**COLLEGE OF BUSINESS EDUCATION**

****

**DODOMA CAMPUS**

*Student’s Name:* **ELISHA STEPHANO MGOVANO**

Reg no: **038332.01.02.2023**

***Course* : BIT**

***Lecturer*:madam ATUPELE CAIRO MWAITETE**

***Subject:* PROGRAMMING IN JAVA**

***Nature of Work:* INDIVIDUAL ASSIGNMENT**

**Question.**

You are required to create a small Java application that addresses an everyday challenge faced by individuals or communities in Tanzania with a theme of **"Digital Solutions for Everyday Challenges in Tanzania"**. Each student should select a specific challenge and provide a software-based solution.

**Guest House Management System**

**1. Introduction**

The Guest House Management System is a Java Swing-based desktop application designed to streamline the process of room bookings in a guest house. This system allows users to create accounts, log in, explore various room categories, and make bookings with a simple and user-friendly interface. It incorporates features such as account management, room selection, and payment processing to enhance the overall customer experience.

**2. Features Implemented**

**2.1 Account Management**

* **User Registration**: Users can create an account by entering their username and password.
* **Login System**: Users log in using their credentials. The system validates the entered details against stored user data.
* **Guest Access**: For users who do not wish to create an account, the system provides the option to proceed as a guest.

**2.2 Room Categories**

* The system offers the following room categories, each with a distinct set of rooms:
  + Single Rooms
  + Double Rooms
  + VIP Rooms
  + VVIP Rooms
  + Family Rooms

**2.3 Room Selection**

* Each room category is displayed as a button with an image and label.
* Clicking a category button opens a panel displaying available rooms with:
  + Room images.
  + Room names.
  + Selection functionality.

**2.4 Payment Processing**

* **Cost Calculation**: Displays the total cost based on selected rooms.
* **Payment Interface**: Users input the amount they wish to pay, and the system calculates the balance or prompts for insufficient funds.
* **Receipt Generation**: After successful payment, a detailed receipt of the transaction is generated and displayed.

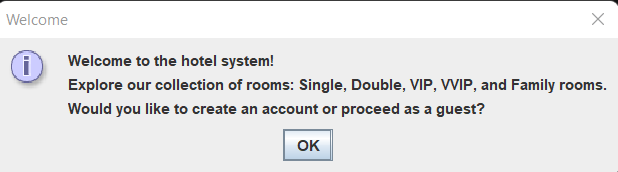
**2.5 User Interaction**

* The system employs **JOptionPane** for user prompts and notifications.
* Custom dialogs provide an intuitive flow for booking and payment processes.

**3. Screenshots of the Interface**

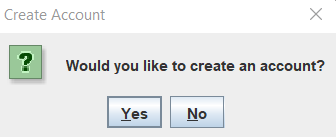
**3.1 Welcome Screen**

* The initial welcome dialog introduces the system to the user.
  + **Screenshot**: Displays a dialog box titled "Welcome," prompting users to create an account or proceed as a guest.

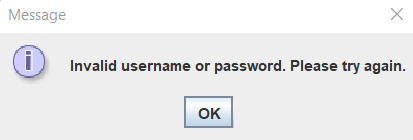


**3.2 Account Creation and Login**

* **Registration Panel**: A simple input dialog allows users to enter their username and password.
  + **Screenshot**: Registration dialog.

