

DSA Final Report

December 22, 2023



Session: 2022 – 2026

Submitted by:

Muskan Awais (2022-CS-38)
Mahrukh Kashan (2022-CS-45)
Bisma Shakeel (2022-CS-50)
Eeman Waheed (2022-CS-52)

Submitted to :
Sir Nazeef Ul Haq

Department of Computer Science

University of Engineering and Technology Lahore, Pakistan

Contents

1 Project Details:	4
1.1 Project Name:	4
1.2 Gitlab Repository Name:	4
1.3 Gitlab Repository Link:	4
2 Introduction:	5
3 Project Overview:	5
4 System Architecture:	5
5 Data Structure and Algorithm:	5
5.1 User Authentication Module using Hashing:	5
5.1.1 Hashing Algorithm:	5
5.1.2 Sign Up and Sign In:	6
5.2 Booking Management Module using Trees:	6
5.2.1 Tree Structure:	6
5.2.2 Booking Algorithm:	6
5.3 Check-In Module:	6
5.3.1 Guest Information Tree:	6
5.3.2 Check-In Algorithm:	6
5.4 Room Service Module:	7
5.5 Place Order Module:	7
5.6 Check Out Module:	7
5.7 Admin Module:	7
6 Interactions Between Modules:	7
6.1 User Registration and Authentication:	7
6.2 Booking and Check-In:	8
7 Scalability and Performance Optimization:	8
8 Security Measures:	8
9 System Limitations:	8
10 Implementation Details:	9
10.1 Programming Language:	9
10.2 Libraries:	9
10.3 Framework:	9

11 Future Enhancements:	9
12 Conclusion:	10
13 User Interface:	10

MMBE Hotel Management System

1 Project Details:

1.1 Project Name:

MMBE Hotel Management System

1.2 Gitlab Repository Name:

CS261F23PID05

1.3 Gitlab Repository Link:

<https://gitlab.com/mahrukhkashan/cs261f23pid05>

2 Introduction:

The documentation of our Hotel Management System (HMS) project serves as a guide to the design and development of Hotel Management System using Python with the help of data structures and algorithms. The user interface will be designed using the Qt framework.

3 Project Overview:

Our Hotel Management System aims to streamline hotel operations, offering features such as guest management, staff management, room booking, and billing. Using Python and Qt for the user interface, we aim to create an efficient, user-friendly, and visually appealing application.

4 System Architecture:

The system architecture will follow a modular structure:

- Presentation Layer: UI designed using Qt Designer
- Application Layer: Business logic implemented in Python

5 Data Structure and Algorithm:

5.1 User Authentication Module using Hashing:

5.1.1 Hashing Algorithm:

- Utilizes Hashing Algorithms for secure storage and retrieval of user credentials.
- Hashed data is stored in a data structure that allows quick access during the authentication process.
- Hashing ensures the confidentiality and integrity of sensitive information.

5.1.2 Sign Up and Sign In:

- During user registration, user details are hashed and stored in the file.
- During sign-in, the entered credentials are hashed and compared with the stored hash for authentication.

5.2 Booking Management Module using Trees:

5.2.1 Tree Structure:

- Implements a Tree Data Structure to organize and manage room booking information.
- Each node in the tree represents a room, and the structure enables quick retrieval and insertion of booking details.
- A booking algorithm optimizes room allocation based on user preferences, availability, and historical data.

5.2.2 Booking Algorithm:

- A booking algorithm optimizes room allocation based on user preferences, availability, and historical data.
- The algorithm ensures optimal room allocation based on user preferences and availability.

5.3 Check-In Module:

5.3.1 Guest Information Tree:

- A separate tree structure is employed to manage guest information during their stay.
- This tree facilitates easy retrieval and updates of guest details during the check-in process.

5.3.2 Check-In Algorithm:

- The check-in algorithm utilizes the guest information tree to streamline the check-in process.

- The system efficiently retrieves guest details upon check-in.

