can comment and rating property, room.

## FAQS

### 7.1 Description

System must provide to Customers a page which displays frequently questions and answers. Administrators can add, modify or delete these.

### Functional Requirements

1. Customers can view FAQs page.
2. Administrators can add, modify or delete these.

## Administrators can manage customer and landlord

### Description

System should provide for Administrators a page which allows they manage Customer and Landlord, such delete a Customer or Landlord, view Customer, Landlord details information, view list of Customer, Landlord

### Functional Requirements

1. Administrators can delete Customer, Landlord.
2. Administrators can view Customer details information.
3. Administrators can view list of Customers.

# APPENDIX

* + - 1. **HARDWARE REQUIREMENT**
* A minimum computer system that will help you access all the tools in the courses is a Core i3 or better.
* 4GB Megabytes of RAM or better.
  + - 1. **SOFTWARE REQUIREMENT**
* Microsoft windows 7 or above.
* Browser: Chrome, Firefox v.20, IE 11 or above.
* Java SDK 1.7 or above
* Glassfish Server v.4.0
* Microsoft SQL Server 2012
* Framework Java Server faces 2.0
* Primefaces 5.2

*--- The End of Review 1---*

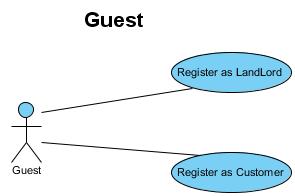
|  |  |  |
| --- | --- | --- |
|  | **Prepared by (Student)** | **Approved by (Faculty)** |
| **Signature** | Team leader  **Ho Chau Can** | **Hoang Duc Quang** |
| **Date** |  |  |

Review 2

Prepare by: **SC Team**

Class: Fat2 F2-13-1402-t0

1. USE CASE AND SEQUENCE DIAGRAM

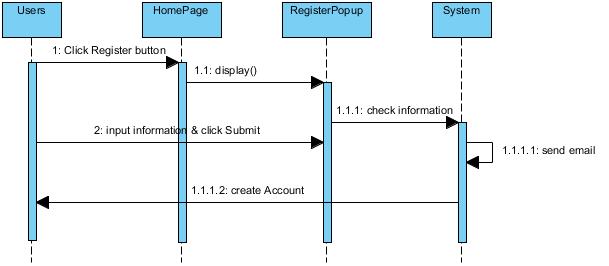
1.1 Users

### Register as Landlord

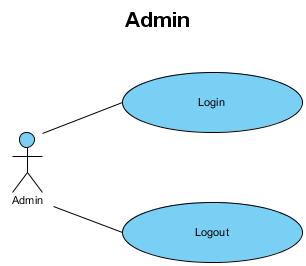
|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Register as Landlord | |
| **Actors** | Guest | |
| **Description** | Guest wants to register with the website to become Landlord or Customer. | |
| **Requirements** | Guest provides Guest’s information | |
| **Pre-conditions** |  | |
| **Post-conditions** | Success: Guest creates Customer or Landlord account in this website  Fail: Refill information | |
| **Basic flow** | Actor Actions:  1. Actor clicks ‘Register’ link in Home page  4. Actor inputs register’s information, select ‘Landlord’ radio button and click ‘Register’ button. [Alternative 1] | System Responses:  2. System redirects to Register page and displays Registration form with the following control :  - ‘User Name’ text field  - ‘Password’ text field  - ‘Confirm Password’ text field  - ‘Name’ text field  - ‘Phone’ text field  - ‘Email’ text field  - ‘Customer’ radio button  - ‘Landlord’ radio button  - ‘Register’ button  - ‘Close’ button  3. System checks the information  [Exception 1]  5. System inserts the account into database, show message: ‘Your account has been registered successfully !’ and then redirects to Home page. |
| **Alternative flow** | Actor Actions:  [Alternative 1]  1. Actor click ‘Close’ button. | System Responses:  2. System returns to Home page. |
| **Exception** | Actor Actions:  [Exception 1]  1. Actor lefts ‘UserName’ text field blank.  2. UserName has already exists.  3. Actor lefts ‘Password’ text field blank.  4. Actor lefts ‘Confirm Password’ text field blank.  5. Password and confirm password don’t match.  6. Actor lefts ‘Name’ text field blank.  7. Actor lefts ‘Address’ text field blank.  8. Actor lefts ‘Phone’ text field blank.  9. Input phone not in numeric or phone length not in range 8 – 14 character.  10. Actor lefts ‘Email’ text field blank or input an invalid email address. | System Responses:  System displays message to Actor :  - ‘UserName cannot be blank’.  - ‘This Username has already exist!’.  - ‘Password cannot be blank’’.  - ‘Confirm Password can’t be blank’.  - ‘Confirm password not match password.’  - ‘Name cannot be blank’.  - ‘Address cannot be blank’.  - ‘Phone is invalid’.  - ‘The phone number should be numeric only and must be between 8 and 14 digits’.  - ‘Email is invalid’. |

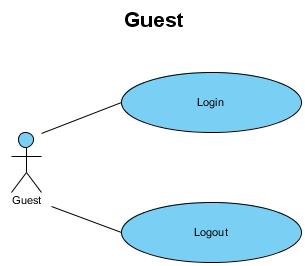
### Register as Customer

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Register as Customer | |
| **Actors** | Guest | |
| **Description** | Guest wants to register with the website to become Landlord or Customer. | |
| **Requirements** | Guest provides Guest’s information | |
| **Pre-conditions** |  | |
| **Post-conditions** | Success: Guest create Customer or Landlord account in this website  Fail: Refill information | |
| **Basic flow** | Actor Actions:  1. Actor clicks ‘Register’ link in Home page  4. Actor inputs register’s information, select ‘Customer’ radio button and click ‘Register’ button. [Alternative 1] | System Responses:  2. System redirects to Register page and displays Registration form with the following control :  - ‘User Name’ text field  - ‘Password’ text field  - ‘Confirm Password’ text field  - ‘Name’ text field  - ‘Phone’ text field  - ‘Email’ text field  - ‘Customer’ radio button  - ‘Landlord’ radio button  - ‘Register’ button  - ‘Close’ button  3. System checks the information  [Exception 1]  5. System inserts the account into database, show message: ‘Your account has been registered successfully!’ and then redirects to Home page. |
| **Alternative flow** | Actor Actions:  [Alternative 1]  1. Actor click ‘Back to Home’ button. | System Responses:  2. System returns to Home page. |
| **Exception** | Actor Actions:  [Exception 1]  1. Actor lefts ‘UserName’ text field blank.  2. UserName has already exists.  3. Actor lefts ‘Password’ text field blank.  4. Actor lefts ‘Confirm Password’ text field blank.  5. Password and confirm password don’t match.  6. Actor lefts ‘Name’ text field blank.  7. Actor lefts ‘Address’ text field blank.  8. Actor lefts ‘Phone’ text field blank.  9. Input phone not in numeric or phone length not in range 8 – 14 character.  10. Actor lefts ‘Email’ text field blank or input an invalid email address. | System Responses:  System displays message to Actor :  - ‘UserName cannot be blank’.  - ‘This Username has already exist!’.  - ‘Password cannot be blank’’.  - ‘Confirm Password can’t be blank’.  - ‘Confirm password not match password.’  - ‘Name cannot be blank’.  - ‘Address cannot be blank’.  - ‘Phone is invalid’.  - ‘The phone number should be numeric only and must be between 8 and 14 digits’.  - ‘Email is invalid’. |



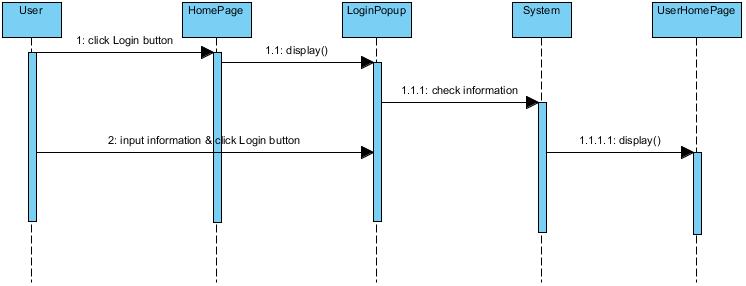
1.2 Admin, Guest





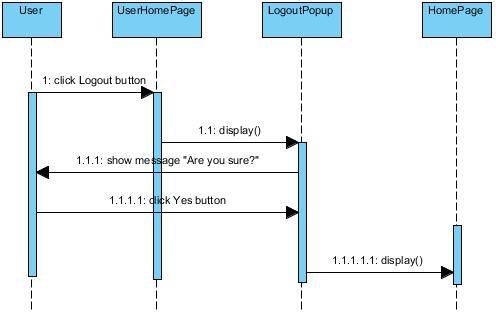
### Login

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Login | |
| **Actors** | Admin or Customer | |
| **Description** | Admin or Customer wants to login to website as a Admin or Customer role. | |
| **Requirements** | Actor provides username and password | |
| **Pre-conditions** |  | |
| **Post-conditions** | Success: User logged in to website.  Fail: Refill information | |
| **Basic flow** | Actor Actions:  1. Actor go to Home page of website.  3. Actor inputs UserName and Password, then click the Login button. | System Responses:  2. System redirects to Home page and displays Login form on the left with the following control:  - ‘UserName’ text field  - ‘Password’ text field  - ‘Login’ button  4. System checks the information  [Exception 1]  5. System redirects to Actor’s Home page. |
| **Alternative flow** | Actor Actions:  [Alternative 2]  1. Actor click ‘OK’ button, inputs UserName and Password, then click the Login button.  [Alternative 1]  1. Actor click ‘Close’ button. | System Responses:  2. System checks the information  [Exception 1]  3. System redirects to Actor’s Home page.  2. System returns to Home page. |
| **Exception** | Actor Actions:  [Exception 1]  1. Actor input invalid username and password. | System Responses:  - System redirect to Login page with the following controls:  - ‘UserName’ text field  - ‘Password’ text field  - ‘Login’ button.  - ‘Close’ button.  System show a message box: “Invalid Username and password”.[Alternative 2] |



### Logout

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Logout | |
| **Actors** | Admin or Customer or Landlord | |
| **Description** | Admin or Customer or Landlord want to log out of the System. | |
| **Requirements** |  | |
| **Pre-conditions** | Actor has logged in. | |
| **Post-conditions** | Success: Actor Logged out. | |
| **Basic flow** | Actor Actions:  1. Actor clicks ‘Log out’ button on the Actor’s Home menu. | System Responses:  2. System redirect to Logout page, remove session, and then redirect to Home page. |
| **Alternative flow** |  |  |
| **Exception** |  |  |

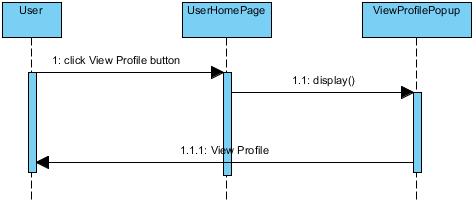


### Manage Account

### Use Case Diagram03

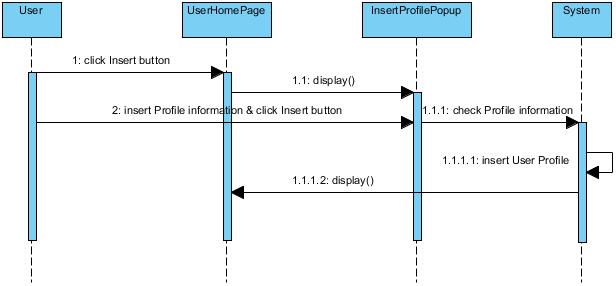
### View Profile

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | View Profile | |
| **Actors** | Landlord or Customer | |
| **Description** | Landlord or Customer want to View profile’s information. | |
| **Pre-conditions** | Actor has logged in. | |
| **Post-conditions** | Success: Profile’s information of this user has been updated.  Fail: Refill information | |
| **Basic flow** | Actor Actions:  1. Actor clicks ‘Profile’ button in Actor’s Home page menu.  3. Actor view information and click ‘Close’ button. [Alternative 1] | System Responses:  2. System redirects to Profile page and displays Profile details with the following control :  - ‘User Name’  - ‘Name  - ‘Phone’  - ‘Email’  - ‘Profile picture’  - ‘Close’ button  4. System checks the information  [Exception 1]  5. System redirects to Actor Home page. |
| **Alternative flow** | Actor Actions:  [Alternative 1]  1. Actor click ‘Close’ button. | System Responses:  2. System returns to Home page. |
| **Exception** |  |  |



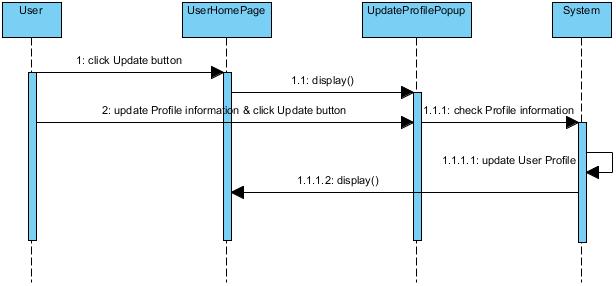
### Insert Profile

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Insert Profile | |
| **Actors** | Landlord or Customer | |
| **Description** | Landlord or Customer want to insert profile’s information. | |
| **Pre-conditions** | Actor has logged in. | |
| **Post-conditions** | Success: Profile’s information of this user has been updated.  Fail: Refill information | |
| **Basic flow** | Actor Actions:  1. Actor clicks ‘Profile’ button in Actor’s Home page menu.  3. Actor view information and click ‘Close’ button. [Alternative 1] | System Responses:  2. System redirects to Profile page and displays Profile details with the following control :  - ‘Name’ text field  - ‘Phone’ text field  - ‘Email’ text field  - ‘Insert’ button  - ‘Close’ button  4. System checks the information  [Exception 1]  5. System redirects to Actor Home page. |
| **Alternative flow** | Actor Actions:  [Alternative 1]  1. Actor click ‘Close’ button. | System Responses:  2. System returns to Home page. |
| **Exception** |  |  |

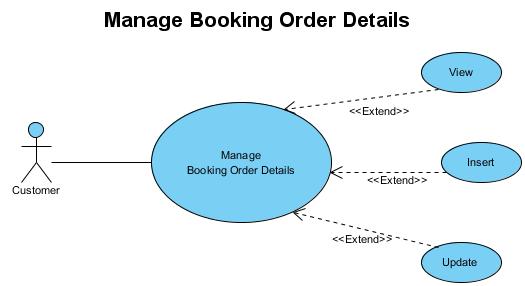


### Update Profile

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Update Profile | |
| **Actors** | Landlord or Customer | |
| **Description** | Landlord or Customer wants to update profile’s information. | |
| **Requirements** | Actor provides profile information | |
| **Pre-conditions** | Actor has logged in. | |
| **Post-conditions** | Success: Profile’s information of this user has been updated.  Fail: Refill information | |
| **Basic flow** | Actor Actions:  1. Actor clicks ‘Profile’ button in Actor’s Home page menu.  3. Actor updates information and click ‘Update’ button. [Alternative 1] | System Responses:  2. System redirects to Update Profile page and displays Update Profile form with the following control :  - ‘UserName’ text field (read only)  - ‘Old Password’ text field  - ‘New Password’ text field  - ‘New Password Confirm’ text field  - ‘Name’ text field  - ‘Phone’ text field  - ‘Email’ text field  - ‘Update’ button  - ‘Close’ button  4. System checks the information  [Exception 1]  5. System update the account into database and then redirects to Actor Home page. |
| **Alternative flow** | Actor Actions:  [Alternative 1]  1. Actor click ‘Back to Home’ button. | System Responses:  2. System returns to Home page. |
| **Exception** | Actor Actions:  [Exception 1]  1. Actor lefts ‘New Password’ text field blank.  2. Actor lefts ‘New Confirm Password’ text field blank.  3. New Password and New Password Confirm don’t match.  4. Actor lefts ‘Name’ text field blank.  6. Actor lefts ‘Phone’ text field blank.  6. Input phone not in numeric or phone length not in range 8 – 14 character.  7. Actor lefts ‘Email’ text field blank or input an invalid email address. | System Responses:  System displays message to Actor :  - ‘Password cannot be blank’’.  - ‘Confirm Password can’t be blank’.  - ‘Confirm password not match Password.’  - ‘Name cannot be blank’.  - ‘Phone is invalid’.  - ‘The phone number should be numeric only and must be between 8 and 14 digits’.  - ‘Email is invalid’. |

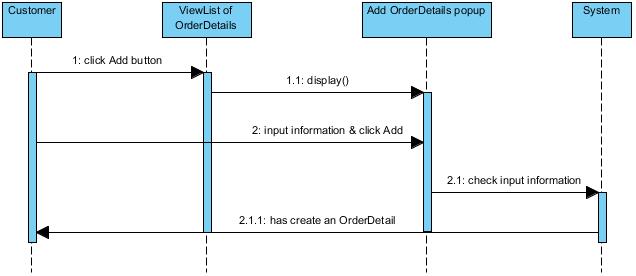


### 1.4 Customer Manage Booking Order Details



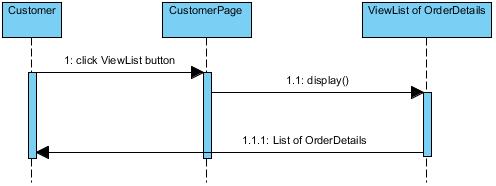
### Insert Booking Order Details

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Insert Booking Order Details | |
| **Actors** | Customer | |
| **Description** | Customer wants to insert booking order details. | |
| **Requirements** | Actor provides profile information | |
| **Pre-conditions** | Actor has logged in. | |
| **Post-conditions** | Success: Insert booking order details of this user has been inserted  Fail: Refill information | |
| **Basic flow** | Actor Actions:  1. Actor clicks ‘Booking Order Details’ button in Actor’s Home page menu.  3. Actor insert information and click ‘Add’ button. [Alternative 1] | System Responses:  2. System redirects to Booking Order Details page and displays form with the following control :  - ‘Order ID’ (Read Only)  - ‘Customer ID’ (Read Only)  ­- ‘Customer Name’ text field  - ‘Phone’ text field  - ‘Email’ text field  - ‘Total Price’(Read Only)  - ‘Status’ (Read Only)  - ‘Order Date” Datepicker  - ‘Booking” button  - ‘Close’ button  4. System checks the information  [Exception 1]  5. System insert booking order details into database and then redirects to Actor Home page. |
| **Alternative flow** | Actor Actions:  [Alternative 1]  1. Actor click ‘Close’ button. | System Responses:  2. System returns to Home page. |
| **Exception** | Actor Actions:  [Exception 1]  1. Actor lefts ‘Name’ text field blank.  2. Actor lefts ‘Phone’ text field blank.  3. Input phone not in numeric or phone length not in range 8 – 14 character.  4. Actor lefts ‘Email’ text field blank or input an invalid email address.  5. Actor lefts ‘Order Date’ blank or pick an invalid date. | System Responses:  System displays message to Actor :  - ‘Name cannot be blank’’.  - ‘Phone can’t be blank’.  - ‘The phone number should be numeric only and must be between 8 and 14 digits’.  - ‘Email is invalid’.  - ‘Order Date is invalid’. |



