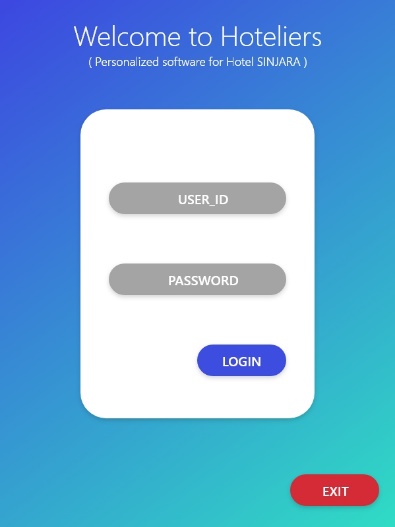
Criterion B – Design

**Prototype of the Application**



Password Field for Administrator / Employees.

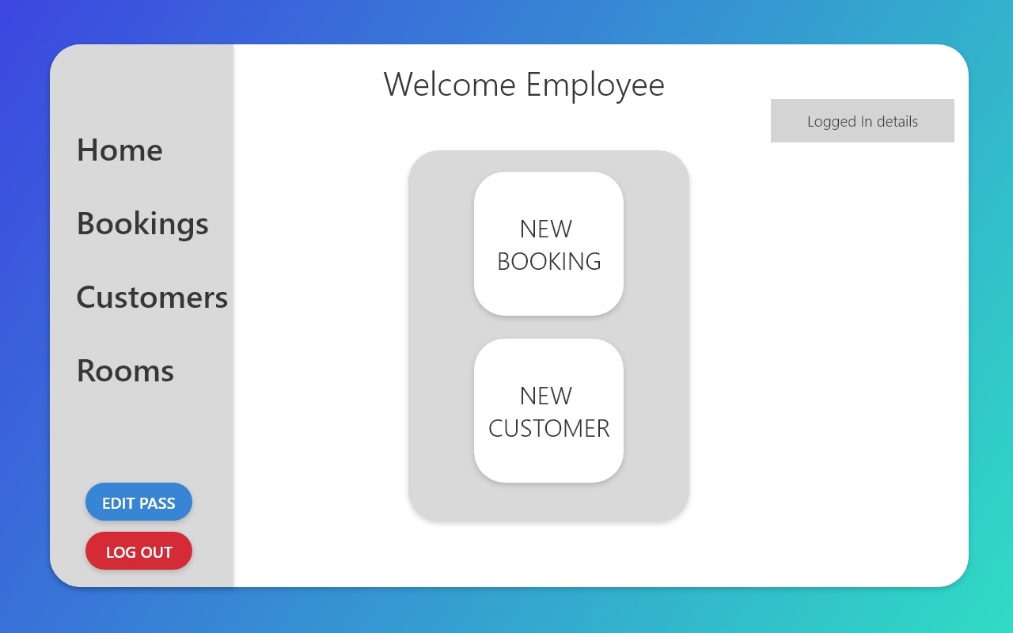
Option for Administrator/ Employees to enter their user ID

Log In with pre-registered Admin ID and Password. These login credentials will be checked with a phpMyAdmin Database.

Login Page of the Application

Quit Application

Button to add New Customer



Edit Password Button

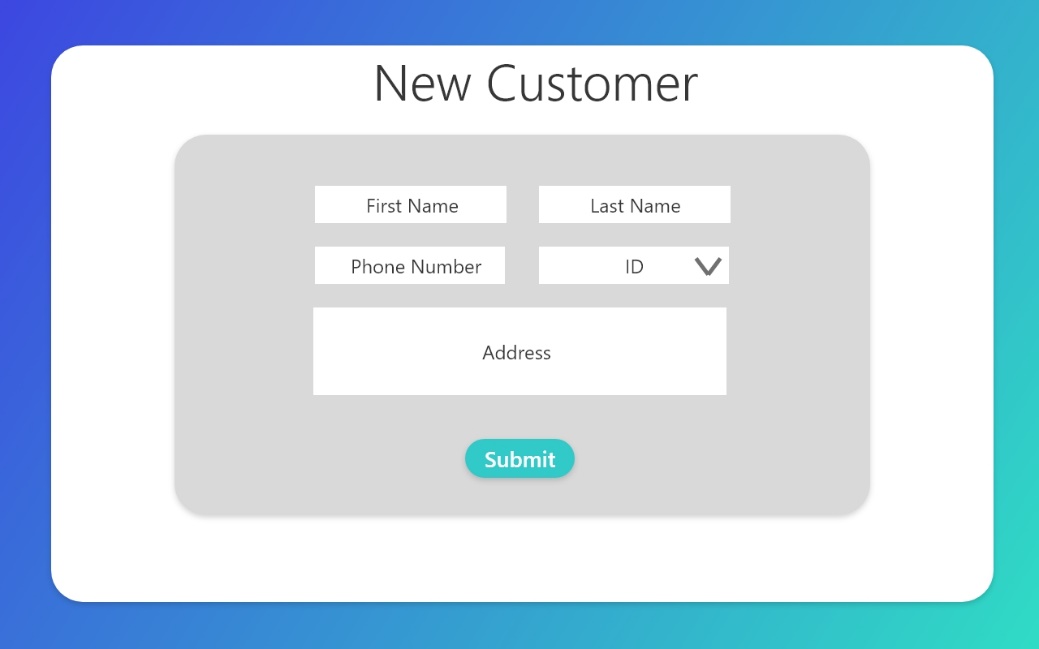
Navigation Panel

Logged In Employee Details

Employee Log In and Landing Page

Button to create New Booking(for existing customer)

