



Name : Kenan Alemam

Student no: 2221251385

Cp2 project

Abstract:

This web-based system tackles the challenge of time-pressed individuals who need to reserve hotel rooms. Designed with convenience in mind, it offers a one-stop shop for efficient hotel management. The system leverages a powerful MySQL database for reliable data storage. Here's what users can expect: Effortlessly search for available rooms, pick their ideal stay duration, and browse various room types to find the perfect fit – all with a user-friendly interface. The online booking system streamlines the reservation process, saving valuable time for busy professionals and anyone seeking a convenient way to book accommodations. IN essence, this user-friendly platform caters to all travelers seeking a smooth and efficient reservation experience, whether you're a business professional on the go or simply someone who values a hassle-free booking process. The robust MySQL database ensures the system runs on a secure and dependable foundation.

Purpose:

A hotel booking system the purpose of automating and streamlining the process of managing hotel reservations and bookings. The main objectives and purposes of a hotel booking system project in Java include:

1. Reservation Management:
 - Provide for customers to make hotel reservations online.
 - Manage and store reservation details, including guest information, check-in and check-out dates, room preferences, etc.
2. Inventory Management:
 - Keep track of available rooms and their types.
 - Update room availability in real-time as reservations are made or canceled.
3. User Authentication and Authorization:
 - Implement a secure login system for both customers and hotel staff.
4. Check-in and Check-out Processes:
 - Streamline the check-in and check-out procedures for both guests and hotel staff.
 - Update room status and generate invoices during check-out.
 - User-Friendly Interface.
 - Develop an intuitive and user-friendly interface for both customers and hotel staff.
 - Ensure responsiveness and accessibility across different devices.
 - Scalability and Robustness.
 - Design the system to be scalable, allowing it to handle a growing number of users and transactions.
 - Implement error handling and recovery mechanisms for system robustness.
5. User Authentication and Authorization:
 - Implement a secure login system for both customers and hotel staff.
6. Check-in and Check-out Processes:

- Streamline the check-in and check-out procedures for both guests and hotel staff



Technologies Used:

Java Swing for the GUI, MySQL for the database, YugSQL database connectivity.

Frame Names:

1. Signup Frame
2. Login Frame
3. Welcome Frame
4. Staff (Tabbed Pane)
5. Staff Log in
6. Servies Frame
7. Rezervatoin Frame



1-Signup frame

A screenshot of a web browser window displaying a 'Sign up' form. The form is set against a light gray background with a large teal vertical bar on the left. The title 'Sign up' is in a large, bold, black font. Below the title are six input fields: 'User name', 'name', 'Phone-number', 'Password', 'Confirm Password', and 'Email'. The 'Password' field has a 'Show' checkbox to its right. At the bottom of the form is a large teal 'Sign Up' button. Below the button, there is a link 'Already have an account ?' followed by two buttons: 'Log in' and 'Cancel'.

The Hotel Management System signup frame streamlines customer registration with essential fields:

Name: Input full name with format validation.

Email: Enter a valid email for unique account identification; email format and uniqueness are verified.

Password: Create a secure password with strength validation. Confirm

Password: Re-enter password for validation.

Phone number: for contact information.

Upon successful signup, access the Login Frame with a Login Button. Use Cancel Button to exit.



Methods():

✓ Go to Signup Method ():

This method is triggered when the user clicks the button to go to the signup page. Creates a new instance of the Signup class, sets its visibility to true, and disposes of the current login frame.

✓ Show/Hide Password Method ():

This method is triggered when the user checks or unchecks the "Show Password" checkbox. It changes the echo character of the password field to either display or hide the entered characters.

✓ DontHaveAccountMouseClicked(): Redirects to the registration frame.

2-The Login Frame

A screenshot of a web application's login frame. The frame has a light blue background and a dark blue sidebar on the left. The main content area contains the text 'Log in' in a large, dark blue font. Below this, there are two input fields: 'Username' and 'Password'. The 'Password' field is masked with asterisks. To the right of the 'Password' field is a checkbox labeled 'Show'. Below the input fields is a green button labeled 'Log in'. At the bottom, there is a link 'You do not have account ?' followed by a green button labeled 'Sign up'.

n the Hotel Management System enables registered users to access their accounts:

Username and Password Fields: Input registered Username and password with format validation.