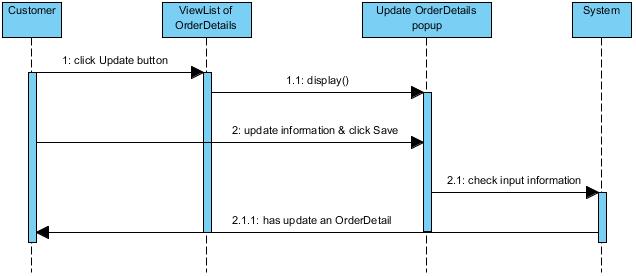
### View Booking Order Details

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | View Booking Order Details | |
| **Actors** | Customer | |
| **Description** | Customer wants to view booking order details. | |
| **Requirements** | Actor provides profile information | |
| **Pre-conditions** | Actor has logged in. | |
| **Post-conditions** | Success: View booking order details of this user has been displayed.  Fail: Refill information | |
| **Basic flow** | Actor Actions:  1. Actor clicks ‘ViewList’ button in Actor’s Home page menu.  3. Actor views Order details and click ‘Close’ button. | System Responses:  2. System redirects to View Booking Order Details page and displays form with the following control :  - ‘Order ID’  - ‘Accommodation ID’  ­- ‘Customer Name’  - ‘Price’  - ‘Booked From Date’  - ‘Booked To Date’  - ‘Close’ button  [Exception 1]  4. System redirects to Actor Home page. |
| **Alternative flow** |  |  |
| **Exception** |  |  |

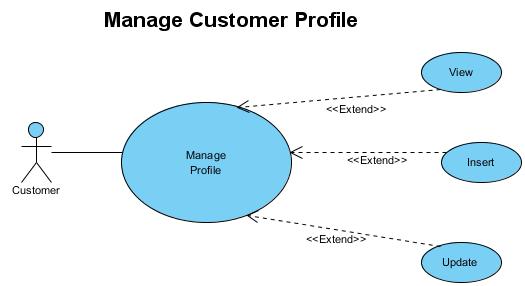


### Update Booking Order Details

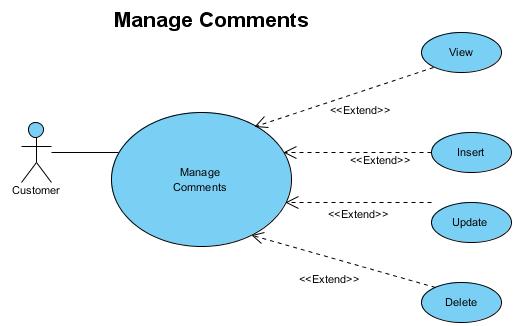
|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Update Booking Order Details | |
| **Actors** | Customer | |
| **Description** | Customer wants to update booking order details. | |
| **Requirements** | Actor provides profile information | |
| **Pre-conditions** | Actor has logged in. | |
| **Post-conditions** | Success: Booking order details of this user has been updated  Fail: Refill information | |
| **Basic flow** | Actor Actions:  1. Actor clicks ‘Update’ button in Actor’s Home page menu.  3. Actor updates information and click ‘Save’ button. [Alternative 1] | System Responses:  2. System redirects to Booking Order Details page and displays form with the following control :  ­- ‘Customer Name’ text field  - ‘Phone’ text field  - ‘Email’ text field  - ‘Total Price’(Read Only)  - ‘Status’ (Read Only)  - ‘Order Date” Datepicker  - ‘Save” button  - ‘Close’ button  4. System checks the information  [Exception 1]  5. System update booking order details and then redirects to Actor Home page. |
| **Alternative flow** | Actor Actions:  [Alternative 1]  1. Actor click ‘Close’ button. | System Responses:  2. System returns to Home page. |
| **Exception** | Actor Actions:  [Exception 1]  1. Actor lefts ‘Name’ text field blank.  2. Actor lefts ‘Phone’ text field blank.  3. Input phone not in numeric or phone length not in range 8 – 14 character.  4. Actor lefts ‘Email’ text field blank or input an invalid email address.  5. Actor lefts ‘Order Date’ blank or pick an invalid date. | System Responses:  System displays message to Actor :  - ‘Name cannot be blank’’.  - ‘Phone can’t be blank’.  - ‘The phone number should be numeric only and must be between 8 and 14 digits’.  - ‘Email is invalid’.  - ‘Order Date is invalid’. |



### 1.5 Customer Manage Profile

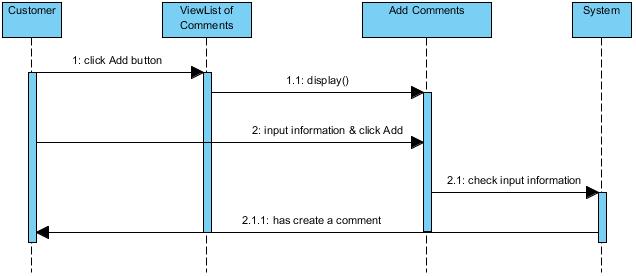


### 1.6 Customer Manage Comments



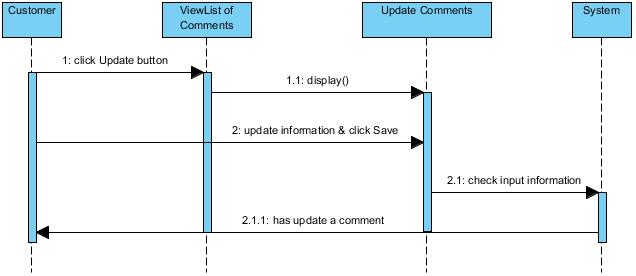
### 1.6.1 Insert Comments

|  |  |  |
| --- | --- | --- |
| **Use case name** | Insert Comments | |
| **Actors** | Customer | |
| **Description** | Add Comments | |
| **Requirements** |  | |
| **Pre-conditions** | Actor has already login into the System as Administrator | |
| **Post-conditions** | Success: Comments added into the database. | |
| **Basic flow** | Actor actions:  1. From ‘Room details’ page, in view comments section, actor clicks on to the ‘Add’ button.  3. Actor:   * Input Comments into ‘Comments’ text field. * Clicks on to ‘Add’ button to finish adding. | System response:  2. System redirects to ‘Add Comments’ page and display form with the following controls:  -‘Comments text field’  -‘Add’ button  -‘Close’ button  4. System checks information. |
| **Alternative flow** | 1. Actor clicks on to ‘Close’ button | 1. System redirects to ‘Room Details’ page |
| **Exception** | Actor lefts ‘Comments’ text field. | Validation for all text field. |



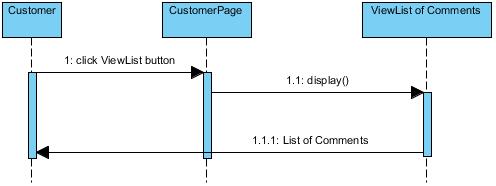
### 1.6.2 Update Comments

|  |  |  |
| --- | --- | --- |
| **Use case name** | Update Comments | |
| **Actors** | Customer | |
| **Description** | Update Comments | |
| **Requirements** |  | |
| **Pre-conditions** | Actor has already login into the System as Administrator | |
| **Post-conditions** | Success: Comments updated into the database. | |
| **Basic flow** | Actor actions:  1. From ‘Room details’ page in view comment section, actor clicks on to the ‘Update’ button.  3. Actor:   * Input Comments into ‘Comments’ text field. * Clicks on to ‘Update’ button to finish adding. | System response:  2. System redirects to ‘Add Comments’ page and display form with the following controls:  -‘Comments text field’  -‘Update’ button  -‘Close’ button  4. System checks information. |
| **Alternative flow** | 1. Actor clicks on to ‘Close’ button | 1. System redirects to ‘Room Details’ page |
| **Exception** |  |  |



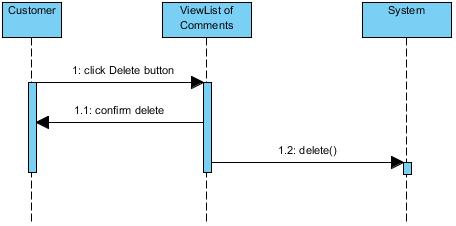
### 1.6.3 View Comments

|  |  |  |
| --- | --- | --- |
| **Use case name** | View Comments | |
| **Actors** | Customer | |
| **Description** | View Comments | |
| **Requirements** |  | |
| **Pre-conditions** | Actor has already login into the System as Administrator | |
| **Post-conditions** | Success: Comments have been displayed | |
| **Basic flow** | Actor actions:  1. From ‘Room details’ page, actor clicks on to the ‘View Comments’ button. | System response:  2. System redirects to ‘Comments’ section and display:  -‘Comments  -‘Update’ button  -‘Delete’ button  -‘Close’ button  3. System checks information. |
| **Alternative flow** | 1. Actor clicks on to ‘Close’ button | 1. System redirects to ‘Room Details’ page |
| **Exception** |  |  |

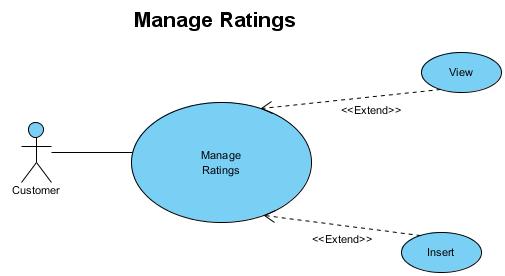


### 1.6.4 Delete Comments

|  |  |  |
| --- | --- | --- |
| **Use case name** | Delete Comments | |
| **Actors** | Customer | |
| **Description** | Delete Comments | |
| **Requirements** |  | |
| **Pre-conditions** | Actor has already login into the System as Administrator | |
| **Post-conditions** | Success: Comments have been deleted | |
| **Basic flow** | Actor actions:  1.From ‘Room details’ page, in view comments section, actor clicks on to the ‘Delete Comments’ button.  3. Actor clicks ‘Yes’ button.  5. Actor clicks ‘No’ button. | System response:  2. A confirms message box appears:”Are you sure that you want to delete this comments?”  4. Comments from database has been deleted.  - System redirects to Comments section |
| **Alternative flow** |  |  |
| **Exception** |  |  |

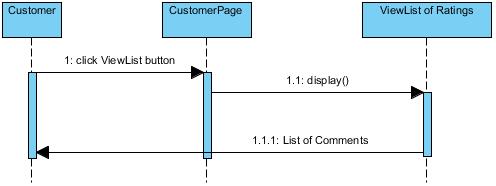


### 1.7 Customer Manage Rating



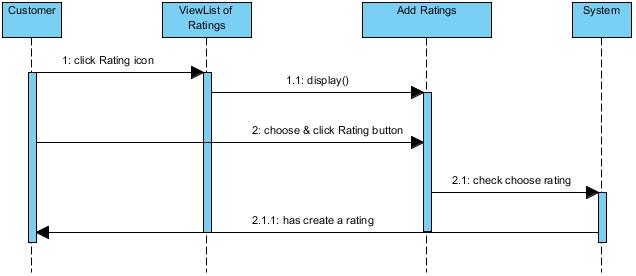
### 1.7.1 View Rating

|  |  |  |
| --- | --- | --- |
| **Use case name** | View Rating | |
| **Actors** | Customer | |
| **Description** | View Rating | |
| **Requirements** |  | |
| **Pre-conditions** | Actor has already login into the System as Administrator | |
| **Post-conditions** | Success: Rating have been displayed | |
| **Basic flow** | Actor actions:  1. From ‘Room details’ page, actor clicks on to the ‘View Rating’ button. | System response:  2. System redirects to ‘Rating’ section and display:  -‘Rating’  -‘Close’ button  3. System checks information. |
| **Alternative flow** | 1. Actor clicks on to ‘Close’ button | 1. System redirects to ‘Room Details’ page |
| **Exception** |  |  |

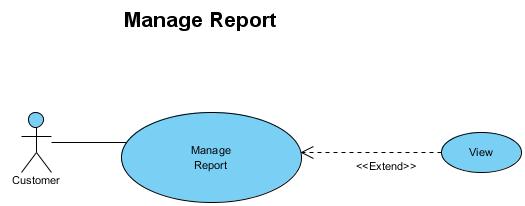


### 1.7.2 Insert Rating

|  |  |  |
| --- | --- | --- |
| **Use case name** | Insert Rating | |
| **Actors** | Customer | |
| **Description** | Add Rating | |
| **Requirements** |  | |
| **Pre-conditions** | Actor has already login into the System as Administrator | |
| **Post-conditions** | Success: Rating score added into the database. | |
| **Basic flow** | Actor actions:  1. From ‘Room details’ page, actor clicks on to the ‘Rating’ icon. | System response:  2. System redirects to ‘Rating’ page and display form with the following controls:  -‘Rating’  -‘Rating’ button  -‘Close’ button  3. System checks information. |
| **Alternative flow** | 1. Actor clicks on to ‘Close’ button | 1. System redirects to ‘Room Details’ page |
| **Exception** |  |  |

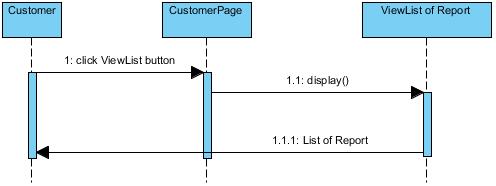


### 1.8 Customer Manage Report

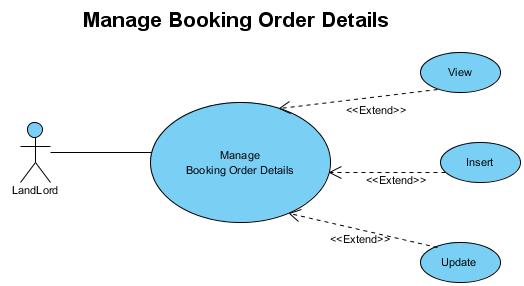


### 1.8.1 View Report

|  |  |  |
| --- | --- | --- |
| **Use case name** | View Report | |
| **Actors** | Customer | |
| **Description** | View Report | |
| **Requirements** |  | |
| **Pre-conditions** | Actor has already login into the System as Administrator | |
| **Post-conditions** | Success: Report have been displayed | |
| **Basic flow** | Actor actions:  1. From Actor home menu page, actor clicks on to the ‘View Report’ button. | System response:  2. System redirects to ‘ViewList of Report’ page and display list of all of Report in table form with control :  ‘Report’ button  ‘Close’ button  ‘Report information’ frame. |
| **Alternative flow** | 1. Actor clicks on to ‘Close’ button | 1. System redirects to ‘Actor Home’ page |
| **Exception** |  |  |

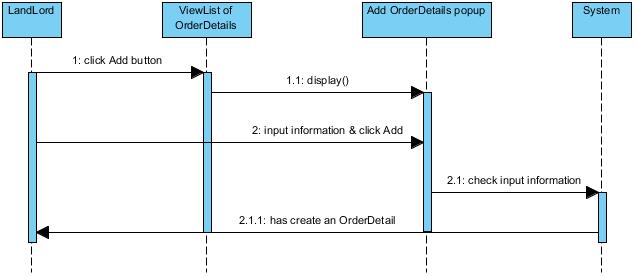


### 1.9 Landlord Manage Booking Order Details



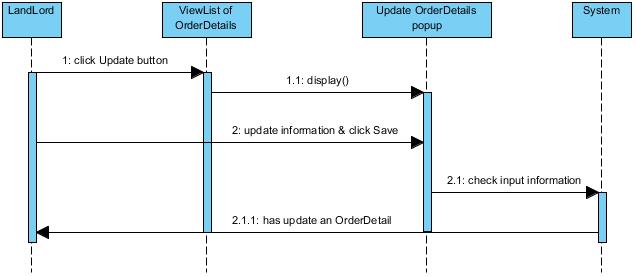
### 1.9.1 Insert Booking Order Details

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Insert Booking Order Details | |
| **Actors** | Landlord | |
| **Description** | Landlord wants to insert booking order details. | |
| **Requirements** | Actor provides profile information | |
| **Pre-conditions** | Actor has logged in. | |
| **Post-conditions** | Success: Insert booking order details of this user has been inserted  Fail: Refill information | |
| **Basic flow** | Actor Actions:  1. Actor clicks ‘Booking Order Details’ button in Actor’s Home page menu.  3. Actor insert information and click ‘Add’ button. [Alternative 1] | System Responses:  2. System redirects to Booking Order Details page and displays form with the following control :  - ‘Order ID’ (Read Only)  - ‘Customer ID’ (Read Only)  ­- ‘Customer Name’ text field  - ‘Phone’ text field  - ‘Email’ text field  - ‘Total Price’(Read Only)  - ‘Status’ (Read Only)  - ‘Order Date” Datepicker  - ‘Booking” button  - ‘Close’ button  4. System checks the information  [Exception 1]  5. System insert booking order details into database and then redirects to Actor Home page. |
| **Alternative flow** | Actor Actions:  [Alternative 1]  1. Actor click ‘Close’ button. | System Responses:  2. System returns to Home page. |
| **Exception** | Actor Actions:  [Exception 1]  1. Actor lefts ‘Name’ text field blank.  2. Actor lefts ‘Phone’ text field blank.  3. Input phone not in numeric or phone length not in range 8 – 14 character.  4. Actor lefts ‘Email’ text field blank or input an invalid email address.  5. Actor lefts ‘Order Date’ blank or pick an invalid date. | System Responses:  System displays message to Actor :  - ‘Name cannot be blank’’.  - ‘Phone can’t be blank’.  - ‘The phone number should be numeric only and must be between 8 and 14 digits’.  - ‘Email is invalid’.  - ‘Order Date is invalid’. |