Log Out Button



Text Area for Address

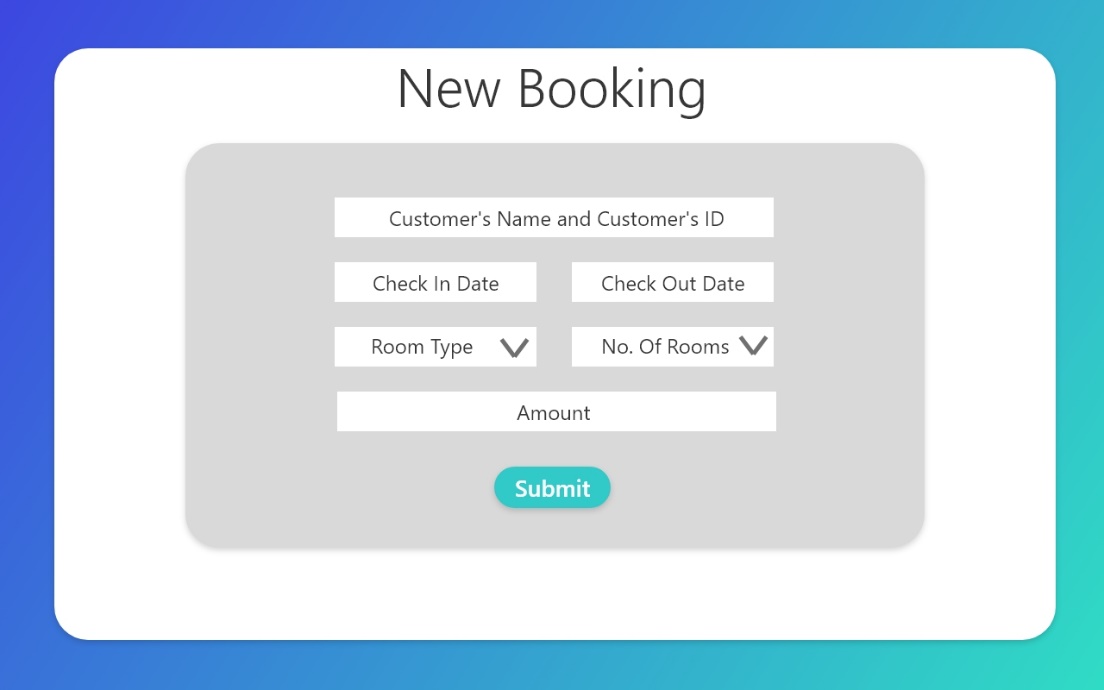
Submit Button navigates to New booking page

Text field for entries.

New Customer

Drop down menu to select ID type (aadhar, voter ID )

New Customer is added to database (if already not present in database).



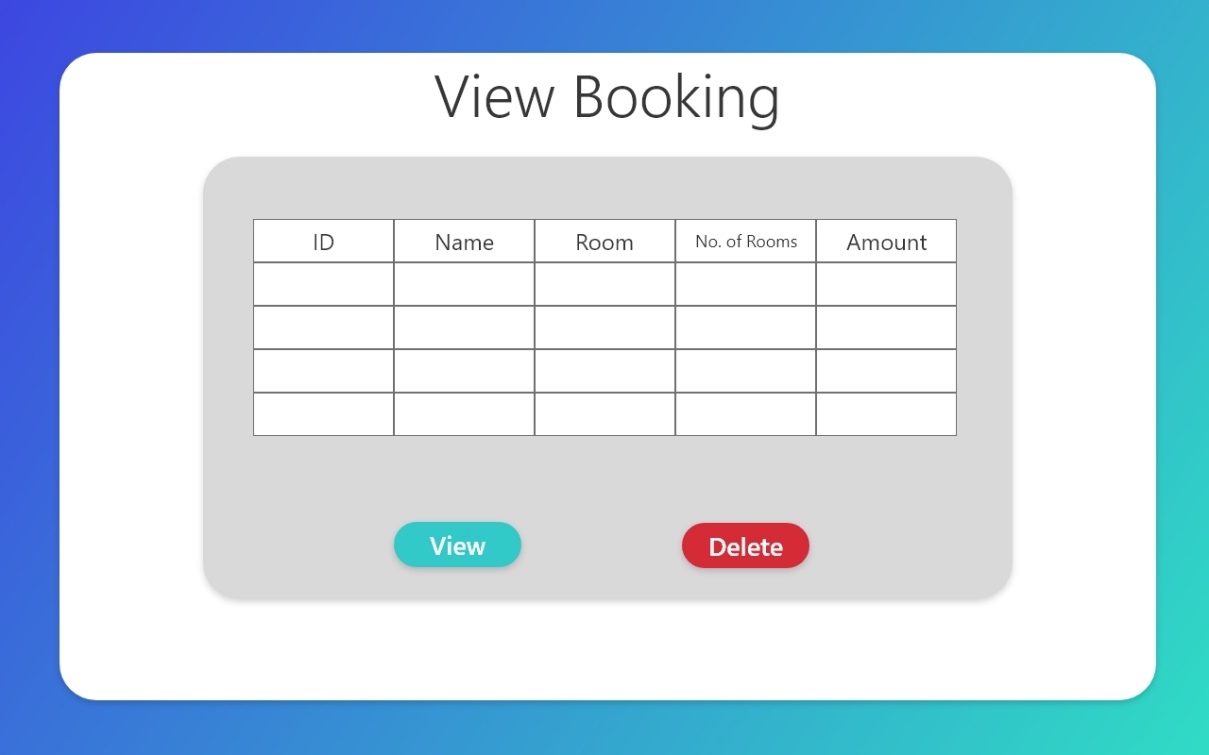
Algorithm automatically generates total amount based on previous entries.

