Temasek Polytechnic

School of Informatics and IT

**Diploma in Information Technology (IT)**

Software Design Specifications (DS)

**Project Particulars**

|  |  |
| --- | --- |
| **Tutor** | Mr Mel Goh |
| **Class** | P02 |
| **Project Title** | Delonix Regia Hotel Management System |

**Project Team’s Particulars**

|  |  |
| --- | --- |
| **Matric Number** | **Student Name** |
| 1500564H | Shawn Yang |
| 1502728G | Xavier See |
| 1500340C | Yeo Kai Sheng |
| 1500858D | Yeo Ai Ling |

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 17/05/2017 | 1.0 | Distribution of Workload  Architecture Design | Shawn |
| 18/05/2017 | 1.1 | Architecture Design | Shawn |
| 18/05/2017 | 1.2 | Distribution of Workload | Xavier |
| 21/05/2017 | 1.3 | Database Design | Ai Ling |
|  |  |  |  |
| 21/05/2017 | 1.4 | Architecture Design | Shawn |
|  |  |  |  |
| 22/05/2017 | 1.5 | User Interface (UI) Design | Kai Sheng |
|  |  |  |  |
| 22/05/2017 | 1.6 | Program Design | Xavier |
|  |  |  |  |
| 23/05/2017 | 1.7 | Database Design | Ai Ling |
|  |  |  |  |
| 23/05/2017 | 1.8 | User Interface (UI) Design | Kai Sheng |
|  |  |  |  |
| 23/05/2017 | 1.9 | Program Design | Xavier |

**Table of Contents**

[1. DISTRIBUTION OF WORKLOAD 3](#_Toc483390438)

[2. ARCHITECTURE DESIGN 4](#_Toc483390439)

[2.1 Architecture Design: 3-Tier Architecture 5](#_Toc483390440)

[2.2 3-Tier Architecture vs 2-Tier Architecture 5](#_Toc483390441)

[2.3 Standalone Application Disadvantages 6](#_Toc483390442)

[2.4 N-Tier Architecture Disadvantages 6](#_Toc483390443)

[3. USER INTERFACE (UI) DESIGN 7](#_Toc483390444)

[3.1 Online Booking module 9](#_Toc483390445)

[3.1.1 Homepage 9](#_Toc483390446)

[3.2 Hotel Management System 12](#_Toc483390447)

[3.2.1 Login Page 12](#_Toc483390448)

[3.2.2 Room availability page 13](#_Toc483390449)

[3.2.3 Room list page 16](#_Toc483390450)

[3.2.4 Housekeeping & Staff management module 18](#_Toc483390451)

[3.2.5 Room status report page 20](#_Toc483390452)

[3.2.6 Listing all the guest in all the room page 22](#_Toc483390453)

[3.2.7 Room occupancy page 23](#_Toc483390454)

[3.2.8 Housekeeping report page 23](#_Toc483390455)

[3.2.9 User Account & Login Creation module 24](#_Toc483390456)

[4. PROGRAM DESIGN 28](#_Toc483390457)

[5. DATABASE DESIGN 66](#_Toc483390458)

# DISTRIBUTION OF WORKLOAD

*[Determine which members of the team will be responsible for which areas of design. Individual’s responsibilities should be clearly spelt out.]*

|  |  |
| --- | --- |
| **Design** | **Members** |
| Architecture Design | Shawn |
| Program Design | Xavier See |
| User Interface Design | Kai Sheng |
| Database Design | Ai Ling |

# 

# ARCHITECTURE DESIGN

*[Describe the proposed system architecture design e.g. 2-tier comprising of client-data? Or 3-tier comprising of client-business-data? Include diagram to illustrate the different tiers in the architecture.]*

## Architecture Design: 3-Tier Architecture

Three Tier Architecture is defined as a client-server architecture in which there is an established 2-way link between the client and server. The 3-Tier Architecture contains the functional logic of a computer system, data access, computer data storage and user interface that are maintained as separate components in different areas of a Desktop Computer system. The 3-Tier Architecture is a software development design pattern and a well-rounded software architecture.

Our project team proposes that Mr Wang’s Hotel brand new Desktop PC should adhere to the 3-Tier Architecture methodology as we firmly believe that this is the best architectural methodology because the 3-Tier Architecture methodology involves a web browser, web application on web server and a database on a database server. The web server and database server would be separate from the computer in the receptionist area as it would be located in the located inside the hotel in an administrative room for administrative authorised users only.

Our computer system located in the receptionist area would be connected to the internet via wireless broadband and would connect to the web server in the administrative room via the online Internet Information Services (IIS) web server. By connecting to IIS, Mr Wang would have full access to the web server and he would have the authority to make configuration changes to the web server when needed such as URL Authorization and IP and Domain Restrictions.

Our computer system in the receptionist area will also be connected to the database server located in an administrative authorised room via wireless broadband via a software application downloaded on the computer called MySQL Workbench database application. The MySQL Workbench would need to be connected to the internet for it to be connected to the database server where it would load data stored in the database server such as staff information and customer information.

## 2.2 3-Tier Architecture vs 2-Tier Architecture

The difference between 3-Tier Architecture and 2-Tier Architecture is that 3-Tier Architecture is web based while 2-Tier Architecture is client server architecture based which means that there is direct communication between the client and server. 2-Tier Architecture does not have an intermediate between a client and server. 2-Tier Architecture would become an inefficient architecture method once more users start to use the system and it is not cost effective.

Meanwhile 3-Tier Architecture is a cost effective architectural method as it is high in performance even when there are more users using the system. 3-Tier Architecture has a high degree of flexibility when used in various platforms and its configuration is also flexible making it easy to be incorporated into a system while 2-Tier Architecture is inflexible and unable to tolerate a high volume of users.

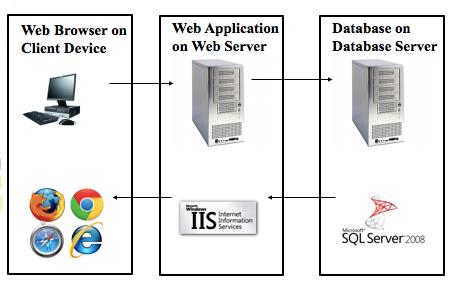
**Standalone Application & N-Tier Architecture disadvantages and why they should not be used**

## 2.3 Standalone Application Disadvantages

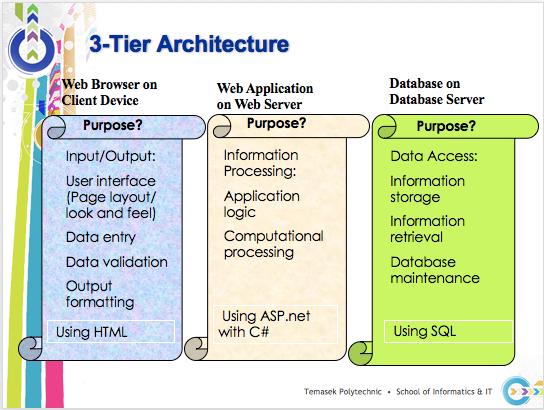
Standalone computers have disadvantages such as users are constricted to only a single computer, user files are not safe and secured as they can be accessed from any connected computer on the same virtual network. A standalone computer also requires any new downloaded applications on other computers to be setup one by one and individually on the network. Therefore with all its drawbacks it was unsuitable and not proposed as the architectural method for the Delonix Regia Hotel Management System.

## 2.4 N-Tier Architecture Disadvantages

Our project group also did not consider or proposed N-Tier Architecture because because it Mr Wang’s Delonix Regia Hotel Management System would be unsuitable for this architectural method because N-Tier Architecture is a complex method, difficult to implement and maintain while Mr Wang’s an architecture that is easy to configure and maintain.



(3 Tier Architecture, L07b Software Design; Building Blocks)



(3 Tier Architecture, L07b Software Design; Building Blocks)

# 

# USER INTERFACE (UI) DESIGN

*[Document the user interface design decisions and considerations. This should not be a mere capture of all UI forms but a reflection of UI design considerations. Is web or windows or other forms of UI used? Why did you choose that particular UI? How does the UI design impact usability and why adopt certain UI controls than others?]*

The Delonix Regia Hotel management system will be a Windows UI design as it improves flexibility and the ability to multitask. Firstly, with a Windows Delonix Regia System, it provides the ability to manipulate easily and instinctively. The ways which allows to have easily manipulation is that it can scroll to any section of the window contents, and adjust to any size.

Additionally, it allows multiple windows to be opened simultaneously, improving the workflow and efficiency as it get lead to better referencing and comparison of reports with side-by-side windows.

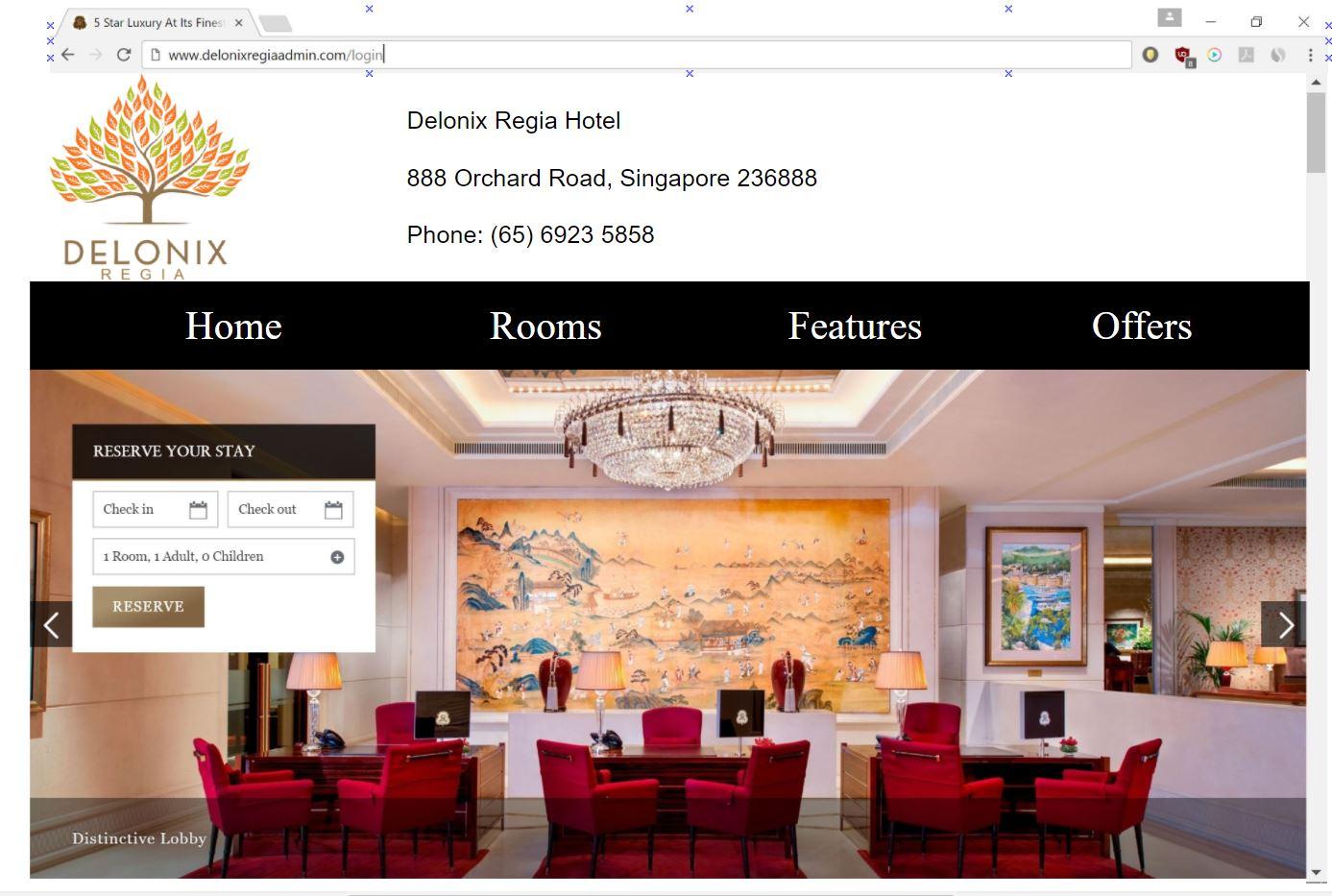
As such having this UI design benefits the 3 users of the system. Firstly, the reception staff will be able to compare the room availability and booking module together with the room status report side-by-side which will improve the efficiency as he or she can find any mistakes indicated in the report.

Secondly, the management users will benefit from a windows UI system as they are able to monitor all the 3 modules in one shot rather than looking inspecting it one at a time.