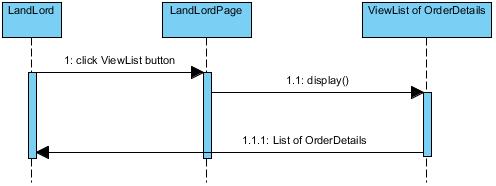
### 1.9.2 Update Booking Order Details

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Update Booking Order Details | |
| **Actors** | Landlord | |
| **Description** | Landlord wants to update booking order details. | |
| **Requirements** | Actor provides profile information | |
| **Pre-conditions** | Actor has logged in. | |
| **Post-conditions** | Success: Booking order details of this user has been updated  Fail: Refill information | |
| **Basic flow** | Actor Actions:  1. Actor clicks ‘Update’ button in Actor’s Home page menu.  3. Actor updates information and click ‘Save’ button. [Alternative 1] | System Responses:  2. System redirects to Booking Order Details page and displays form with the following control :  ­- ‘Customer Name’ text field  - ‘Phone’ text field  - ‘Email’ text field  - ‘Total Price’(Read Only)  - ‘Status’ (Read Only)  - ‘Order Date” Datepicker  - ‘Save” button  - ‘Close’ button  4. System checks the information  [Exception 1]  5. System update booking order details and then redirects to Actor Home page. |
| **Alternative flow** | Actor Actions:  [Alternative 1]  1. Actor click ‘Close’ button. | System Responses:  2. System returns to Home page. |
| **Exception** | Actor Actions:  [Exception 1]  1. Actor lefts ‘Name’ text field blank.  2. Actor lefts ‘Phone’ text field blank.  3. Input phone not in numeric or phone length not in range 8 – 14 character.  4. Actor lefts ‘Email’ text field blank or input an invalid email address.  5. Actor lefts ‘Order Date’ blank or pick an invalid date. | System Responses:  System displays message to Actor :  - ‘Name cannot be blank’’.  - ‘Phone can’t be blank’.  - ‘The phone number should be numeric only and must be between 8 and 14 digits’.  - ‘Email is invalid’.  - ‘Order Date is invalid’. |

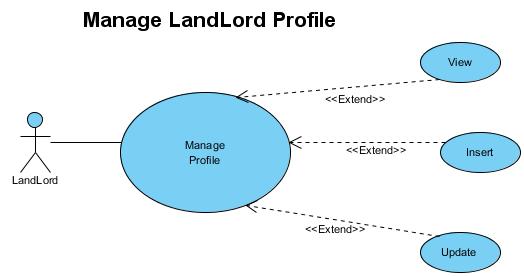


### 1.9.3 View Booking Order Details

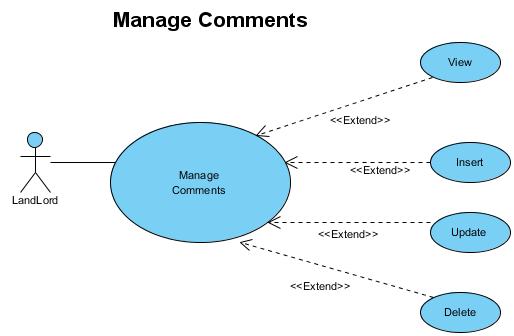
|  |  |  |
| --- | --- | --- |
| **Use Case Name** | View Booking Order Details | |
| **Actors** | Landlord | |
| **Description** | Landlord wants to view booking order details. | |
| **Requirements** | Actor provides profile information | |
| **Pre-conditions** | Actor has logged in. | |
| **Post-conditions** | Success: View booking order details of this user has been displayed.  Fail: Refill information | |
| **Basic flow** | Actor Actions:  1. Actor clicks ‘ViewList’ button in Actor’s Home page menu.  3. Actor views Order details and click ‘Close’ button. | System Responses:  2. System redirects to View Booking Order Details page and displays form with the following control :  - ‘Order ID’  - ‘Accommodation ID’  ­- ‘Customer Name’  - ‘Price’  - ‘Booked From Date’  - ‘Booked To Date’  - ‘Close’ button  [Exception 1]  4. System redirects to Actor Home page. |
| **Alternative flow** |  |  |
| **Exception** |  |  |



### 1.10 Landlord Manage Profile

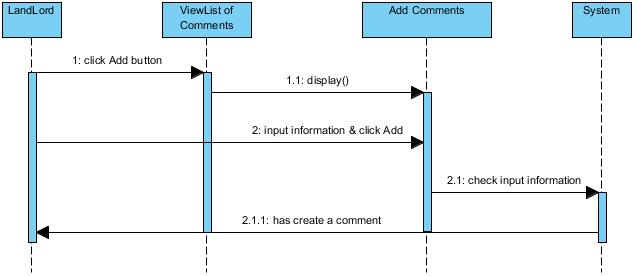


### 1.11 Landlord Manage Comments



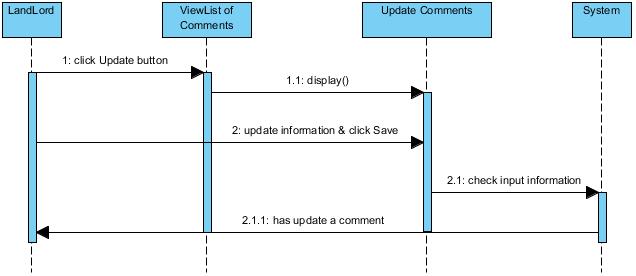
### 1.11.1 Insert Comments

|  |  |  |
| --- | --- | --- |
| **Use case name** | Insert Comments | |
| **Actors** | Landlord | |
| **Description** | Add Comments | |
| **Requirements** |  | |
| **Pre-conditions** | Actor has already login into the System as Administrator | |
| **Post-conditions** | Success: Comments added into the database. | |
| **Basic flow** | Actor actions:  1. From ‘Room details’ page, in view comments section, actor clicks on to the ‘Add’ button.  3. Actor:   * Input Comments into ‘Comments’ text field. * Clicks on to ‘Add’ button to finish adding. | System response:  2. System redirects to ‘Add Comments’ page and display form with the following controls:  -‘Comments text field’  -‘Add’ button  -‘Close’ button  4. System checks information. |
| **Alternative flow** | 1. Actor clicks on to ‘Close’ button | 1. System redirects to ‘Room Details’ page |
| **Exception** | Actor lefts ‘Comments’ text field. | Validation for all text field. |



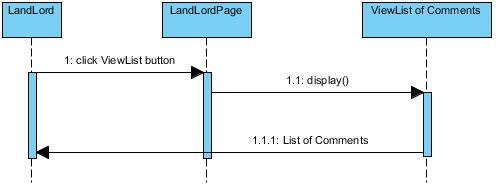
### 1.11.2 Update Comments

|  |  |  |
| --- | --- | --- |
| **Use case name** | Update Comments | |
| **Actors** | Customer | |
| **Description** | Update Comments | |
| **Requirements** |  | |
| **Pre-conditions** | Actor has already login into the System as Administrator | |
| **Post-conditions** | Success: Comments updated into the database. | |
| **Basic flow** | Actor actions:  1. From ‘Room details’ page in view comment section, actor clicks on to the ‘Update’ button.  3. Actor:   * Input Comments into ‘Comments’ text field. * Clicks on to ‘Update’ button to finish adding. | System response:  2. System redirects to ‘Add Comments’ page and display form with the following controls:  -‘Comments text field’  -‘Update’ button  -‘Close’ button  4. System checks information. |
| **Alternative flow** | 1. Actor clicks on to ‘Close’ button | 1. System redirects to ‘Room Details’ page |
| **Exception** |  |  |



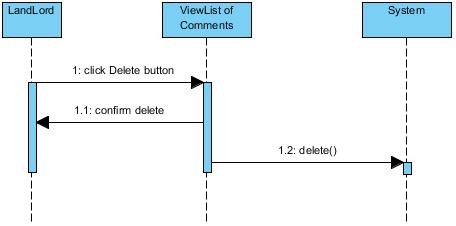
### 1.11.3 View Comments

|  |  |  |
| --- | --- | --- |
| **Use case name** | View Comments | |
| **Actors** | Landlord | |
| **Description** | View Comments | |
| **Requirements** |  | |
| **Pre-conditions** | Actor has already login into the System as Administrator | |
| **Post-conditions** | Success: Comments have been displayed | |
| **Basic flow** | Actor actions:  1. From ‘Room details’ page, actor clicks on to the ‘View Comments’ button. | System response:  2. System redirects to ‘Comments’ section and display:  -‘Comments  -‘Update’ button  -‘Delete’ button  -‘Close’ button  3. System checks information. |
| **Alternative flow** | 1. Actor clicks on to ‘Close’ button | 1. System redirects to ‘Room Details’ page |
| **Exception** |  |  |

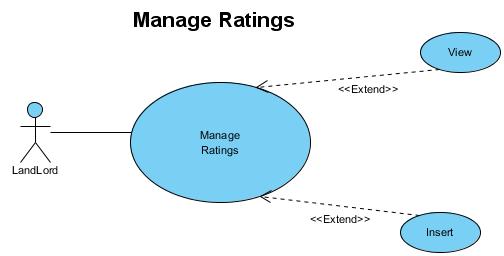


### 1.11.4 Delete Comments

|  |  |  |
| --- | --- | --- |
| **Use case name** | Delete Comments | |
| **Actors** | Landlord | |
| **Description** | Delete Comments | |
| **Requirements** |  | |
| **Pre-conditions** | Actor has already login into the System as Administrator | |
| **Post-conditions** | Success: Comments have been deleted | |
| **Basic flow** | Actor actions:  1.From ‘Room details’ page, in view comments section, actor clicks on to the ‘Delete Comments’ button.  3. Actor clicks ‘Yes’ button.  5. Actor clicks ‘No’ button. | System response:  2. A confirms message box appears:”Are you sure that you want to delete this comments?”  4. Comments from database has been deleted.  - System redirects to Comments section |
| **Alternative flow** |  |  |
| **Exception** |  |  |

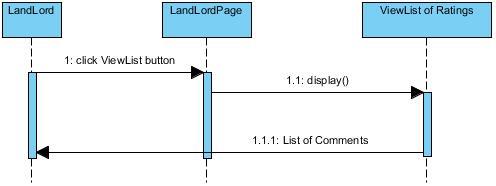


### 1.12 Landlord Manage Rating



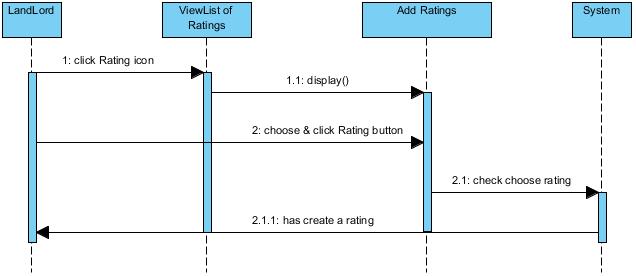
### 1.12.1 View Rating

|  |  |  |
| --- | --- | --- |
| **Use case name** | View Rating | |
| **Actors** | Landlord | |
| **Description** | View Rating | |
| **Requirements** |  | |
| **Pre-conditions** | Actor has already login into the System as Administrator | |
| **Post-conditions** | Success: Rating have been displayed | |
| **Basic flow** | Actor actions:  1. From ‘Room details’ page, actor clicks on to the ‘View Rating’ button. | System response:  2. System redirects to ‘Rating’ section and display:  -‘Rating’  -‘Close’ button  3. System checks information. |
| **Alternative flow** | 1. Actor clicks on to ‘Close’ button | 1. System redirects to ‘Room Details’ page |
| **Exception** |  |  |

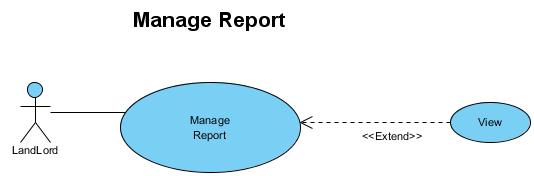


### 1.12.2 Insert Rating

|  |  |  |
| --- | --- | --- |
| **Use case name** | Insert Rating | |
| **Actors** | Landlord | |
| **Description** | Add Rating | |
| **Requirements** |  | |
| **Pre-conditions** | Actor has already login into the System as Administrator | |
| **Post-conditions** | Success: Rating score added into the database. | |
| **Basic flow** | Actor actions:  1. From ‘Room details’ page, actor clicks on to the ‘Rating’ icon. | System response:  2. System redirects to ‘Rating’ page and display form with the following controls:  -‘Rating’  -‘Rating’ button  -‘Close’ button  3. System checks information. |
| **Alternative flow** | 1. Actor clicks on to ‘Close’ button | 1. System redirects to ‘Room Details’ page |
| **Exception** |  |  |

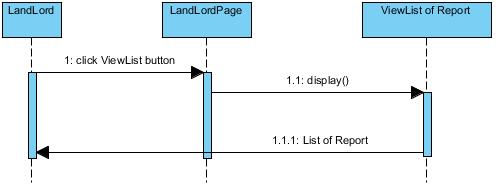


### 1.13 Landlord Manage Report

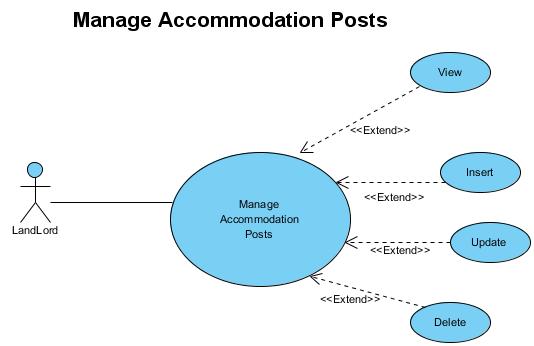


### 1.13.1 View Report

|  |  |  |
| --- | --- | --- |
| **Use case name** | View Report | |
| **Actors** | Landlord | |
| **Description** | View Report | |
| **Requirements** |  | |
| **Pre-conditions** | Actor has already login into the System as Administrator | |
| **Post-conditions** | Success: Report have been displayed | |
| **Basic flow** | Actor actions:  1. From Actor home menu page, actor clicks on to the ‘View Report’ button. | System response:  2. System redirects to ‘ViewList of Report’ page and display list of all of Report in table form with control :  ‘Report’ button  ‘Close’ button  ‘Report information’ frame. |
| **Alternative flow** | 1. Actor clicks on to ‘Close’ button | 1. System redirects to ‘Actor Home’ page |
| **Exception** |  |  |

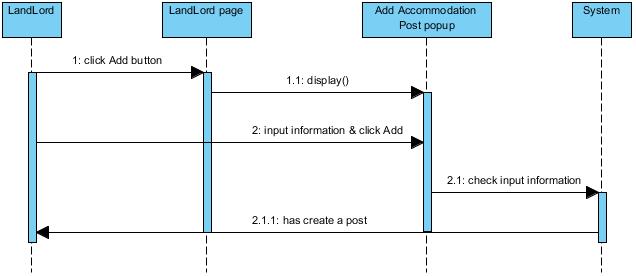


### 1.14 Landlord Manage Accommodation Posts



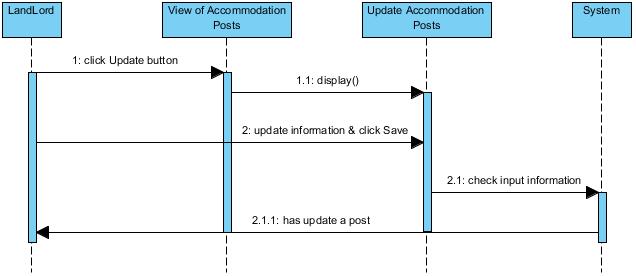
### 1.14.1 Insert Accommodation Posts

|  |  |  |
| --- | --- | --- |
| **Use case name** | Add Accommodation Posts | |
| **Actors** | LandLord | |
| **Description** | Add Accommodation Posts | |
| **Requirements** |  | |
| **Pre-conditions** | Actor has already login into the System as LandLord | |
| **Post-conditions** | Success: A Accommodation Post is added into the database. | |
| **Basic flow** | Actor actions:  1. From ‘LandLord’ page, actor clicks on to the ‘Add’ button.  3. Actor:   * Inputs Accommodation Post information into textfield. * Clicks on to ‘Add’ button to finish adding. | System response:  2. System redirects to ‘Add Accommodation Post’ popup and display form with the following controls:  - AccomName textfield  - Image is type thumbnail  - Price textfield  - Address textfield  - Description textfield  - NoOfBed, NoOfPersons, NoOfToilet textfield  - CreatedDate is used to getdate()  - CreatedBy is used itseft  4. System checks information. |
| **Alternative flow** | 1. Actor clicks on to ‘Cancel’ button | 1. System redirects to ‘LandLord’ page |
| **Exception** | 1. Actor lefts or wrong input text field. | Validation for all text field. |



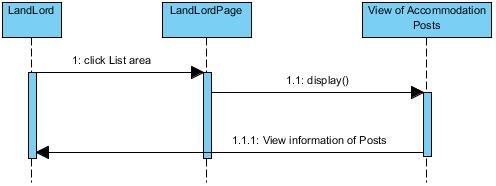
### 1.14.2 Update Accommodation Posts

|  |  |  |
| --- | --- | --- |
| **Use case name** | Update Accommodation Post | |
| **Actors** | LandLord | |
| **Description** | Update Accommodation Post | |
| **Requirements** |  | |
| **Pre-conditions** | Actor has already login into the System as LandLord | |
| **Post-conditions** | Success: A Accommodation Post is updated in the database. | |
| **Basic flow** | Actor actions:  1. From ‘LandLord’ page, actor clicks on to the ‘Update’ button.  3. Actor:   * Inputs Accommodation Post information into textfield. * Clicks on to ‘Update’ button to finish update. | System response:  2. System redirects to ‘Update Accommodation Post’ popup and display form with the following controls:  - AccomName textfield  - Image is type thumbnail  - Price textfield  - Address textfield  - Description textfield  - NoOfBed, NoOfPersons, NoOfToilet textfield  - CreatedDate is used to getdate()  - CreatedBy is used itseft  4. System checks information. |
| **Alternative flow** | 1. Actor clicks on to ‘Cancel’ button | 1. System redirects to ‘LandLord’ page |
| **Exception** | 1. Actor lefts or wrong input textfield. | Validation for all textfield. |