Button to confirm Booking

Customer’s name and ID automatically taken from database on based on phone number (Entered at home page)

Booking Page

Drop down menu to select number of room.

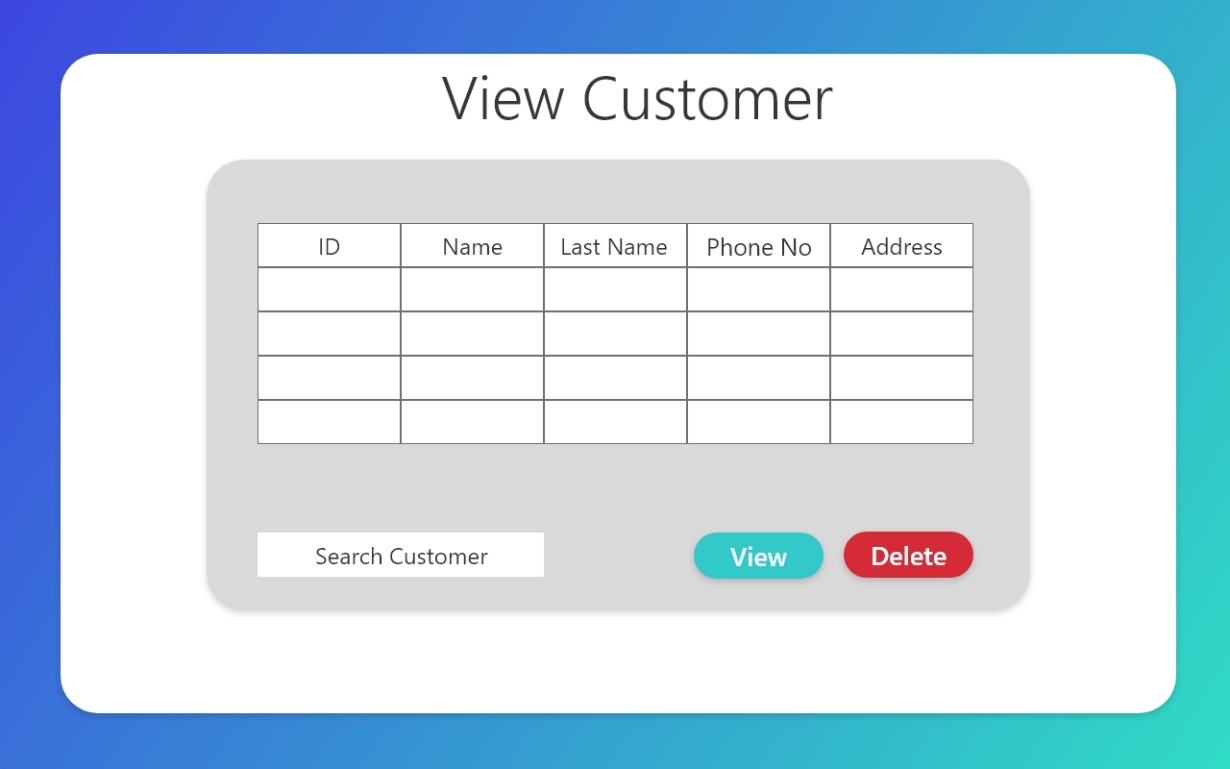


Employee can delete Bookings

Employee can View Bookings

The table displays the essential fields for the booking that the employee decides to view.

View Bookings Page



Employee can delete Bookings

Employee can View Bookings

The table displays the essential fields for the booking that the employee decides to view.

View Customer’s Info Page

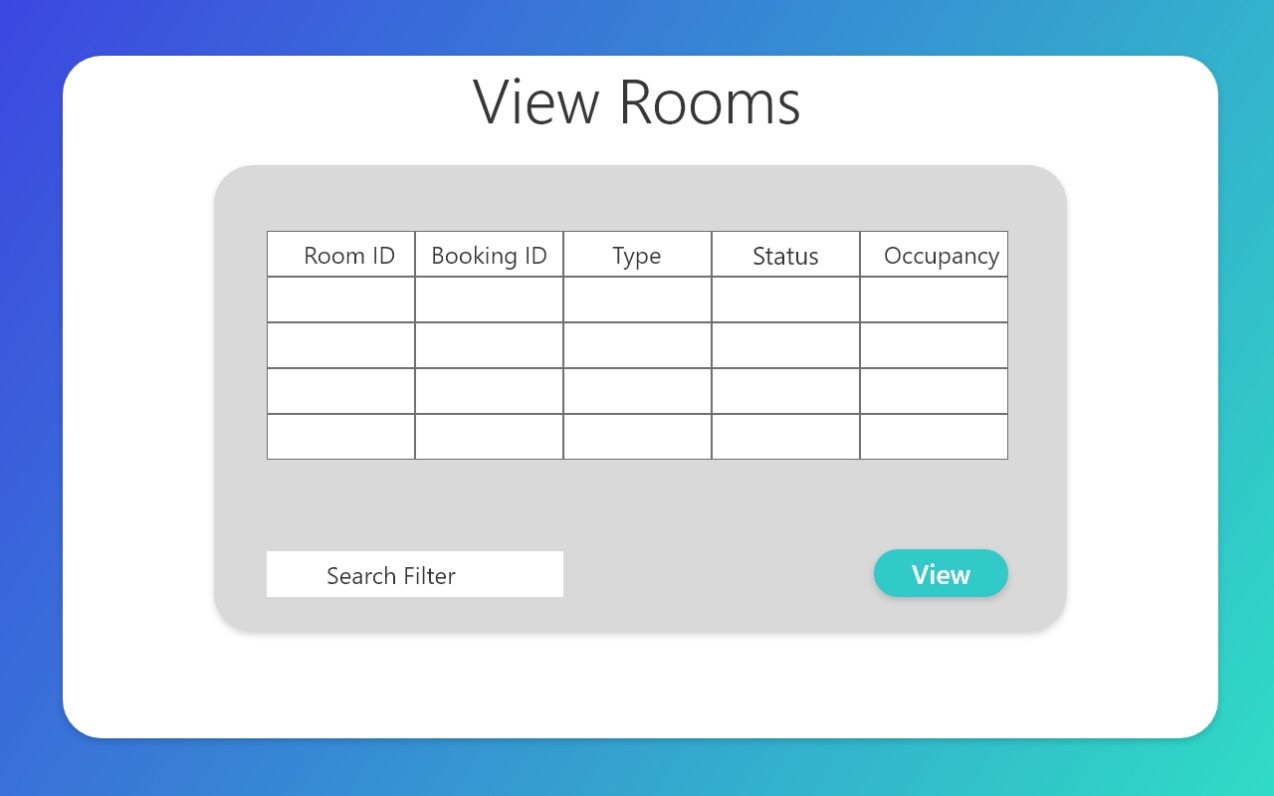
Specific customer’s details can be displayed