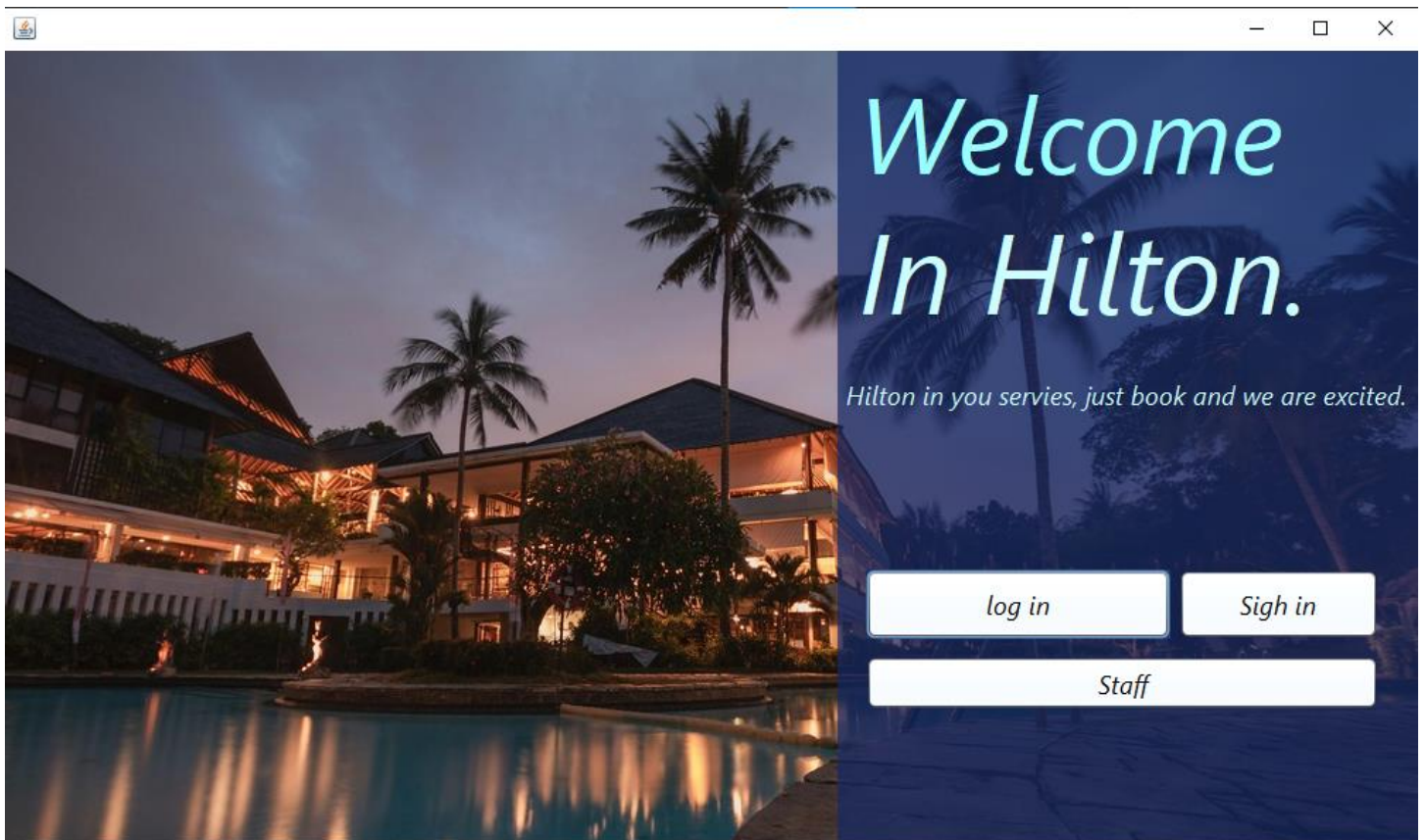
Login Button: Click to access the system; error handling informs users of any issues like incorrect credentials or locked accounts.

- **Method ():**

- ✓ **Go to Login Method ():**

This method is triggered when the user clicks the button to go back to the login page. Creates a new instance of the Login class, sets its visibility to true, and disposes of the current registration frame.

3- **Welcome frame:**



Log In : to go to lonin page

Sign up: to go to sgin up page

Staff : to go to staff page



4- Staff Log in frame

The image shows a staff login interface with a teal background. At the top, the text 'Log in for Staff' is displayed in a large, white, sans-serif font. Below this, there are two input fields. The first is labeled 'Employee Name' in a small, italicized, white font, followed by a dark gray rectangular input field. The second is labeled 'Password' in a small, italicized, white font, followed by a dark gray rectangular input field containing ten dots. To the right of the password field is a small dark gray square button labeled 'Show' in a small, italicized, white font. At the bottom, there are two dark gray rectangular buttons. The left button is labeled 'Enter' in a small, italicized, white font, and the right button is labeled 'Cancel' in a small, italicized, white font.