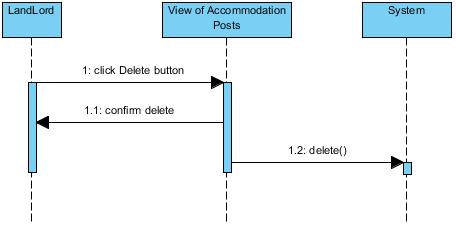
### 1.14.3 View Accommodation Posts

|  |  |  |
| --- | --- | --- |
| **Use case name** | View List Accommodation Posts | |
| **Actors** | LandLord | |
| **Description** | View List Accommodation Posts | |
| **Requirements** |  | |
| **Pre-conditions** | Actor has already login into the System as LandLord | |
| **Post-conditions** | Success: View List of Accommodation Posts | |
| **Basic flow** | Actor actions:  1. From ‘LandLord’ page, actor clicks on to the ‘List’ area. | System response:  2. System redirects to ‘View of Accommodation Post’ page and display list of all of accommodation post in table form with control :  ‘Accommodation Post’ information  ‘Update Accommodation Post’ button  ‘Delete Accommodation’ button |
| **Alternative flow** |  |  |
| **Exception** |  |  |

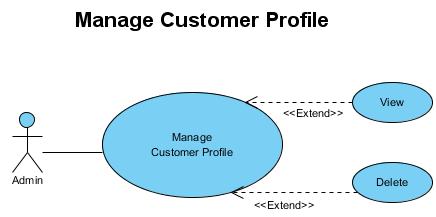


### 1.14.4 Delete Accommodation Posts

|  |  |  |
| --- | --- | --- |
| **Use case name** | Delete Accommodation Post | |
| **Actors** | LandLord | |
| **Description** | Delete Accommodation Post | |
| **Requirements** |  | |
| **Pre-conditions** | Actor has already logged into the System as LandLord | |
| **Post-conditions** | Success: A Accommodation Post is deleted | |
| **Basic flow** | Actor actions:  - From ‘View of Accommodation Post’ page, actor clicks on to the ‘Delete’ button of the question that actor wants to delete.  - Actor clicks ‘Yes’ button.  - Actor clicks ‘No’ button. | System response:  - A confirms message box appears:”Are you sure that you want to delete this Accommodation Post?”  - System saves question information.  - System redirects to View of Accommodation Post page. |
| **Alternative flow** |  |  |
| **Exception** |  |  |

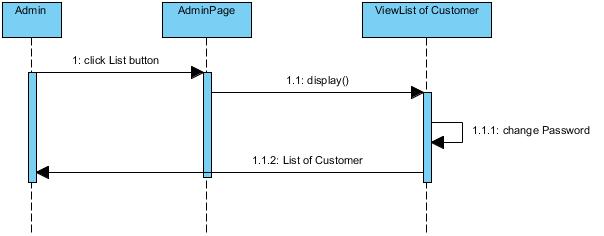


### 1.15 Admin Manage Customer Profile



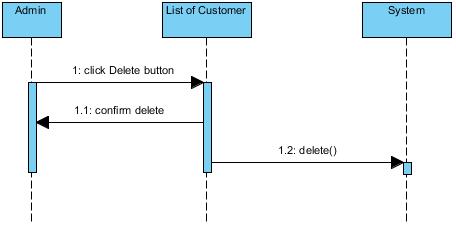
### 1.15.1 View Customer Profile

|  |  |  |
| --- | --- | --- |
| **Use case name** | View List of Customers | |
| **Actors** | Administrator | |
| **Description** | View List of Customers | |
| **Requirements** |  | |
| **Pre-conditions** | Actor has already logged into the System as Administrator | |
| **Post-conditions** | Success: View List of Customers | |
| **Basic flow** | Actor actions:  1. From ‘Admin’ page, actor clicks on to ‘List’ button. | System response:  2. System redirects to ‘ViewList Customers’ page  3. Appear icon “Customer”  4. Display list of Customers in table form:   * CustName * AccountCustomer * Phone * Email * Profile Picture (that may or may not) * RegisterDate * Notified * IsActived (yes or no) * RoleName * Requirement of Customers * Default password is room4u (Admin can change the password for the customer, then auto send notification mail is “Password has been changed. Your new password is…”) |
| **Alternative flow** |  |  |
| **Exception** |  |  |

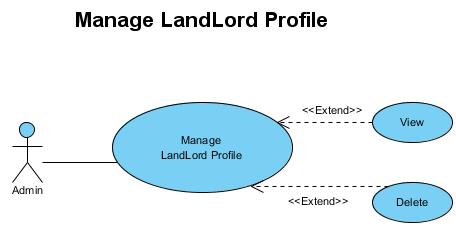


### 1.15.2 Delete Customer Profile

|  |  |  |
| --- | --- | --- |
| **Use case name** | Delete Customer | |
| **Actors** | Administrator | |
| **Description** | Delete Customer | |
| **Requirements** |  | |
| **Pre-conditions** | Actor has already logged into the System as Administrator | |
| **Post-conditions** | Success: A Customer is deleted | |
| **Basic flow** | Actor actions:  1. From ‘List of Customers’ page, actor clicks on to the ‘Delete’ button of the Customer whom actor wants to delete (symbol of Status button is activate).  3. Actor clicks ‘Yes’ button.  5. Actor clicks ‘No’ button. | System response:  2. A confirms message box appears:”Are you sure that you want to delete this Customer?”  4. System saves Customers information.  6. System redirects to List of Customer page. |
| **Alternative flow** |  |  |
| **Exception** |  |  |



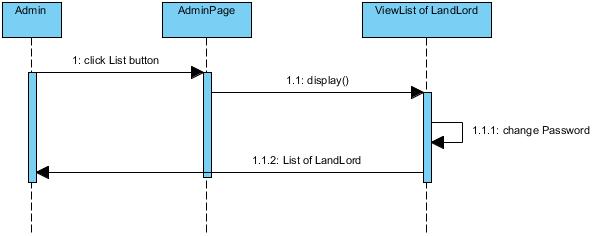
### 1.16 Admin Manage Landlord Profile



### 1.16.1 View Landlord Profile

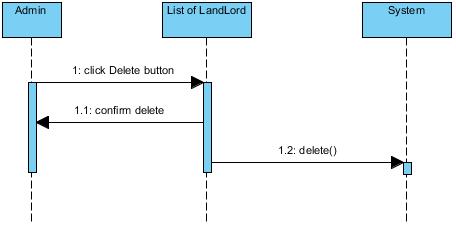
|  |  |  |
| --- | --- | --- |
| **Use case name** | View List of Landlord | |
| **Actors** | Administrator | |
| **Description** | View List of LandLord | |
| **Requirements** |  | |
| **Pre-conditions** | Actor has already logged into the System as Administrator | |
| **Post-conditions** | Success: View List of LandLord | |
| **Basic flow** | Actor actions:  1. From ‘Admin’ page, actor clicks on to ‘List’ button. | System response:  2. System redirects to ‘ViewList of LandLord’ page  3. Appear icon “LandLord”  4. Display list of LandLord in table form similar to “Customer profile” but only have “list room is booking” is showed. |
| **Alternative flow** |  |  |
| **Exception** |  |  |

### 

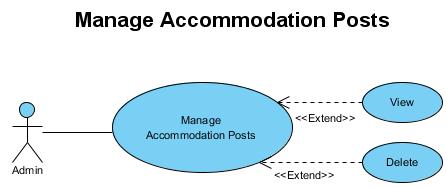


### 1.16.2 Delete Landlord Profile

|  |  |  |
| --- | --- | --- |
| **Use case name** | Delete LandLord | |
| **Actors** | Administrator | |
| **Description** | Delete LandLord | |
| **Requirements** |  | |
| **Pre-conditions** | Actor has already logged into the System as Administrator | |
| **Post-conditions** | Success: A LandLord is deleted | |
| **Basic flow** | Actor actions:  - From ‘List of LandLord’ page, actor clicks on to the ‘Delete’ button of the LandLord whom actor wants to delete (symbol of Status button is activate).  - Actor clicks ‘Yes’ button.  - Actor clicks ‘No’ button. | System response:  - A confirms message box appears:”Are you sure that you want to delete this LandLord?”  - System saves LandLord information.  - System redirects to List of LandLord page. |
| **Alternative flow** |  |  |
| **Exception** |  |  |

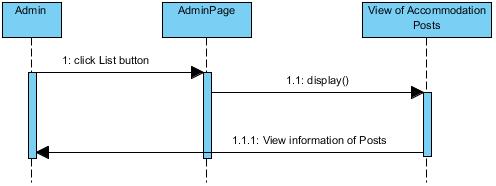


### 1.17 Admin Manage Accommodation Post



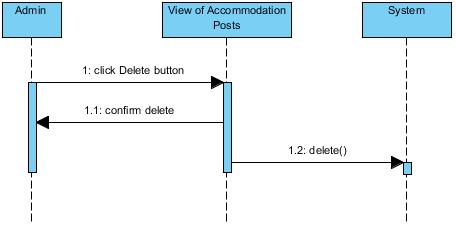
### 1.17.1 View Accommodation Post

|  |  |  |
| --- | --- | --- |
| **Use case name** | View List of Accommodation Posts | |
| **Actors** | Administrator | |
| **Description** | View List of Accommodation Posts | |
| **Requirements** |  | |
| **Pre-conditions** | Actor has already logged into the System as Administrator | |
| **Post-conditions** | Success: View List of Accommodation Posts | |
| **Basic flow** | Actor actions:  1. From ‘Admin’ page, actor clicks on to ‘List’ button. | System response:  2. System redirects to ‘View of Accommodation Posts’ page  3. Appear icon “LandLord”  4. Display list of Accommodation Posts in table form:   * AccomName * Images is type thumbnail * Price * Address * Description * Number of Bed * Number of Persons * Number of Toilet * Created Date * Created By |
| **Alternative flow** |  |  |
| **Exception** |  |  |

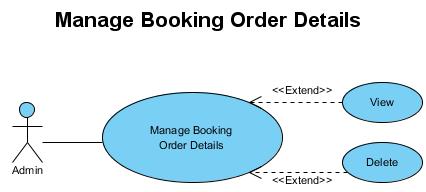


### 1.17.2 Delete Accommodation Post

|  |  |  |
| --- | --- | --- |
| **Use case name** | Delete Accommodation Post | |
| **Actors** | Administrator | |
| **Description** | Delete Accommodation Post | |
| **Requirements** |  | |
| **Pre-conditions** | Actor has already logged into the System as Administrator | |
| **Post-conditions** | Success: A Accommodation Post is deleted | |
| **Basic flow** | Actor actions:  - From ‘View of Accommodation Post’ page, actor clicks on to the ‘Delete’ button of the Accommodation Post that actor wants to delete.  - Actor clicks ‘Yes’ button.  - Actor clicks ‘No’ button. | System response:  - A confirms message box appears:”Are you sure that you want to delete this Accommodation Post?”  - System saves Accommodation Post information.  - System redirects to View of Accommodation Post page. |
| **Alternative flow** |  |  |
| **Exception** |  |  |

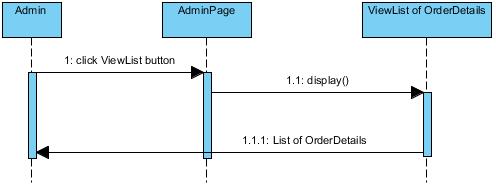


### 1.18 Admin Manage Booking Order Details



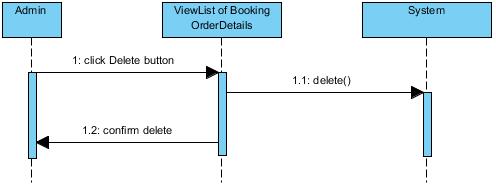
### 1.18.1 View Booking Order Details

|  |  |  |
| --- | --- | --- |
| **Use case name** | View List of Booking Order Details | |
| **Actors** | Administrator | |
| **Description** | View List of Booking Order Details | |
| **Requirements** |  | |
| **Pre-conditions** | Actor has already logged into the System as Administrator | |
| **Post-conditions** | Success: View List of Booking Order Details | |
| **Basic flow** | Actor actions:  1. From ‘Admin’ page, actor clicks on to ‘ViewList’ button.  5. If Status is booked, Date to Book and Order Date would visible, otherwise was invisible. | System response:  2. System redirects to ‘ViewList of Order Details’ page  3. Appear icon “LandLord” or “Customer”  4. Display list of Order Details in table form:   * Price * Booked From Date * Booked To Date * Order Date * Name, Phone, Email of guest (may or may not) * Status (is booked or not booked) |
| **Alternative flow** |  |  |
| **Exception** |  |  |

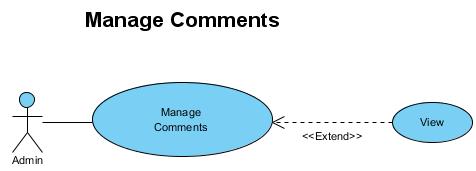


### 1.18.2 Delete Booking Order Details

|  |  |  |
| --- | --- | --- |
| **Use case name** | Delete Booking Order Detail | |
| **Actors** | Administrator | |
| **Description** | Delete Booking Order Detail | |
| **Requirements** |  | |
| **Pre-conditions** | Actor has already logged into the System as Administrator | |
| **Post-conditions** | Success: A Booking Order Detail is deleted | |
| **Basic flow** | Actor actions:  - From ‘ViewList of Booking Order Details’ page, actor clicks on to the ‘Delete’ button of the Booking Order Detail that actor wants to delete.  - Actor clicks ‘Yes’ button.  - Actor clicks ‘No’ button. | System response:  - A confirms message box appears:”Are you sure that you want to delete this Booking Order Detail?”  - System saves Booking Order Detail information.  - System redirects to View of Booking Order Detail page. |
| **Alternative flow** |  |  |
| **Exception** |  |  |

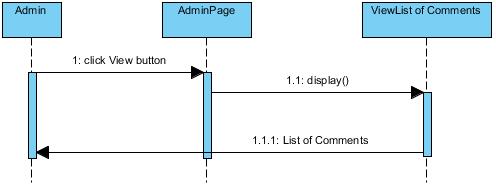


### 1.19 Admin Manage Comments

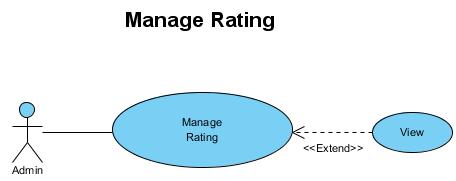


### 1.19.1 View Comments

|  |  |  |
| --- | --- | --- |
| **Use case name** | View List of Comments | |
| **Actors** | Administrator | |
| **Description** | View List of Comments | |
| **Requirements** |  | |
| **Pre-conditions** | Actor has already login into the System as Administrator | |
| **Post-conditions** | Success: View List of Comments | |
| **Basic flow** | Actor actions:  1. From ‘Admin’ page, actor clicks on to the ‘View Comment’ button. | System response:  2. System redirects to ‘ViewList of Comments’ page and display list of all of Comments in table form with control :  ‘Comment’ label  ‘Comment’ information |
| **Alternative flow** |  |  |
| **Exception** |  |  |

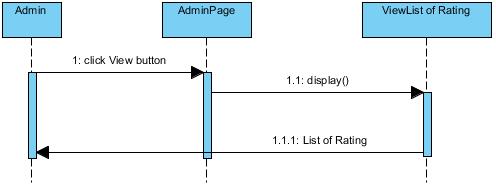


### 1.20 Admin Manage Rating

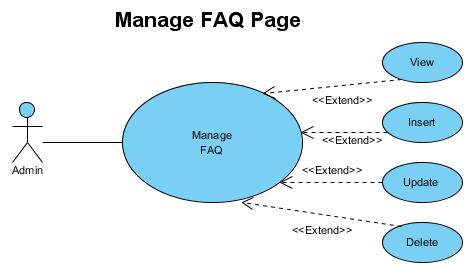


### 1.20.1 View Rating

|  |  |  |
| --- | --- | --- |
| **Use case name** | View List of Rating | |
| **Actors** | Administrator | |
| **Description** | View List of Rating | |
| **Requirements** |  | |
| **Pre-conditions** | Actor has already login into the System as Administrator | |
| **Post-conditions** | Success: View List of Rating | |
| **Basic flow** | Actor actions:  1. From ‘Admin’ page, actor clicks on to the ‘View Rating’ button. | System response:  2. System redirects to ‘ViewList of Rating’ page and display list of all of Rating in table form with control :  ‘Rating’ label  ‘Rating’ score |
| **Alternative flow** |  |  |
| **Exception** |  |  |

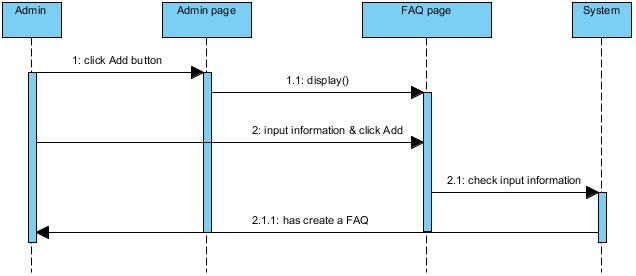


### 1.21 Admin Manage FAQ



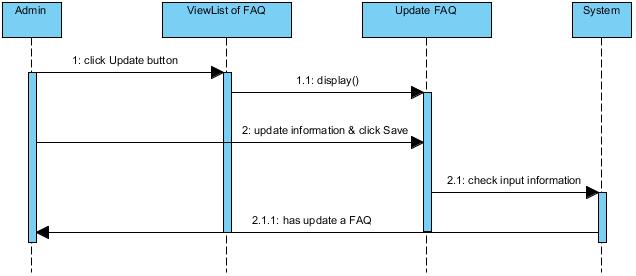
### 1.21.1 Insert FAQ

|  |  |  |
| --- | --- | --- |
| **Use case name** | Add Question | |
| **Actors** | Administrator | |
| **Description** | Add a Question | |
| **Requirements** |  | |
| **Pre-conditions** | Actor has already login into the System as Administrator | |
| **Post-conditions** | Success: A Question is added into the database. | |
| **Basic flow** | Actor actions:  1. From ‘Admin’ page, actor clicks on to the ‘Add Question’ button.  3. Actor:   * Inputs Question into ‘Question’ textfield. * Inputs Answer into ‘Answer’ textfield. * Clicks on to ‘Add’ button to finish adding. | System response:  2. System redirects to ‘Add Question’ page and display form with the following controls:  ‘Question’ textfield.  ‘Answer’ textfield.  ‘Add’ button.  ‘Clear’ button  ‘Cancel’  4. System checks information. |
| **Alternative flow** | 1. Actor clicks on to ‘Cancel’ button | 1. System redirects to ‘Admin’ page |
| **Exception** | 1. Actor lefts ‘Question’ textfield.  2. Actor lefts ‘Answer’ textfield. | Validation for all textfield. |



