**Hotel Booking Management**

**System**

Author : <Group 2>

Creation Date :

Last Updated : 20 Dec, 2017

**ABSTRACT**

This project is aimed at developing a Hotel booking management system for hotels those are using a manual system to handle hotel processes. There are three main users in hotel management system, administrator, employee and customer. Administrator can maintain daily updates in the hotel records. This system will automate the major operations of the hotel. Administrator can access to all system functionalities without any restrictions. Administrator must be an authorized user. He can further change the password. Employee can only access to the Reservation Management section. The main aim of the entire activity is to automate the process of day to day activities of Hotel.  
The software is to keep track in room and check availability. Using this system user can check which room is reserved and which room is available. He can reserve room as per vacant rooms available. The main objective of the entire activity is to automate the process of day to day activities of Hotel. Using this system you can manage room activities and keep track of admission of a New Customer. Using this system you can check rooms according to customer’s need and can assign easily room to customer. Checkout of a customer and updating the releasing room information in the system is very easy. When user do checkout system will generate final bill after calculating all the bills. User can also check online, all the packages available. He can also book rooms online. He can also cancel booked room online.

**INTRODUCTION**

Currently in hotel all the work done manually. When a guest make a reservation, all the reservation details (including guest details) are recorded in a hotel register. At the time of checkout of customer, calculations of bills and inventory items are done manually too.  Doing all the work manually and storing information on register takes much time and wastes much precious man hours. Manually calculation of bill is also error prone. If management want any old information like room record or reservation details then finding old records is very tiresome task and it takes a lot of time to find records form old files.

In order to avoid the manual tiresome work the online Hotel Booking management system has come in the picture. This software will not only reduce the time required to add new details but would also help register error free data. Users can browse into the site and find their requirements and check the availability sitting back home. Retrieving customer details would become easy and lesser time would be needed. Information at the same time will be secured.

**PROBLEM STATEMENT**

**Existing System**

Currently in hotels all the work is done manually. When a guest make a reservation, all the reservation details (including guest details) are recorded in a hotel register. At the time of checkout of customer, calculations of bills and inventory items are done manually too. User has find manually whether room is available or not.

**Proposed System**

Following is a list of functionalities of the system. Wherever, the description of functionality is not adequate; you can make appropriate assumptions and proceed .There are 3 types of users. The first group of users are “Admin” who has access to add, modify as well as delete records. The second group of users are the “Employee” who has the access to modify records and add them when record. The third group of users are the “Customers” who would book their rooms and avail other facilities as per requirement. Assume that credential details of all the employees are existing in the database.

**Features of the proposed system**

* **Admin** 
  + Login to the system using his/her credentials
  + Perform Hotel Management (add/delete/modify Hotel info like description, any special offers etc)
  + Perform Room Management (add/delete/modify Room info like revised tariff)
  + Generate various reports like:
    - View List of Hotels
    - View Bookings of specific hotel
    - View guest list of specific hotel
    - View bookings for specified date
* **Customer And Employee**
  + Register into the system.
  + Login to the system using his/her credentials.
  + Search for hotel rooms.
  + Book hotel rooms.
  + View Booking Status

**SYSTEM REQUIREMENTS**

The following are the system requirements for feedback management systems

**Hardware Requirements**

* Processor : Intel P-IV system
* Processor Speed : 250MHz to 833MHz
* RAM : 512MB RAM
* Hard Disk : 40GB