5.4 Room Service Module:

- Implements dynamic lists to manage and prioritize room service requests.
- Algorithms optimize the assignment and fulfillment of room service requests.

5.5 Place Order Module:

- The place order module is using a queue to take orders and then confirming the order
- Algorithm checks the queue and give orders according to order taken.

5.6 Check Out Module:

- This module is used for checking out. In this module the guest check outs and make payment and add rating.
- The information of payment and rating is later saved in csv file.

5.7 Admin Module:

- Admin functions include view, modify, remove, search and sort guest information.
- Modification is done based on any attribute of guest personal information. Removal, Sorting and Searching is based on username.

6 Interactions Between Modules:

6.1 User Registration and Authentication:

- Users register through the authentication module, and their credentials are securely hashed and stored.
- During login, the hashing algorithm verifies the entered credentials against the stored hashed values.

6.2 Booking and Check-In:

- The booking module interacts with the check-in module to update room availability and guest information.
- Algorithms ensure the seamless transition of a booked room to an occupied status during check-in.

7 Scalability and Performance Optimization:

- DSA is employed to optimize algorithms for quick data retrieval, insertion, and updates.
- Tree structures and dynamic lists are designed to efficiently handle a growing volume of data, ensuring scalability.

8 Security Measures:

- Hashing algorithms are implemented for secure storage of sensitive user information.
- Access controls and encryption mechanisms are incorporated to protect data integrity and confidentiality.

9 System Limitations:

- Real-Time Inventory Updates: The system may not provide real-time updates on room availability, and there could be a slight delay in reflecting changes.
- Internet Dependency: The system might be dependent on internet connectivity for certain functionalities
- Customization Limitations: Extensive customization of the system by individual hotels may be limited, and certain features may not be adjustable based on specific hotel requirements.
- Scalability: The system may face limitations in handling a large volume of simultaneous users or extensive data, affecting its scalability.

- Support and Maintenances: Continuous support and maintenance may be necessary to address bugs, implement updates, and provide assistance to users.

10 Implementation Details:

10.1 Programming Language:

- Python

10.2 Libraries:

- from logging import root
- from tkinter import messagebox, Label, Spinbox, Button, Entry, Toplevel, Tk
- from tkinter.ttk import Treeview
- from PyQt5.QtWidgets import QComboBox, QApplication, QMessageBox, QMainWindow, QInputDialog, QLineEdit, QDialog, QVBoxLayout, QPushButton, QTableWidgetItem, QTableWidget, QSizePolicy, QLabel, QWidget, QSpinBox
- from PyQt5 import uic
- import csv
- import hashlib
- from collections import OrderedDict

10.3 Framework:

- QT designer for user interface

11 Future Enhancements:

Future iterations may explore the integration of advanced algorithms, machine learning models, and real-time analytics to further enhance the

efficiency and personalized experience provided by the Hotel Management System. Additionally, ongoing optimizations based on user feedback and technological advancements should be considered for continuous improvement.

12 Conclusion:

In conclusion, our Hotel Management System emphasizes simplicity and efficiency by incorporating basic DSA concepts. By using fundamental data structures and algorithms, we aim to provide a reliable and user-friendly solution for meeting essential operational needs and ensuring a positive experience for both guests and admin.