Employee can view Rooms

Filter to see occupied Rooms

The table displays the essential fields for the available rooms in Hotel that the employee decides to view.

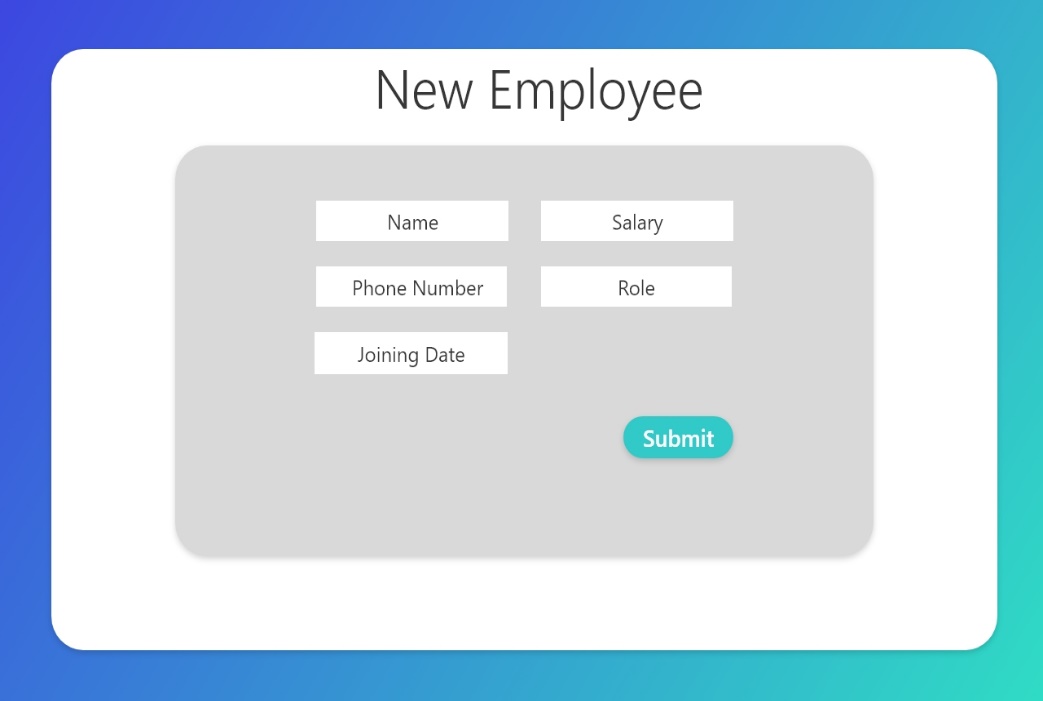
View Rooms Page

****

All of the above diagrams have explained the employee side of the application. I will now move onto explaining the administrator side of the application. This includes equally complex functionality such as searching and leaving a message.

Administrator has all the privileges as that of an employee and can also Add/View/Delete an employee from hotel’s database.