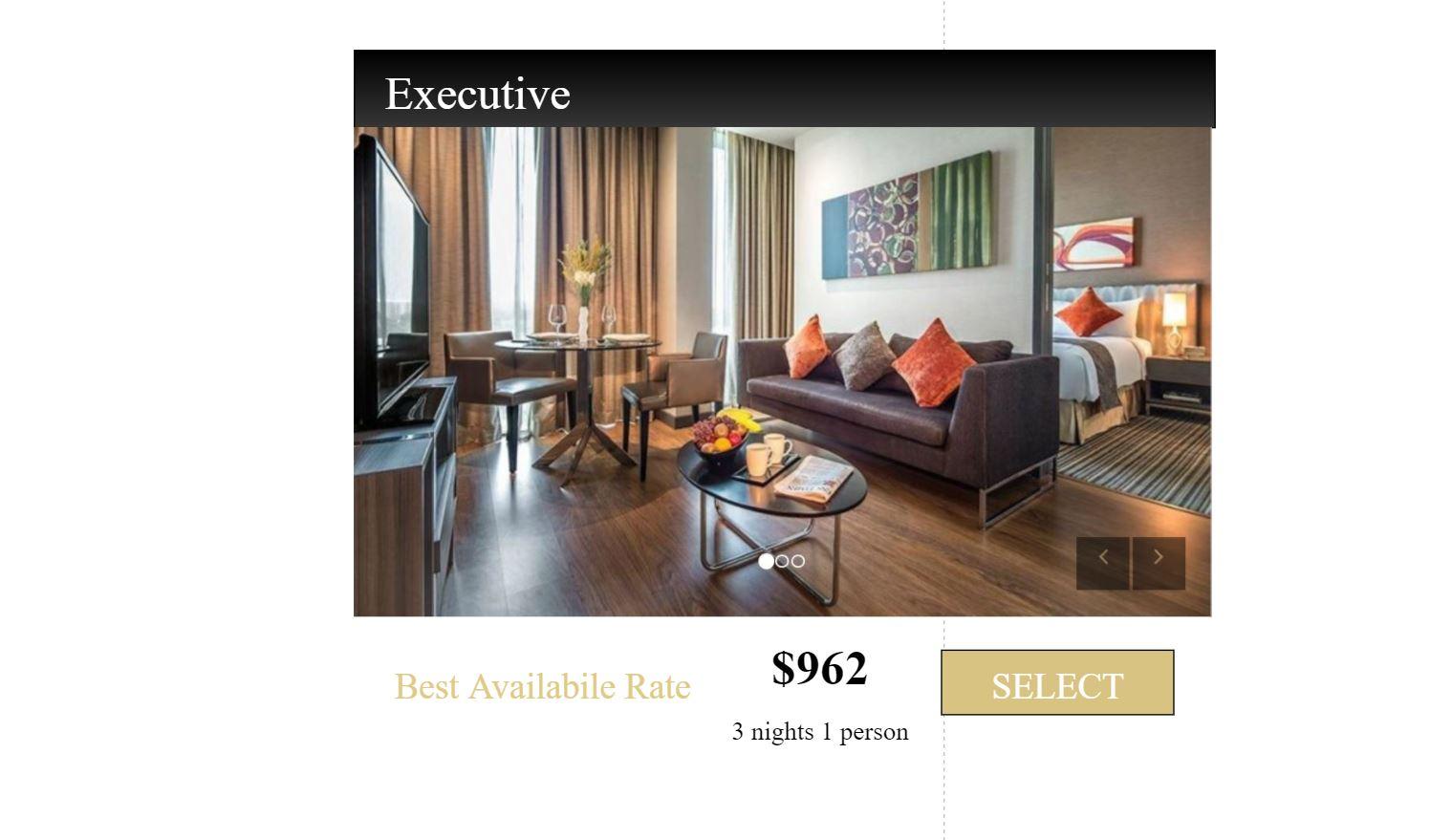
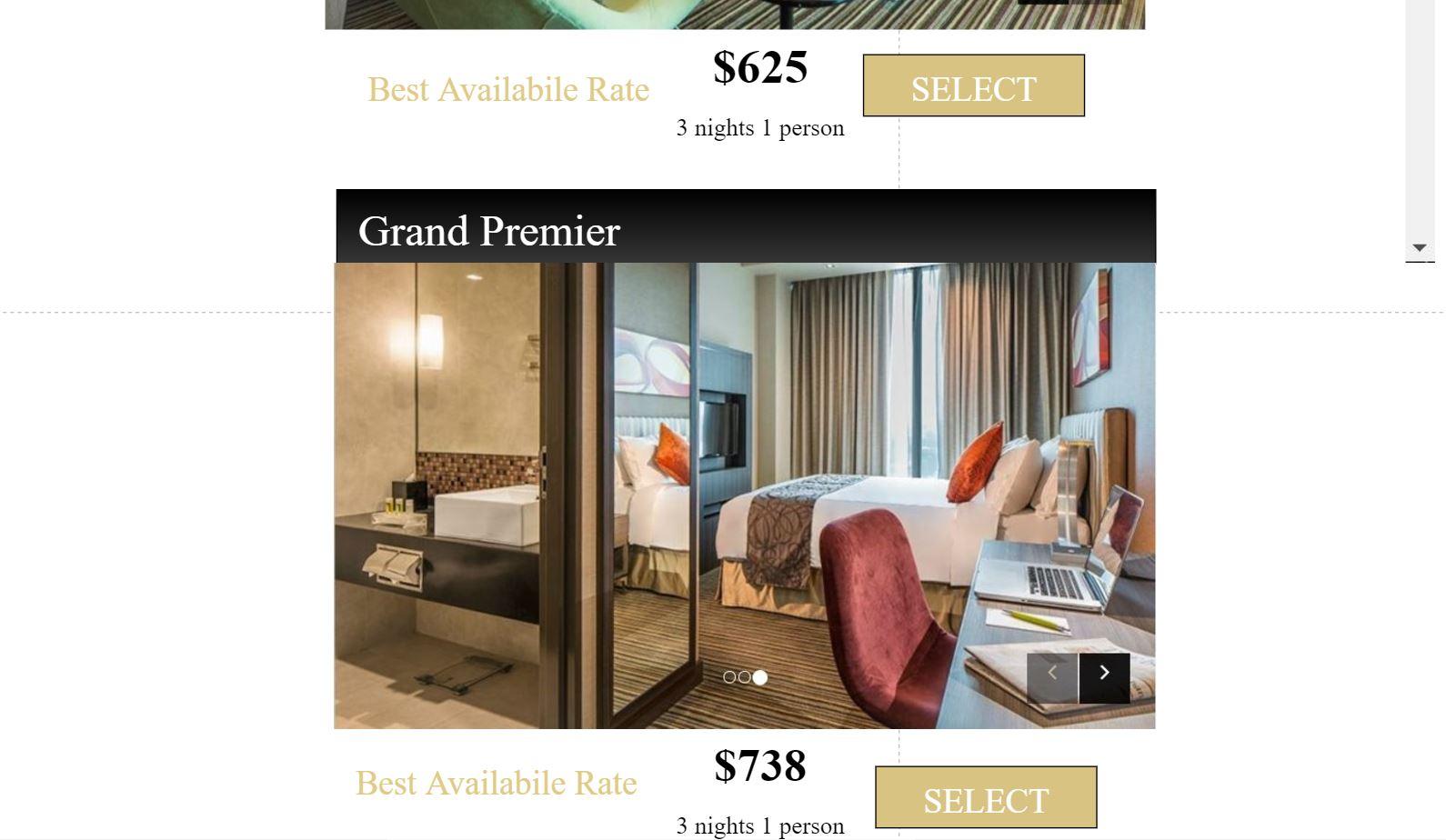
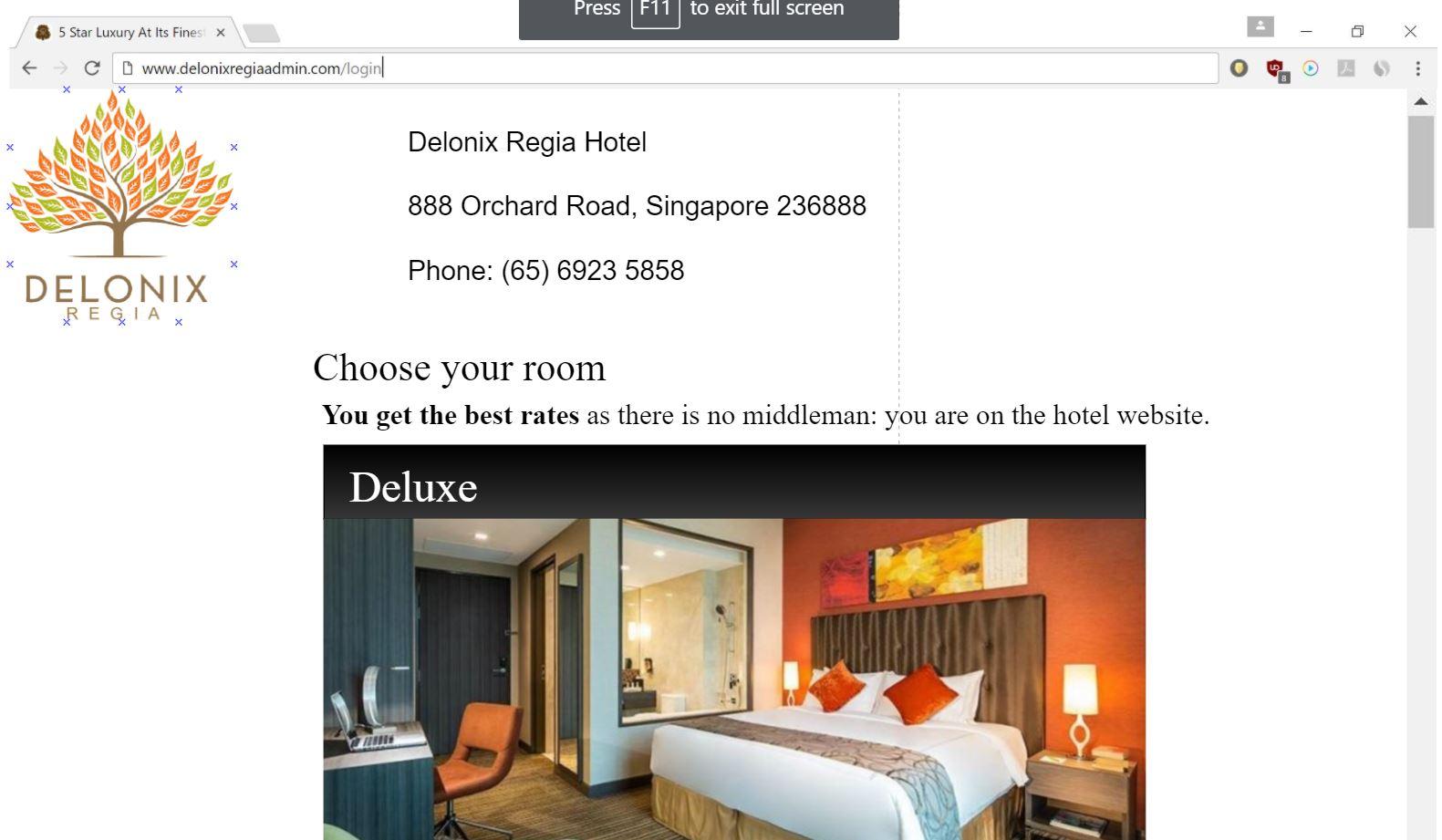
Lastly, similar to the management users, Mr Wang will also be able monitor and inspect all modules. This UI system also allows Mr Wang to view on reports efficiently.

## 3.1 Online Booking module

### 3.1.1 Homepage



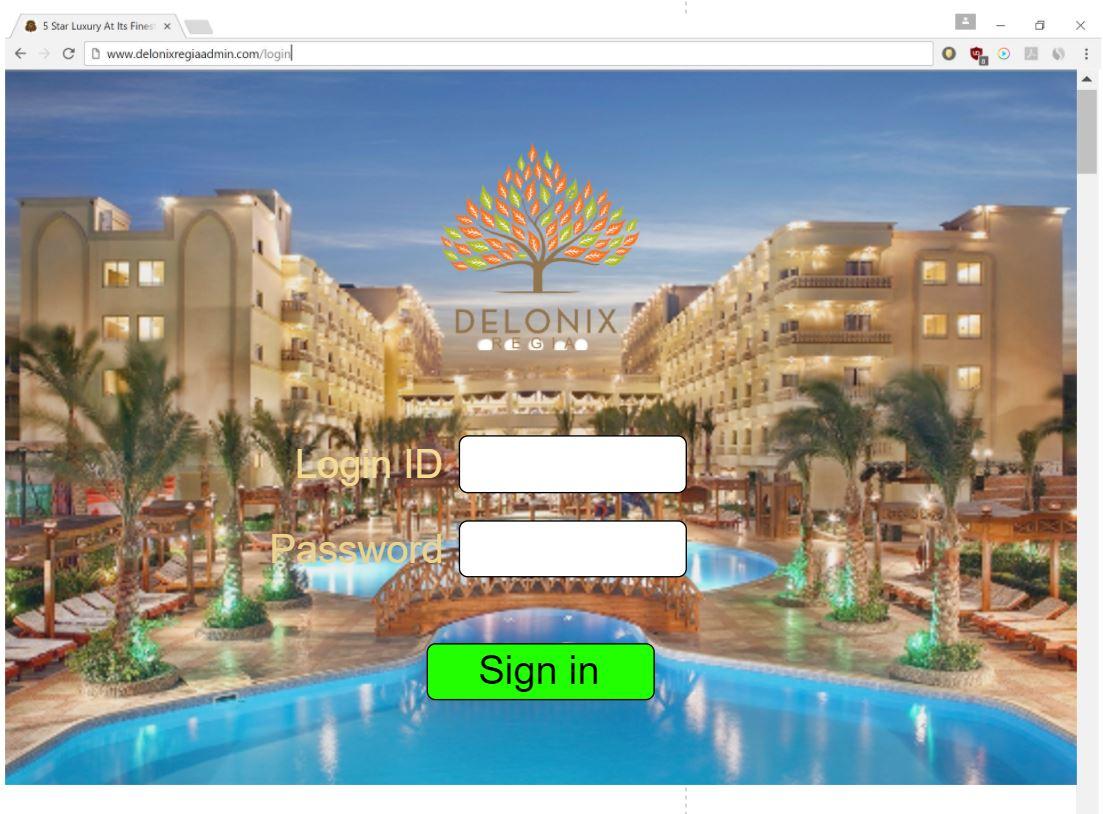
This will be a brief capture of Delonix Regia Hotel homepage which allows users to book a hotel room online as well as view on the various rooms, features, offers and promotion offered by Delonix Regia Hotel.



Afterwards, when a user has selected his check-in date, check-out date and occupancy, he or she will be prompted to the next page to select the 3 rooms catered by Delonix Regia Hotel. We also aim to provide the customer with ease by providing him or her the best available rate.

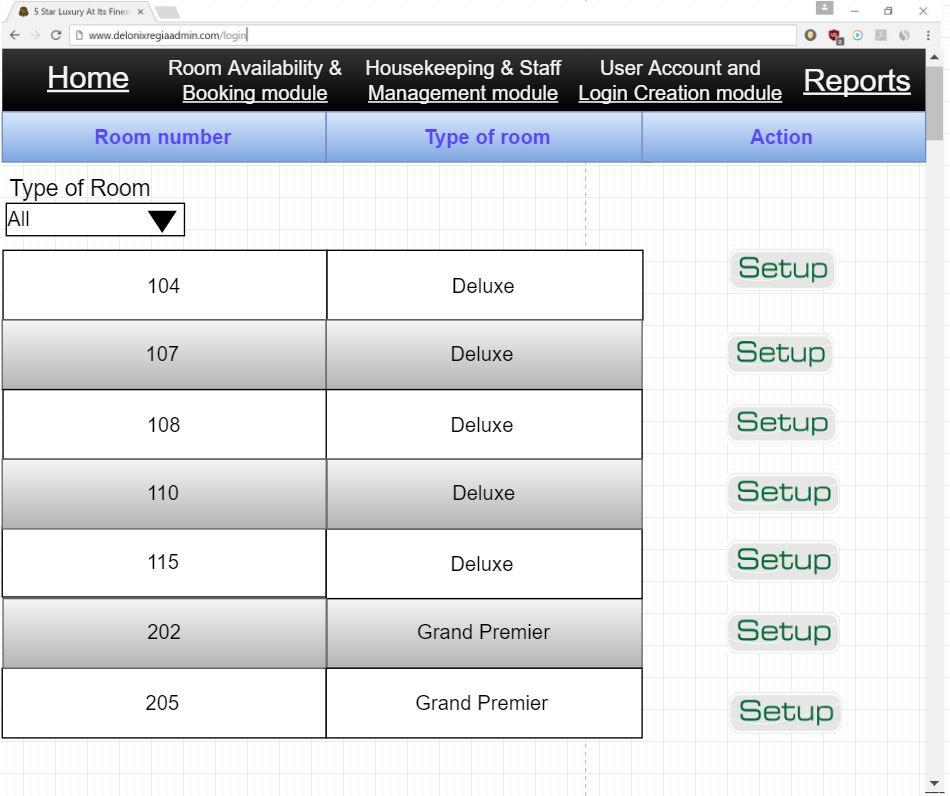
## 3.2 Hotel Management System

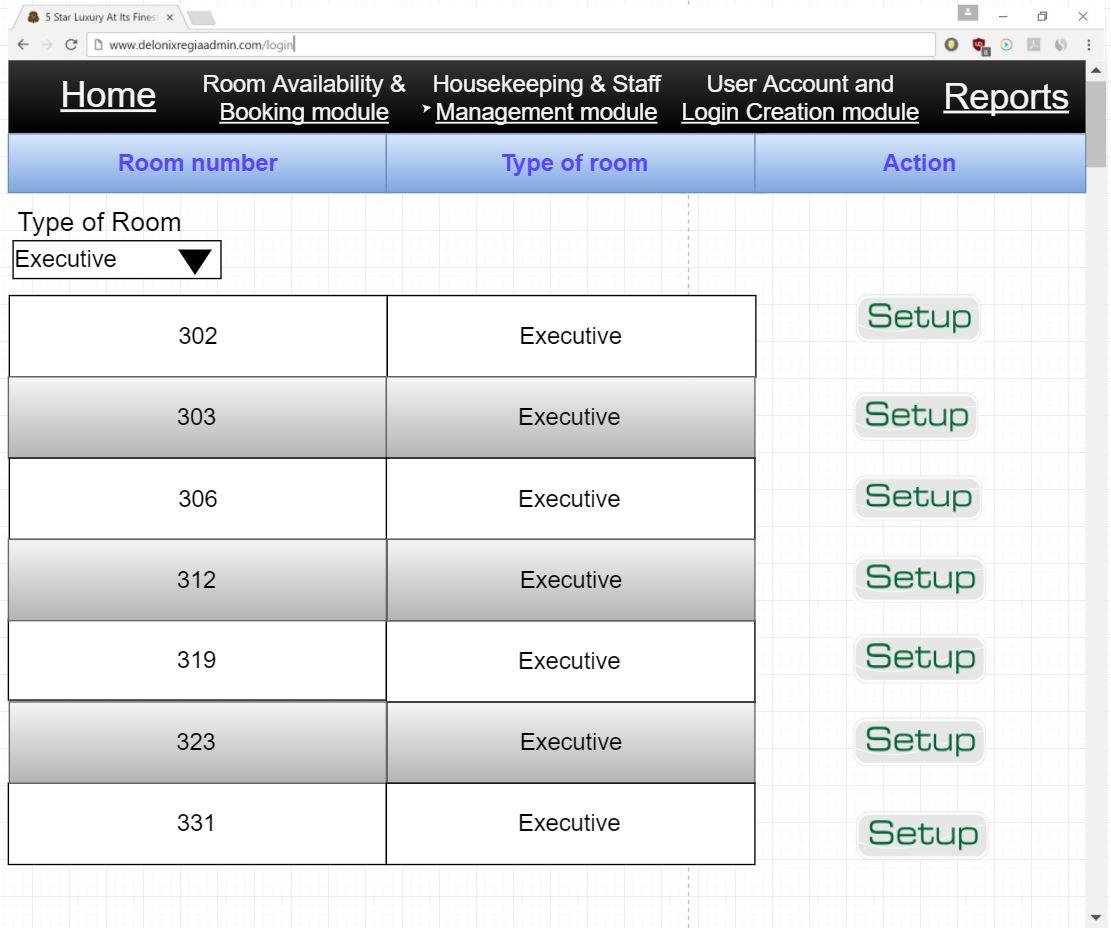
### 3.2.1 Login Page



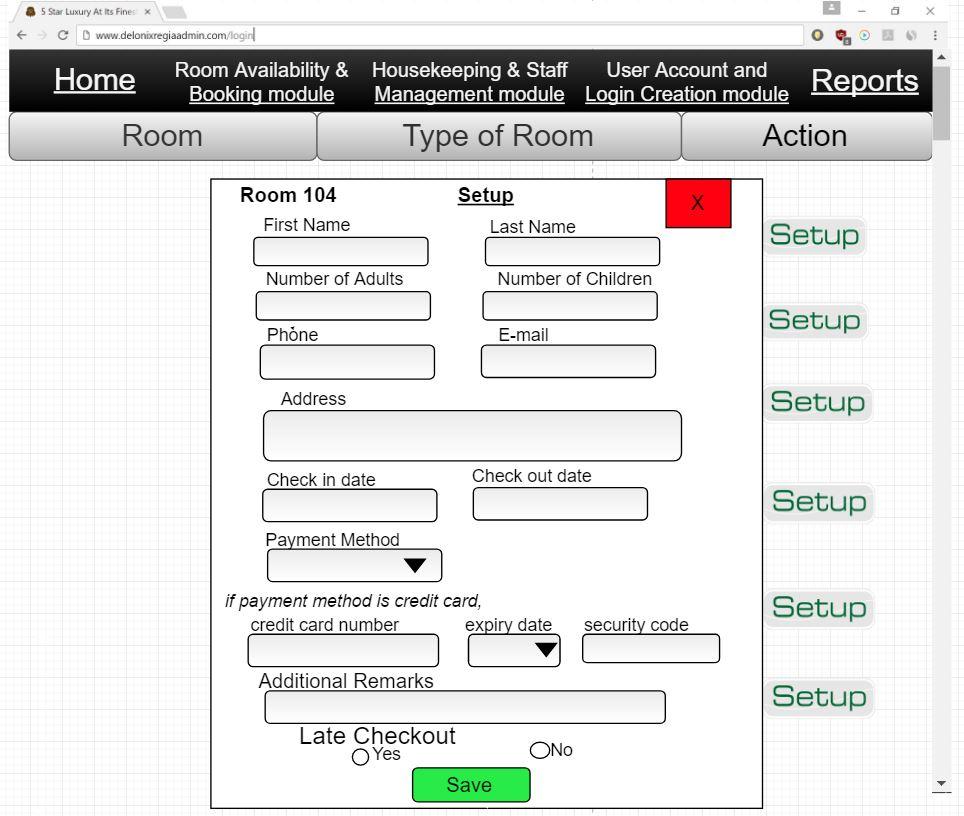
At the start of the hotel management system, the user will be prompted to the login screen where he or she must key in the username and password. For the password, it will be their staff id.