13 User Interface:

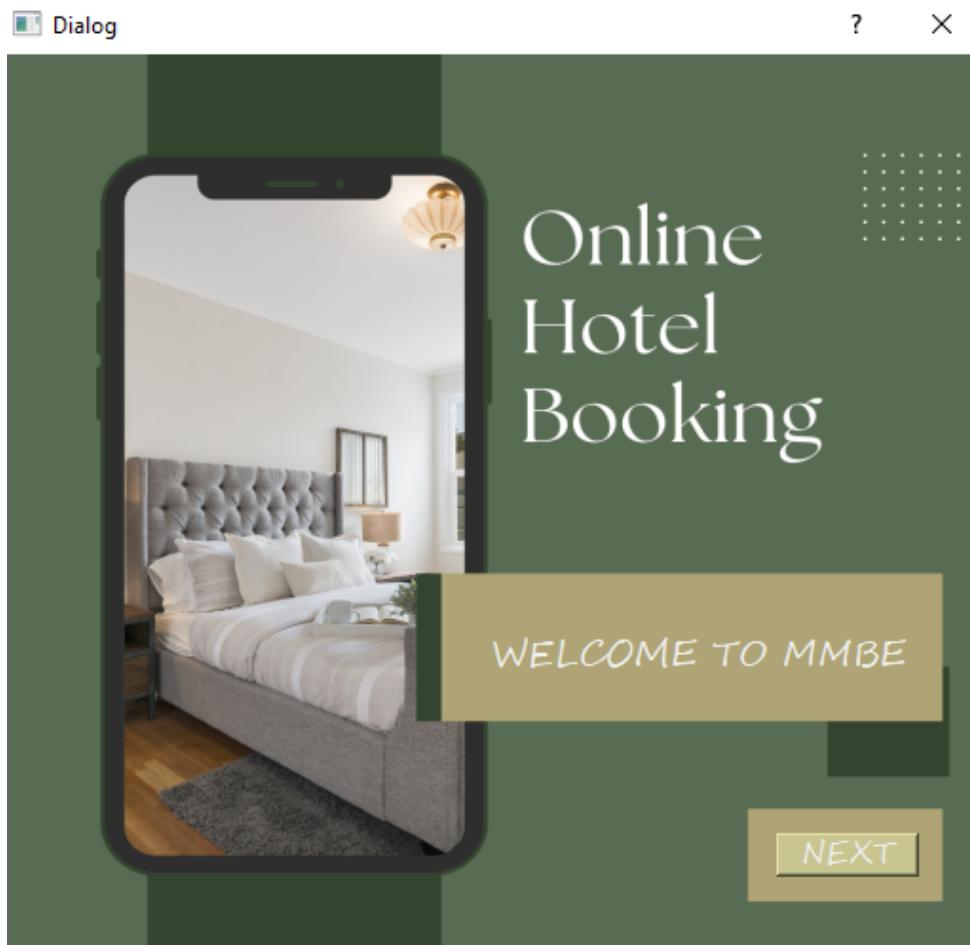


Figure 1: Welcome UI

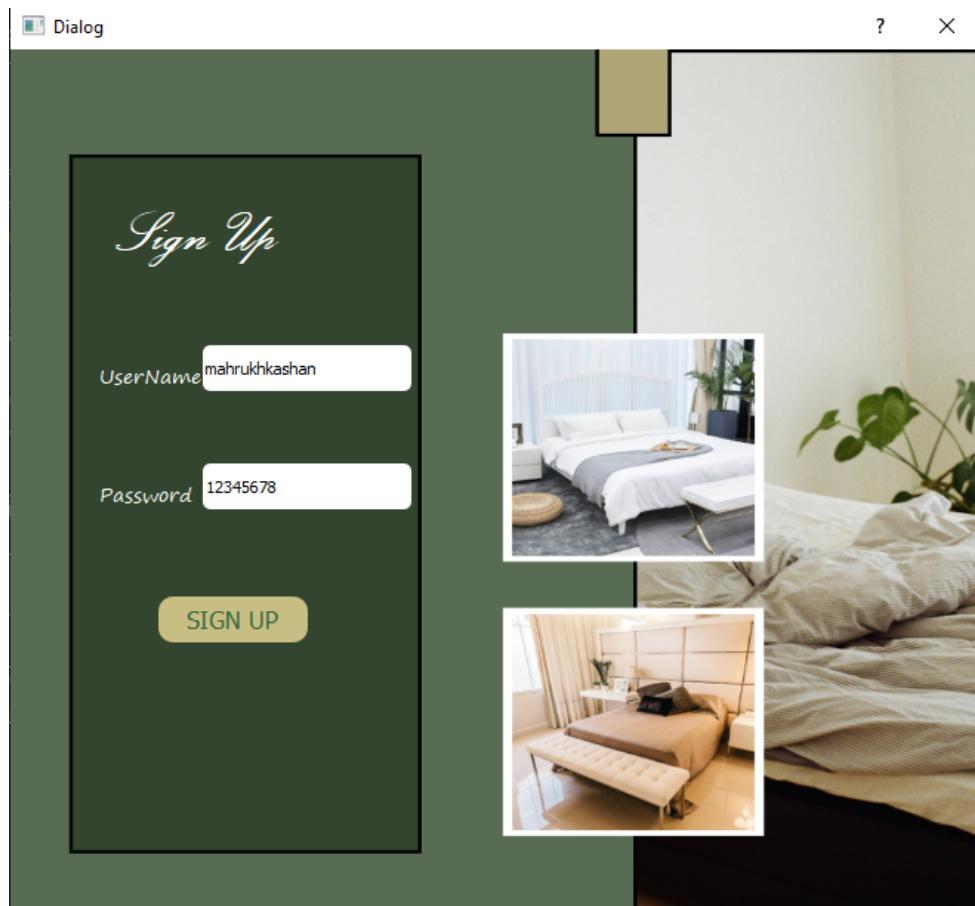


