

Hidalgo Hotel Website

Rafael Hidalgo

Montclair State University

Web Development

Dr. Jenq

CSIT537-01

Contents

[Description of Hotel System 3](#_Toc146554829)

[System Components 4](#_Toc146554830)

[Database Design 4](#_Toc146554831)

[Assumptions 5](#_Toc146554832)

[Version 1 ERD 7](#_Toc146554833)

[Version 2 ERD 9](#_Toc146554834)

[Version 3 ERD 10](#_Toc146554835)

[Relational Schema 11](#_Toc146554836)

[MVC 12](#_Toc146554837)

[Testing 12](#_Toc146554838)

[Test Plan: User Interface Testing 12](#_Toc146554839)

[Objectives: 12](#_Toc146554840)

[Scope: 12](#_Toc146554841)

[Test Cases: 13](#_Toc146554842)

[Test Steps: 13](#_Toc146554843)

[Expected Results: 15](#_Toc146554844)

[Actual Results: 15](#_Toc146554845)

[Test Discussion 44](#_Toc146554846)

[Conclusion 45](#_Toc146554847)

[References 45](#_Toc146554848)

[Video Link For Demo 45](#_Toc146554849)

# Description of Hotel System

The hotel booking system is a comprehensive web application that facilitates a wide range of functionalities for guests and hotel staff. It is built using several programming languages including PHP, HTML, CSS, and JavaScript, and it utilizes a MySQL database to store relevant information. The system enables guests to register and create an account by providing their personal information. This enables them to log in and make future bookings more efficiently.

The room booking component of the system is one of the most critical functionalities. Guests can search for available rooms based on their preferred check-in and check-out dates, room type, and number of occupants. The system ensures that rooms are not double-booked by checking the check-in and check-out dates of other bookings. Once guests have selected their preferred room, they can complete their booking and receive a confirmation number.

In addition to the room booking functionality, the system also allows guests to search for their reservation details without having to log in. This component of the system is crucial for guests who may have forgotten their login credentials or who have lost their confirmation number.

The Admin Panel is another essential component of the hotel booking system. It allows hotel staff to manage the system efficiently. The admin panel enables staff to view room information, edit room information, add new rooms, and delete rooms from the system. The Admin Panel is password protected to ensure that only authorized personnel have access to it.

The security and privacy of guest information is paramount to the system. To ensure this, the system implements password hashing and encryption when storing sensitive data in the database. This ensures that guest information is secure and cannot be easily accessed by unauthorized personnel.

Overall, the hotel booking system is user-friendly and convenient for guests while enabling hotel staff to manage the system efficiently. The system provides a comprehensive set of functionalities, including guest registration, room booking, reservation search, and an admin panel.

# System Components

## Database Design

In the hotel booking system, the database design is an important component that ensures the system functions efficiently and accurately. To design the database, a list of assumptions was first made using the project requirements. Once the assumptions were made, three versions of an entity relationship diagram (ERD) were made. Finally, with the ERD’s ready, a relationship diagram was made to help us visualize how our data will look once it is in SQL.

Assumptions are made to help visualize how the database system should look and how the components of a database should interact with one another. They serve as a starting point for creating the database schema, and they help to identify the necessary entities and attributes required to store and manage the data. Assumptions also help to identify any potential problems or limitations that may arise during the database design process.

After the assumptions the first version of the entity relationship diagram (ERD) was created to identify each entity individually and determine the attributes associated with it. Entities such as guests, rooms, reservations, and admins were identified, and attributes such as guest names, room numbers, check-in and check-out dates, and admin login details were identified for each entity. The second version of the ERD was created to visualize the relationship between the entities and to eliminate any redundant relationships.

The final version of the ERD was created to view the constraints and cardinalities of each relationship. This helped to identify which attributes should be used as keys to ensure data integrity and accuracy. For example, the reserve\_room entity has a composite key consisting of the confirmation ID and room number. This helps us monitor what rooms are reserved and at that time.

Finally a relationship diagram was made by consolidating all of the ERD’s and assumptions. With a relationship diagram, were are able to visualize what schemas required foreign keys, and what relationships should be treated as schema themselves. This ultimately helps us to make a reliable database with the least amount of resources as possible. Please see below for the assumptions, ERD’s and relationship diagram used.

### Assumptions

1. Room
   1. Room\_number
   2. Room\_type
   3. {Room\_imgs}
   4. Room\_availability
   5. Max\_occupancy
   6. Price\_per\_night
2. Guest User
   1. Has Name
      1. First
      2. Middle
      3. Last
   2. Address
      1. Apartment Number
      2. Street Number
      3. Street Name
      4. City
      5. Zip Code
      6. State
      7. Country
   3. {Has Phone Number}
   4. {Has Email Address}
   5. User\_name
   6. Password
   7. User I.D.
3. Reservation
   1. Confirmation number
   2. Check in Date
   3. Check out Date
   4. Number of adults
   5. Number of Children
   6. Total\_cost
   7. Time stamp created
4. Admin User
   1. Has Name
      1. First
      2. Middle
      3. Last
   2. Address
      1. Apartment Number
      2. Street Number
      3. Street Name
      4. City
      5. Zip Code
      6. State
      7. Country
   3. {Has Phone Number}
   4. {Has Email Address}
   5. User\_name
   6. Password
   7. User I.D.