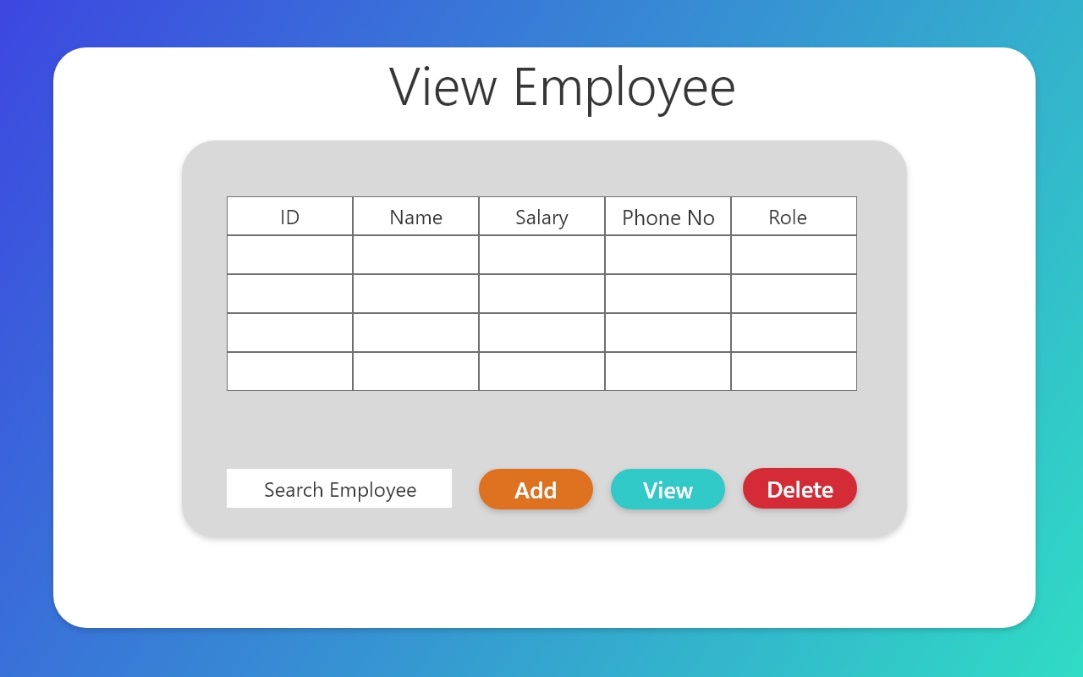
Administrator can also delete customer’s details.



Text field for entries.

Employee (Admin Special Feature)

Button to add new employee to database



Admin can Add/View and Delete Employees

Employee can be searched

The table displays the essential fields for the booking that the employee decides to view.

View Bookings Page

**Table Design**

**Customers Table:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Description** | **Other Information** |
| cust\_id | Integer | Customer ID. | Primary Key, Auto Increment |
| cust\_hotel\_id | Integer | Respective customer’s hotel ID. | Foreign Key |
| cust\_address | String | Customer’s address. |  |
| cust\_id\_type | String | Type of ID proof used by customer. |  |
| cust\_phone | String | Customer’s Phone. | Unique |
| cust\_fname | String | Customer’s First Name. |  |
| cust\_lname | String | Customer’s Last Name. |  |

**Employee Table:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Description** | **Other Information** |
| emp\_id | Integer | Employee ID. | Primary Key, Auto Increment |
| emp\_hotel\_id | Integer | Respective employee hotel ID. | Foreign Key |
| emp\_name | String | Employee Name. |  |
| emp\_salary | Integer | Employee Salary |  |
| emp\_joining\_date | Date | Joining Date of employee |  |
| emp\_role | String | Role/Designation of the employee. |  |
| emp\_phone | String | Contact Number of employee. |  |
| is\_admin | Boolean | Check whether the employee is admin or not. |  |
| emp\_pass | String | Password for login of employee. |  |

**Booking Table:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Description** | **Other Information** |
| book\_id | Integer | Booking ID of the made booking. | Primary Key, Auto Increment |
| book\_cust\_id | Integer | Customer ID of the respective booking. | Foreign Key |
| book\_hotel\_id | Integer | Hotel ID of the respective booking. | Foreign Key |
| book\_room\_count | Integer | Number of rooms booked in the booking. |  |
| book\_amount | Integer | Booking Amount. |  |
| check\_in\_date | Date | Check in Date when customer will arrive. |  |
| check\_out\_date | Date | Check Out Date when customer will leave. |  |

**Rooms Table:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Description** | **Other Information** |
| room\_id | Integer | Room ID/Number. | Primary Key, Auto Increment |
| room\_hotel\_id | Integer | Hotel ID of the respective room. | Foreign Key |
| room\_occupancy | Integer | Number of people which can stay in room together. |  |
| room\_status | String | Vacancy status of the room. |  |
| room\_type | String | Type of the room. |  |
| room\_book\_id | Integer | Book ID of the respected room. | Foreign Key |

**Hotel Table:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Description** | **Other Information** |
| hotel\_id | Integer | Hotel ID. | Primary Key, Auto Increment |
| hotel\_pass | Integer | Hotel password. |  |
| hotel\_name | String | Name of the Hotel. |  |
| hotel\_desc | String | Description about the Hotel. |  |

**Logged In Hotel Id Table:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Description** | **Other Information** |
| id | Integer | ID of the hotel which is currently logged in. |  |
| cust\_id | Integer | Customer ID who is currently logged in. |  |
| emp\_id | Integer | Employee ID who is currently logged in. |  |

**Algorithms & Pseudocode:**

* The following depicts the algorithm and Pseudo code for the Login Functionality:

|  |
| --- |
| Algorithms |
| 1. Program will take input from the user via input fields. 2. Initialize a connection with SQL back end database. 3. Check whether the login user ID and password in the database and login the respective user else show the error dialog. 4. Store the ID of the logged in employee in emp\_id field of logged\_in\_hotel\_id table to store the state of the logged in user. |

|  |
| --- |
| Pseudocode |
| INPUT USER\_ID, PASSWORD  IF USER\_ID == EMPLOYEE.ID AND PASSWORD == EMPLOYEE.PASS:  LOGIN USER, REDIRECT TO NEXT PAGE AND STORE EMP\_ID IN LOGGED\_IN\_HOTEL\_ID TABLE  ELSE:  DIALOG (‘INVALID ENTRIES’) |