Figure 2: Sign Up UI

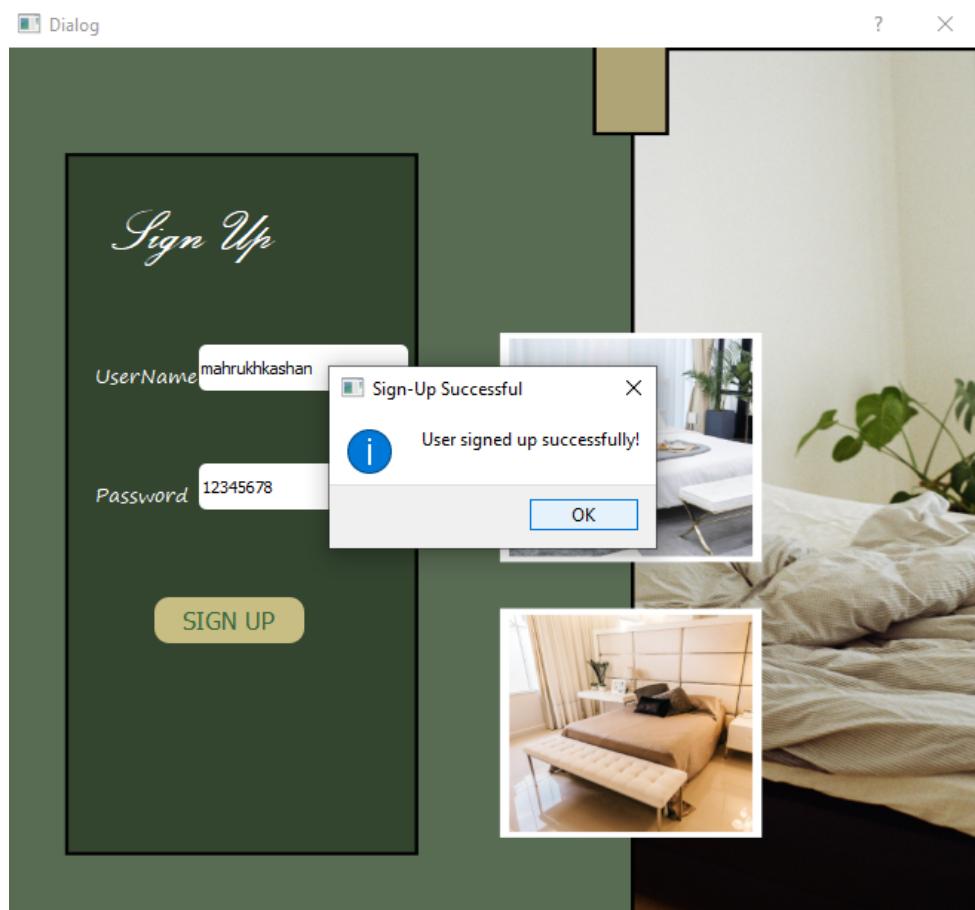


Figure 3: Successfully