Interactions of hotel Database

Admin User manages rooms

Admin User can manage reservation

Guest User can book Reservations

Reservations reserve a room.

### Version 1 ERD

Diagram, schematic

Description automatically generated

### Version 2 ERD

Diagram, schematic, map

Description automatically generated

### Version 3 ERD

Map

Description automatically generated

### Relational Schema

Graphical user interface

Description automatically generated

## MVC

The hotel management system employs SQL queries, HTML, and PHP to implement an MVC (Model-View-Controller) structure. This structure is characterized by dividing the application into three separate components. The first component, the model, represents the business logic of the system. It comprises the database schema and all the methods for interacting with the database. In the hotel management system, the model incorporates tables for reservations, rooms, guests, and other necessary information. The model is basically the SQL database itself, along with the PHP code and PDO/SQL commands used to retrieve the data,

The second component is the view, which handles the user interface and presentation of data to the user. In the hotel management system, the view includes the HTML templates and CSS stylesheets for displaying information about reservations, rooms, and guests.

The third component is the controller, which manages the flow of data between the model and the view. The controller contains the application logic for handling user input and generating appropriate responses. In the hotel management system, the controller receives user input from the search form and uses the model to retrieve and display reservation data in the view. This was done mostly by using PHP.

# Testing

## Test Plan: User Interface Testing

### Objectives:

* To ensure that the user interface of the hotel management system is user-friendly and intuitive.
* To ensure that all the functionalities of the system are easily accessible to the users.
* To ensure that the system is responsive and works well on different devices and screen sizes.

### Scope:

* The testing will be performed on the user and admin interface of the hotel management system.
* The testing will include both desktop and mobile versions of the system.

### Test Cases:

1. Navigation: Test the navigation of the system to ensure that all the functionalities of the system can be easily accessed by the users. Verify that all links and buttons are working correctly and that they lead to the appropriate pages.
2. User Input: Test the input fields of the system to ensure that they are working correctly and that the users can input their information without any errors. Verify that input fields have appropriate labels and placeholders.
3. Responsiveness: Test the responsiveness of the system to ensure that it works well on different devices and screen sizes. Verify that the system's layout adjusts appropriately to different screen sizes.
4. User Feedback: Test the feedback mechanisms of the system to ensure that the users are notified of any errors or successful actions. Verify that error messages are clear and concise and that success messages are displayed appropriately.

### Test Steps:

#### User

##### Navigation:

1. Click on login button on the homepage. Verify that the login page is loaded.
2. Click on Register button on the login page. Verify that the registration page is loaded.
3. Click on Rooms button on the homepage. Verify that the Rooms page is loaded.
4. Click on the Reservation Search page. Verify the page is loaded.
5. Click on Facilities, Local Travel, and Fine Dining buttons on the homepage. Verify that the respective pages are loaded.
6. In the Rooms Page, Click on Book Now while the user is logged off. Verify that the user is redirected to the login page.
7. Click on Book Now! button on the Rooms page while the user is logged on. Verify that the user is redirected to the booking page.

##### Input Data:

1. Enter valid details in the registration form and click on the register button. Verify that the user is registered and redirected to the login page.
2. Enter valid credentials in the login form and click on the login button. Verify that the user is logged in and redirected to the homepage.
3. In the Rooms page, enter valid check-in and check-out dates, room type, and click on the Book Now! button. Verify that the booking is confirmed, and the user is redirected to the booking page.
4. In the booking page, input 1 adult and 1 child. Click on book now. Should bring you to the confirmation page.

##### Responsive Design:

1. Resize the browser window to various screen sizes and verify that the layout adjusts accordingly.

##### Error Handling:

1. Enter invalid credentials in the login form and click on the login button. Verify that the system displays an error message.
2. Submit the registration form with invalid details and verify that the system displays appropriate error messages.
3. In the booking page submit the booking form with invalid details and verify that the system displays appropriate error messages.

#### Admin

##### Navigation:

1. Log in to the admin panel and verify that the dashboard is loaded.
2. Click on Rooms on the sidebar menu. Click on Add Room and Edit. Verify that the respective pages are loaded.

##### Input Data:

1. Add a new room by filling in the necessary details in the Add Room form and clicking on the Add Room button. Verify that the room is added to the system.
2. Update the details of an existing room by editing the details in the Edit Room form and clicking on the Edit Room button. Verify that the room details are updated.
3. Delete a room by clicking on the Delete button for that room in the Rooms page and verifying that the room is removed from the system.

##### Responsive Design:

1. Resize the browser window to various screen sizes and verify that the layout adjusts accordingly.

##### Error Handling:

1. Add a new room with invalid details and verify that the system displays appropriate error messages.
2. Update the details of a room with invalid details and verify that the system displays appropriate error messages.
3. Attempt to delete a room that has active bookings and verify that the system displays an appropriate error message.