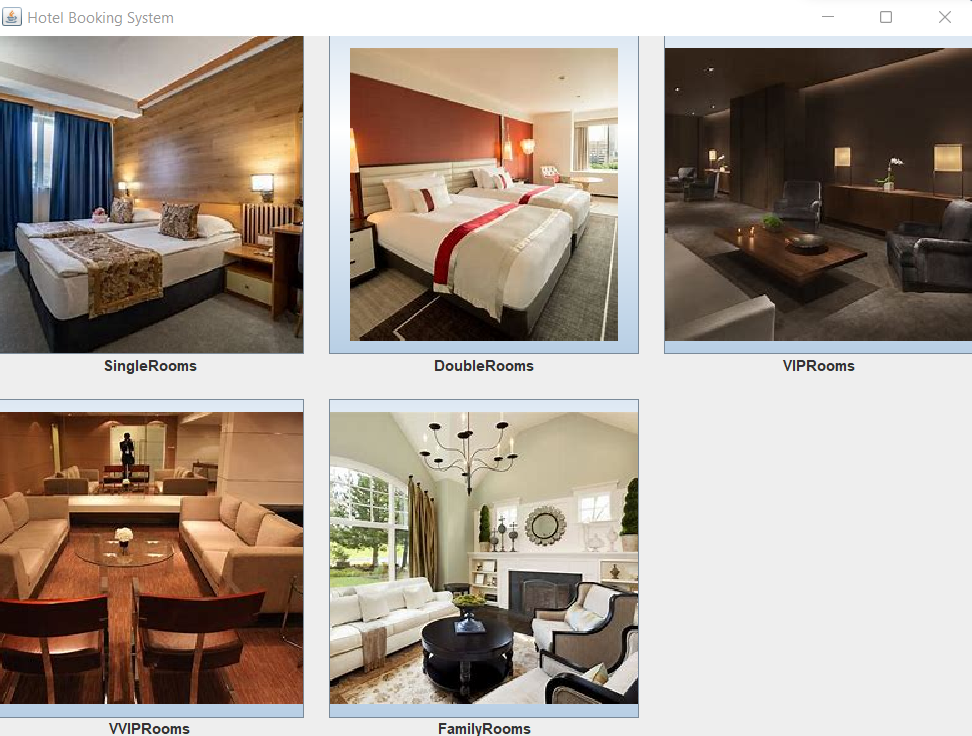


* **Login Panel**: Users log in using their registered credentials.
  + **Screenshot**: Login dialog with error handling for incorrect credentials.

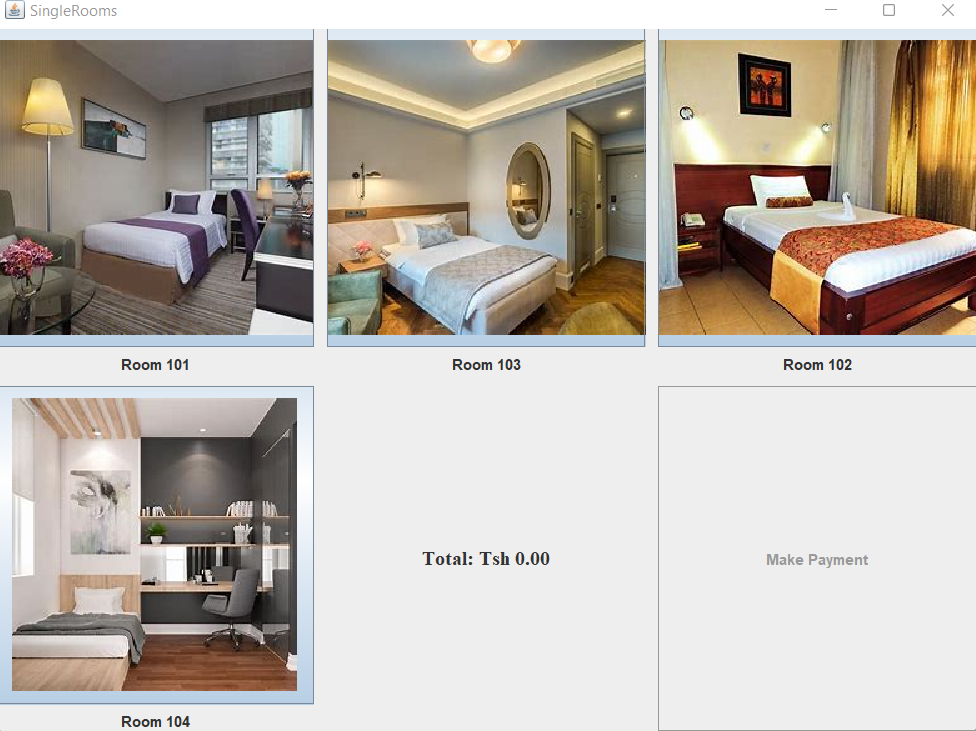


**3.3 Room Selection**

* **Category Panel**: Displays room categories in a grid layout with category images and labels.
  + **Screenshot**: Main menu with buttons for "Single Rooms," "Double Rooms," etc.