Here , this frame for staff to enter to the sistem and make bookin and edit information of customs.

Jtextfield

JPasswordField.

Jbutton.



5-Staff Manage Frame.

A-BookingTabbed:

A screenshot of a web application window titled 'Booking'. The window has a tabbed interface with 'Booking' and 'Customers' tabs. The 'Booking' tab is active, showing a form with fields for 'Name', 'ID', 'Room No', 'Check-in', and 'Check-out'. There is also a 'Room Type' dropdown menu set to 'Single => 100'. Two buttons, 'Add customer' and 'Back To Home', are at the bottom right of the form. The left side of the window is a dark blue sidebar.

Components:

Text Fields: Customer ID, Room Number.

JDate Choosers: Check-In Date, Check-Out Date. Buttons: Add, Edit,

Delete.

JTable: Displays Customer ID, Room Number, Check-In Date, and Check-Out Date.

Functionality:

Add Button: Adds a new customer with validated information. Edit Button:

Modifies existing customer details.

Delete Button: Removes selected customer with confirmation.

JTable Display: Real-time updates for viewing and managing customer.

B-Customers Tabbed:

The screenshot shows a web application window with a title bar and two tabs: 'Booking' and 'Customers'. The 'Customers' tab is active. The main content area has a light blue background with the word 'Customers' in a large, bold, dark blue font. On the left side, there is a form with the following fields and controls:

- Name**: A text input field.
- ID**: A text input field.
- Room No**: A text input field.
- Room Type**: A dropdown menu showing 'Single' and '=> 100'.
- Check-in**: A text input field with a calendar icon to its right.
- Check-out**: A text input field with a calendar icon to its right.
- Delete**: A green button.
- Edit**: A green button.
- Done**: A green button.

On the right side, there is a table with the following columns: Name, ID, Room no, Room Type, Check-in, Check-out, and Rezervation... The table is currently empty.

Components:

Text Fields: Customer ID, Room Number.

JDate Choosers: Check-In Date, Check-Out Date. Buttons: Add, Edit,

Delete.

JTable: Displays Customer ID, Room Number, Check-In Date, and Check-Out Date.

Functionality:

Add Button: Adds a new check-in record with validation. Edit Button:

Modifies existing check-in details.

Delete Button: Removes selected check-in record.

Check Room Availability: Validates room availability for specified dates. JTable Display: Real-time updates for viewing and managing check-in details.

Booking second tabbed pane offers a streamlined interface for administrators to manage check-in details. Its efficient layout and functionality make it easy to add, edit, and delete check-in records while providing crucial information about room availability during specific dates.

Methods():

Add Booking ():

- Adds a new booking to the database.
- Retrieves and validates input data.
- Updates the JTable to reflect the addition.

✓ Edit Booking ():

- Modifies details of an existing booking.
- Retrieves and displays selected details for editing.
- Updates the database and JTable.

✓ Delete Booking ():

- Removes a booking from the database.
- Retrieves selected details and requests confirmation.
- Updates the JTable after deletion.

✓ Booking Table():

✓ checkOverlappingReservations():

- This method assesses the reservation system to identify and report any instances of overlapping reservations, ensuring efficient management of booking schedules and preventing conflicts.



6- Customer Reservation Frame:

A screenshot of a web application window titled 'Rezervation' (sic). The window has a light gray background and a blue sidebar on the left. The main content area contains a form with the following fields: 'Name/Surname' (text input), 'ID' (text input), 'Check-in' (text input with a calendar icon), 'Check-out' (text input with a calendar icon), and 'Room type' (dropdown menu showing 'Single => 100'). Below the form are two buttons: 'Done' and 'Cancel/Back'. At the bottom, there is a 'Mode' section with two radio buttons: 'Dark' and 'Normal' (selected).

Rezervation

Name/Surname

ID

Check-in

Check-out

Room type Single => 100 ▼

Done **Cancel/Back**

Mode ☐ Dark ☒ Normal



In the Customer Reservation frame, customers are provided with a simplified interface to facilitate the reservation process. The frame incorporates essential components, including text fields and date choosers, empowering customers to specify their reservation details.

Reservation Process:

Customer ID and Room Number: Customers enter their unique identifier (Customer ID) and select their desired room number using dedicated text fields.

Check-In and Check-Out Dates: The reservation process involves choosing the intended check-in and check-out dates through intuitive date choosers.

Add Button: The "Add" button triggers the reservation submission, capturing the provided details for further processing.