Sign Up Message Box

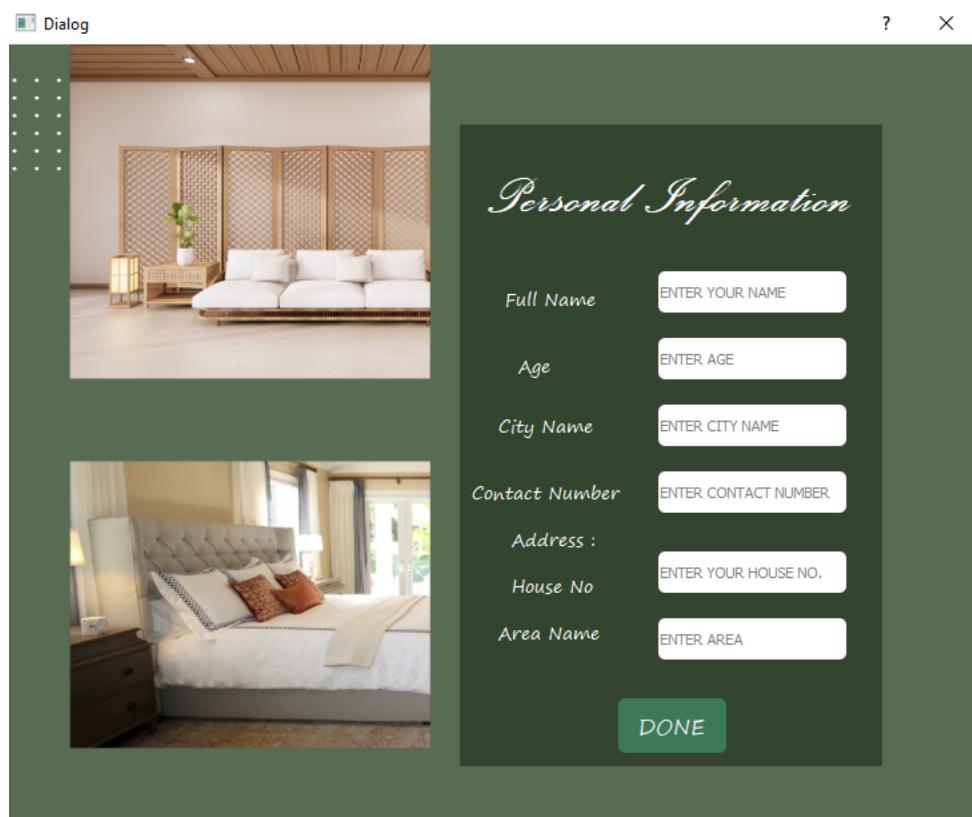


Figure 4: Personal Information UI