### 3.2.2 Room availability page



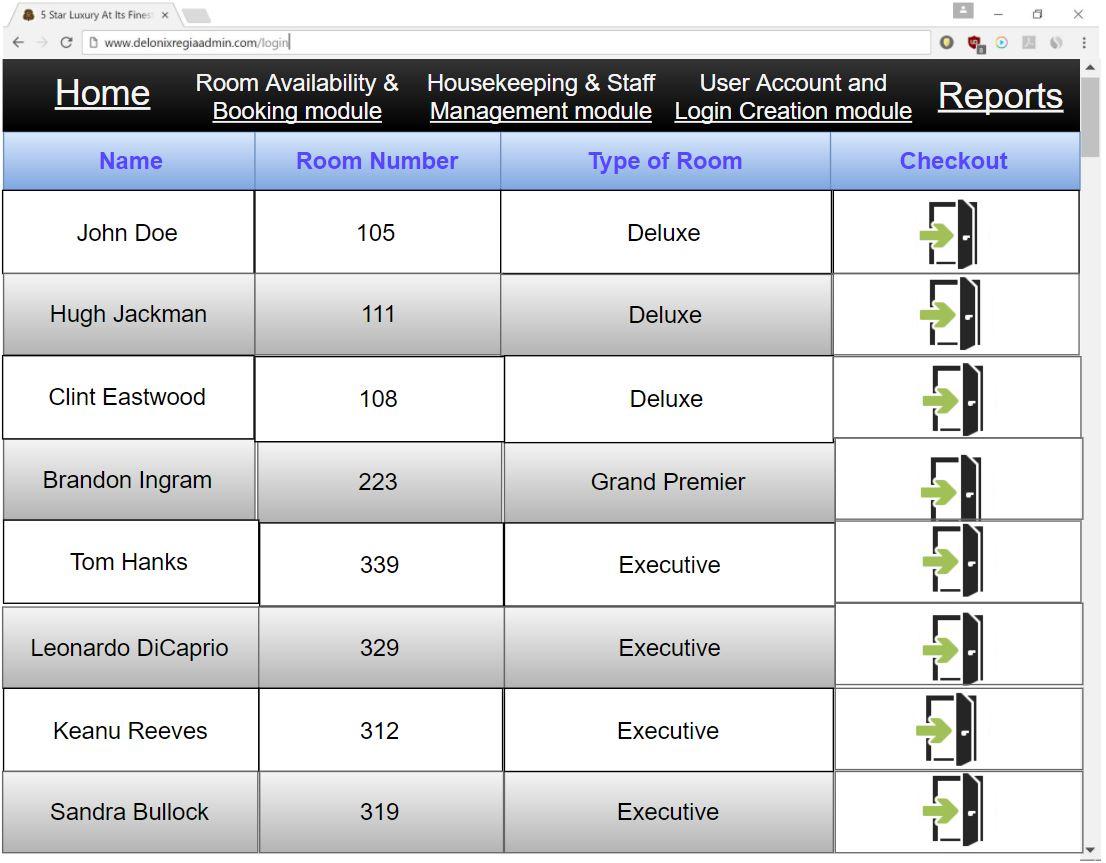


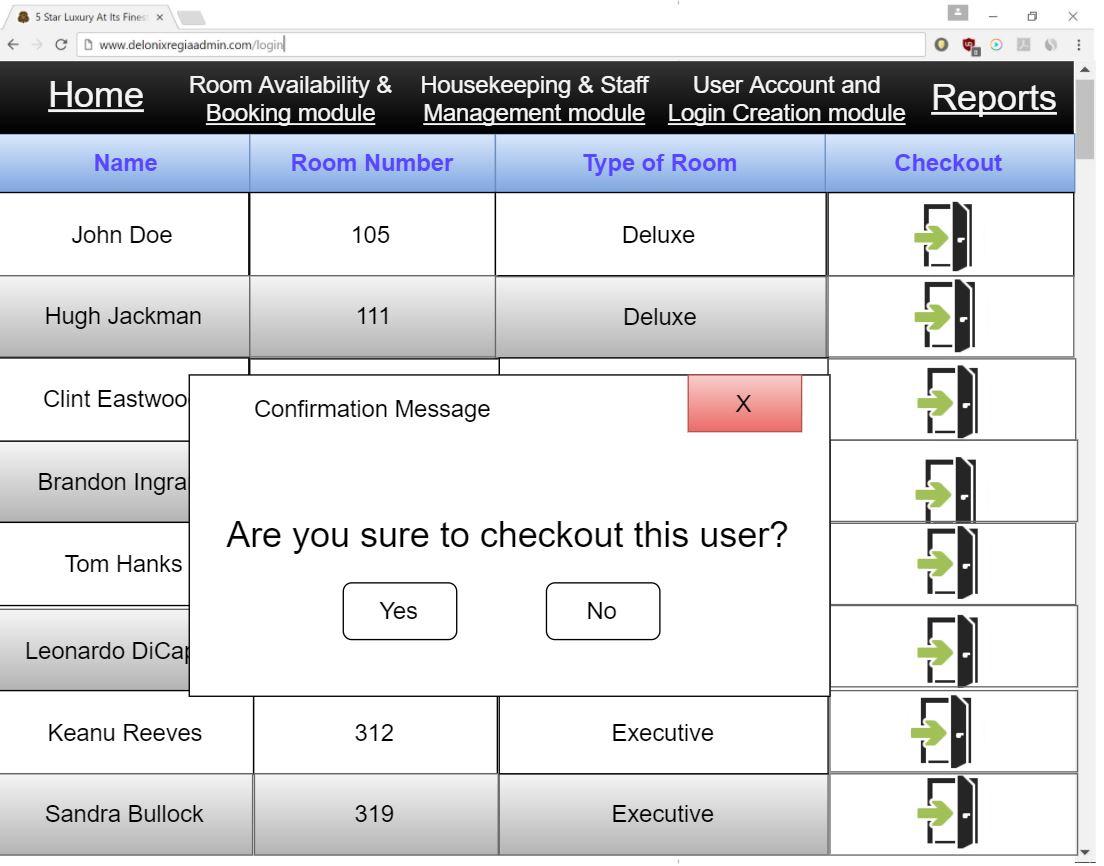
At the room availability page, the receptionist is able to see the list of rooms which are available and based on the room type. The receptionist can also filter by the type of rooms which they want using the the drop-down box on the top left.

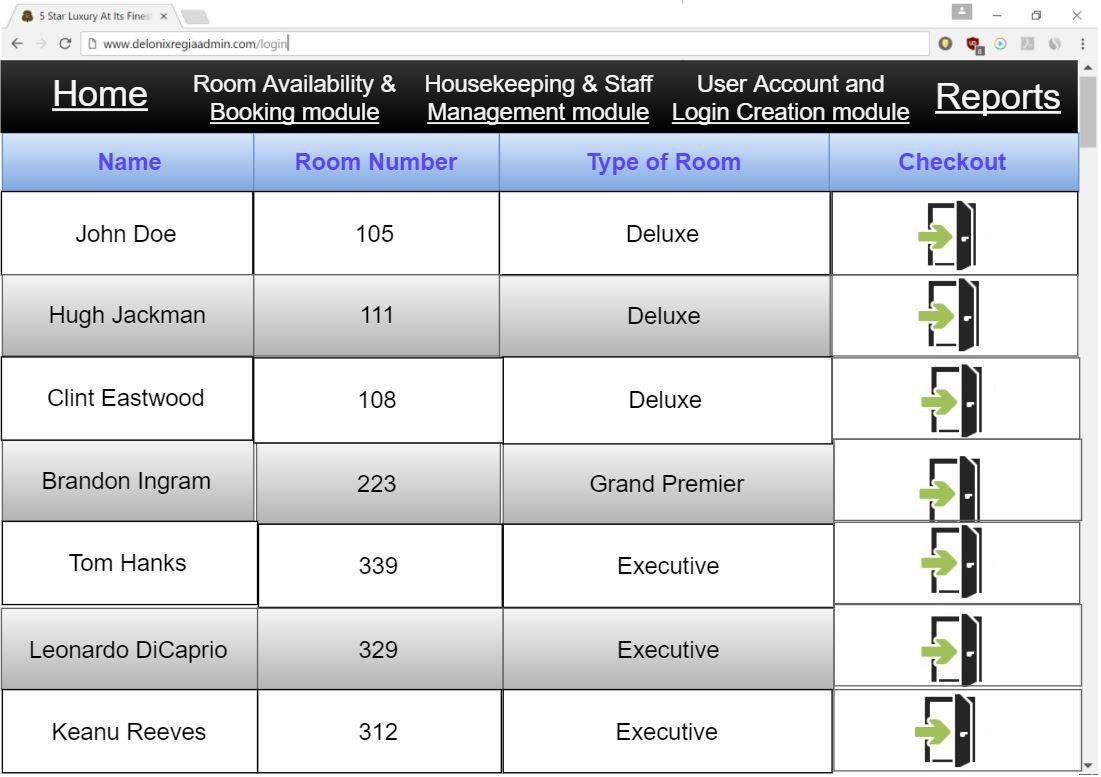


Once the receptionist clicks on the setup icon, a pop-up window will appear and the receptionist will have to key down all the customer’s necessary information.

### 3.2.3 Room list page







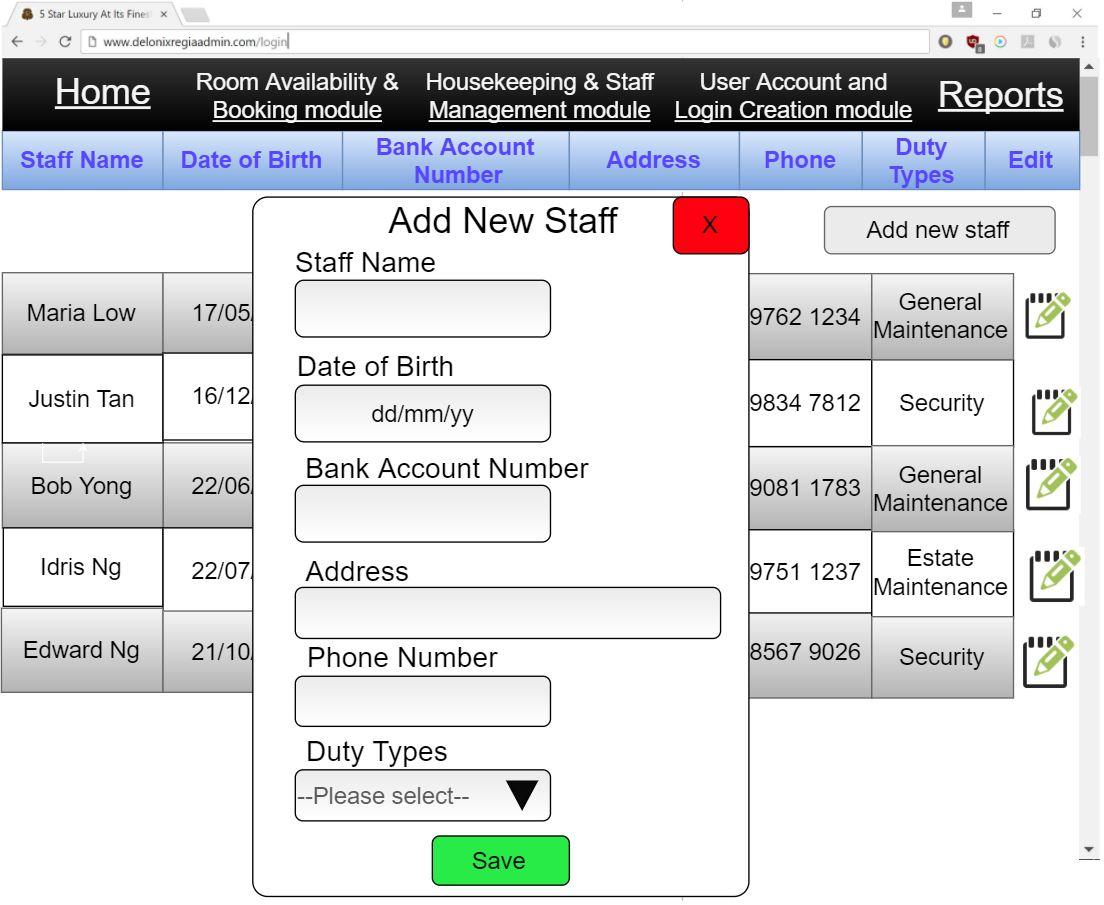
Once the setup is completed and the customer’s information has been saved, the room list page will display the customer name, room number, the type of room, and the checkout icon.

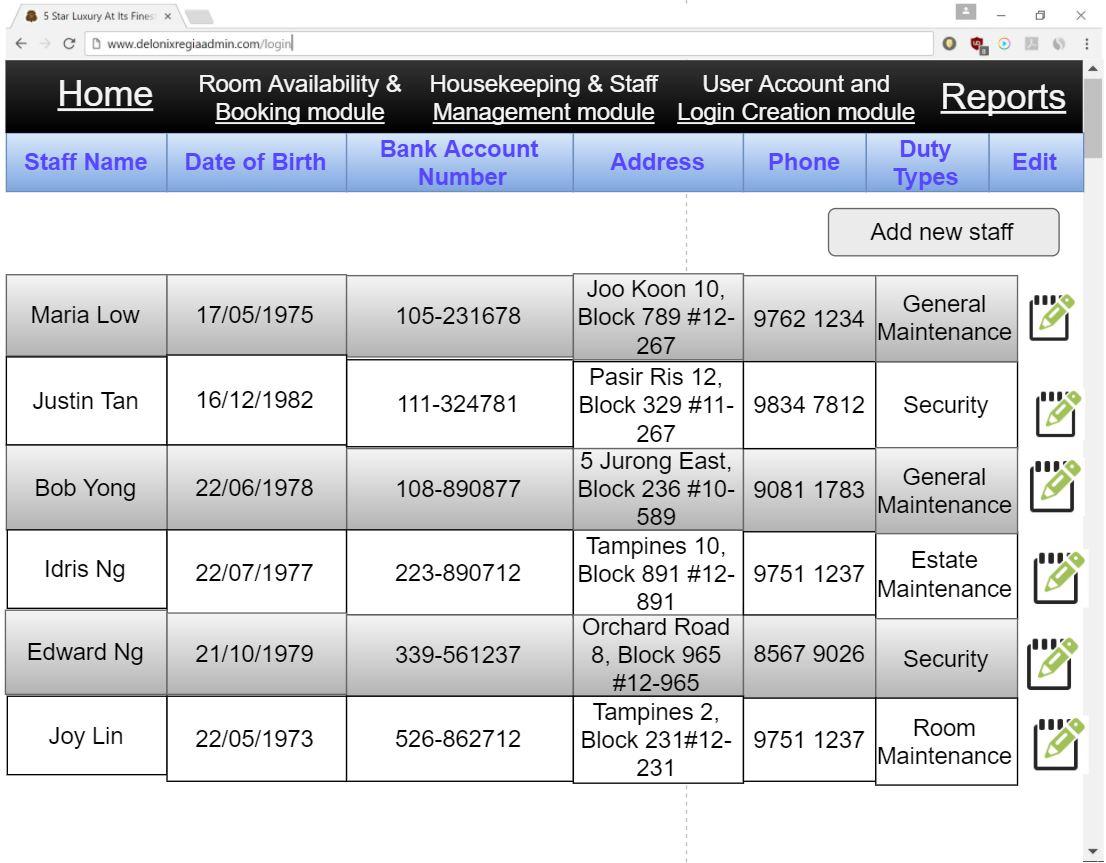
The room list page also interfaces with the room availability and room status page because when the customer has checkout, the receptionist can click on the checkout icon, indicating that the room is now available and can be scheduled for cleaning.

For example, once the receptionist clicks on the checkout icon, a confirmation message will appear ensuring to check out the guest. Afterwards, once the receptionist clicks “Yes” the guest’s information will have removed from the room list page.