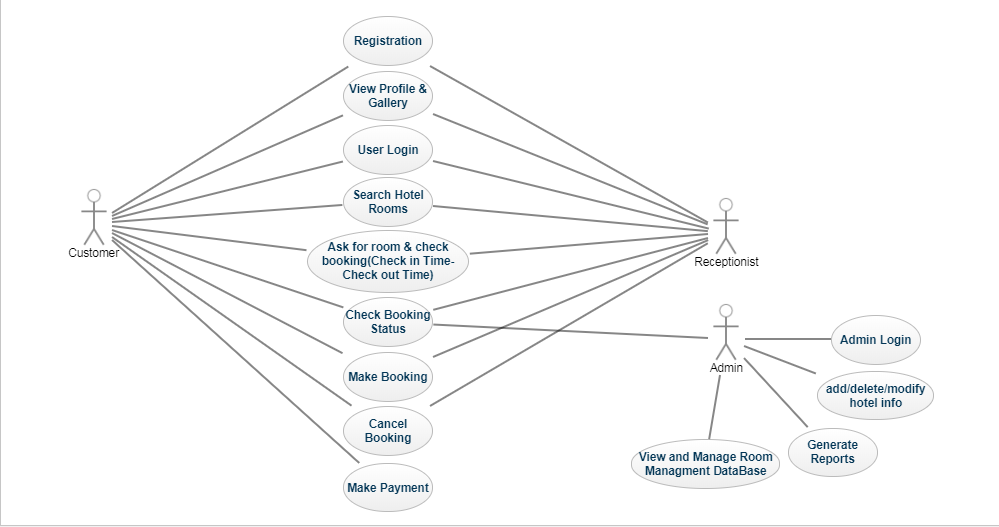
**Software Requirements**

* Front End : Class with main() function
* Business Logic Components
* and Services : Java Beans
* Databases : Oracle 11g

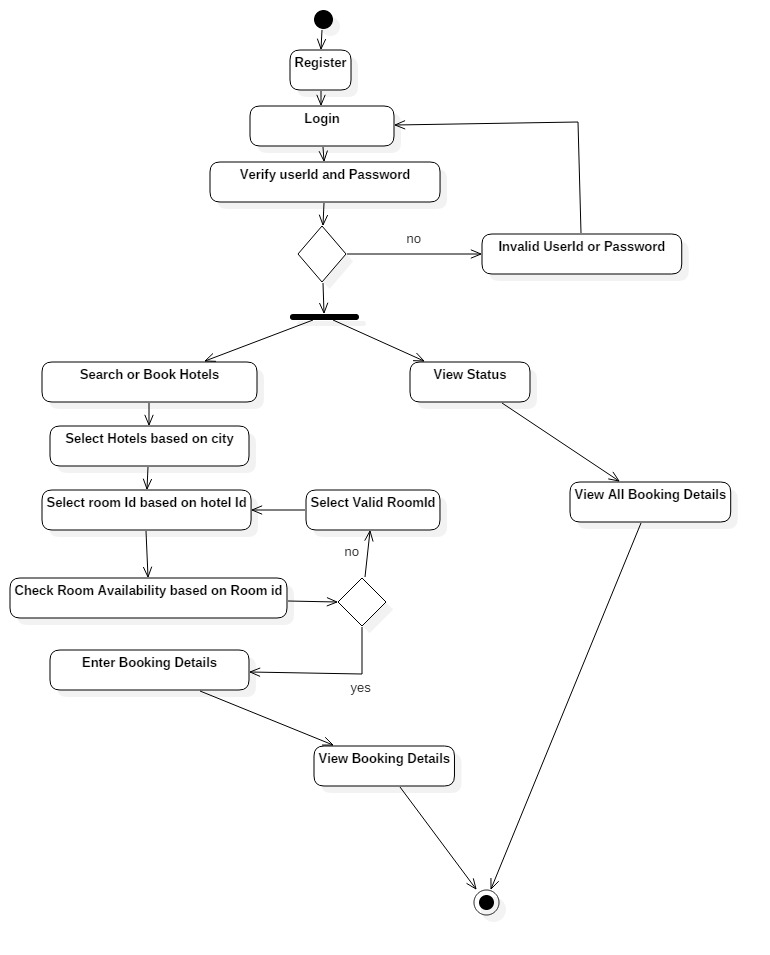
**DESIGN**

This section deals with the activities performed and diagrams created during the design phase of the system. Use case diagram, class diagram and sequence diagrams are created.