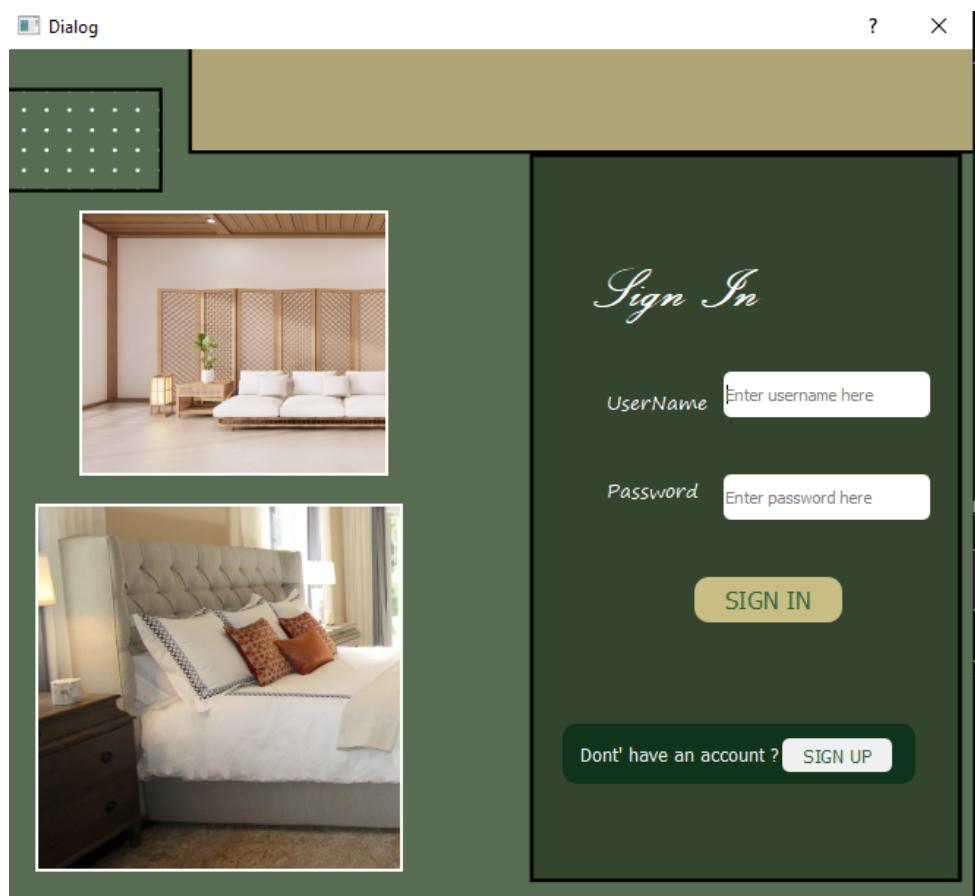


Figure 5: Sign In UI

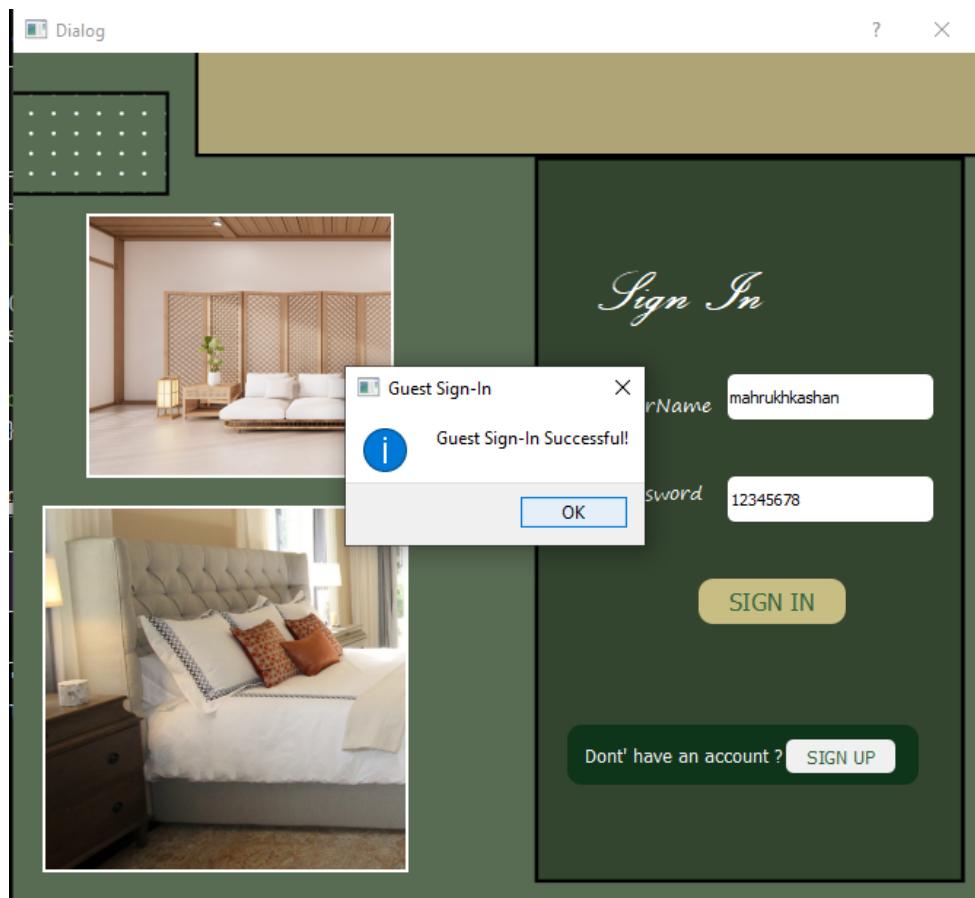