### 3.2.4 Housekeeping & Staff management module



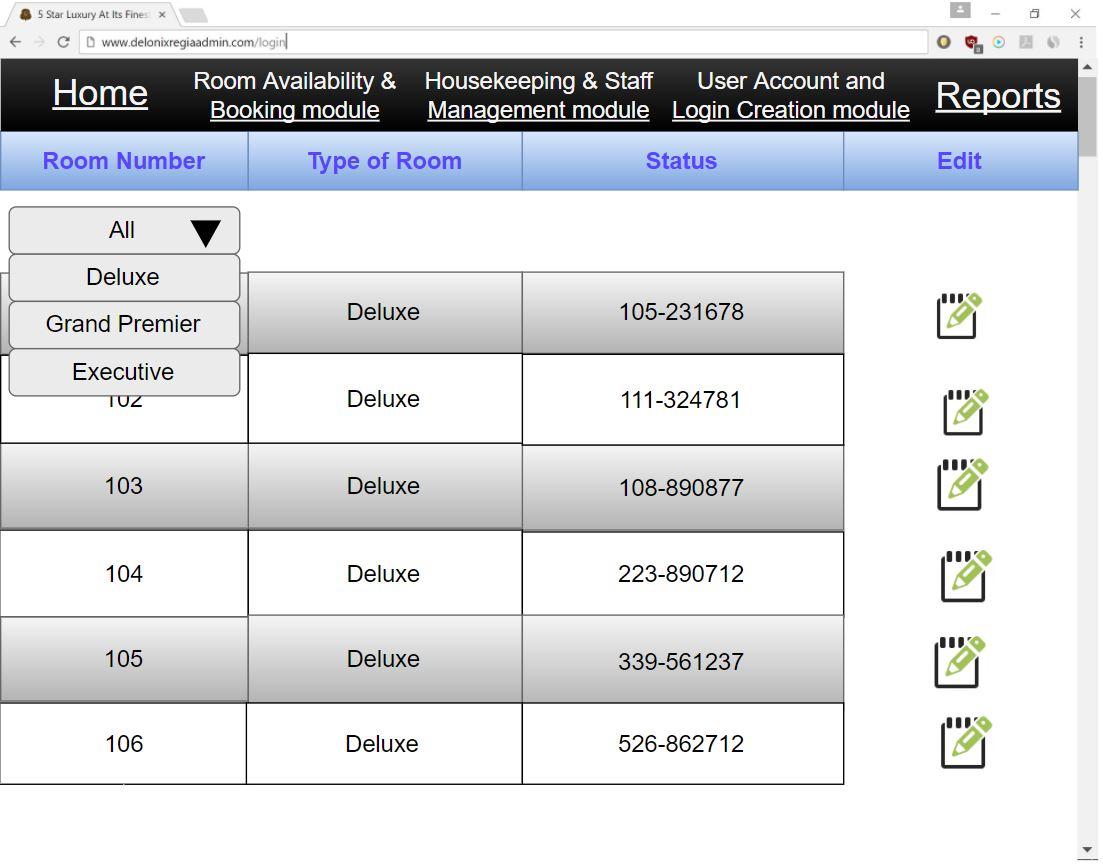


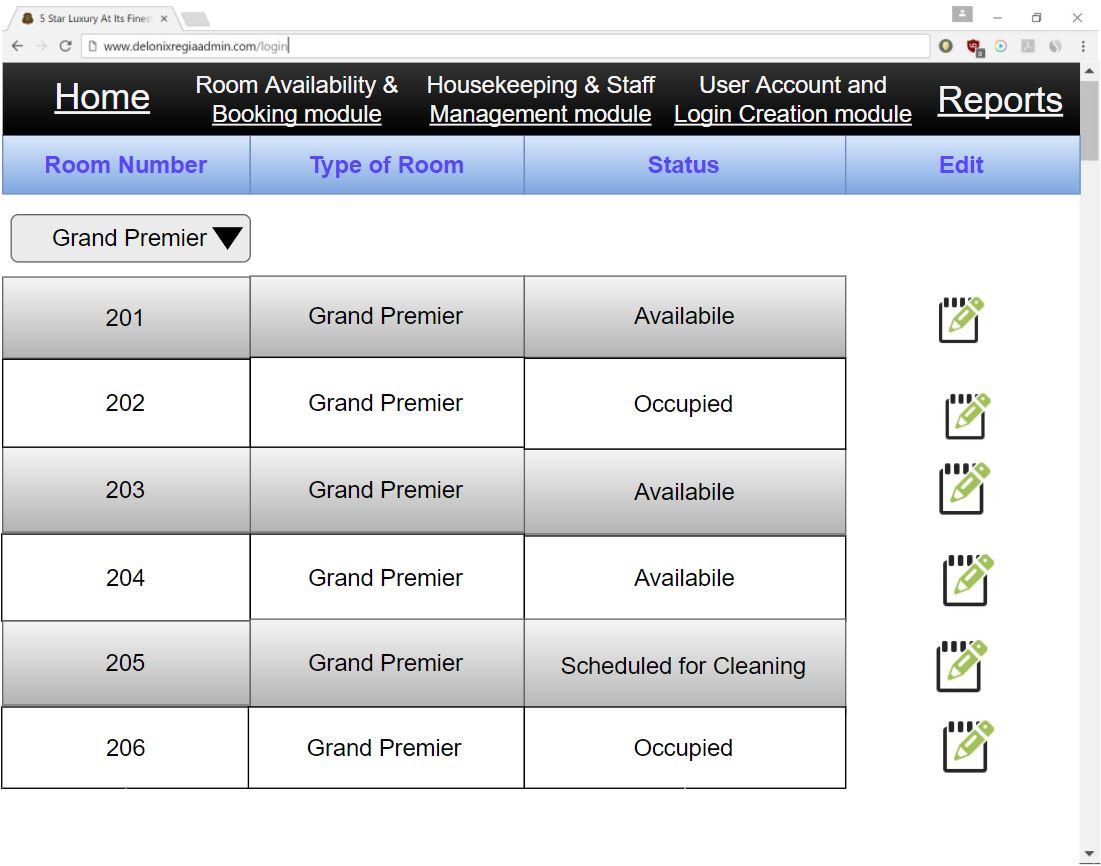


For the housekeeping and staff management module page, it displays the staff name, date of birth, bank account number, address, phone and duty types of the an individual staff.

It also provides an edit icon which allows the management users and administrators to change any information as well as an “Add new staff” button for new staffs.

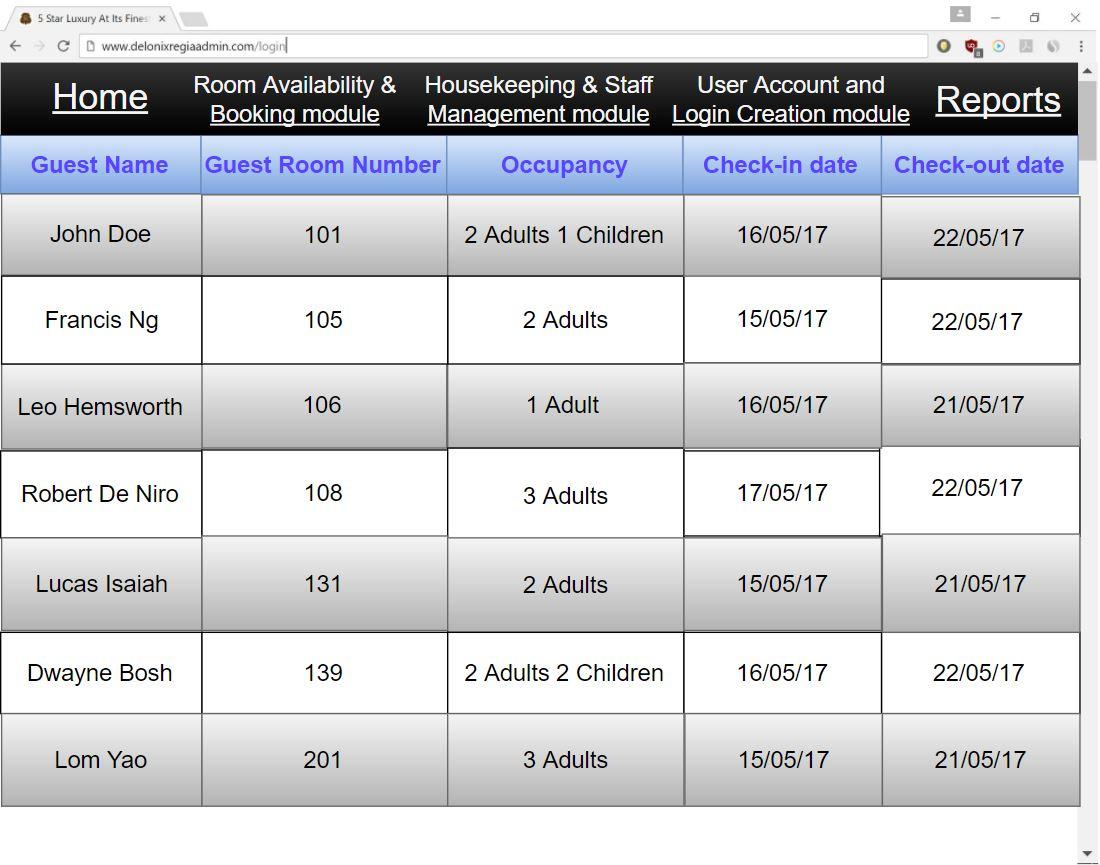
### 3.2.5 Room status report page





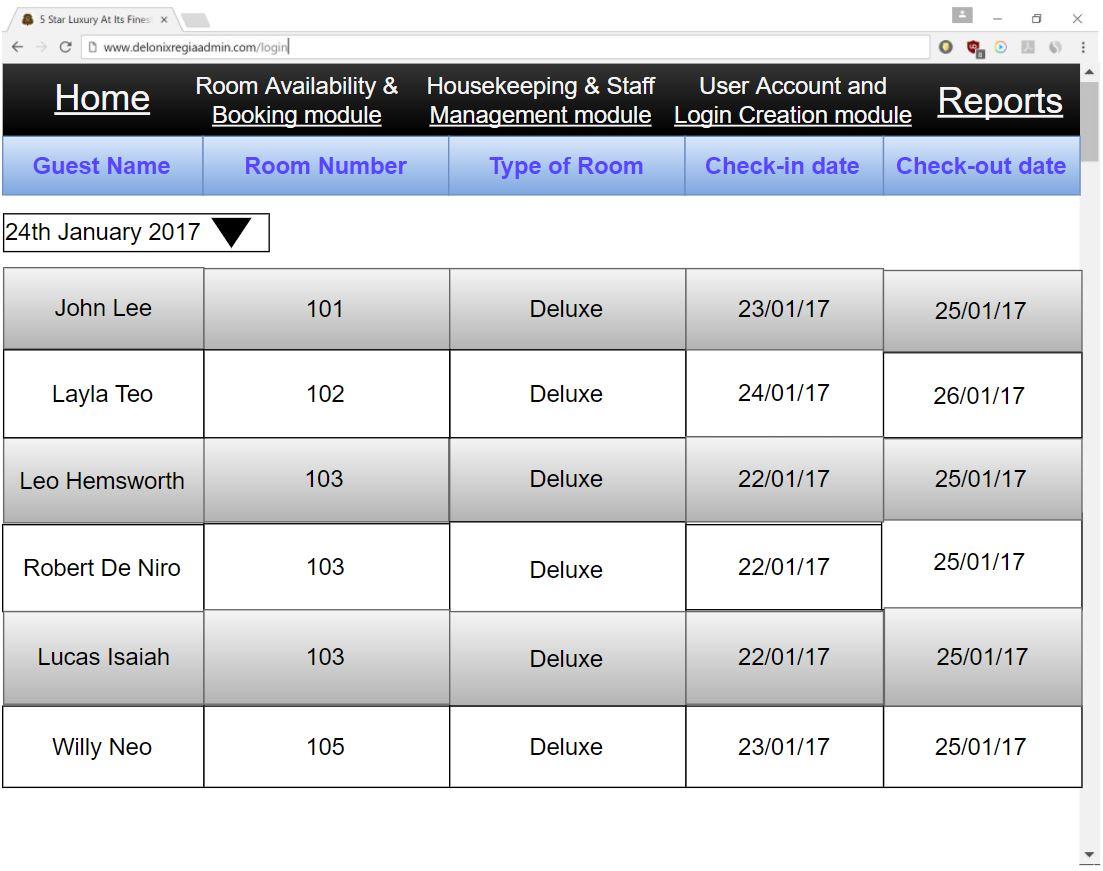
The room status report displays the room number, type of room and status. It has a drop down box which allows one to see room status by the type of room.

Guest status page



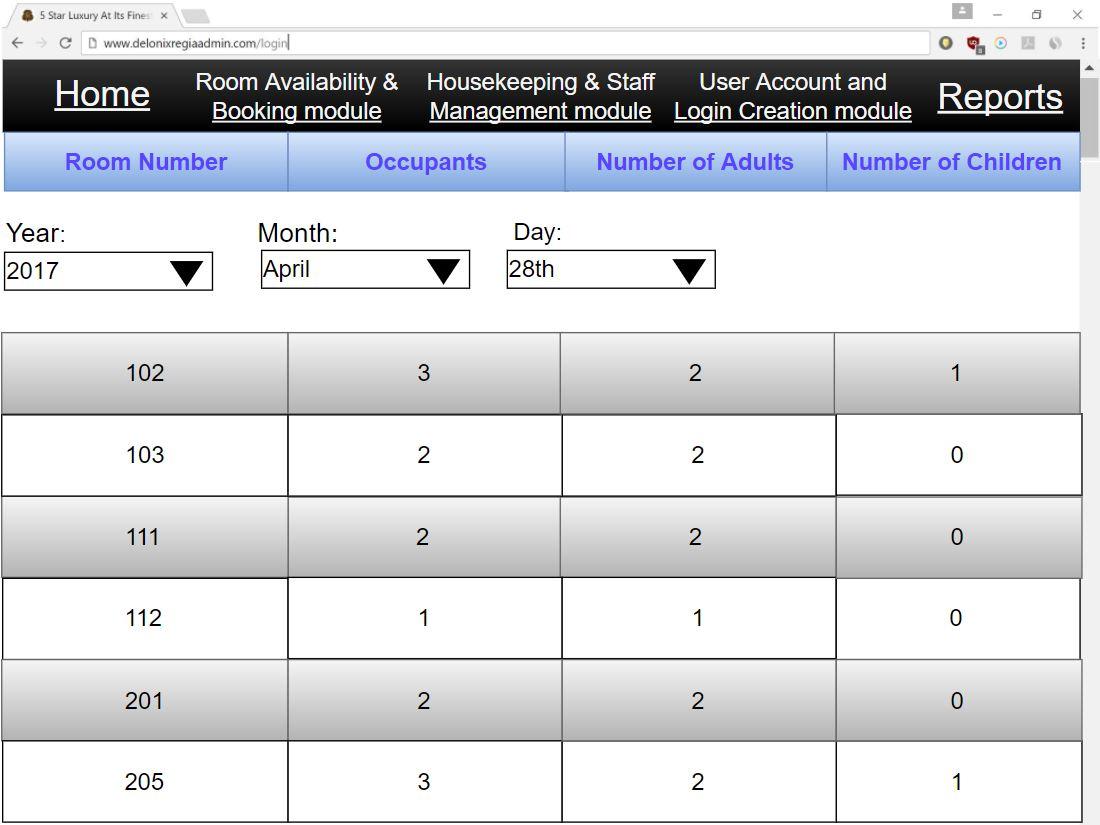
The guest status page displays the guest name, guest room number, occupancy, check in date, check-out date.

### 3.2.6 Listing all the guest in all the room page



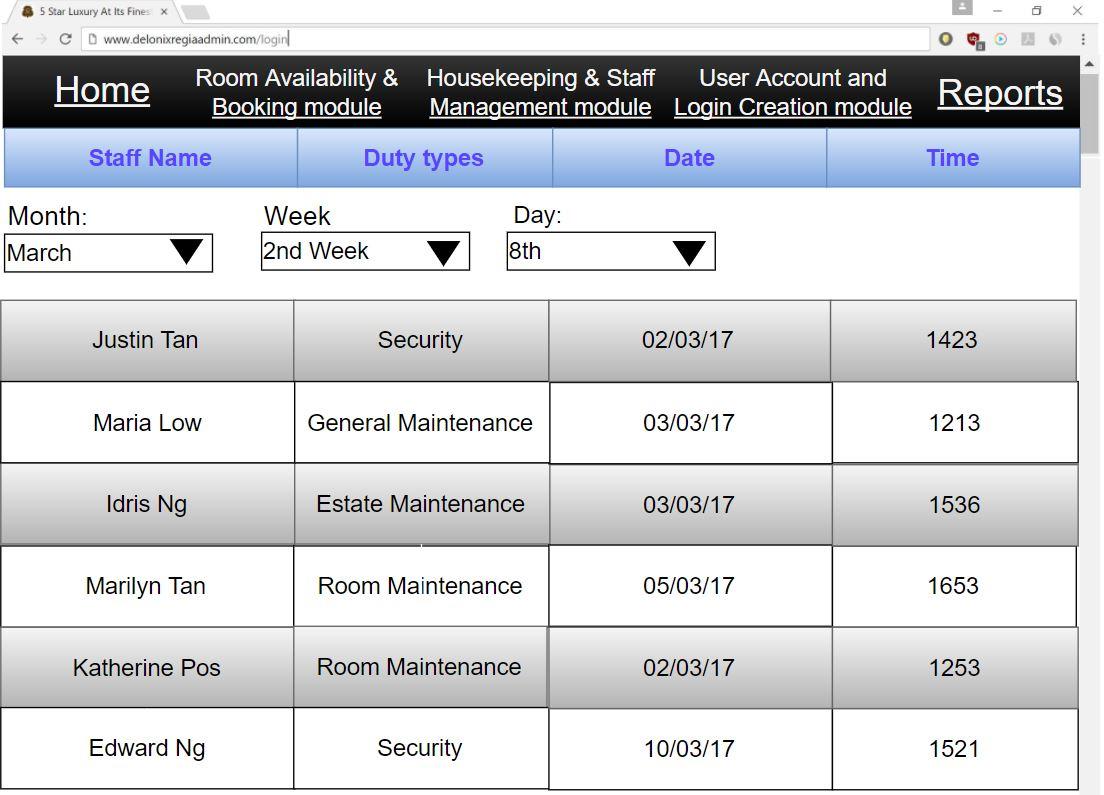
This page provides a list of guest based on their guest name, room number, type of room, check-in date, check-out date on a particular date, Using the drop-down box, one can select a specific date to view guest information.