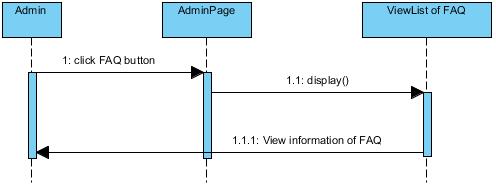
### 1.21.2 Update FAQ

|  |  |  |
| --- | --- | --- |
| **Use case name** | Update Question | |
| **Actors** | Administrator | |
| **Description** | Update a Question | |
| **Requirements** |  | |
| **Pre-conditions** | Actor has already login into the System as Administrator | |
| **Post-conditions** | Success: A Question is updated in the database. | |
| **Basic flow** | Actor actions:  1. From ‘ViewList of FAQ’ page, actor clicks on to the ‘Update’ button  3. Actor:   * Inputs Question into ‘Question’ textfield. * Inputs Answer into ‘Answer’ textfield. * Clicks on to ‘Save’ button to finish adding. | System response:  2. System redirects to ‘Update Question’ page and display form with the following controls:  ‘Question’ textfield.  ‘Answer’ textfield.  ‘Update’ button.  ‘Cancel’ button  4. System checks information. |
| **Alternative flow** | 1. Actor clicks on to ‘Cancel’ button | 1. System redirects to ‘View List of FAQ’ page  2. Reload data to all field |
| **Exception** | 1. Actor lefts ‘Question’ textfield.  2. Actor lefts ‘Answer’ textfield. | Validation for all textfield. |



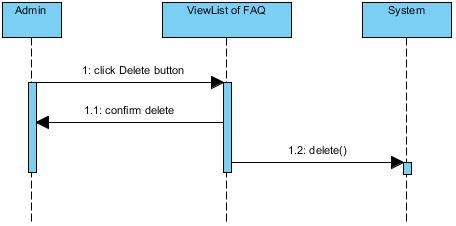
### 1.21.3 View FAQ

|  |  |  |
| --- | --- | --- |
| **Use case name** | View List of Questions | |
| **Actors** | Administrator | |
| **Description** | View List of Questions | |
| **Requirements** |  | |
| **Pre-conditions** | Actor has already login into the System as Administrator | |
| **Post-conditions** | Success: View List of Questions | |
| **Basic flow** | Actor actions:  1. From ‘Admin’ page, actor clicks on to the ‘FAQ’ button. | System response:  2. System redirects to ‘ViewList of FAQ’ page and display list of all of Questions in table form with control :  ‘Question’ label  ‘Update Question’ button  ‘Delete Question’ button  ‘Add Question’ button |
| **Alternative flow** |  |  |
| **Exception** |  |  |



### 1.21.4 Delete FAQ

|  |  |  |
| --- | --- | --- |
| **Use case name** | Delete Question | |
| **Actors** | Administrator | |
| **Description** | Delete Question | |
| **Requirements** |  | |
| **Pre-conditions** | Actor has already logged into the System as Administrator | |
| **Post-conditions** | Success: A Question is deleted | |
| **Basic flow** | Actor actions:  - From ‘ViewList of FAQ’ page, actor clicks on to the ‘Delete’ button of the question that actor wants to delete.  - Actor clicks ‘Yes’ button.  - Actor clicks ‘No’ button. | System response:  - A confirms message box appears:”Are you sure that you want to delete this question?”  - System saves question information.  - System redirects to View of Question page. |
| **Alternative flow** |  |  |
| **Exception** |  |  |

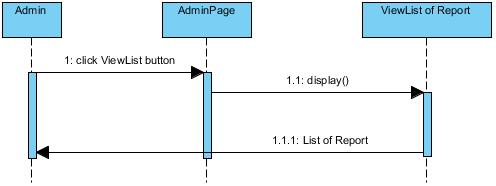


### 1.22 Admin Manage Report



### 1.22.1 View Report

|  |  |  |
| --- | --- | --- |
| **Use case name** | View List of Report | |
| **Actors** | Administrator | |
| **Description** | View List of Report | |
| **Requirements** |  | |
| **Pre-conditions** | Actor has already login into the System as Administrator | |
| **Post-conditions** | Success: View List of Report | |
| **Basic flow** | Actor actions:  1. From ‘Admin’ page, actor clicks on to the ‘ViewList’ button. | System response:  2. System redirects to ‘ViewList of Report’ page and display list of all of Report in table form with control :  ‘Report’ button  ‘Close’ button  ‘Report information’ frame |
| **Alternative flow** |  |  |
| **Exception** |  |  |



*--- The End of Review 2---*

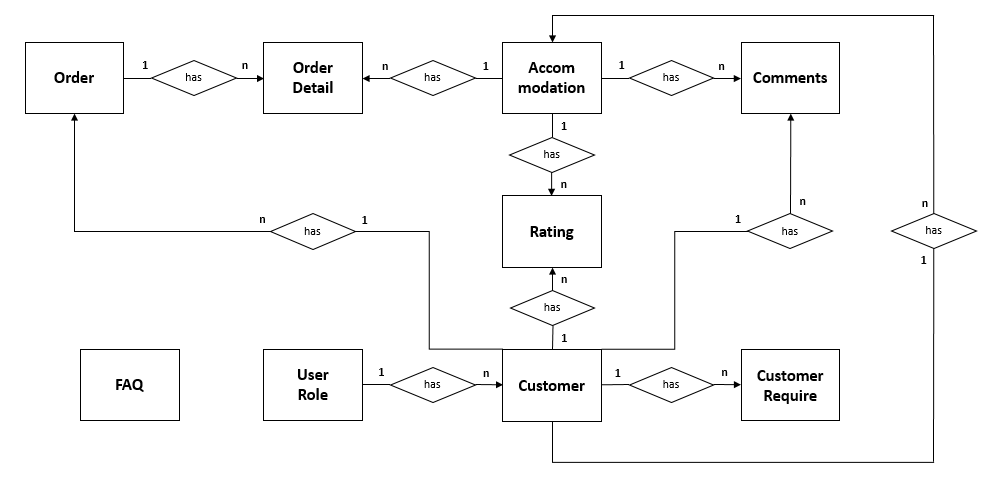
|  |  |  |
| --- | --- | --- |
|  | **Prepared by (Student)** | **Approved by (Faculty)** |
| **Signature** | Team leader  **Ho Chau Can** | **Hoang Duc Quang** |
| **Date** |  |  |

Review 3

Prepare by: **SC Team**

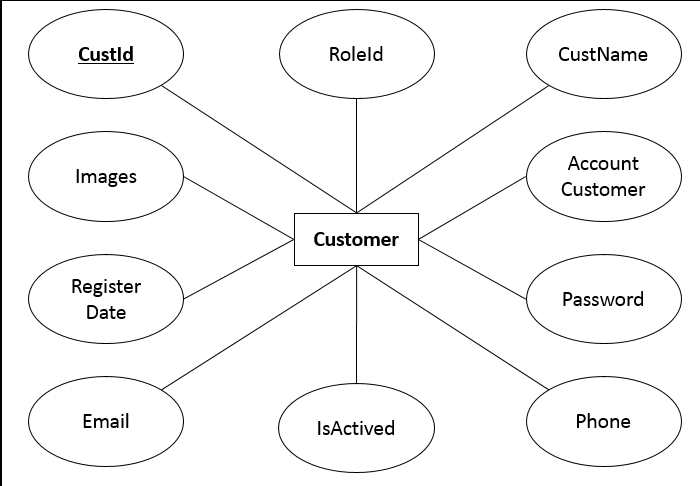
Class: Fat2 F2-13-1402-t0

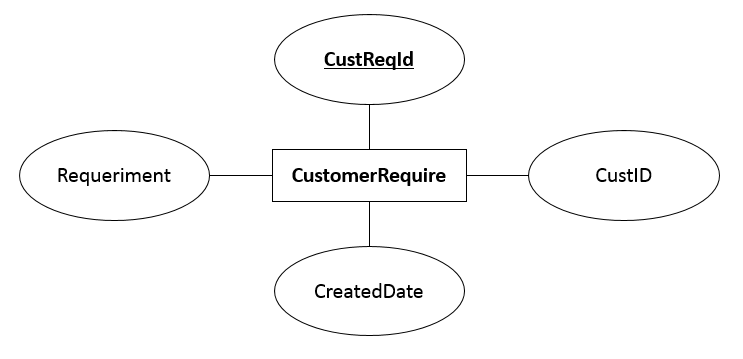
1. ENTITY RELATIONSHIP DIAGRAM



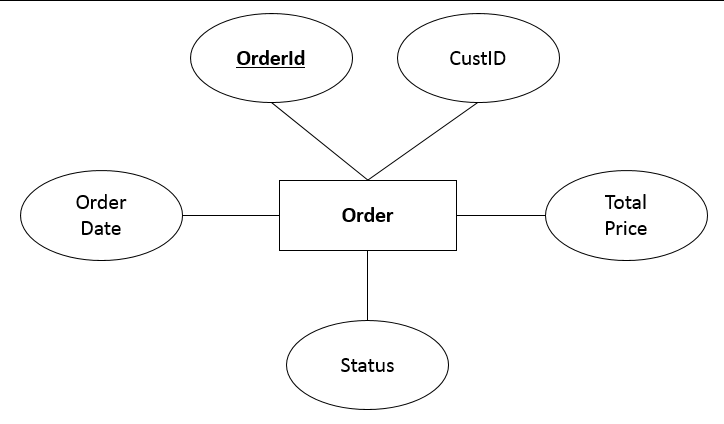
1. ENTITY PROPERTIES

CUSTOMER

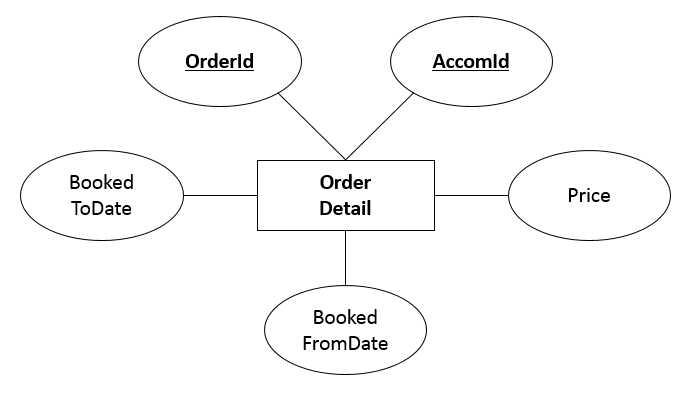


CUSTOMER REQUIRE

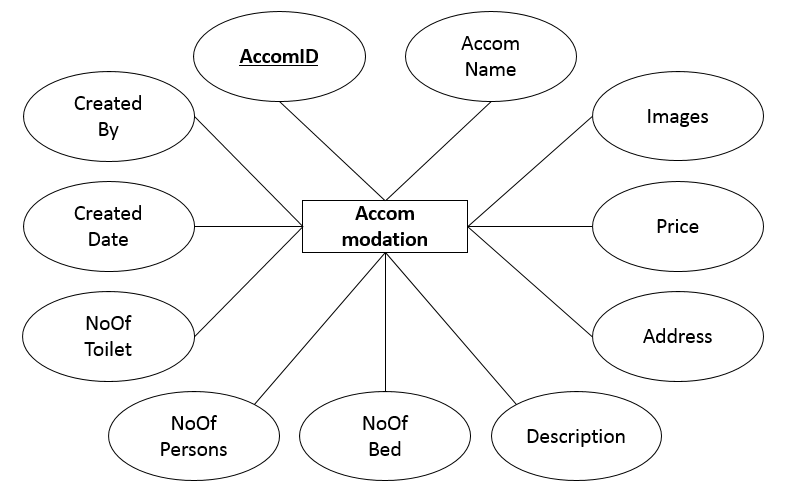
ORDER



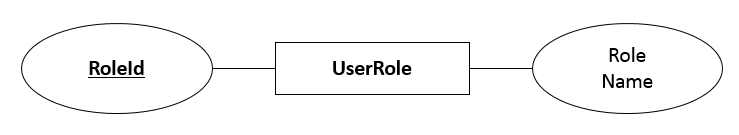
ORDER DETAIL



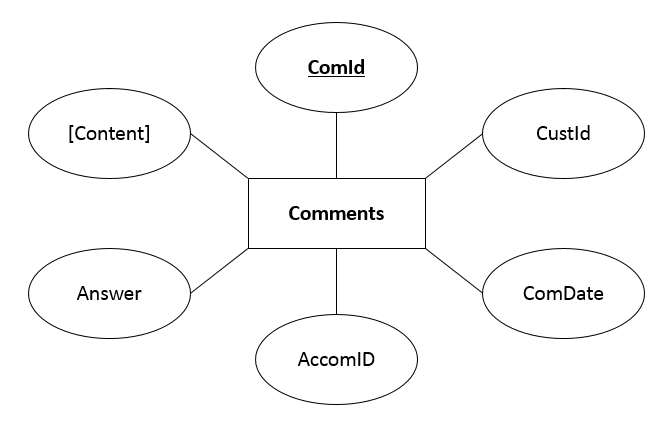
ACCOMMODATION



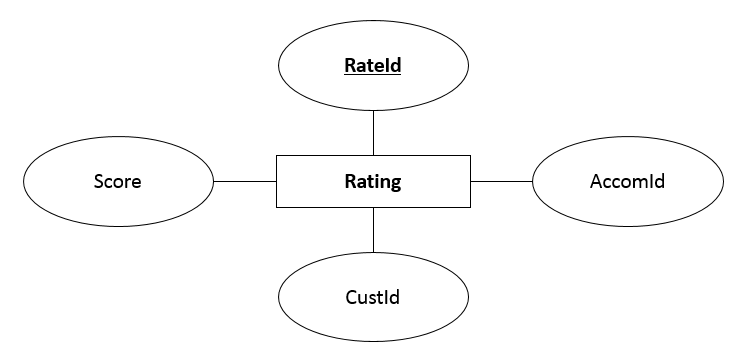
USER ROLE



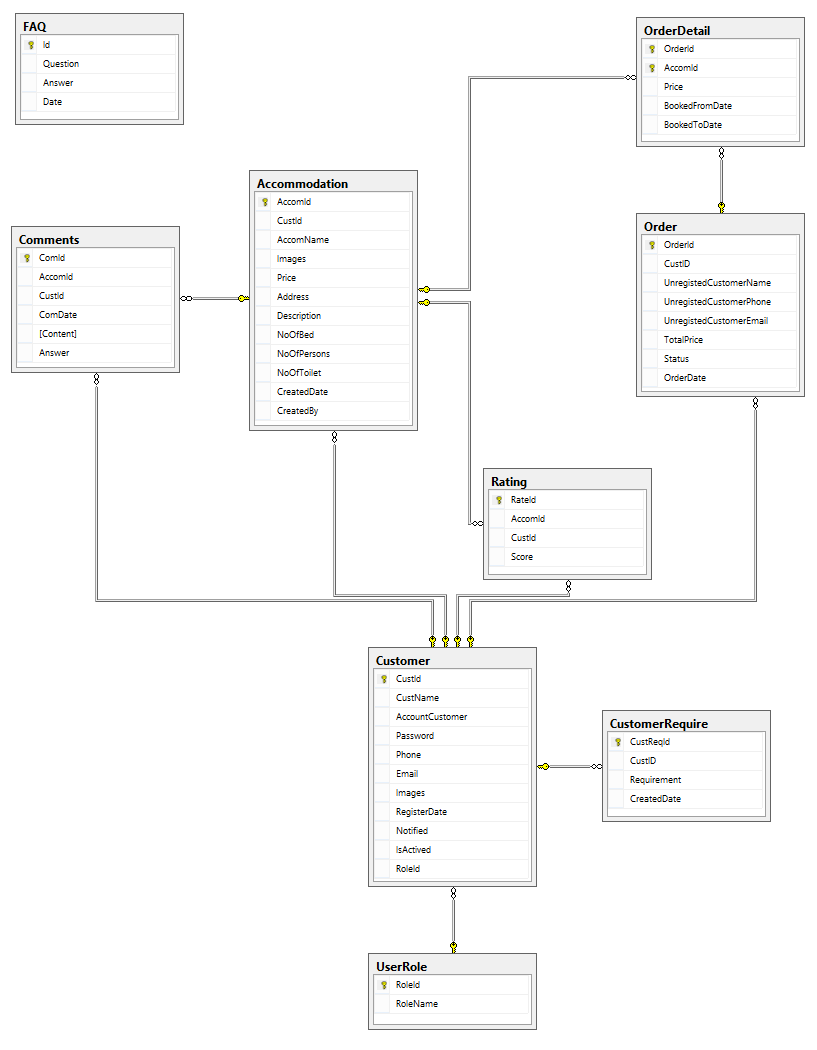
COMMENTS



RATING



1. DATABASE RELATIONSHIP DIAGRAM



1. DATABASE STRUCTURE

* + - 1. CUSTOMER

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Allow Null** | **Default Value** | **Key** | **Validation** | **Description** |
| CustId | int | No |  | Primary | Unique | Customer ID Identity |
| CustName | nvarchar(200) | No |  |  |  | Customer’s Name |
| AccountCustomer | nvarchar(200) | No |  |  |  |  |
| Password | nvarchar(200) | No |  |  |  | Customer’s Password |
| Phone | nvarchar(50) | No |  |  | Valid phone | Customer’s Phone |
| Email | nvarchar(200) | No |  |  | No duplication;  Valid email | Customer’s Email |
| RegisterDate | datetime | No |  |  |  | Registered Date |
| IsActive | Bit | No | 1 |  |  | Receiving Notification:  0: disabled  1: enabled |
| Image | nvarchar(200) | No |  |  |  |  |
| RoleId | int | No |  | Foreign | Unique |  |