* The following depicts the algorithm and Pseudo code for the Amount Calculation:

|  |
| --- |
| Algorithms |
| 1. Input the check-in, check-out dates, room type and number of rooms from the user via input fields. 2. Let check-in date = cid, check-out date = cod, room type factor = rtf, number of rooms = nr. 3. There will be a room type factor by which the whole amount will be multiplied:   Total Amount = (cod - cid) \* rtf \* nr.   1. Will display the generated amount on the GUI when user clicks generate amount. |

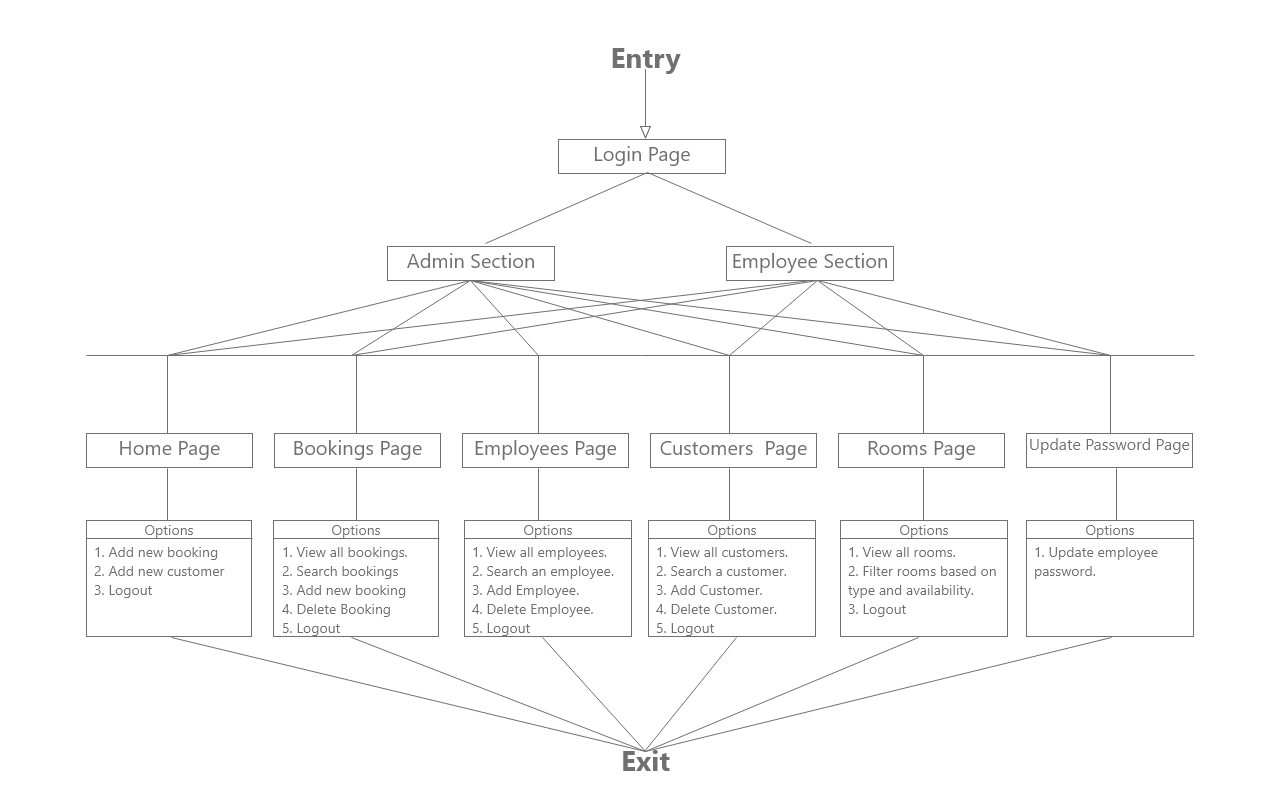
|  |
| --- |
| Pseudocode |
| INPUT CHECK-IN DATE, CHECK-OUT DATE, ROOM TYPE, NUMBER OF ROOMS  COD = CHECK-IN DATE  CID = CHECK-OUT DATE  RTF = ROOM TYPE FACTOR  NR = NUMBER OF ROOMS  AMOUNT = (COD - CID) \* RTF \* NR |