### 3.2.7 Room occupancy page



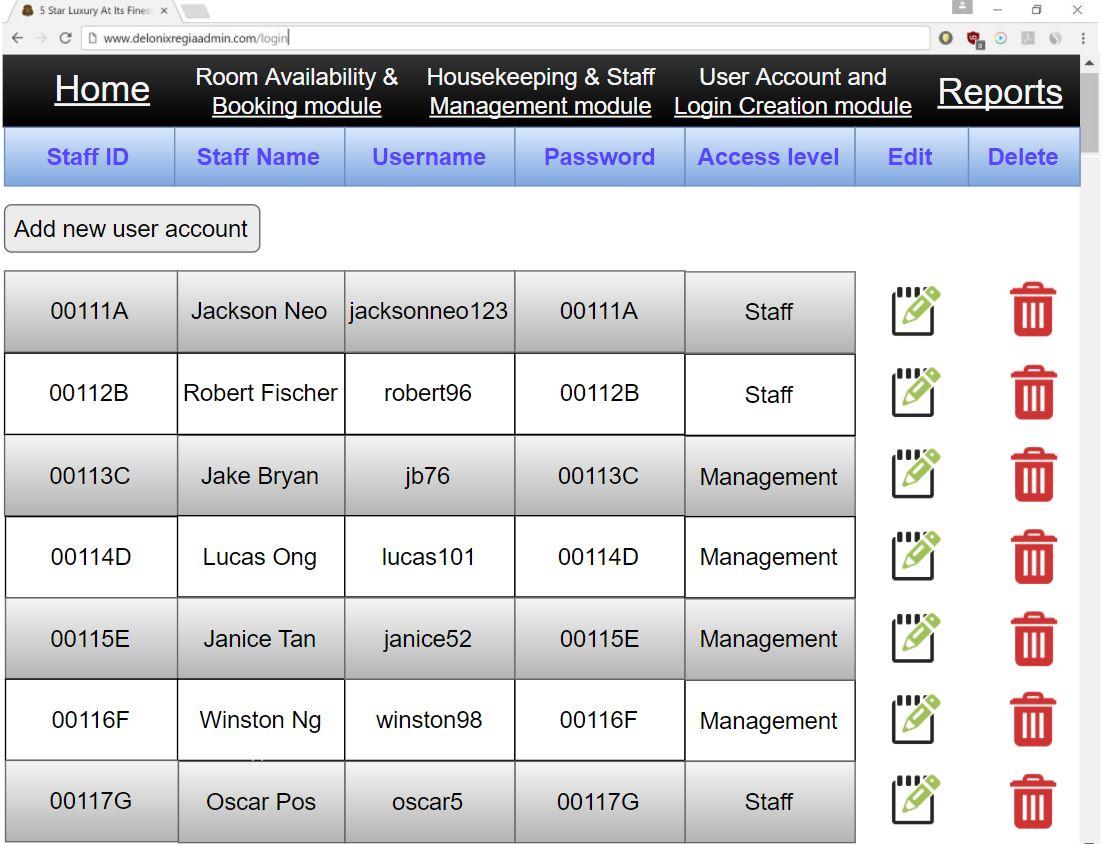
The room occupancy page displays the room number, occupants, number of adults and number of children. it calculates the total occupants in a room during a year, month or day.

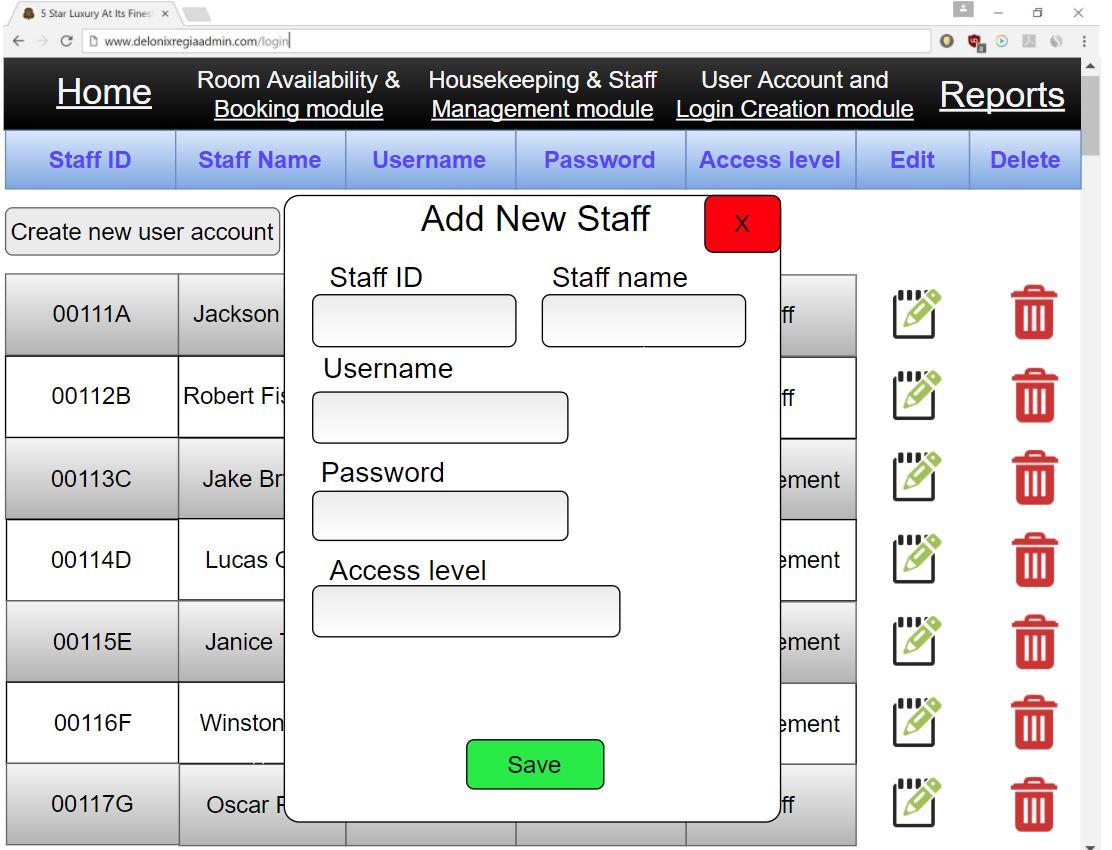
### 3.2.8 Housekeeping report page

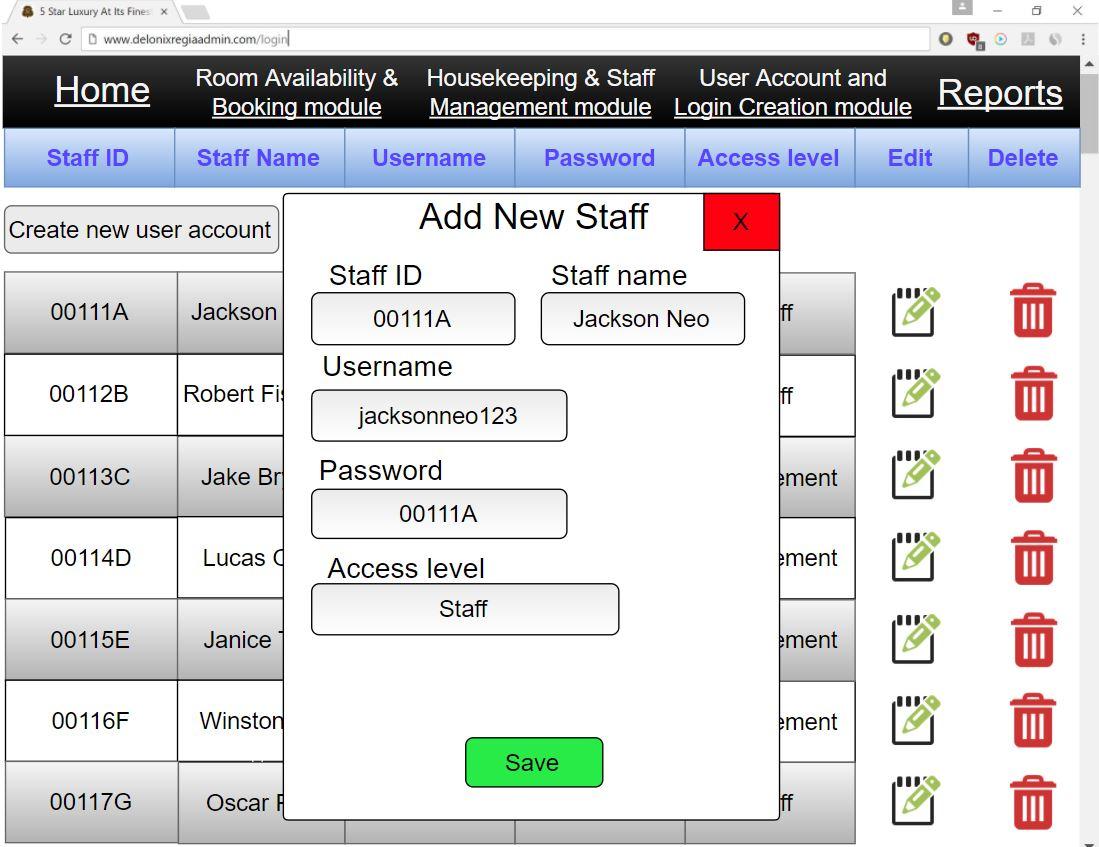


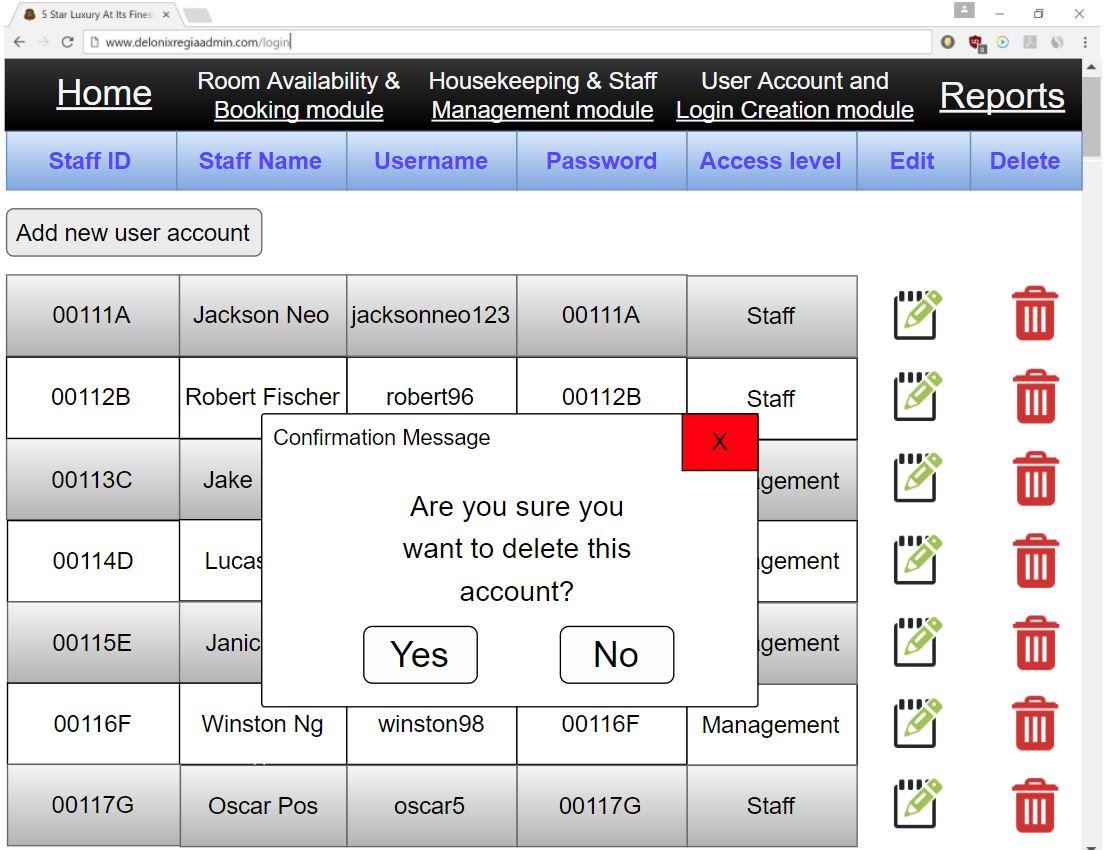
The housekeeping report page shows the staff name, duty types, date and time of an individual staff. One can check using the any month, any week and any day.

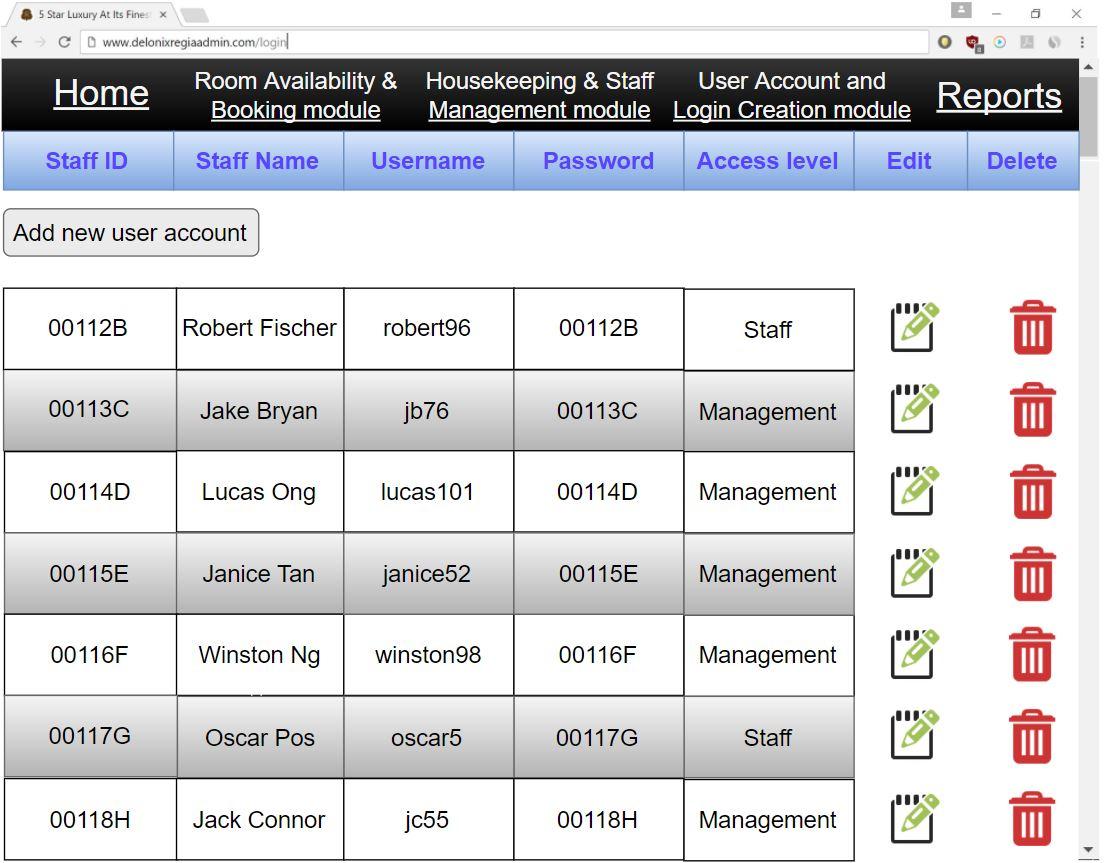
### 3.2.9 User Account & Login Creation module









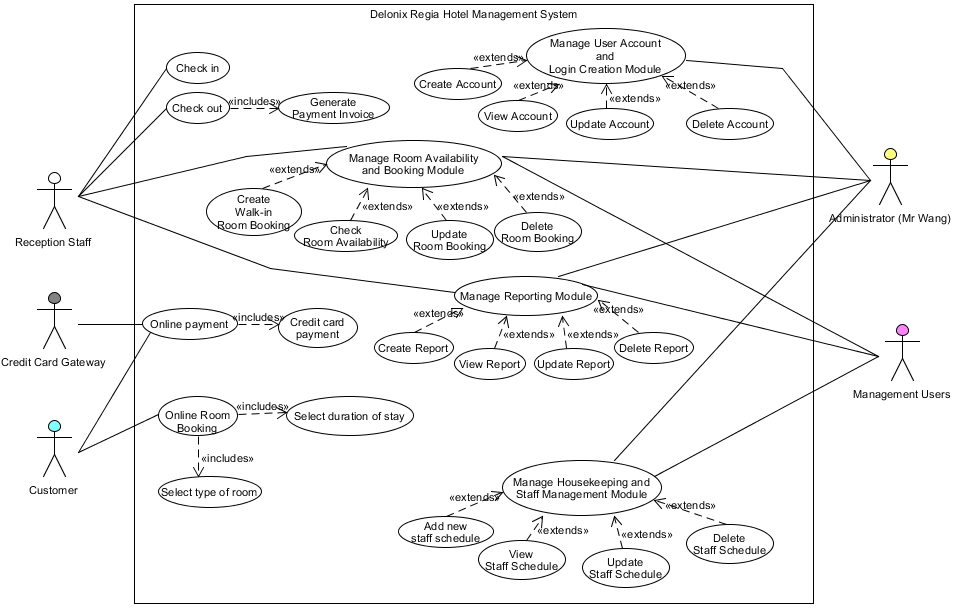


The user account and login creation module helps to record the staff id and staff name of an employee and their username and password with the access level. This is beneficial for the system as this can be a helpful if a staff forgets his or her username and password. It also provides an edit function to make changes to any information, a delete function to remove a staff user account, and also the create function which allows Mr Wang to create a new user account for new staffs employed.