✓ `checkOverlappingReservations()`:

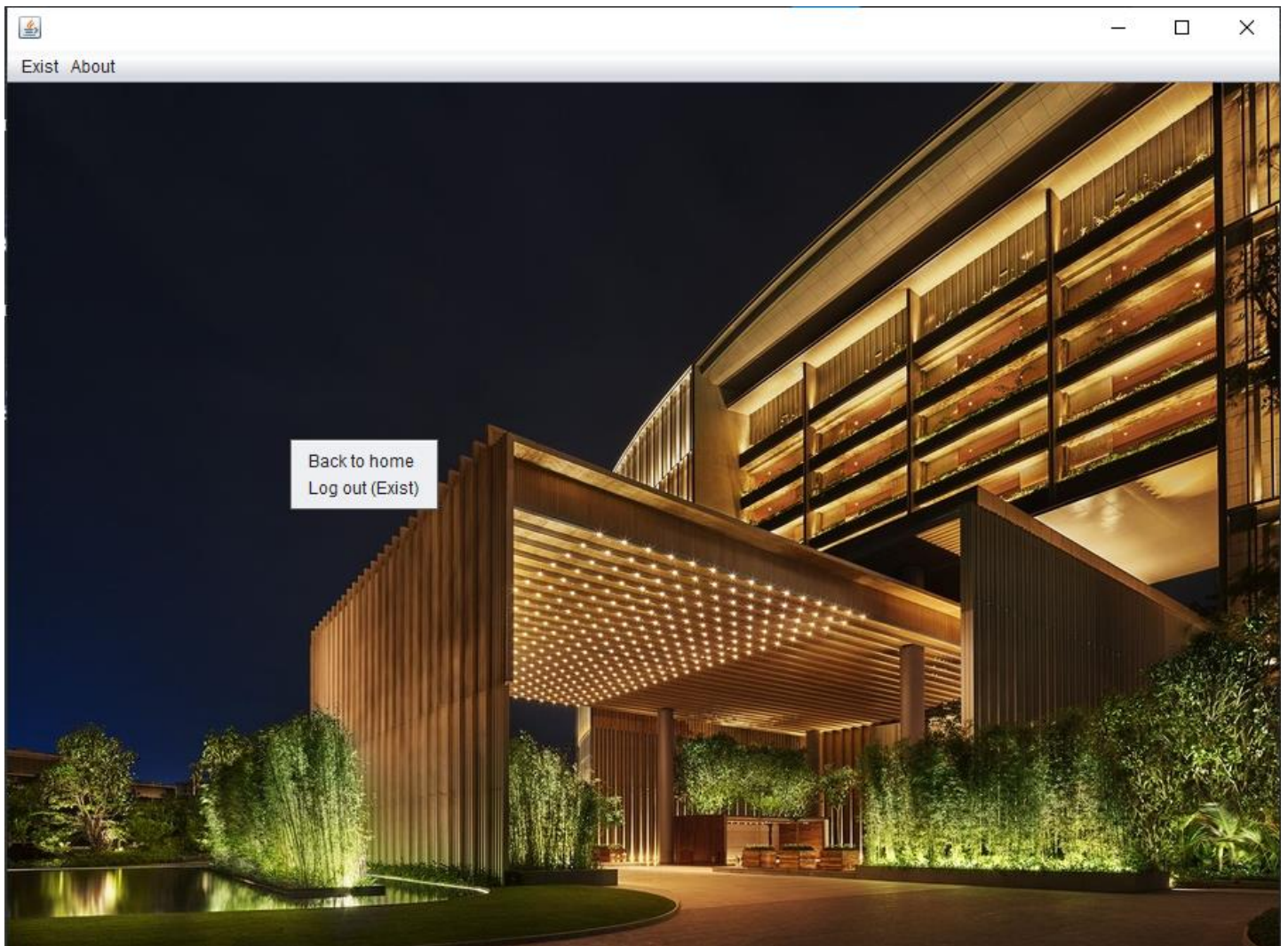
- This method assesses the reservation system to identify and report any instances of overlapping reservations, ensuring efficient management of booking schedules and preventing conflicts.

✓ `checkBookingReservationStatus()`:

The `checkBookingReservationStatus` method evaluates and reports the status of booking reservations, providing real-time insights into the progress and availability of reserved accommodations within the Hotel Management System



7- Servies staff frame



Menu Bar:

1. Exsit Menu:
 - Exit: Allows administrators to exit the application with a confirmation prompt.
2. About the founder Menu:
 - About Us: Provides information about the application and its development team.

Poupu Menu :

1. Services Button:
 - Provides quick access to service management.
2. Logout Button:
 - Allows administrators to securely log o