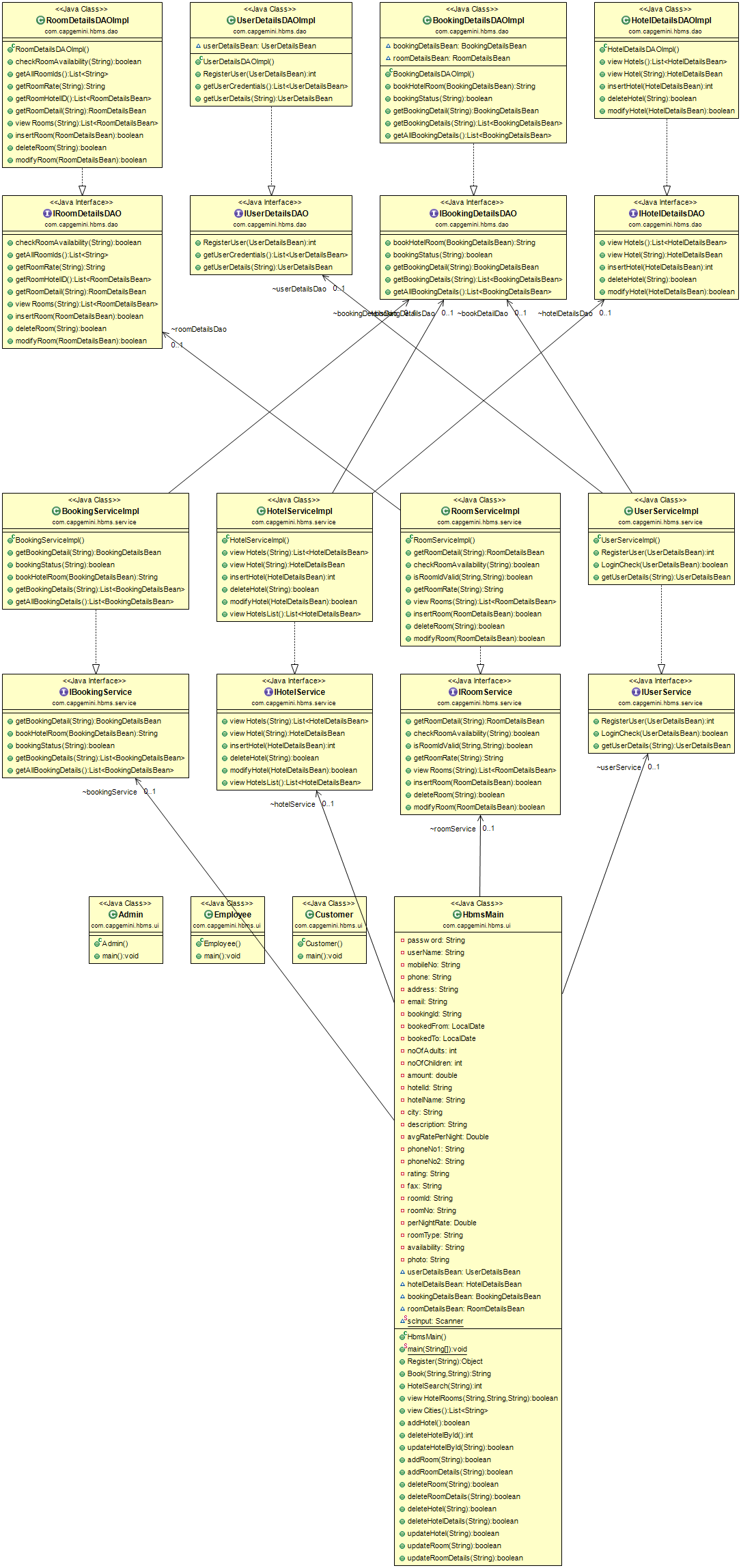
**Use Case Diagram**

****

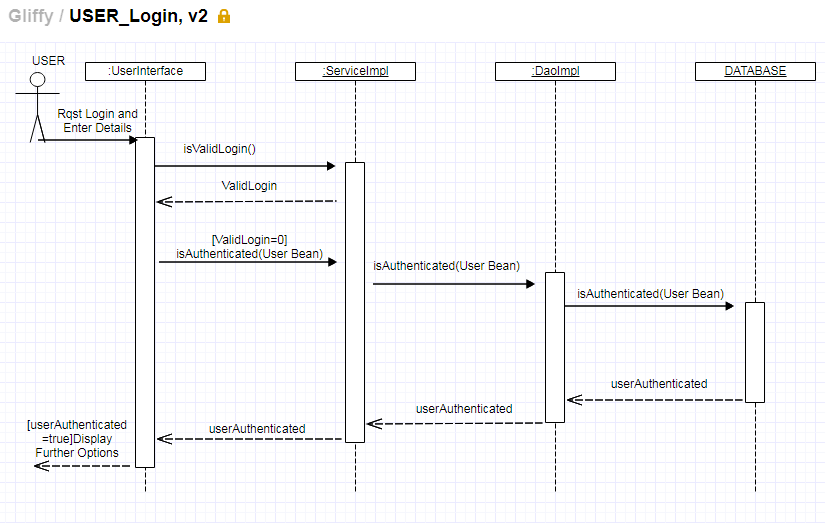
**Activity Diagram**

****

**Class Diagram**

****

**Sequence Diagram**



**Database Design**

Tables Used

1. Name of the table: users

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Description** | **Datatype** | **Size** |
| user\_id | User’s Id | varchar | 4 |
| password | User’s password | varchar | 7 |
| role | Role=’Customer’ || ‘Employee’ | varchar | 10 |
| user\_name | Username of user | varchar | 20 |
| mobile\_no | Mobile number of the user | varchar | 10 |
| phone | Phone number of the user | varchar | 10 |
| address | Address of the user | varchar | 25 |
| email | Email address of the user | varchar | 15 |

1. Name of the table: hotel

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Description** | **Datatype** | **Size** |
| hotel\_id | Hotel Id | varchar | 4 |
| city | City of the hotel | varchar | 10 |
| hotel\_name | Name of the hotel | varchar | 20 |
| address | Address of the Hotel | varchar | 25 |
| description | Description of hotel | varchar | 50 |
| avg\_rate\_per\_night | Average per night of hotel room | number | 6 |
| phone\_no1 | Phone number of hotel | varchar | 10 |
| Phone\_no2 | Alternate phone number | varchar | 10 |
| rating | Rating of hotel | varchar | 4 |
| email | Email of hotel | varchar | 15 |
| fax | Fax of hotel | varchar | 15 |

1. Name of the table: roomdetails

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Description** | **Datatype** | **Size** |
| hotel\_id | Id of hotel | varchar | 4 |
| room\_id | Id of room of the hotel | varchar | 4 |
| room\_no | Room number | varchar | 3 |
| room\_type | Type=’AC’ || ‘Non-AC’ | varchar | 20 |
| per\_night\_rate | Per night rate of hotel | number | 6 |
| availability | Availability of the room | varchar | 3 |
| photo | Future use | blob | - |