# PROGRAM DESIGN

*[Describe how program design is done. What are the programs making up each system/software module?]*

* 1. Use Case Diagram



* 1. List of Actors and their respective description
* Reception Staff
  + Human actor who is an employee taking an office/administrative support position. They manage things like customer’s information and handles checking in and out for them. They are also able to manage reports
* Administrator
  + Human actor who is in charge of maintaining and managing the system, for example managing user accounts and login creation, managing reports, managing housekeeping and staff, etc.
* Management users
  + Human actor who is an employee taking an office/administrative position which helps the administrator in managing the system.
* Customer
  + Human actor that uses the system to accomplish his/her task which in this case booking a hotel room online
* Credit Card Gateway (Secondary Actor)
  + An object actor that allows customers to be able to make online payments
  1. List of Use Cases and their respective description
* Check in
  + The use case is to record the customer’s information into the system upon arrival
* Check out
  + It is to check the customer out of the hotel and generate a payment invoice of the customer’s spending, so that they can settle their bill.
* Manage User Account and Login Creation Module
  + Use case is to allow Administrators to manage the user account creation for new staffs and new users
* Mange Room Availability and Booking Module
  + The use case is to allow Administrators, Management users, and Reception Staff to manage room bookings. They can check the room availability, create a walk-in room booking, update an existing booked room, or delete a booked room.
* Manage Reporting Module
  + Use case is to allow Administrators, Management users, and Reception staff to manage reports. They can view a report, create a new report, update report, or delete a report.
* Manage Housekeeping and Staff Management Module
  + Use case is to allow Administrators and Management users to manage housekeeping and staff schedules. They can assign duties to staff as well as view staff schedules, create new staff schedule, update staff schedule, or delete staff schedule.
* Online Room Booking
  + Use Case is for customers to make online booking on the hotel’s website. It includes selecting the type of room they wish to book and selecting their duration of stay.
* Online Payment
  + Use case where customer pays with their credit card after booking a room
  1. Basic Flow and Alternative Flow of Use Cases

**Use Case 1: Check-in**

**Brief Description**

Reception staff needs to help customers to check in upon arrival to the hotel

**Actors**

Reception staff

**Flow of Events**

Basic Flow

|  |  |  |  |
| --- | --- | --- | --- |
| **Step No.** | **Actor’s Action** | **Step No.** | **System’s Response** |
| **1** | This use case begins when the receptionist wants to help the customer to check in |  |  |
| **2** | Receptionist retrieves customer’s booking information | **3** | System retrieves booking information from database |
|  |  | **4** | System displays customer’s booking information |
| **5** | Receptionist retrieves room availability information from system | **6** | System retrieves room availability information from database |
|  |  | **7** | System displays room availability information |
| **8** | Receptionist checks if the rooms are vacant or currently occupied |  |  |
| **9** | Receptionist selects an available room and confirms with customer on the room and the customer’s information | **10** | Customer has successfully been checked in. System records customer as checked in. Use case is done. |

Alternate Flow

An alternate flow will occur if the type of room the customer desired is not available. The customer will be prompt to either wait for the room to be available or to choose another type of room (receptionist may offer an upgrade as an apology from the hotel) – Which in this case the receptionist will repeat step 5 until the customer’s desired room is booked and checked in.

**Use Case 2: Check-out**

**Brief Description**

Receptionist will help customers check out and generate a payment invoice for them to pay their bills

**Actors**

Reception staff

**Flow of Events**

Basic Flow

|  |  |  |  |
| --- | --- | --- | --- |
| **Step No.** | **Actor’s Action** | **Step No.** | **System’s Response** |
| **1** | This use case begins when the receptionist wants to help the customer check out |  |  |
| **2** | Receptionist retrieves customer’s information from the system | **3** | System retrieves information from database |
|  |  | **4** | System displays customer’s information |
| **5** | Receptionist checks if customer checked out late or spent on stuff that requires additional payment, and request for a payment invoice from the system | **6** | System calculate customer’s spending, and generate and print a payment invoice |
| **7** | Receptionist gives the customer the payment invoice and receives payment and checks them out | **8** | Check out is successful. System updates and customer is checked out. Use case is done. |

**Use Case 3: Manage Room Availability and Booking Module**

**Brief Description**

It is for managing room availability and booking for walk-in customers. The use case is for managing and making changes to room bookings such as creating a new walk-in room booking, checking room availability, updating a booked room, and deleting a booked room.

**Actors**

Administrator, Management users, Reception Staff

**Flow of Events**

Basic Flow

|  |  |  |  |
| --- | --- | --- | --- |
| **Step No.** | **Actor’s Action** | **Step No.** | **System’s Response** |
| **1** | This use case begins when the Administrator/Management users/ Reception Staff wishes to create a walk-in room booking, check room availability, update, or delete a booked room |  |  |
| **2** | Administrator retrieves room availability information from system | **3** | System retrieves room availability information |
|  |  | **4** | System displays room availability information |
| **5** | Administrator chooses to create a new walk-in room booking, view, update or delete a room. | **6** | All changes has been updated and saved successfully. Use case is done. |

Alternate Flows

1. Create new walk-in room booking

|  |  |  |  |
| --- | --- | --- | --- |
| **Step No.** | **Actor’s Action** | **Step No.** | **System’s Response** |
| **1** | This use case begins when the receptionist wants to book a room for a walk-in customer |  |  |
| **2** | Receptionist requests for the customer’s information and their desired type of room |  |  |
| **3** | Receptionist retrieves room availability information from system | **4** | System retrieves room availability information |
|  |  | **5** | System displays room availability information |
| **6** | Receptionist checks if the type of room is available and selects the room that is not occupied |  |  |
| **7** | Receptionist confirms with customer on the room and enters the customer’s information as well as the duration of stay | **8** | Room has successfully been booked. System records booked room and customer as checked in. Use case is done. |

Alternate Flow

An alternate flow will take place if the type of room the customer desired is not available and the reception staff is unable to select a room. The reception staff would have to repeat from step 2 and request from the customer another desired type of room.

2. Check Room Availability

|  |  |  |  |
| --- | --- | --- | --- |
| **Step No.** | **Actor’s Action** | **Step No.** | **System’s Response** |
| **1** | This use case begins when the Administrator/Management users/ Reception Staff wishes to check the room availability |  |  |
| **2** | Administrator/Management users/ Reception Staff retrieves room availability information from system | **3** | System retrieves room availability information |
|  |  | **4** | System displays room availability information |
| **5** | Administrator/Management users/ Reception Staff searches for the rooms’ availability and view the details. Use case is done |  |  |

3. Update Room Booking

|  |  |  |  |
| --- | --- | --- | --- |
| **Step No.** | **Actor’s Action** | **Step No.** | **System’s Response** |
| **1** | This use case begins when the Administrator/Management users/ Reception Staff wishes to update a room booking |  |  |
| **2** | Administrator/Management users/ Reception Staff retrieved room booking information from system | **3** | System retrieves room booking information |
|  |  | **4** | System displays room booking information |
| **5** | Administrator/Management users/ Reception Staff searches for the specific room and updates the details in it | **6** | System successfully updated and changes made were saved. Use case is done |

4. Delete Room Booking

|  |  |  |  |
| --- | --- | --- | --- |
| **Step No.** | **Actor’s Action** | **Step No.** | **System’s Response** |
| **1** | This use case begins when the Administrator/Management users/ Reception Staff wishes to delete a booked room |  |  |
| **2** | Administrator/Management users/ Reception Staff retrieves room booking information from system | **3** | System retrieves room booking information |
|  |  | **4** | System displays room booking information |
| **5** | Administrator/Management users/ Reception Staff searches for the specific room booking and deletes it | **6** | System successfully updated and changes made were saved. Use case is done. |

**Use Case 4: Manage User Account and Login Creation Module**

**Brief Description**

It is for managing and creating new accounts for new staff and users. The use case is for managing and making changes to the user accounts such as creating a new user account, viewing a user account, updating a user account, and deleting a user account.

**Actors**

Administrator

**Flow of Events**

Basic Flow

|  |  |  |  |
| --- | --- | --- | --- |
| **Step No.** | **Actor’s Action** | **Step No.** | **System’s Response** |
| **1** | This use case begins when the Administrator wishes to create, view, update, or delete an account |  |  |
| **2** | Administrator retrieves user accounts database from system | **3** | System retrieves user accounts database |
|  |  | **4** | System displays user accounts database |
| **5** | Administrator chooses to create, view, update or delete an account | **6** | All changes has been updated and saved successfully. Use case is done. |

Alternate Flows

1. Create new user account

Alternate flow begins from step 5 of basic flow

|  |  |  |  |
| --- | --- | --- | --- |
| **Step No.** | **Actor’s Action** | **Step No.** | **System’s Response** |
| **1** | This use case begins when the Administrator wants to create a account for new staff/user |  |  |
| **2** | Administrator retrieves account creation page from system | **3** | System retrieves account creation page |
|  |  | **4** | System displays account creation page |
| **5** | Administrator enters new staff/user’s information | **6** | New account has been created successfully. System saves and updates new account into the database. Use case is done. |

2. View user account

|  |  |  |  |
| --- | --- | --- | --- |
| **Step No.** | **Actor’s Action** | **Step No.** | **System’s Response** |
| **1** | This use case begins when the Administrator wishes to view a user account |  |  |
| **2** | Administrator retrieves user accounts database from system | **3** | System retrieves user accounts database |
|  |  | **4** | System displays user accounts database |
| **5** | Administrator searches for the user account and view the details.  Use case is done |  |  |

3. Update user account

|  |  |  |  |
| --- | --- | --- | --- |
| **Step No.** | **Actor’s Action** | **Step No.** | **System’s Response** |
| **1** | This use case begins when the Administrator wishes update an account |  |  |
| **2** | Administrator retrieves user accounts database from system | **3** | System retrieves user accounts database |
|  |  | **4** | System displays user accounts database |
| **5** | Administrator searches for the specific user account and updates the details in it | **6** | System successfully updated and changes made were saved. Use case is done |

4. Delete a user account

|  |  |  |  |
| --- | --- | --- | --- |
| **Step No.** | **Actor’s Action** | **Step No.** | **System’s Response** |
| **1** | This use case begins when the Administrator wishes to delete an account |  |  |
| **2** | Administrator retrieves user accounts database from system | **3** | System retrieves user accounts database |
|  |  | **4** | System displays user accounts database |
| **5** | Administrator searches for a specific user account and deletes it | **6** | System successfully updated and changes made were saved. Use case is done. |

**Use Case 5: Manage Reporting Module**

**Brief Description**

The use case is for managing and making changes to the reports such as creating a new report, viewing the report, updating the report, and deleting a report.

**Actors**

Administrators, Management users and Reception Staff

**Flow of Events**

Basic Flow

|  |  |  |  |
| --- | --- | --- | --- |
| **Step No.** | **Actor’s Action** | **Step No.** | **System’s Response** |
| **1** | This use case begins when the Administrator/Management users/Reception Staff wishes to create, view, update, or delete a report from the reports database |  |  |
| **2** | Administrator/Management users/Reception Staff retrieves report database from system | **3** | System retrieves report database |
|  |  | **4** | System displays report database |
| **5** | Administrator/Management users/Reception Staff decides whether to create, view, update or delete a report | **6** | System updates and saves all changes made successfully. Use case is done |

Alternate Flows

1. Create a report

|  |  |  |  |
| --- | --- | --- | --- |
| **Step No.** | **Actor’s Action** | **Step No.** | **System’s Response** |
| **1** | This use case begins when the Administrator/Management users/Reception Staff wishes to create a report |  |  |
| **2** | Administrator/Management users/Reception Staff retrieves report database from system | **3** | System retrieves report database |
|  |  | **4** | System displays report database |
| **5** | Administrator/Management users/Reception Staff creates a new report | **6** | New report has been created successfully. System saves and updates new report into the database. Use case is done. |

2. View a report

|  |  |  |  |
| --- | --- | --- | --- |
| **Step No.** | **Actor’s Action** | **Step No.** | **System’s Response** |
| **1** | This use case begins when the Administrator/Management users/Reception Staff wishes to view a report |  |  |
| **2** | Administrator/Management users/Reception Staff retrieves report database from system | **3** | System retrieves report database |
|  |  | **4** | System displays report database |
| **5** | Administrator/Management users/Reception Staff searches and view the specific report. Use case is done |  |  |

3. Update a report

|  |  |  |  |
| --- | --- | --- | --- |
| **Step No.** | **Actor’s Action** | **Step No.** | **System’s Response** |
| **1** | This use case begins when the Administrator/Management users/Reception Staff wishes to update a report |  |  |
| **2** | Administrator/Management users/Reception Staff retrieves report database from system | **3** | System retrieves report database |
|  |  | **4** | System displays report database |
| **5** | Administrator/Management users/Reception Staff searches for specific report and updates it | **6** | System successfully updated and changes made were saved. Use case is done |

4. Delete a report

|  |  |  |  |
| --- | --- | --- | --- |
| **Step No.** | **Actor’s Action** | **Step No.** | **System’s Response** |
| **1** | This use case begins when the Administrator/Management users wishes to delete a report |  |  |
| **2** | Administrator/Management users retrieves report database from system | **3** | System retrieves report database |
|  |  | **4** | System displays report database |
| **5** | Administrator/Management users searches for specific report and deletes it | **6** | System successfully updated and changes made were saved. Use case is done |

**Use Case 6: Manage Housekeeping and Staff Management Module**

**Brief Description**

The use case is for managing and making changes to the staff schedules such as creating a new staff schedule, viewing the staff schedule, updating the staff schedule, and deleting a staff schedule.

**Actors**

Administrators and Management users

**Flow of Events**

Basic Flow

|  |  |  |  |
| --- | --- | --- | --- |
| **Step No.** | **Actor’s Action** | **Step No.** | **System’s Response** |
| **1** | This use case begins when the Administrator/Management users wishes to create, view, update, or delete a staff schedule from the staff schedule database |  |  |
| **2** | Administrator/Management users retrieves staff schedule database from system | **3** | System retrieves staff schedule database |
|  |  | **4** | System displays staff schedule database |
| **5** | Administrator/Management users decides whether to create, view, update or delete the staff schedule | **6** | System updates and saves all changes made successfully. Use case is done |

Alternate Flows

1. Create a staff schedule

|  |  |  |  |
| --- | --- | --- | --- |
| **Step No.** | **Actor’s Action** | **Step No.** | **System’s Response** |
| **1** | This use case begins when the Administrator/Management users wishes to create a new staff schedule |  |  |
| **2** | Administrator/Management users retrieves report database | **3** | System retrieves staff schedule database |
|  |  | **4** | System displays staff schedule database |
| **5** | Administrator/Management users creates a new staff schedule. | **6** | New staff schedule has been created successfully. System saves and updates new staff schedule into the database. Use case is done. |

2. View a staff schedule

|  |  |  |  |
| --- | --- | --- | --- |
| **Step No.** | **Actor’s Action** | **Step No.** | **System’s Response** |
| **1** | This use case begins when the Administrator/Management users/ Reception Staff wishes to view a staff schedule |  |  |
| **2** | Administrator/Management users/Reception Staff retrieves staff schedule database | **3** | System retrieves staff schedule database |
|  |  | **4** | System displays staff schedule database |
| **5** | Administrator/Management users/Reception Staff searches and view the specific staff duty schedule. Use case is done |  |  |

3. Update a staff schedule

|  |  |  |  |
| --- | --- | --- | --- |
| **Step No.** | **Actor’s Action** | **Step No.** | **System’s Response** |
| **1** | This use case begins when the Administrator/Management users wishes to update a staff schedule |  |  |
| **2** | Administrator/Management users retrieves staff schedule database | **3** | System retrieves staff schedule database |
|  |  | **4** | System displays staff schedule database |
| **5** | Administrator/Management users searches for specific staff schedule and updates it | **6** | System successfully updated and changes made were saved. Use case is done |

4. Delete a report

|  |  |  |  |
| --- | --- | --- | --- |
| **Step No.** | **Actor’s Action** | **Step No.** | **System’s Response** |
| **1** | This use case begins when the Administrator/Management users wishes to delete a staff schedule |  |  |
| **2** | Administrator/Management users retrieves staff schedule database | **3** | System retrieves staff schedule database |
|  |  | **4** | System displays staff schedule database |
| **5** | Administrator/Management users searches for specific staff schedule and deletes it | **6** | System successfully updated and changes made were saved. Use case is done |

**Use Case 7: Online Room Booking**

**Brief Description**

The use case is for customers to make an online booking on the hotel’s website. It includes selecting the type of room they want to book and the duration of their stay.

**Actors**

Customer

**Flow of Events**

Basic Flow

|  |  |  |  |
| --- | --- | --- | --- |
| **Step No.** | **Actor’s Action** | **Step No.** | **System’s Response** |
| **1** | This use case begins when the Customer wishes to book a room online |  |  |
| **2** | Customer clicks on room lists | **3** | System retrieves list of rooms |
|  |  | **4** | System displays list of rooms |
| **5** | Customer selects the type of room to book and selects when to check in and when to check out | **6** | System prompts for booking confirmation and payment |
| **7** | Customer confirms booking and goes through online payment | **8** | Room has been successfully booked and recorded into database. Customer will be sent a pin/confirmation number via SMS where they can update or cancel their bookings. Use case is done. |

**Use Case 8: Online Payment**

**Brief Description**

The use case is for customers to make an online payment after room booking confirmation.