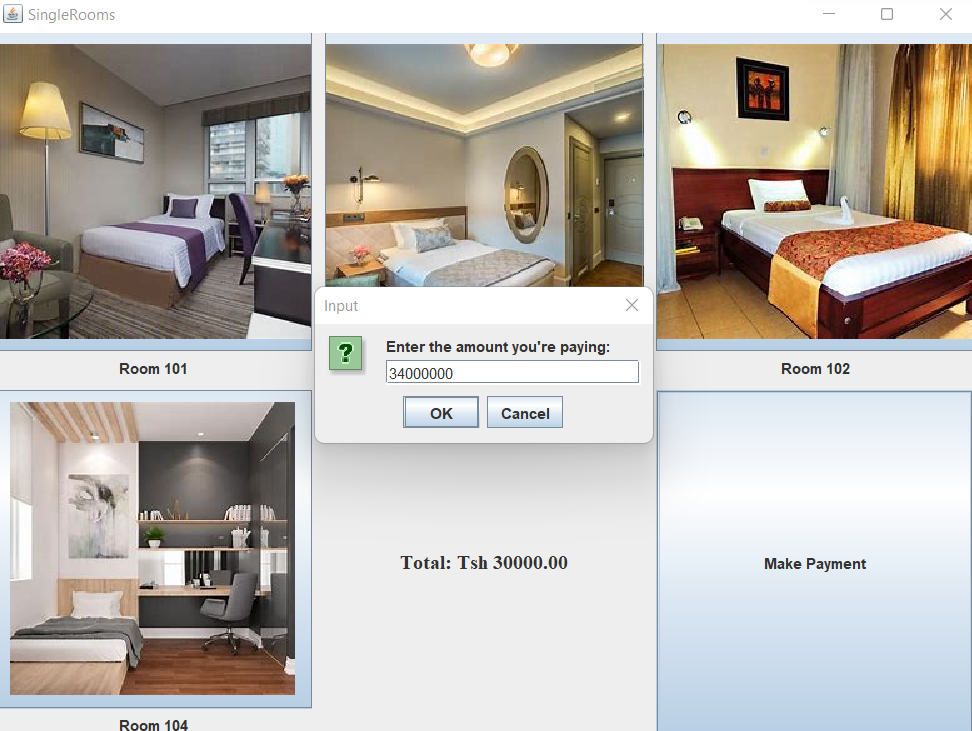


* **Room Display Panel**: A grid of room images, names, and selection buttons for a specific category.
  + **Screenshot**: Grid layout showing available rooms in the "Single Rooms" category.



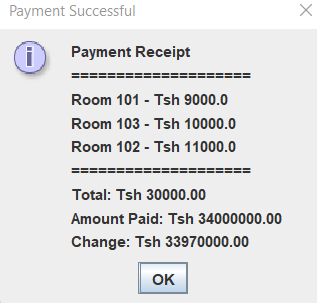
**3.4 Payment Panel**

* Displays the total cost, payment input field, and "Make Payment" button.
  + **Screenshot**: Payment dialog with "Total Cost" label and payment options.



**3.5 Receipt Generation**

* After successful payment, a detailed receipt is shown in a dialog box.
  + **Screenshot**: Receipt dialog showing room names, total cost, amount paid, and change.



**4. Challenges Faced During Development**

**4.1 User Interface Design**

* Designing an intuitive and aesthetically pleasing user interface using **Java Swing** was a challenge, as Swing has limited built-in styling options.
* Ensuring proper alignment and spacing in panels, especially for different screen resolutions, required extensive testing and adjustments.

**4.2 Dynamic Room Loading**

* Dynamically loading room details and images for each category required careful management of **HashMap** data structures and file paths.
* Handling missing or incorrect image paths during testing highlighted the importance of proper file management.

**4.3 Payment Calculation**

* Implementing real-time cost calculation based on selected rooms involved writing efficient and accurate code to handle room selection and deselection.
* Managing edge cases, such as insufficient payment or invalid input, required robust error handling.

**4.4 Data Persistence**

* Storing user accounts and session data only in memory limited the system's scalability.
* Implementing basic user authentication without encryption was identified as a security risk.

**4.5 Testing and Debugging**

* Ensuring the application behaved as expected across different scenarios, such as multiple room selections or partial payments, required extensive manual testing.
* Debugging logical errors in receipt generation and total cost calculation was time-consuming but crucial for the system’s reliability.

**5. Future Enhancements**

**5.1 Database Integration**

* Replace in-memory data structures with a relational database (e.g., MySQL) for storing user accounts, room details, and transaction records.

**5.2 Improved UI/UX**

* Utilize modern UI libraries or frameworks (e.g., JavaFX) for a more polished user interface.

**5.3 Advanced Authentication**

* Implement secure user authentication with encrypted password storage.

**5.4 Reporting and Analytics**

* Add features for generating and exporting detailed booking reports and transaction summaries.

**5.5 Mobile Compatibility**

* Develop a mobile application version of the system for greater accessibility.

**6. Conclusion**

The Guest House Management System serves as a robust foundation for managing room bookings efficiently. Despite certain limitations, it offers a functional and user-friendly experience. Future developments will focus on enhancing security, scalability, and usability to meet the evolving needs of guest house operations.