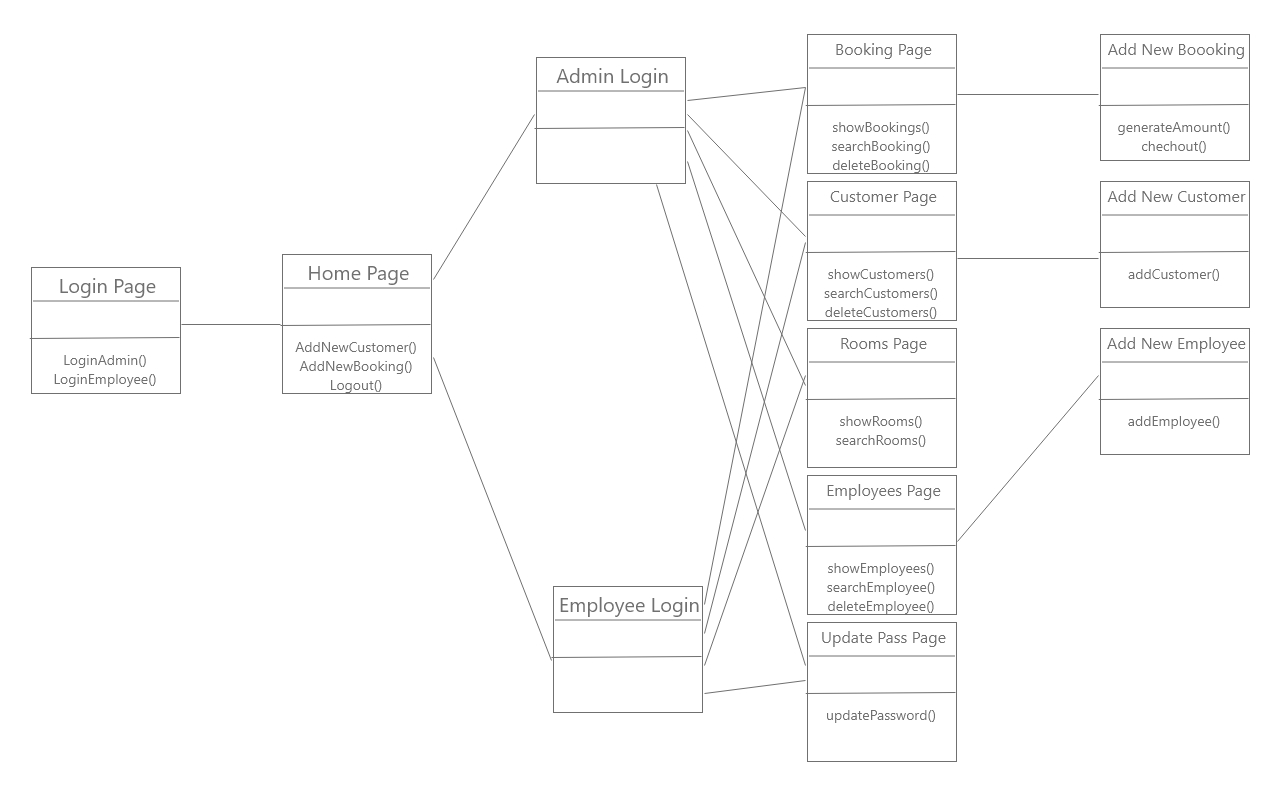
* The following depicts the algorithm and Pseudo code for fetching the specific customer name and ID for which the booking is going:

|  |
| --- |
| Algorithms |
| 1. Fetch the customer ID from the logged\_in\_hotel\_id table in which the customer ID is stored and employee adds a customer to make a booking. 2. Store it in a variable. 3. Fetch the employee details required like emp\_fname from the employee table querying via emp\_id stored in the previous step. 4. Set text of the label with employee name and employee ID to show on the New Booking Form. |

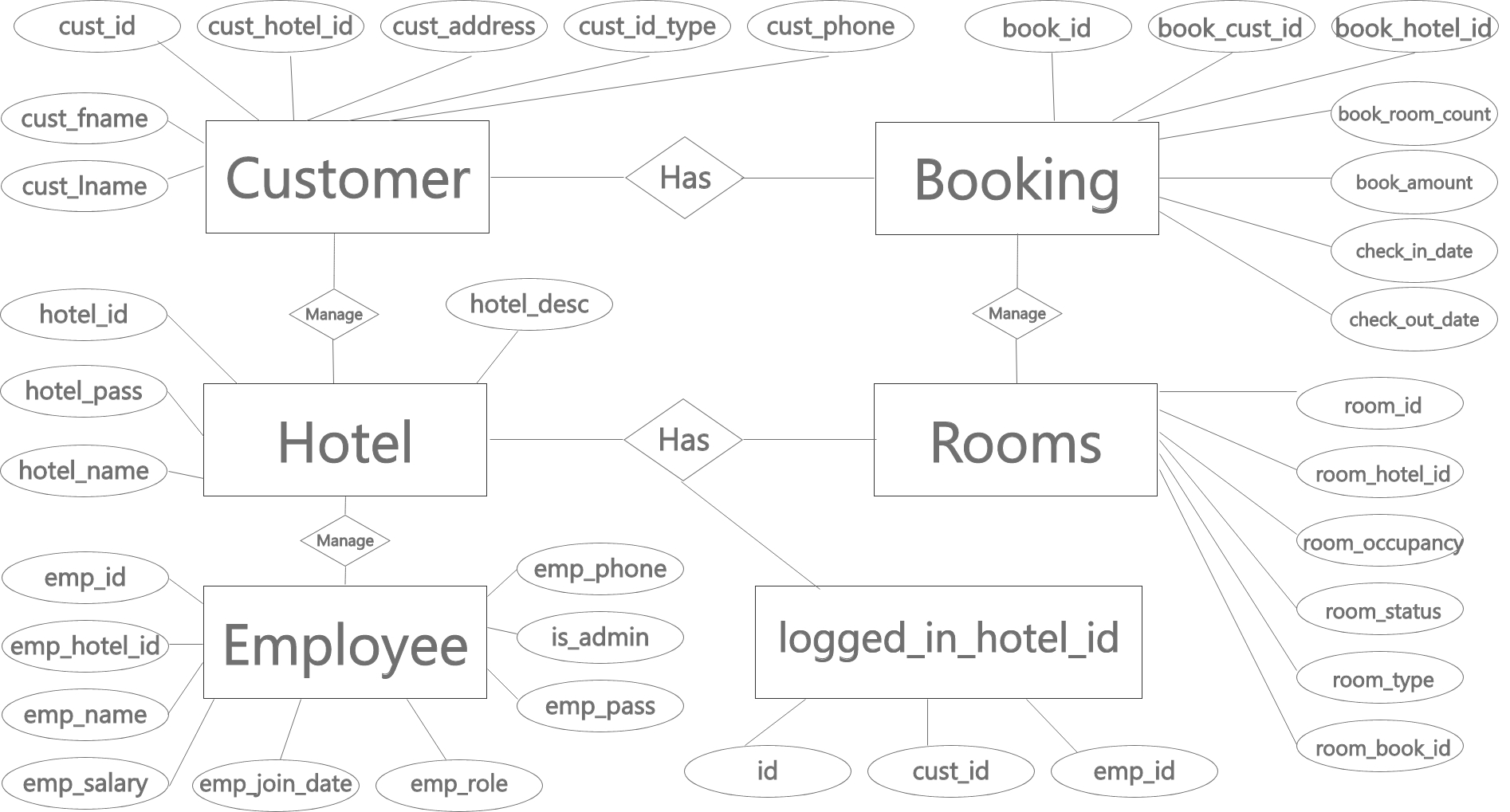
|  |
| --- |
| Pseudocode |
| RECENT\_EMP\_ID = SELECT EMP\_ID FROM LOGGED\_IN\_HOTEL\_ID  SELECT EMP\_FNAME, EMP\_LNAME FROM EMPLOYEE WHERE EMP\_ID = RECENT\_EMP\_ID  SET TEXT (EMP\_FNAME)  SET TEXT (RECENT\_EMP\_ID) |