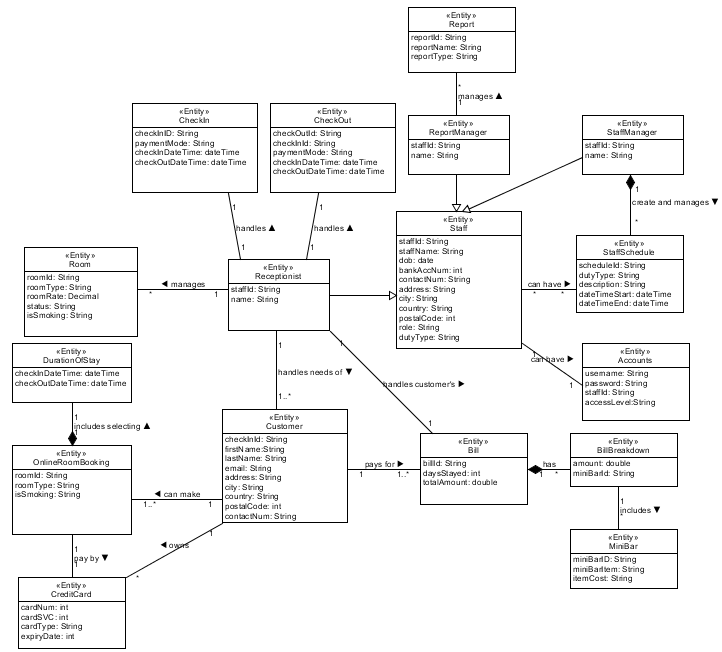
**Actors**

Customer

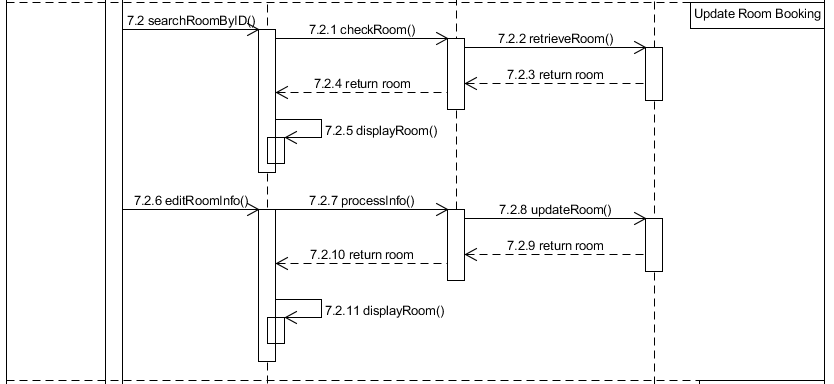
**Flow of Events**

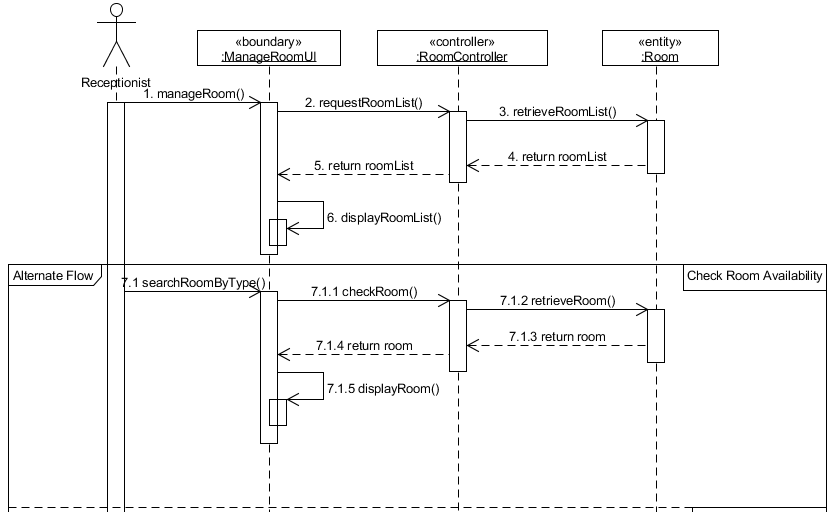
Basic Flow

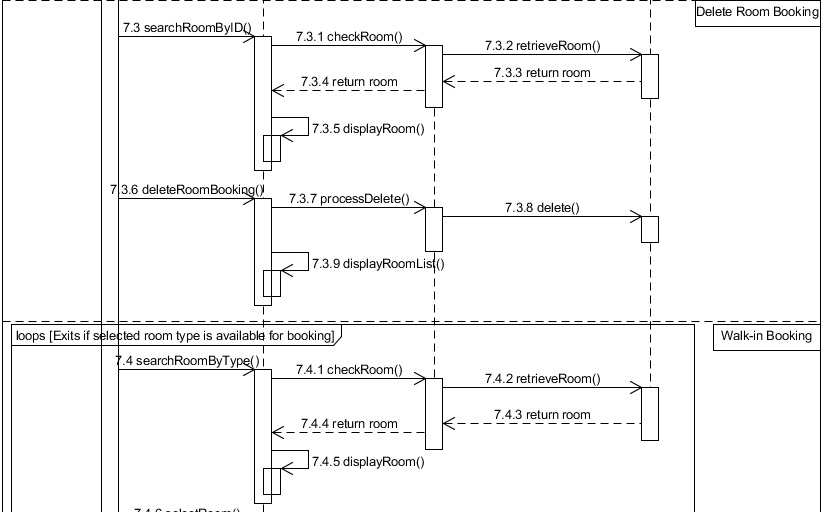
|  |  |  |  |
| --- | --- | --- | --- |
| **Step No.** | **Actor’s Action** | **Step No.** | **System’s Response** |
| **1** | This use case begins when the Customer finished room booking and makes online payment | **2** | System prompts for type of credit card payment |
| **3** | Customer selects type of card payment | **4** | System prompts for credit card details |
| **5** | Customer enters credit card details | **6** | System prompt for confirmation and submit |
| **7** | Customer submits payment information | **6** | System displays booking summary, payment has been successfully made. Use case is done |

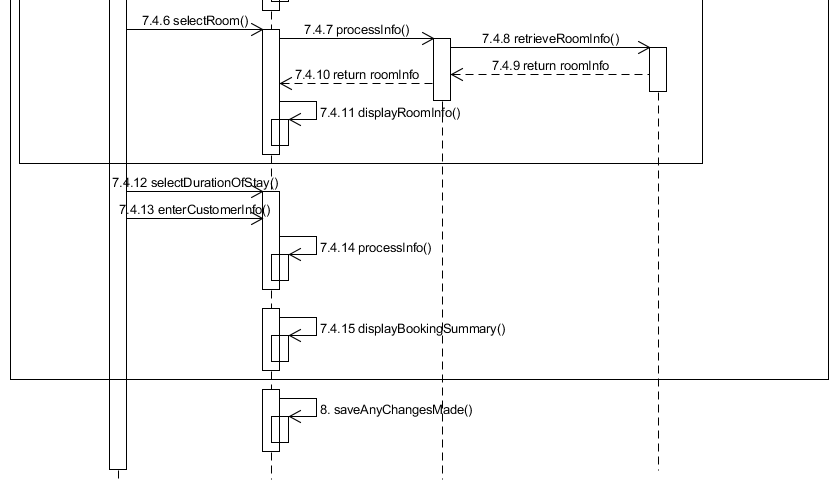
* 1. Domain Model
  2. Sequence Diagrams and VOPC