### Expected Results:

* The system's user interface should be intuitive and easy to use.
* The system should be responsive and work well on different devices and screen sizes.
* The system should provide clear and concise feedback to the users.

### Actual Results:

#### User

##### Navigation:

1. Click on login button on the homepage. Verify that the login page is loaded.

Graphical user interface, application

Description automatically generated

1. Click on Register button on the login page. Verify that the registration page is loaded.

Graphical user interface, application

Description automatically generated

1. Click on Rooms button on the homepage. Verify that the Rooms page is loaded.

Graphical user interface, website

Description automatically generated

1. Click on the Reservation Search page. Verify the page is loaded.

Graphical user interface, text, application, email

Description automatically generated

1. Click on Facilities, Local Travel, and Fine Dining buttons on the homepage. Verify that the respective pages are loaded.

Graphical user interface, text, application, email

Description automatically generated

1. In the Rooms Page, Click on Book Now while the user is logged off. Verify that the user is redirected to the login page.

Graphical user interface

Description automatically generated

1. Click on Book Now! button on the Rooms page while the user is logged on. Verify that the user is redirected to the booking page.

Graphical user interface

Description automatically generated with medium confidence

##### Input Data:

1. Enter valid details in the registration form and click on the register button. Verify that the user is registered and redirected to the login page.

Graphical user interface

Description automatically generated

1. Enter valid credentials in the login form and click on the login button. Verify that the user is logged in and redirected to the homepage.

Graphical user interface

Description automatically generated

1. In the Rooms page, enter valid check-in and check-out dates, room type, and click on the Book Now! button. Verify that the booking is confirmed, and the user is redirected to the booking page.

Graphical user interface, text

Description automatically generated

1. In the booking page, input 1 adult and 1 child. Click on book now. Should bring you to the confirmation page.

Graphical user interface

Description automatically generated

##### Responsive Design:

1. Resize the browser window to various screen sizes and verify that the layout adjusts accordingly.

Graphical user interface, application

Description automatically generated Graphical user interface, application

Description automatically generated

##### Error Handling:

1. Enter invalid credentials in the login form and click on the login button. Verify that the system displays an error message.

Graphical user interface, website

Description automatically generated

1. Submit the registration form with invalid details and verify that the system displays appropriate error messages.

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

1. In the booking page submit the booking form with invalid details and verify that the system displays appropriate error messages.

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, application, Word

Description automatically generated

#### Admin

##### Navigation:

1. Log in to the admin panel and verify that the dashboard is loaded.

Graphical user interface

Description automatically generated

Text

Description automatically generated

1. Click on Rooms on the sidebar menu. Click on Add Room and Edit. Verify that the respective pages are loaded.

Graphical user interface, text, website

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

##### Input Data:

1. Add a new room by filling in the necessary details in the Add Room form and clicking on the Add Room button. Verify that the room is added to the system.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

1. Update the details of an existing room by editing the details in the Edit Room form and clicking on the Edit Room button. Verify that the room details are updated.

Graphical user interface, website

Description automatically generated

Graphical user interface, text, website

Description automatically generated

1. Delete a room by clicking on the Delete button for that room in the Rooms page and verifying that the room is removed from the system.

Graphical user interface

Description automatically generated

Graphical user interface, text

Description automatically generated

##### Responsive Design:

1. Resize the browser window to various screen sizes and verify that the layout adjusts accordingly.

Graphical user interface

Description automatically generated

##### Error Handling:

1. Add a new room with invalid details and verify that the system displays appropriate error messages.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

1. Update the details of a room with invalid details and verify that the system displays appropriate error messages.

A screenshot of a computer

Description automatically generated with medium confidence

A computer screen capture

Description automatically generated with medium confidence

Graphical user interface, text, timeline

Description automatically generated

### Test Discussion

#### User

Upon testing the User system, no apparent bugs were found. Please note however that any bugs found during development were fixed and accounted for at the time of development.

#### Admin

Some bugs were found when testing the admin system. Namely the way the admin system interacts with negative numbers. For instance, upon inputting a negative number for the price of a room, the system accept the number. This is obviously a problem since the hotel would need to pay the guest for staying with them. These bugs will be accounted for prior to future test runs.

# Conclusion

In conclusion, the hotel management system is a comprehensive web application that provides a wide range of functionalities for both guests and hotel staff. The system's critical components include room booking, reservation search, and the admin panel, which allows hotel staff to manage the system efficiently. The system's security features, such as password hashing and encryption, ensure that guest information is protected from unauthorized access. The system is user-friendly and intuitive, making it easy for guests to navigate and make bookings efficiently. The system’s testing phase of the user view was completely successful. However, there still is work to be done in the admin side to make it completely viable.

# References

<https://www.youtube.com/watch?v=BLqDewjag48&list=PLWxTHN2c_6cbh1C7yIskoXszoTl-okogt&index=3&ab_channel=TJWEBDEV>

# Video Link For Demo

<https://youtu.be/rNu2pauM7HY>