**Structure Diagram:**

****

**UML Diagram:**

**E-R Diagram:**

****

**Test Plan:**

|  |  |
| --- | --- |
| **Success Criteria To Be Tested** | **Example** |
| **Linked to Success Criteria 1**  User login test – to test whether the User is able to login to the system. | **Correct Data Input**  User ID: 7  Password: admin123  Click: Login 🡪 Login Successful message shown  **Incorrect Data Input**  User ID: 7  Password: admin1234  Click: Login 🡪 Login Unsuccessful message shown |
| **Linked to Success Criteria 2, 3**  Employee login test – to test whether the employee is able to login to the system. | **Correct Data Input**  Customer’s Phone Number: 6367262034  Click: Book Now 🡪 Redirect to create new booking page  Click: Add New Customer 🡪 Redirect to add new customer page  **Incorrect Data Input**  Customer’s Phone Number: 6367262028  Click: Book Now 🡪 Shows an error.  Click: Add New Customer 🡪 Redirect to add new customer page |
| **Linked to Success Criteria 4**  Test for new customer to sign up and to be added to the system. | **Correct Data Input**  First Name: Aviral  Last Name: Singh  Phone Number: 6367262026  Customer ID: PAN  Address: 313 amar nagar c  Click: ADD 🡪 confirm message displayed and customer added  **Incorrect Data Input**  First Name: Aviral  Last Name: Singh  Phone Number: dfsjaldjfsadf  Customer ID: PAN  Address: 313 amar nagar cx  Click: ADD 🡪 error message displayed  **Validation Check** takes place at phone number does not have integer values. |
| **Linked to Success Criteria 5**  Test for new booking to be added to the system. | **Correct Data Input**  Check-In Date: 2015-05-16  Check-Out Date: 2015-08-13  Room Type: Double  Number of Rooms: 2  Click: Generate Amount 🡪 amount will be displayed and Check Out button will display.  Click: Check Out 🡪 Booking added to system and redirect to booking listing page.  **Incorrect Data Input**  Check-In Date: 2015-05-16  Check-Out Date: 8-05-15  Room Type: Double  Number of Rooms: 2  Click: Generate Amount 🡪 error will be displayed  **Validation Check** takes place at date field doesn’t have date input in desired format. |
| **Linked to Success Criteria 6**  Test for new employee to sign up and to be added to the system. | **Correct Data Input**  Name: Shreeram  Salary: 10000  Phone Number: 9632587410  Role: Chef  Joining Date: 2015-02-19  Click: ADD 🡪 Show success message and add employee and redirect to employee listing table page  **Incorrect Data Input**  Name: Shreeram  Salary: ten thousand  Phone Number: 9632587410  Role: Chef  Joining Date: 2015-02-19  Click: ADD 🡪 show error message  **Validation Check** takes place at phone number field doesn’t have date input in desired format. |
| **Linked to Success Criteria 7,8,10,13,14,15**  Test for admin and employee to see the list of customers, bookings and employees | Click: Show Records 🡪 Fetch the records from database and display it in the table.  Click: Add Records 🡪 Redirect to the add new record page based on the current page user is on.  Click: Delete Employee 🡪 Delete Record from the table and from the database.  Search Field: Name of the suitable query as indicated in the text field placeholder. |
| **Linked to Success Criteria 9, 16**  Test for admin and employee to see the list of rooms available in the hotel | Click: Show Rooms 🡪 Fetch the rooms from the database and display it in the table.  Room Type Dropdown: Single  Show Available Rooms Checkbox: check the checkbox  Click: Search 🡪 Will filter out the result based on the type and availability of the rooms. |
| **Linked to Success Criteria 12**  Test for update password for admin and employee | **Correct Data Input**  Old Password: pass123  New Password: password12345  Re-Type New Password: password12345  Click: Update 🡪 show success message and close the update page window  **Incorrect Data Input**  Old Password: pass123  New Password: password123  Re-Type New Password: password1234  Click: Update 🡪 show error message  **Validation Check** takes place at Re-Type password field don’t having same password as New password field |
| **Linked to Success Criteria 17**  Test for logout button to log the user out of application | Click: Logout 🡪 log the user out, clear the user session from database and redirect the user to log in page. |