**Manage Room Availability and Booking Module**

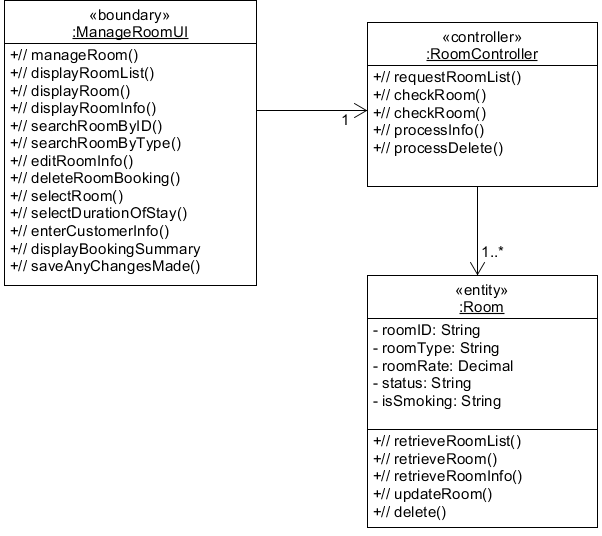
Sequence Diagram





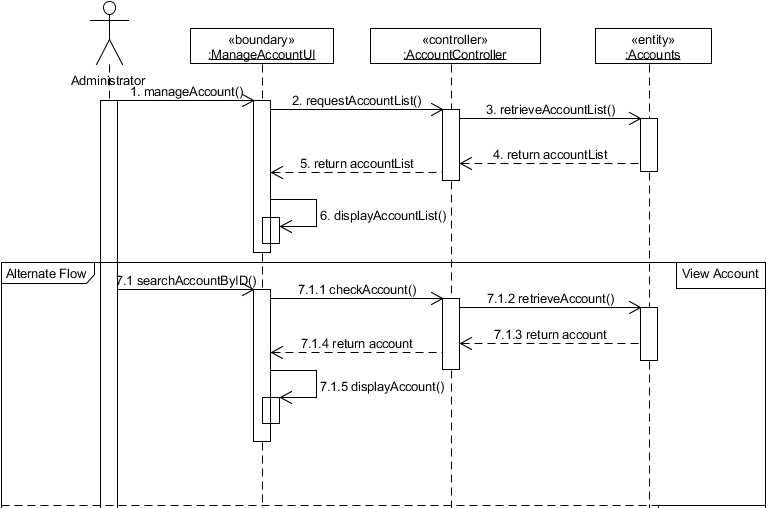


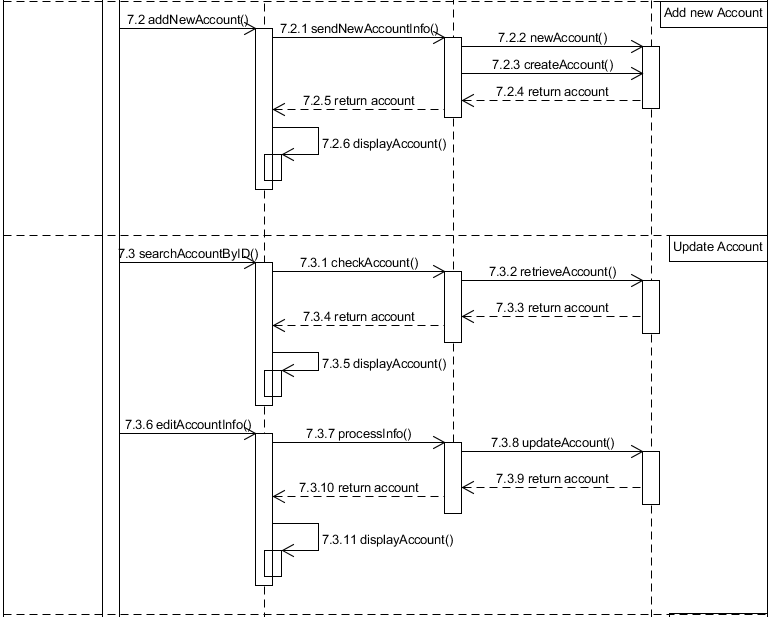
VOPC

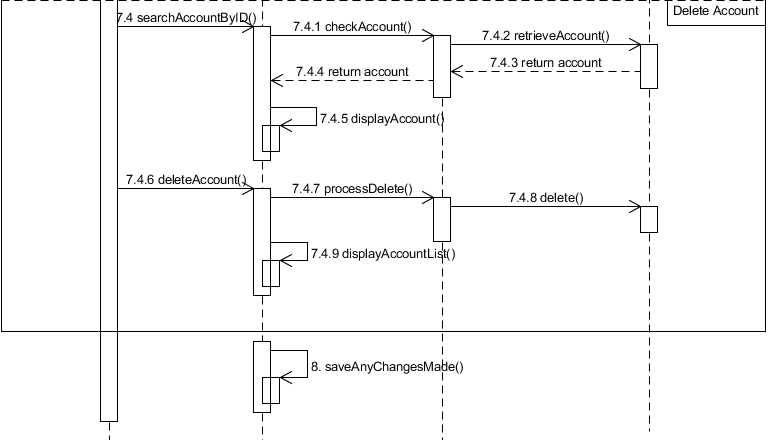


**Manage User Account and Login Creation Module**

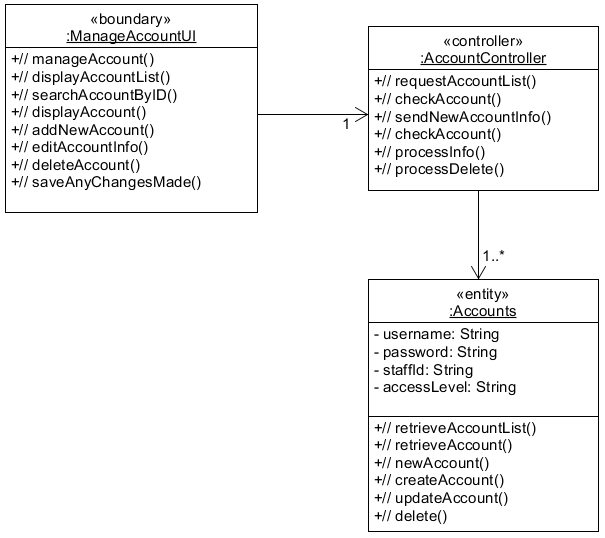
Sequence Diagram





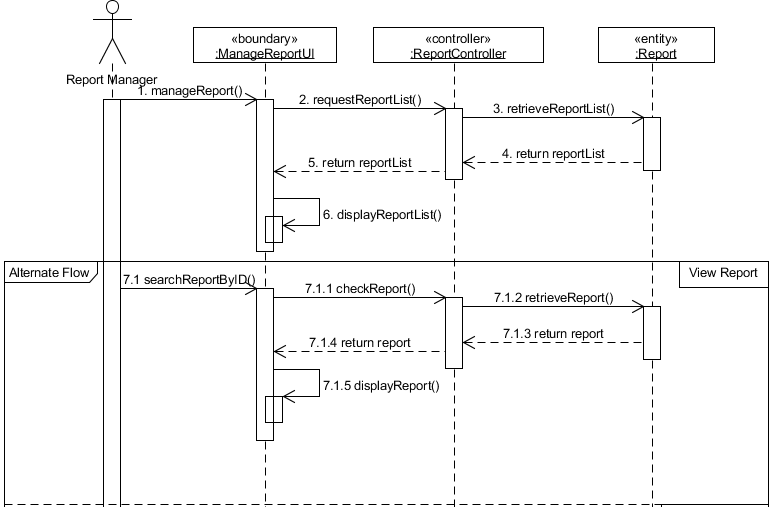


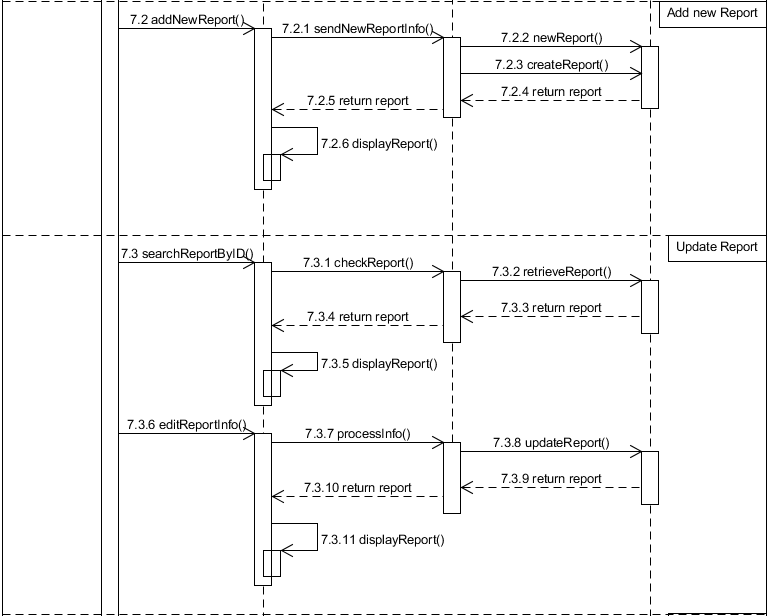
VOPC

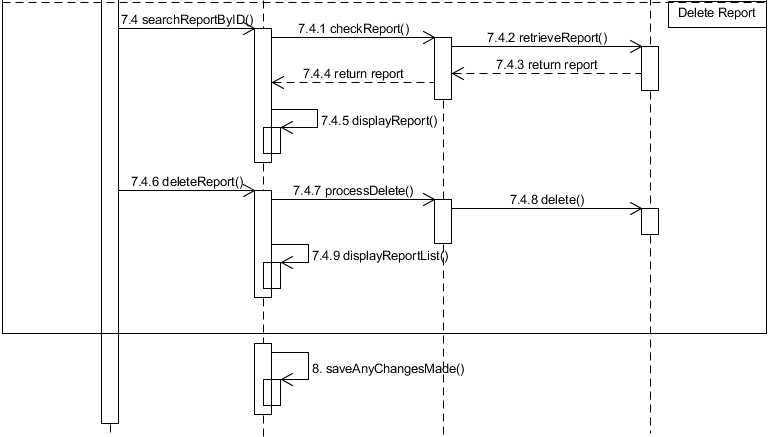


**Manage Reporting Module**

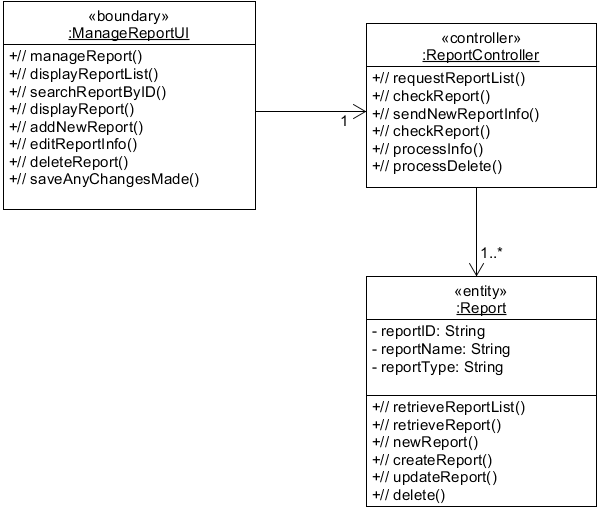
Sequence Diagram





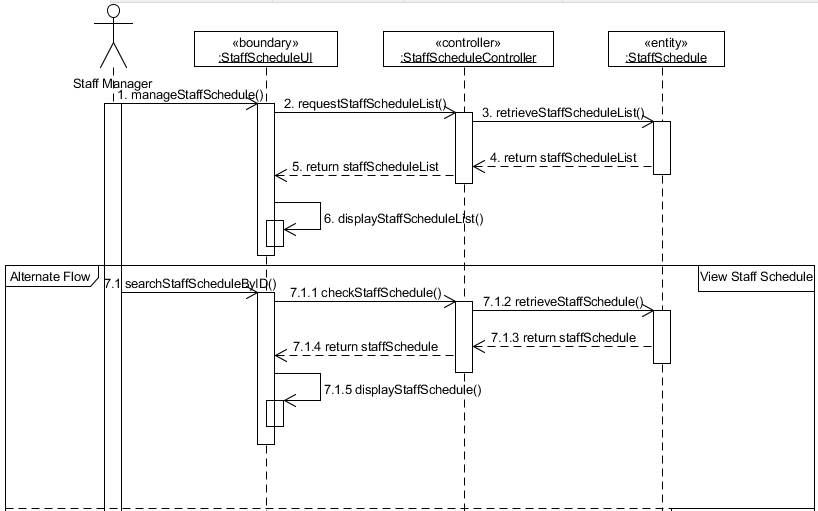


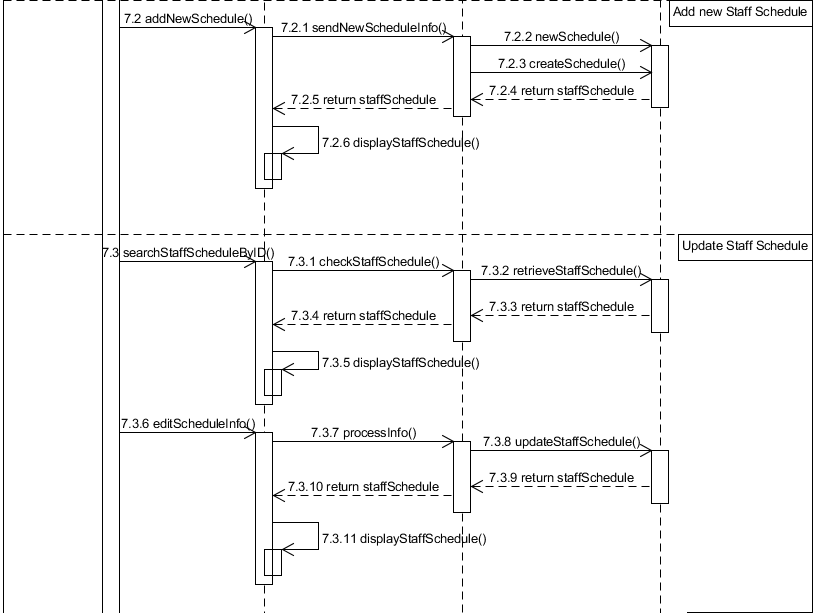
VOPC

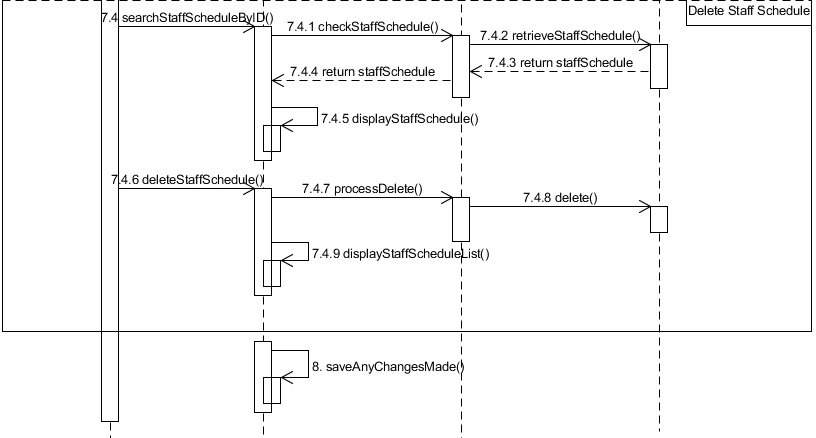


**Manage Housekeeping and Staff Management Module**

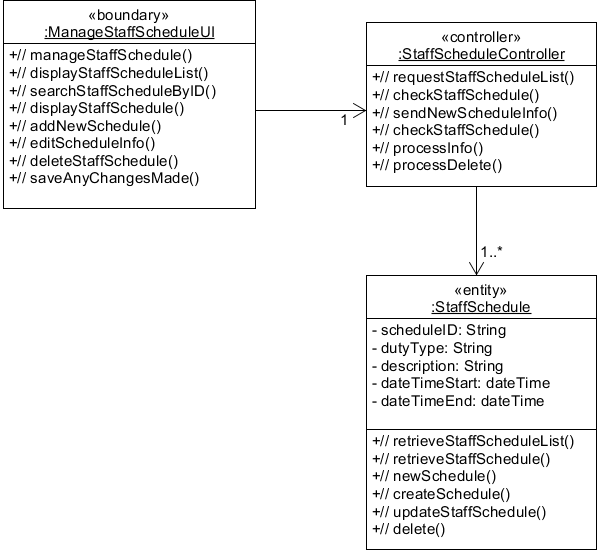
Sequence Diagram





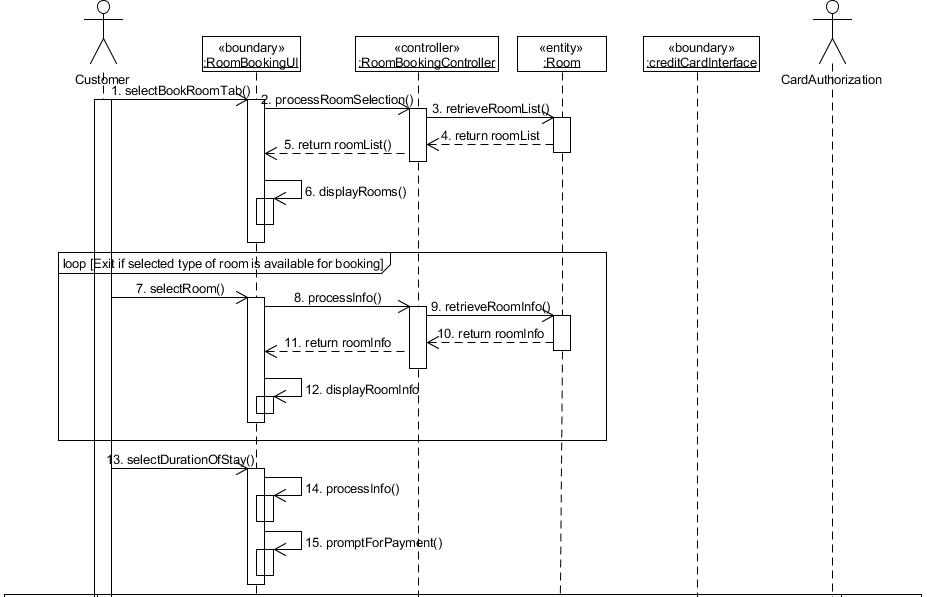


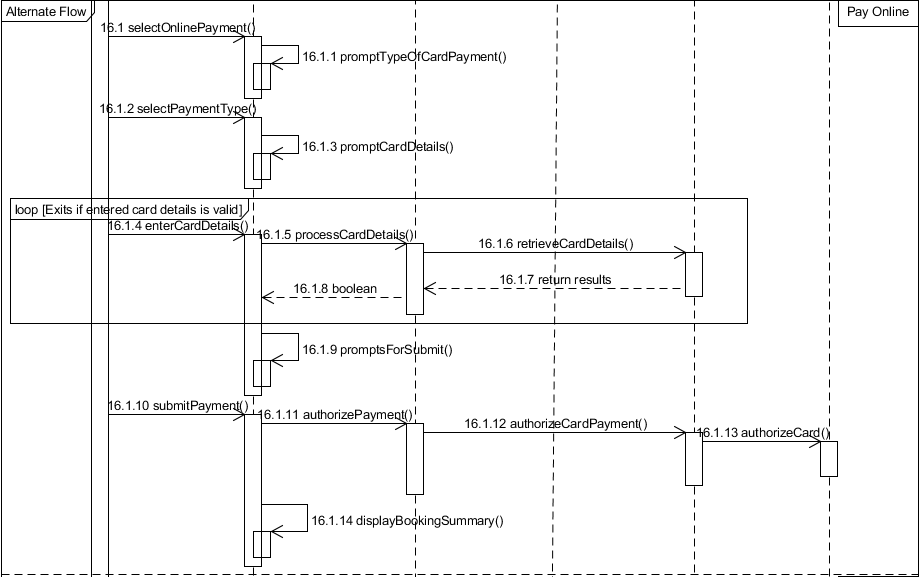
VOPC

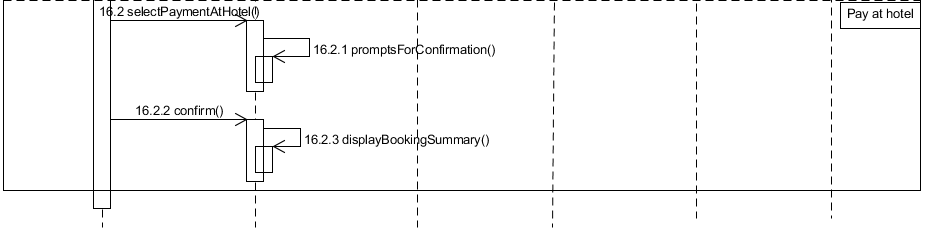


**Online Room Booking**

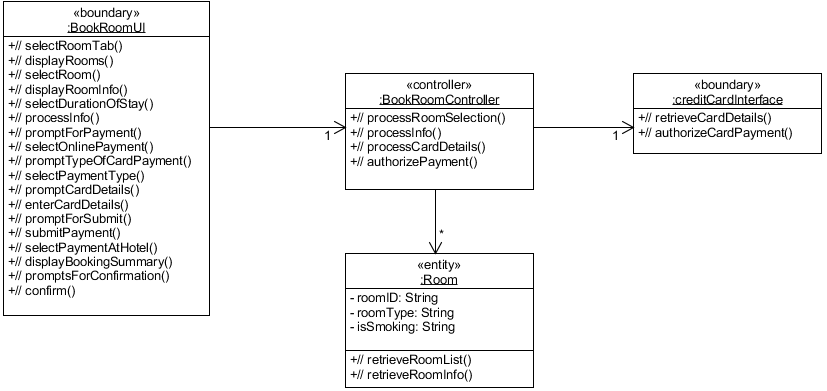
Sequence Diagram







VOPC



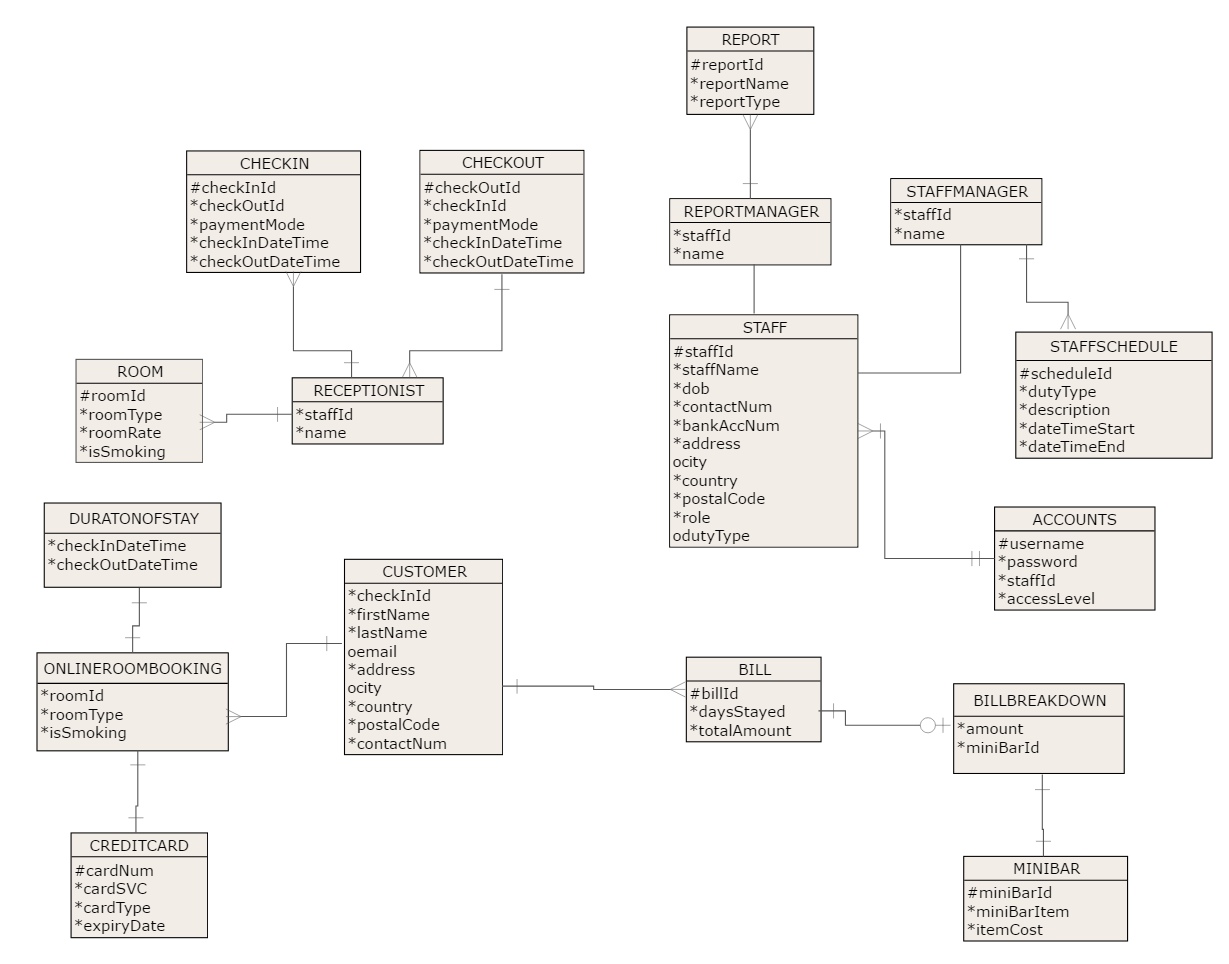
# DATABASE DESIGN

*[Document the database design. What considerations and decisions went into the design of the database schema? How does your database design solve your data storage requirements? Are there any limitations? Highlight portions of the database schema that needs explanation. Highlight also interesting/innovative portions of your database design.]*

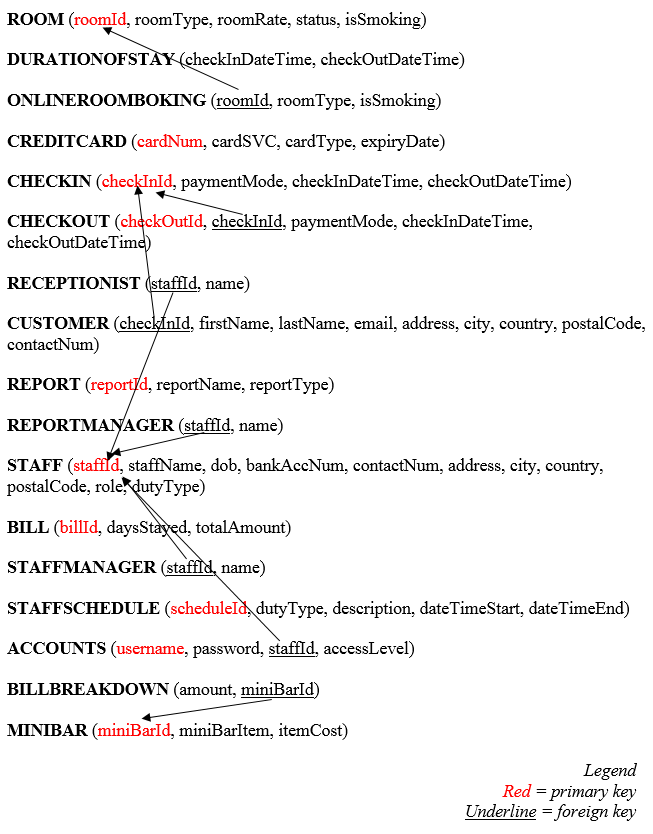
We will be using MySQL to store the database in, to minimize data storage; I’ve separated some of tables to prevent repeat of data or dull data types as it isn’t necessary always filled. For example, MINIBAR table, it is not very frequent that guest will eat from the minibar, so I’ve separate it from the CHECKOUT table. CREDIT is also separated from CHECKOUT table, as some people might use cash to pay and credit card information will be left null, which cause empty field. BILL is separated from CHECKOUT table as I would want BILL to be a separate table for easy extraction.

VARCHAR is used for most data type as it is able to store variable length of alphanumeric data, while CHAR is not used as it has to be a string of fixed length, means that when I put firstName to be CHAR(10) they must put 10 characters no matter how short their name is, this causes inflexibility in name usage, as some name need not be so long. With VARCHAR this is possible as it is able to store a variable length of alphanumeric data. Limitations is that storage of VARCHAR uses dynamic which is 50% slower than CHAR which uses static memory location. For postalCode, I’ve limited it to 10 only as, I’ve searched that the maximum number for a country’s postal code is 10. For phone number it is also up till 15 as I’ve found that the max number for phone is 15. As for the rest of the data type limit, is by assumption.

**Entity Diagram**



**Database Schema**



**Database Dictionary**