* + - 1. CUSTOMER REQUIRE

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Allow Null** | **Default Value** | **Key** | **Validation** | **Description** |
| CustReqId | int | No |  | Primary | Unique | Customer Require ID Identity |
| CustId | int |  |  | Foreign | Unique | Customer ID, FK of Customers table |
| Requirement | nvarchar(MAX) |  |  |  |  | Content Of The Requirement |
| CreatedDate | datetime | No |  |  |  | Submitted Date |

* + - 1. ORDER

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Allow Null** | **Default Value** | **Key** | **Validation** | **Description** |
| OrderId | int | No |  | Primary | Unique | Order ID Identity |
| CustId | int | No |  | Foreign | Unique | Customer ID, FK of Customers table |
| TotalPrice | decimal(18,2) |  |  |  | Valid price;  VNĐ currency | Total Amount |
| Status | nvarchar(100) |  |  |  |  | Status of the order:  Paid or Not Paid |
| OrderDate | datetime |  |  |  |  | Submitted Date |

* + - 1. FAQ

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Allow Null** | **Default Value** | **Key** | **Validation** | **Description** |
| Id | int | No |  | Primary | Unique | FAQ ID Identity |
| Question | nvarchar(MAX) |  |  |  |  | FAQ Question |
| Answer | nvarchar(MAX) |  |  |  |  | FAQ Answer |
| Date | datetime |  |  |  |  | FAQ Created On Date |

* + - 1. ACCOMMODATION

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Allow Null** | **Default Value** | **Key** | **Validation** | **Description** |
| AccomId | int | No |  | Primary | Unique | Accommodation ID Identity |
| AccomName | nvarchar(200) | No |  |  |  | Accommodation’s Name |
| Images | nvarchar(200) | No |  |  |  | Accommodation’s Images |
| Price | float | No |  |  | Valid price;  VNĐ currency | Accommodation’s Price |
| Address | nvarchar(MAX) | No |  |  |  | Accommodation’s Address |
| Description | nvarchar(MAX) |  |  |  |  | Accommodation’s Description |
| NoOfBed | int | No |  |  | only digit | Number of Bed |
| NoOfPersons | int |  |  |  | only digit | Number of Person |
| NoOfToilet | int |  |  |  | only digit | Number of Toilet |
| CreatedDate | datetime | No |  |  |  | Submitted Date |
| CreatedBy | nvarchar(200) | No |  |  |  | Created By: Name of the use belong to Landlord Role |

* + - 1. ORDER DETAIL

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Allow Null** | **Default Value** | **Key** | **Validation** | **Description** |
| OrderId | int | No |  | Foreign | Unique | Order ID Identity |
| AccomId | int | No |  | Foreign | Unique | Accommodation ID, FK of Accommodation table |
| Price | decimal(18,2) |  |  |  | Valid price;  VNĐ currency | Pricing |
| BookedFromDate | datetime |  |  |  |  | Booked From Date |
| BookedToDate | datetime |  |  |  |  | Booked To Date |

* + - 1. RATING

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Allow Null** | **Default Value** | **Key** | **Validation** | **Description** |
| RateId | int | No |  | Primary | Unique | Rate ID Identity |
| AccomId | int |  |  | Foreign | Unique | Accommodation ID, FK of Accommodation table |
| CustId | int |  |  | Foreign | Unique | Customer ID, FK of Customer table |
| Score | int |  |  |  |  |  |

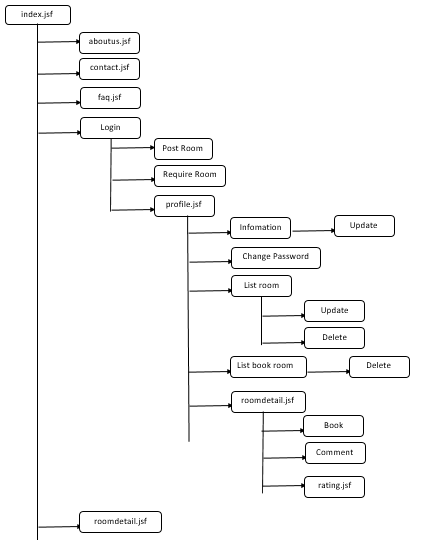
* + - 1. COMMENTS

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Allow Null** | **Default Value** | **Key** | **Validation** | **Description** |
| ComId | int | No |  | Primary | Unique | Comment ID Identity |
| AccomId | int |  |  | Foreign | Unique | Accommodation ID, FK of Accommodation table |
| CustId | int |  |  | Foreign | Unique | Customer ID, FK of Customer table |
| ComDate | datetime |  |  |  |  | Comment Submitted On Date |
| [Content] | nvarchar(MAX) |  |  |  |  | Content Of The Comment |
| Answer | nvarchar(MAX) |  |  |  |  | Reply by the user belong to Landlord Role |

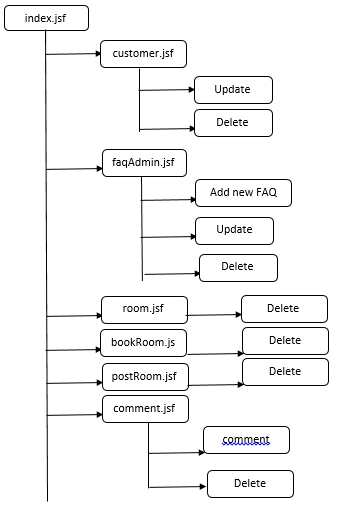
* + - 1. USER ROLE

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Allow Null** | **Default Value** | **Key** | **Validation** | **Description** |
| ComId | int | No |  | Primary | Unique |  |
| AccomId | int |  |  | Foreign | Unique |  |

1. SITE MAP
2. User Side

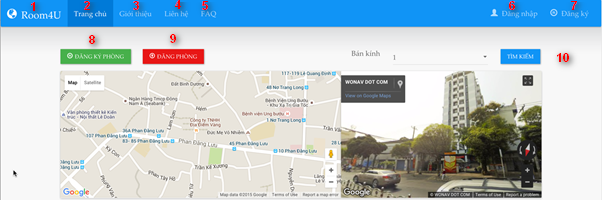


1. Admin Side



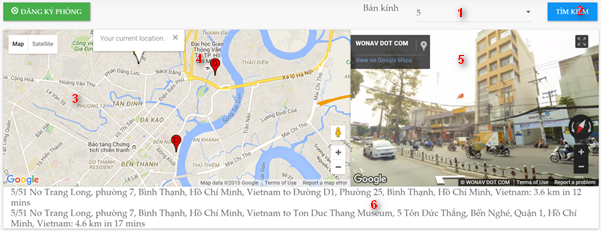
1. GUI DESIGN
   * + 1. User Side

1. 1 Home Page



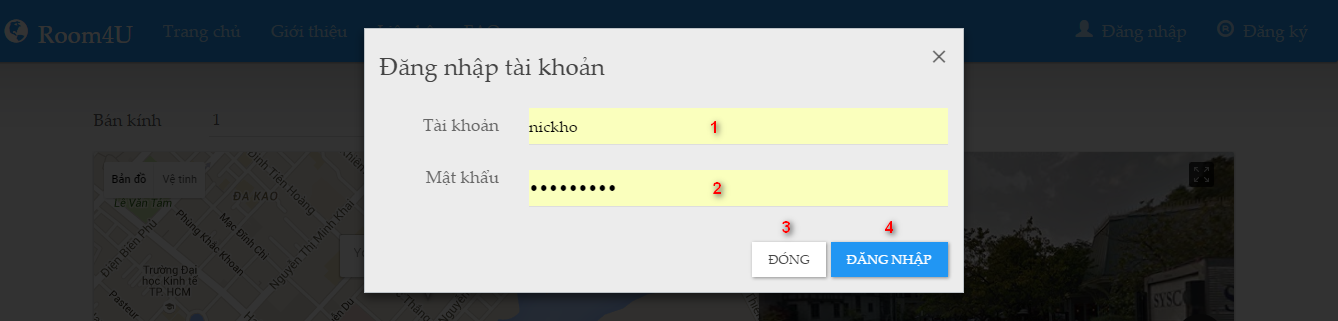
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Validation** | **Event** | **Description** | **Status** |
| 1 | Room4U | Link |  | Click | Redirect to Home page |  |
| 2 | Trang chủ | Link |  | Click | Redirect to Home page |  |
| 3 | Giới thiệu | Link |  | Click | Redirect to Introduction page |  |
| 4 | Liên hệ | Link |  | Click | Redirect to Contact page |  |
| 5 | FAQ | Link |  | Click | Redirect to FAQ page |  |
| 6 | Đăng nhập | Link |  | Click | Show login modal |  |
| 7 | Đăng ký | Link |  | Click | Show registration modal |  |
| 8 | Đăng ký phòng | Buttom |  | Click | Show booking room modal |  |
| 9 | Đăng phòng | Button |  | Click | Show posting room modal |  |
| 10 | Tìm kiếm | Button |  | Click | Search with the specified range (by select bán kính) |  |

1.2 Search Accommodation



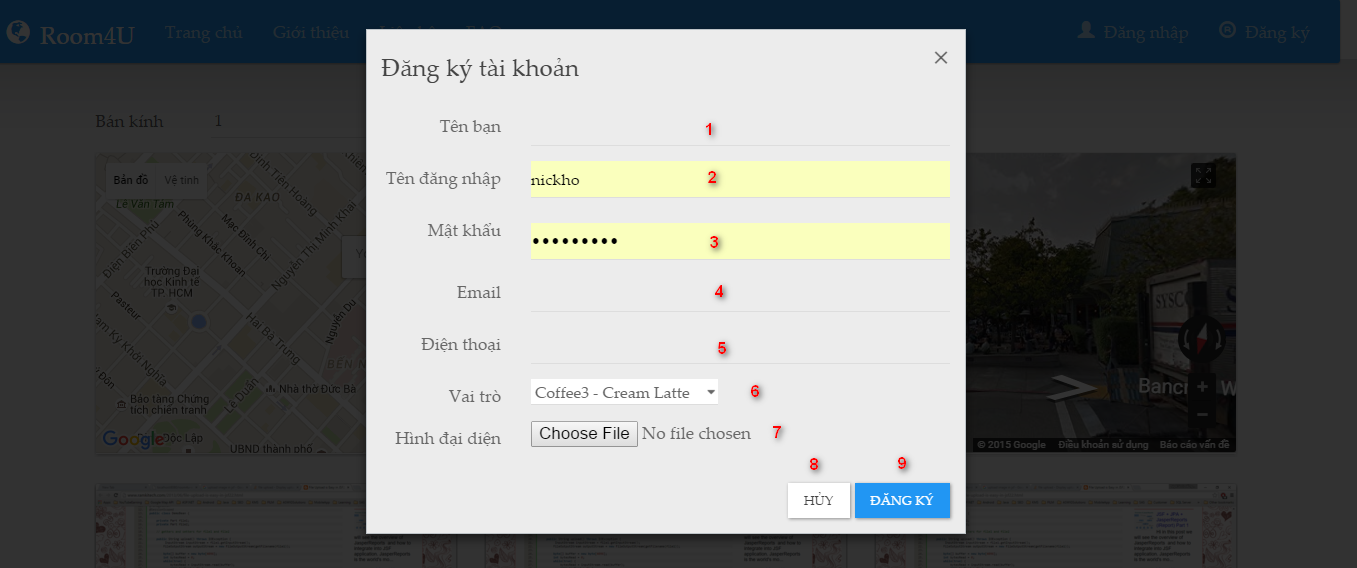
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Validation** | **Event** | **Description** | **Status** |
| 1 | Bán kính | Combo box |  | Click | Select a search range by km |  |
| 2 | Tìm kiếm | Button |  | Click | Start searching with the specified search range |  |
| 3 | Map | Map |  |  | Show the map with the current position and mark each search result with marker |  |
| 4 | Marker |  |  |  | Mark the destination with a marker sign |  |
| 5 | Street view | Street view |  |  | Click < and > sign to turn the view, to take a look around |  |
| 6 | List of search result | Grid view |  |  | Redirect to Contact page |  |

1.3 Login Page



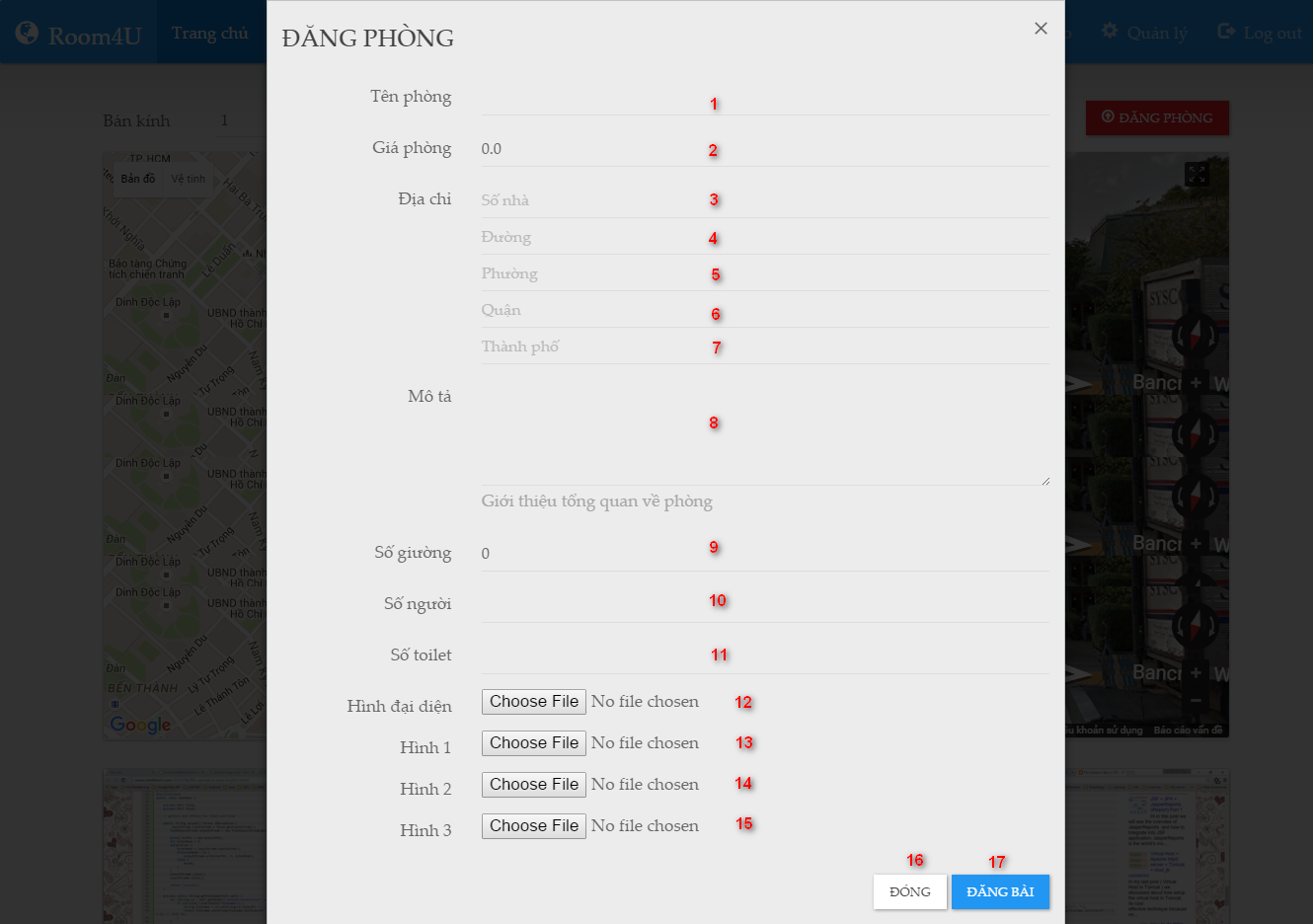
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Validation** | **Event** | **Description** | **Status** |
| 1 | Tài khoản | Textbox | Empty, >3, <25 |  | Input login name |  |
| 2 | Mật khẩu | Textbox | Empty, >3, <20 |  | Input password |  |
| 3 | Đóng | Button |  | Click | Close the modal, cancel login process |  |
| 4 | Đăng nhập | Button |  | Click | Submit the login form |  |