1. Name of the table: bookingdetails

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Description** | **Datatype** | **Size** |
| booking\_id | Booking id | varchar | 4 |
| room\_id | Room id | varchar | 4 |
| user\_id | User id | varchar | 4 |
| booked\_from | Date from | date | - |
| booked\_to | Date to | date | - |
| no\_of\_adults | Number of adults | number | 6 |
| no\_of\_users | Number of users | number | 6 |
| no\_of\_children | Number of children | number | 6 |

**TESTING AND RESULTS**

The system developed has to be tested to check whether it confirms to the specification. The testing is done to guarantee that the system is functioning as defined in the requirements specification and is free of known errors and bugs. Using test data and examining the outputs of the system helps to track errors that may present in the system. This is done at various stages of development to ensure that each stage is free of errors.

The different testing strategies used are:

1. Unit testing: Different units of the systems are tested separately. Certain set of inputs are given to the different modules of the system, and output from each module is verified

2. System testing: Entire system is tested as a whole unit. Selected set of inputs are given to the system .Some feedback forms are filled by entering in to the system in user login, then the feedbacks are verified, by analyzing them, by login in to system as administrator

3. Black box testing: Various inputs are given and the outputs are verified. Here the functionality of the system is verified, by giving certain feedback as input, and verifying the retrieved feedback, which is the output. The functionality of the system is tested without referring to the source code of the system

4. White box testing: Different sections of the code is analyzed and tested, that it will work correctly, without any errors and bugs