1.4 Register Page



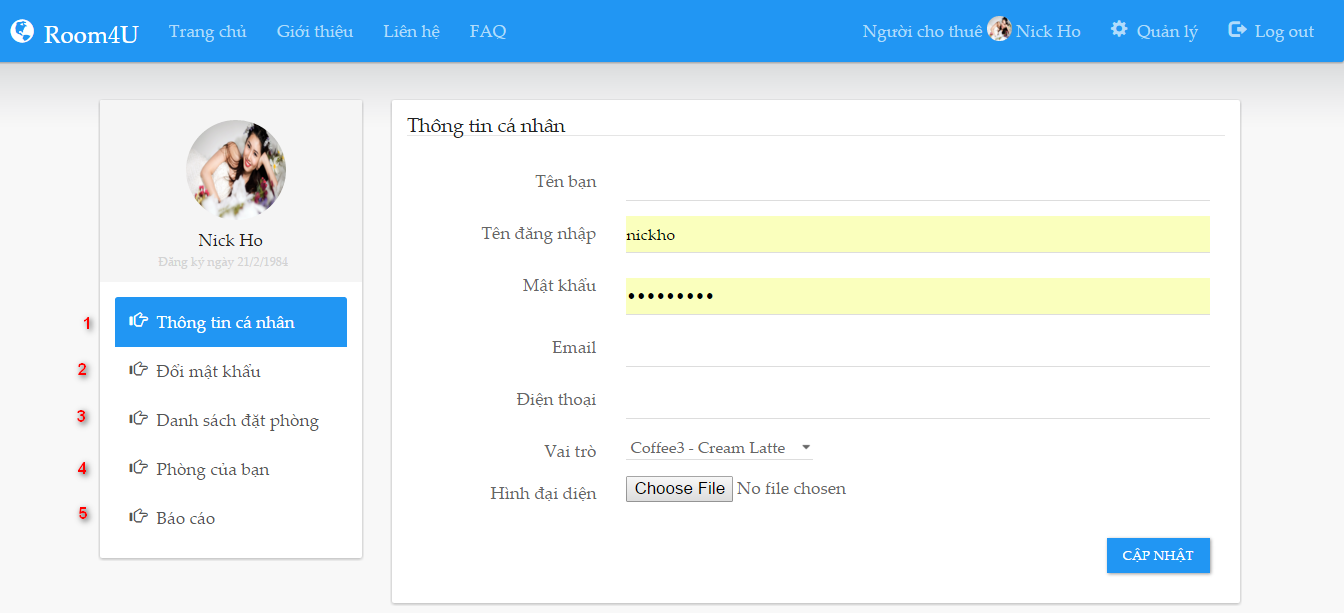
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Validation** | **Event** | **Description** | **Status** |
| 1 | Tên bạn | Textbox | Empty, character |  | Input account name |  |
| 2 | Tên đăng nhập | Textbox | Empty, character |  | Input login name |  |
| 3 | Mật khẩu | Textbox | Empty, >=3, <=20 |  | Input Login password |  |
| 4 | Email | Textbox | Empty, email |  | Input email address |  |
| 5 | Điện thoại | Textbox | Empty, >=8, <=12 |  | Input phone number |  |
| 6 | Vai trò | Combo box | Select role: Landlord or Customer | Click | Select role: Landlord or Customer |  |
| 7 | Hình đại diện | Button |  | Click | Choose a photo to upload |  |
| 8 | Đóng | Button |  | Click | Cancel registration process |  |
| 9 | Đăng ký | Button |  | Click | Submit the registration form |  |

1.5 Publish Accommodation



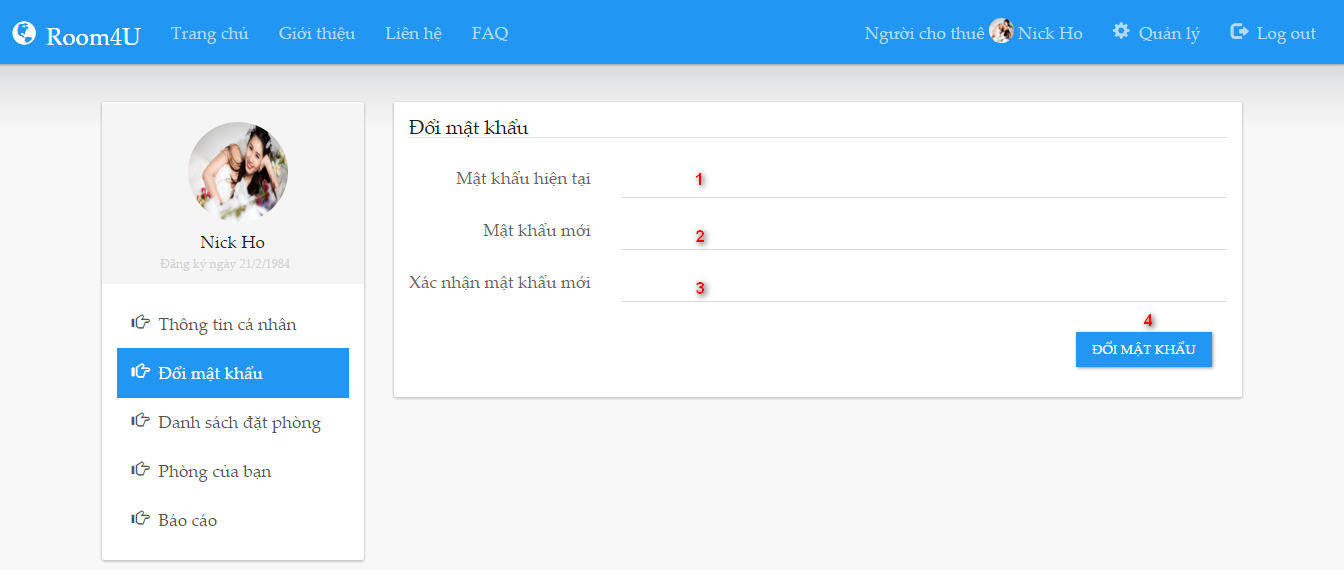
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Validation** | **Event** | **Description** | **Status** |
| 1 | Tên phòng | Textbox | Empty |  | Input room name |  |
| 2 | Giá phòng | Textbox | Empty, number, >0 |  | Input price |  |
| 3 | Số nhà | Textbox | Empty, number, >0 |  | Input address |  |
| 4 | Đường | Textbox | Empty |  | Input street |  |
| 5 | Phường | Textbox | Empty |  | Input ward |  |
| 6 | Quận | Textbox | Empty |  | Input district |  |
| 7 | Thành phố | Textbox | Empty |  | Input city |  |
| 8 | Mô tả | Textarea | Empty |  | Input room’s description |  |
| 9 | Số Giường | Textbox | Empty, >0, <50 |  | Input number of bed |  |
| 10 | Số người | Textbox | Empty, >0, <50 |  | Input number of person |  |
| 11 | Số Toilet | Textbox | Empty, >0, <50 |  | Input number of toilet |  |
| 12 | Hình đại diện | Button | \*.gif, \*.jpg | Click | Photo |  |
| 13 | Hình 1 | Button | \*.gif, \*.jpg | Click | Photo |  |
| 14 | Hình 2 | Button | \*.gif, \*.jpg | Click | Photo |  |
| 15 | Hình 3 | Button | \*.gif, \*.jpg | Click | Photo |  |
| 16 | Đóng | Button |  | Click | Cancel posting process |  |
| 17 | Đăng Bài | Button |  | Click | Submit the room posting form |  |

1.6 Account Profile



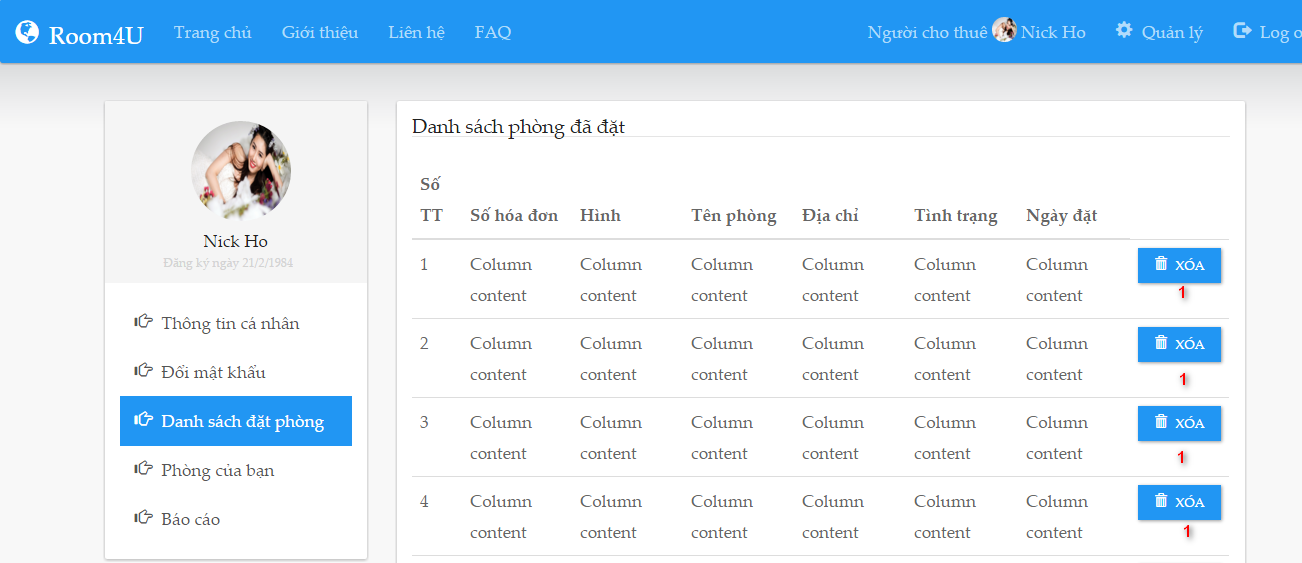
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Validation** | **Event** | **Description** | **Status** |
| 1 | Thông tin cá nhân | Link |  | Click | Display personal information |  |
| 2 | Đổi mật khẩu | Link |  | Click | Show change password modal |  |
| 3 | Danh sách đặt phòng | Link |  | Click | Show the list of booking room |  |
| 4 | Phòng của bạn | Link |  | Click | Show user’s own room |  |
| 5 | Báo cáo | Link |  | Click | Show Landlord’s report |  |

1.6.1 Change password



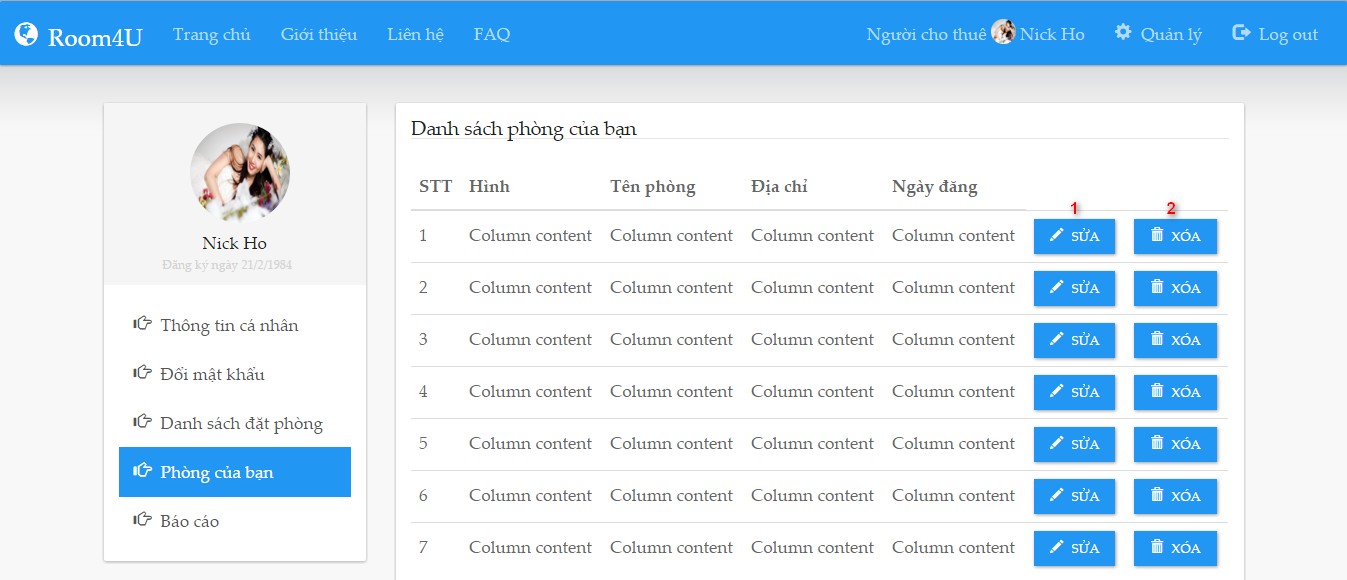
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Validation** | **Event** | **Description** | **Status** |
| 1 | Mật khẩu hiện tại | Textbox | Empty |  | Input current password |  |
| 2 | Mật khẩu mới | Textbox | Empty, >3, <50 |  | Input new password |  |
| 3 | Xác nhận mật khẩu mới | Textbox | Empty, >3, <50 |  | Input confirm password |  |
| 4 | Đổi mật khẩu | Button |  | Click | Submit the change password form |  |

1.6.2 List of booking room



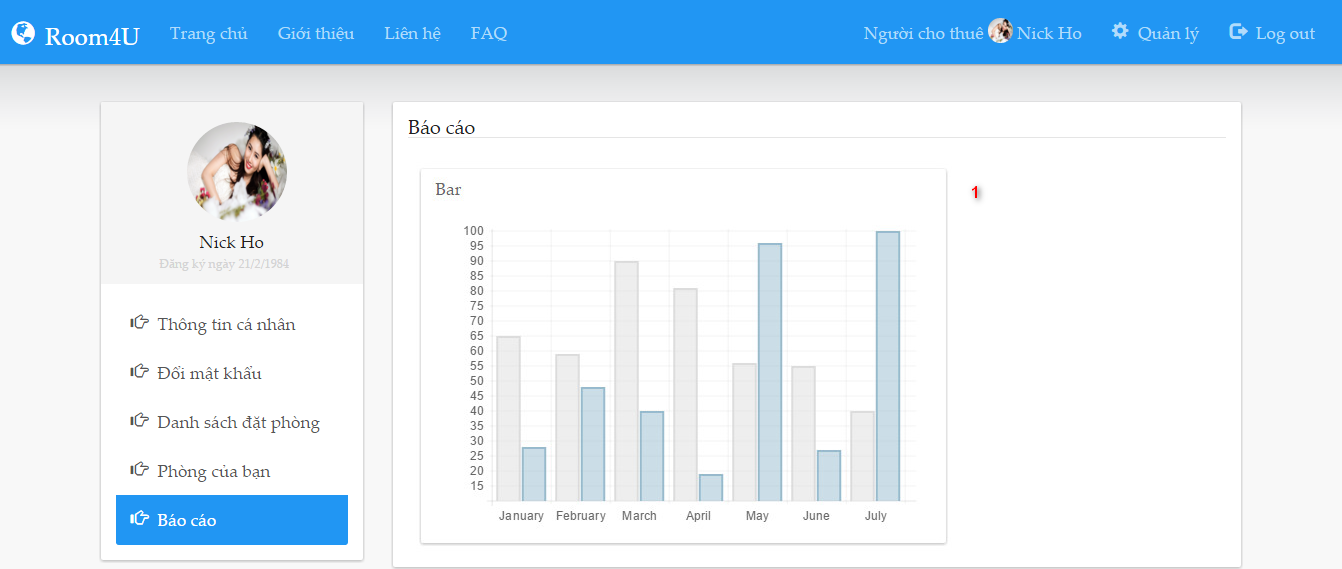
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Validation** | **Event** | **Description** | **Status** |
| 1 | Xóa | Button |  |  | Delete record on specified row |  |

1.6.3 User’s own room



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Validation** | **Event** | **Description** | **Status** |
| 1 | Sửa | Button |  |  | Edit record on specified row |  |
| 2 | Xóa | Button |  |  | Delete record on specified row |  |

1.6.4 Landlord’s Report

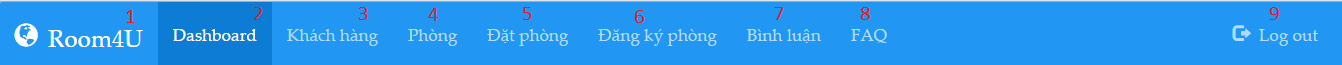


|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Validation** | **Event** | **Description** | **Status** |
| 1 | Báo cáo | Image |  |  |  |  |

1. Admin Side

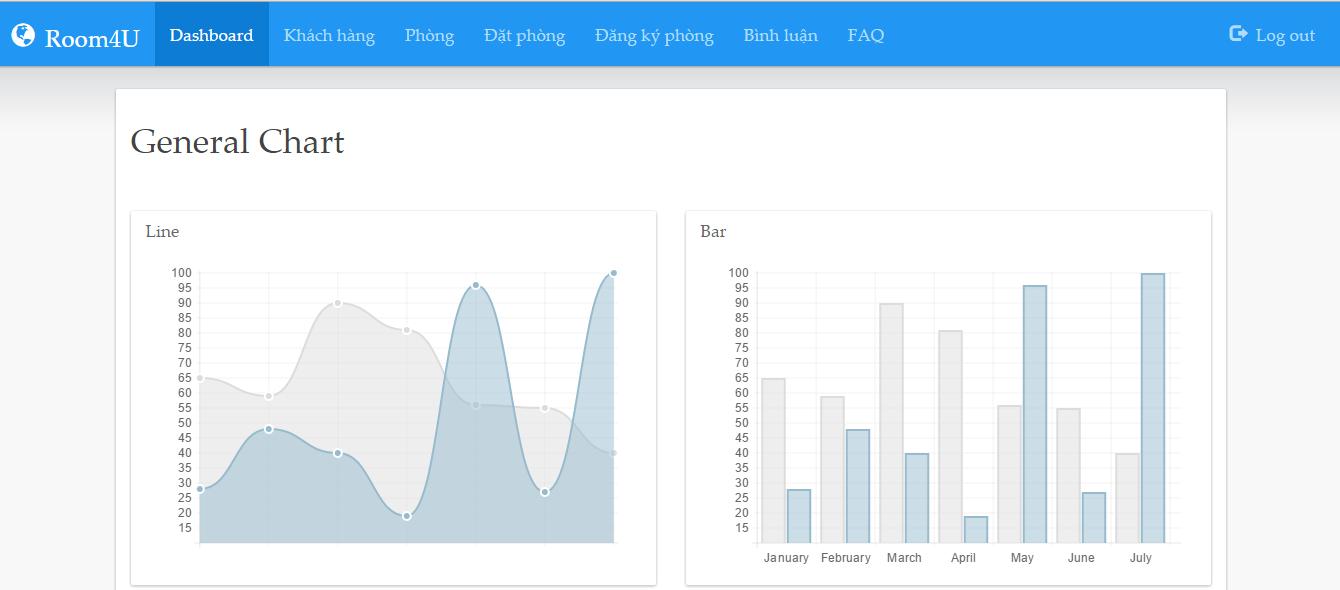
**3.1 After user login with admin account. Click link ‘Quản lý’ to go to admin page.**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Validation** | **Event** | **Description** | **Status** |
| 1 | btnAdmin | commandLink |  | Click | Redirect to admin page |  |

**3.2 Admin page’s menu look like this**

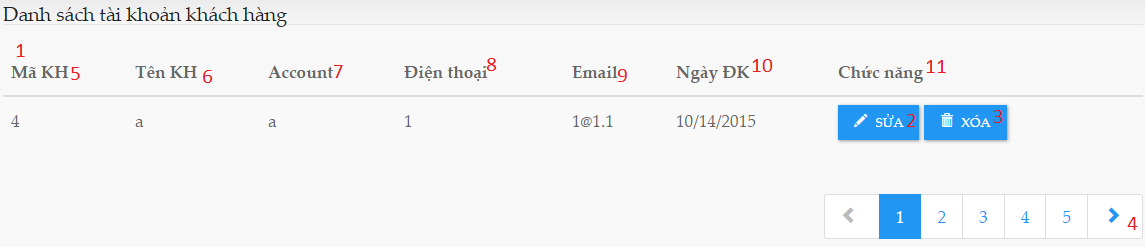
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Validation** | **Event** | **Description** | **Status** |
| 1 | btnIndex | Hyperlink |  | Click | Redirect to Home page |  |
| 2 | btnDashboard | Hyperlink |  | Click | Show summary, statistic report |  |
| 3 | btnCustomer | Hyperlink |  | Click | Redirect to List of customer management page |  |
| 4 | btnRoom | Hyperlink |  | Click | Redirect to List of room management page |  |
| 5 | btnBookRoom | Hyperlink |  | Click | Redirect to List of book room management page |  |
| 6 | btnPostRoom | Hyperlink |  | Click | Redirect to List of post room management page |  |
| 7 | btnComment | Hyperlink |  | Click | Redirect to List of comment management page |  |
| 8 | btnFAQ | Hyperlink |  | Click | Redirect to List of FAQ management page |  |
| 9 | btnLogout | Hyperlink |  | Click | Processing log out, clear session, and redirect to Home page |  |

3.3 Dashboard



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Validation** | **Event** | **Description** | **Status** |
| 1 | Dashboard | canvas |  |  | Statistic information |  |

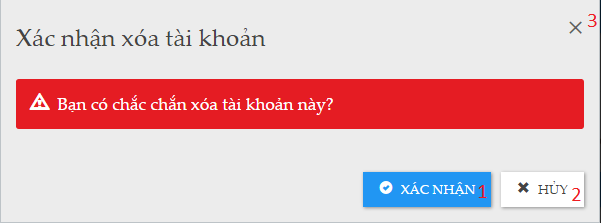
3.4 Customer



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Validation** | **Event** | **Description** | **Status** |
| 1 | dbtCust | Datatable |  |  | Datatalbe load database for customer |  |
| 2 | btnEdit | button |  | Click | Load edit customer form |  |
| 3 | btnDelete | button |  | Click | Delete customer |  |
| 4 | ulCust | unorderedlist |  | Click | View next or previous list customer |  |
| 5 | lblCustID | label |  |  | Customer ID |  |
| 6 | lblCustName | label |  |  | Customer Name |  |
| 7 | lblAccount | label |  |  | Customer Account |  |
| 8 | lblPhone | label |  |  | Customer phone |  |
| 9 | lblEmail | label |  |  | Customer email |  |
| 10 | lblDate | label |  |  | Customer register date |  |
| 11 | lblfunction | label |  |  | Contains function control: 2,3 |  |



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Validation** | **Event** | **Description** | **Status** |
| 1 | txtCustName | inputText |  |  | Customer Name |  |
| 2 | txtPhone | inputText |  |  | Customer phone |  |
| 3 | txtRegisterDate | inputText |  |  | Customer register date |  |
| 4 | txtCustRole | inputText |  |  | Customer role |  |
| 5 | btnUpdate | button |  | Click | Update database and close form |  |
| 6 | btnClose | button |  | Click | Close form |  |

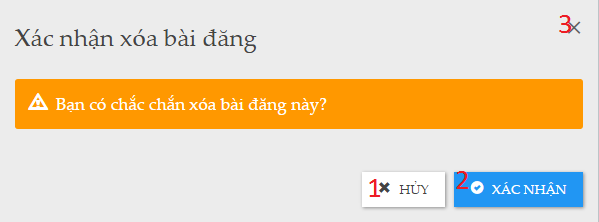


|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Validation** | **Event** | **Description** | **Status** |
| 1 | btnAccept | button |  | Click | Accept to delete customer |  |
| 2 | btnCancel | Button |  | Click | Cancel delete |  |
| 3 | btnClose | Button |  | Click | Close form |  |

3.5 Accommodation

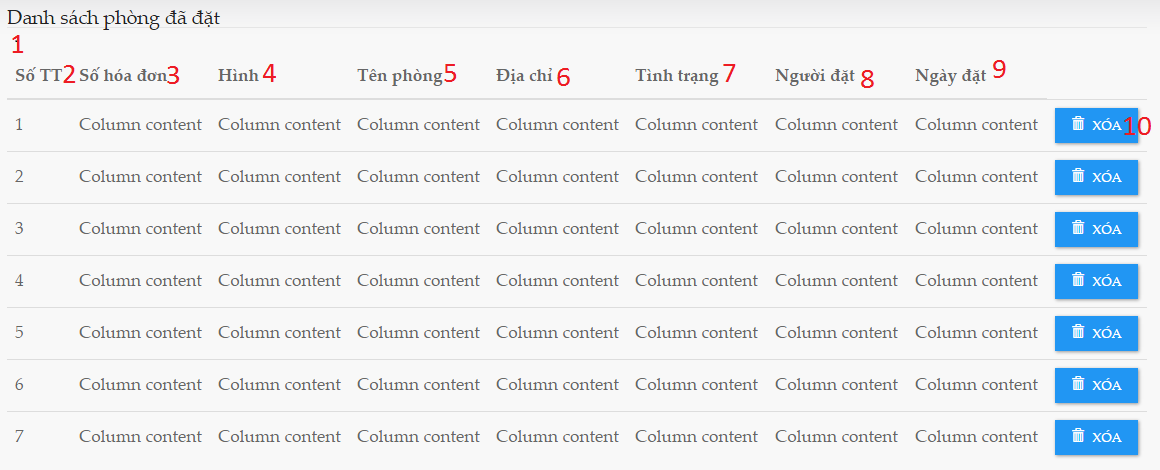


|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Validation** | **Event** | **Description** | **Status** |
| 1 | dbtRoom | Table |  |  | Table load database form accommodation |  |
| 2 | lblID | label |  |  | Room ID |  |
| 3 | lblImages | label |  |  | Images of room |  |
| 4 | lblRoomName | label |  |  | Room name |  |
| 5 | lblAddress | label |  |  | Room address |  |
| 6 | lblUsername | label |  |  | User post room |  |
| 7 | lblDatePost | label |  |  | Date post room |  |
| 8 | btnDeleteRoom | button |  | Click | Delete room |  |

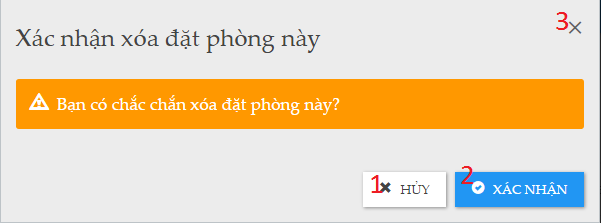


|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Validation** | **Event** | **Description** | **Status** |
| 1 | btnAccept | button |  | Click | Accept to delete accommodation |  |
| 2 | btnCancel | Button |  | Click | Cancel delete |  |
| 3 | btnClose | Button |  | Click | Close form |  |

3.6 Booking

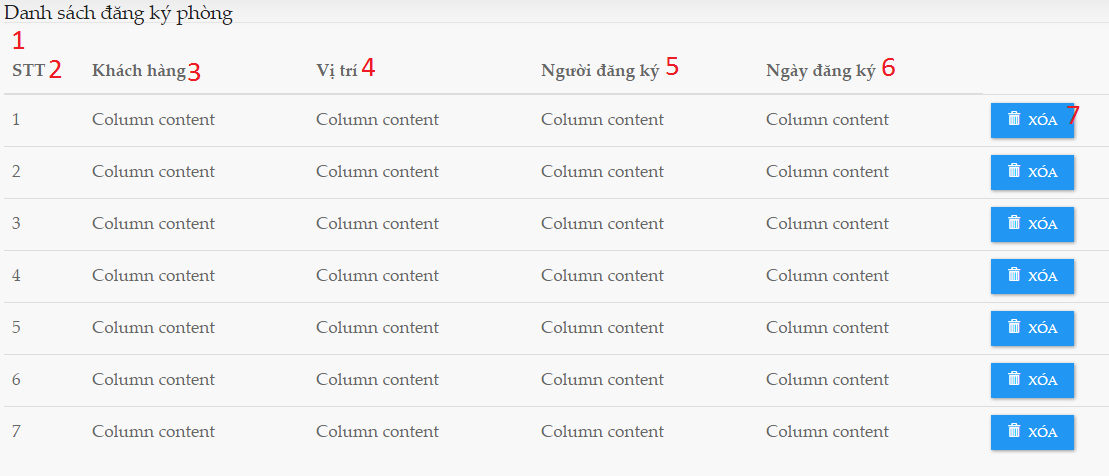


|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Validation** | **Event** | **Description** | **Status** |
| 1 | dbtBookRoom | Table |  |  | Table load database form order |  |
| 2 | lblNo | label |  |  | Number of book |  |
| 3 | lblID | label |  |  | Book ID |  |
| 4 | lblImages | label |  |  | Images of room |  |
| 5 | lblRoomName | label |  |  | Room name |  |
| 6 | lblAddress | label |  |  | Room address |  |
| 7 | lblStatus | label |  |  | Status of room |  |
| 8 | lblUsername | label |  |  | User book room |  |
| 9 | lblDateBook | label |  |  | Date book room |  |
| 10 | btnDeleteBook | button |  | Click | Delete book |  |

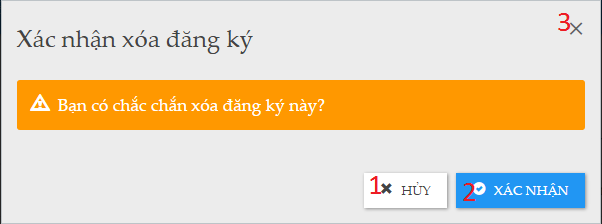


|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Validation** | **Event** | **Description** | **Status** |
| 1 | btnAccept | button |  | Click | Accept to delete Order |  |
| 2 | btnCancel | Button |  | Click | Cancel delete |  |
| 3 | btnClose | Button |  | Click | Close form |  |

3.7 Booking - Waiting List

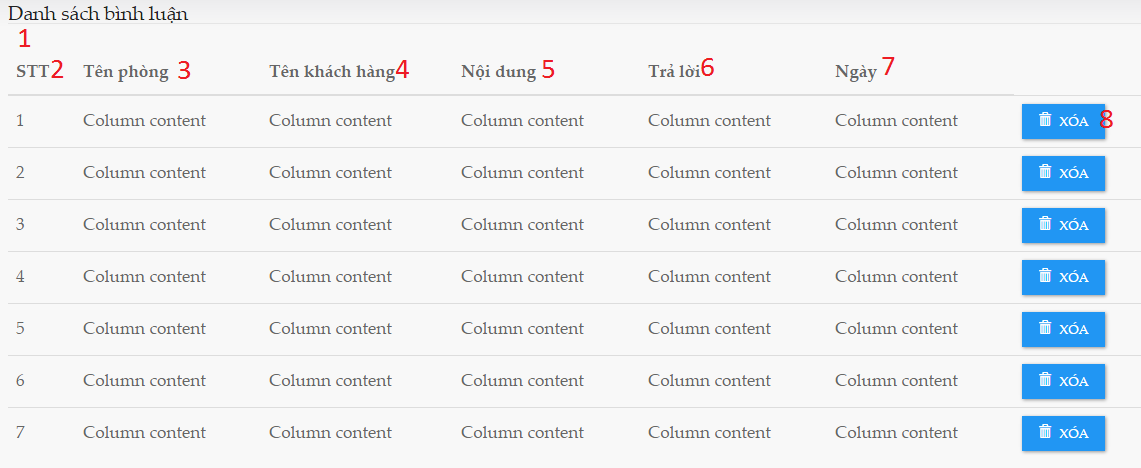


|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Validation** | **Event** | **Description** | **Status** |
| 1 | dbtRequireRoom | Table |  |  | Table load database from CustomerRequire |  |
| 2 | lblNo | label |  |  | Number of book require |  |
| 3 | lblCustName | label |  |  | Customer require |  |
| 4 | lblLocation | label |  |  | Location require |  |
| 5 | lblUsername | label |  |  | User require room |  |
| 6 | lblDateRequire | label |  |  | Date require room |  |
| 7 | btnDeleteRequire | button |  | Click | Delete require |  |

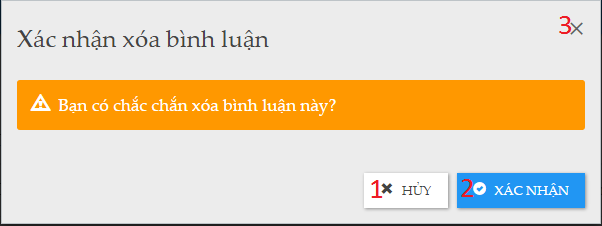


|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Validation** | **Event** | **Description** | **Status** |
| 1 | btnAccept | button |  | Click | Accept to delete CustomerRequire |  |
| 2 | btnCancel | Button |  | Click | Cancel delete |  |
| 3 | btnClose | Button |  | Click | Close form |  |

3.8 Comment

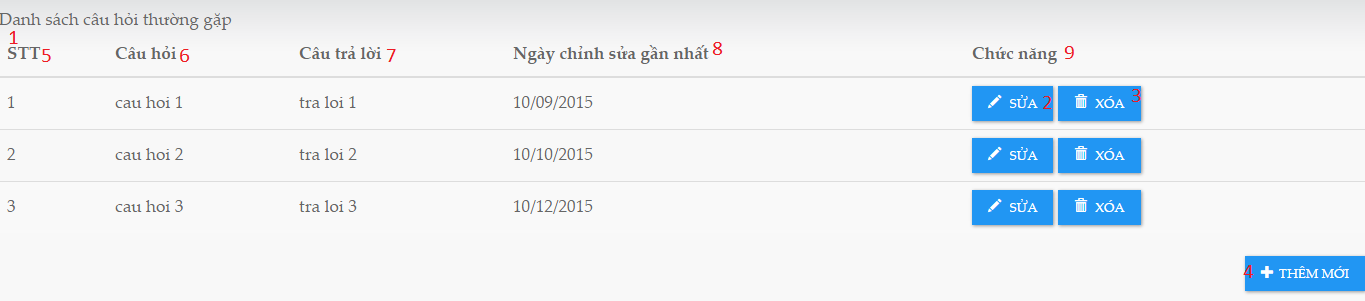


|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Validation** | **Event** | **Description** | **Status** |
| 1 | dbtComment | datatable |  |  | Datatable load database form Comment |  |
| 2 | lblNo | label |  |  | Number of comment |  |
| 3 | lblRoomName | label |  |  | Room name |  |
| 4 | lblUsername | label |  |  | Name of user comment |  |
| 5 | lblContent | label |  |  | Content |  |
| 6 | lblComment | label |  |  | Comment |  |
| 7 | lblDate | label |  |  | Date of comment |  |
| 8 | btnDeleteRequire | button |  | Click | Delete require |  |



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Validation** | **Event** | **Description** | **Status** |
| 1 | btnAccept | button |  | Click | Accept to delete Comment |  |
| 2 | btnCancel | Button |  | Click | Cancel delete |  |
| 3 | btnClose | Button |  | Click | Close form |  |

3.9 FAQ



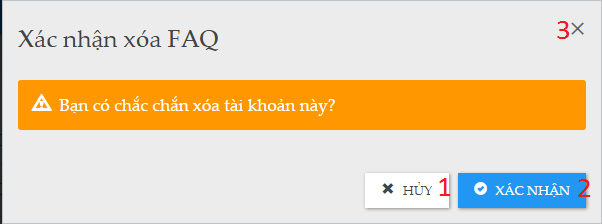
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Validation** | **Event** | **Description** | **Status** |
| 1 | dbtFAQ | Datatable |  |  | Datatalbe load database for FAQ |  |
| 2 | btnEdit | button |  | Click | Load edit FAQ form |  |
| 3 | btnDelete | button |  | Click | Delete FAQ |  |
| 4 | btnAddFAQ | btn |  | Click | Add new FAQ |  |
| 5 | lblFAQID | label |  |  | FAQ ID |  |
| 6 | lblQuestion | label |  |  | Question |  |
| 7 | lblAnswer | label |  |  | Answer for question |  |
| 8 | lblDate | label |  |  | Last modifier date |  |
| 9 | lblfuntion | label |  |  | Contains function control: 2,3 |  |



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Validation** | **Event** | **Description** | **Status** |
| 1 | txtQuestion | inputText | Not null |  | Question of FAQ |  |
| 2 | txtAnswer | inputText | Not null |  | Answer of the question |  |
| 3 | btnCancel | button |  | Click | Cancel add new FAQ |  |
| 4 | btnAdd | button |  | Click | Add new FAQ |  |
| 5 | btnClose | button |  | Click | Close form |  |



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Validation** | **Event** | **Description** | **Status** |
| 1 | txtQuestion | inputText | Not null |  | Question of FAQ |  |
| 2 | txtAnswer | inputText | Not null |  | Answer of the question |  |
| 3 | btnCancel | button |  | Click | Cancel update FAQ |  |
| 4 | btnAdd | button |  | Click | Update FAQ |  |
| 5 | btnClose | button |  | Click | Close form |  |



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Validation** | **Event** | **Description** | **Status** |
| 1 | btnCancel | button |  | Click | Cancel delete |  |
| 2 | btnAccept | Button |  | Click | Accept to delete FAQ |  |
| 3 | btnClose | Button |  | Click | Close form |  |

*--- The End of Review 3---*

|  |  |  |
| --- | --- | --- |
|  | **Prepared by (Student)** | **Approved by (Faculty)** |
| **Signature** | Team leader  **Ho Chau Can** | **Hoang Duc Quang** |
| **Date** |  |  |