Figure 6: Successfully Sign In Message Box

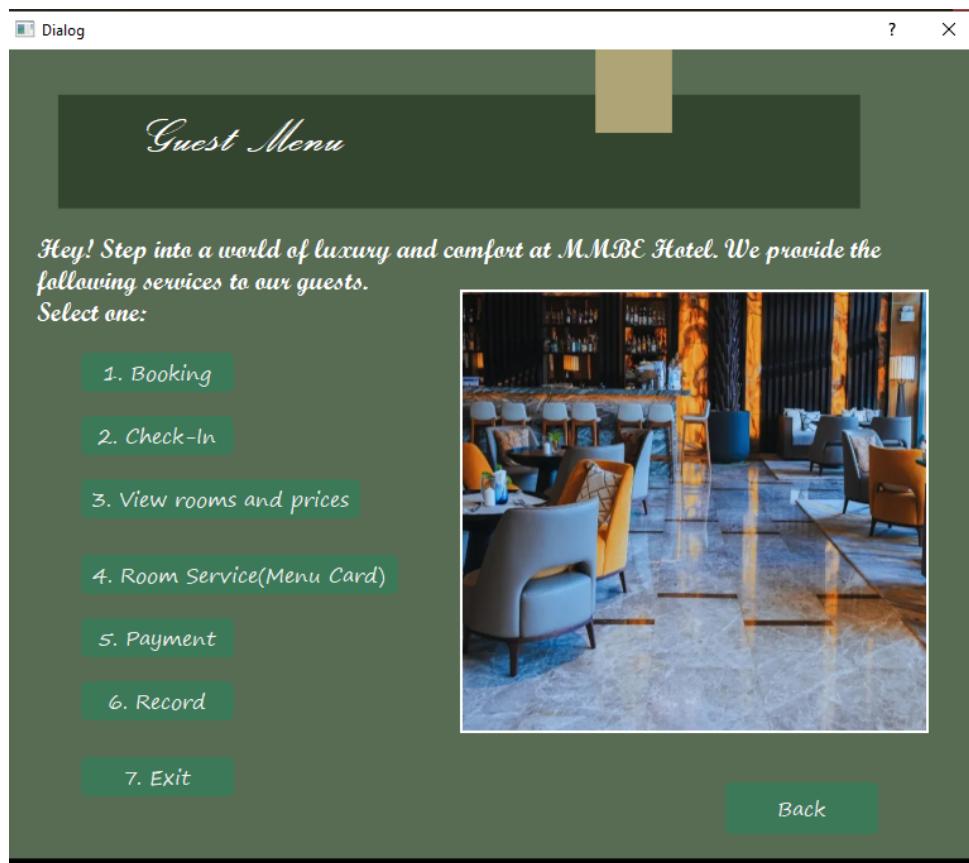


Figure 7: Guest Menu UI

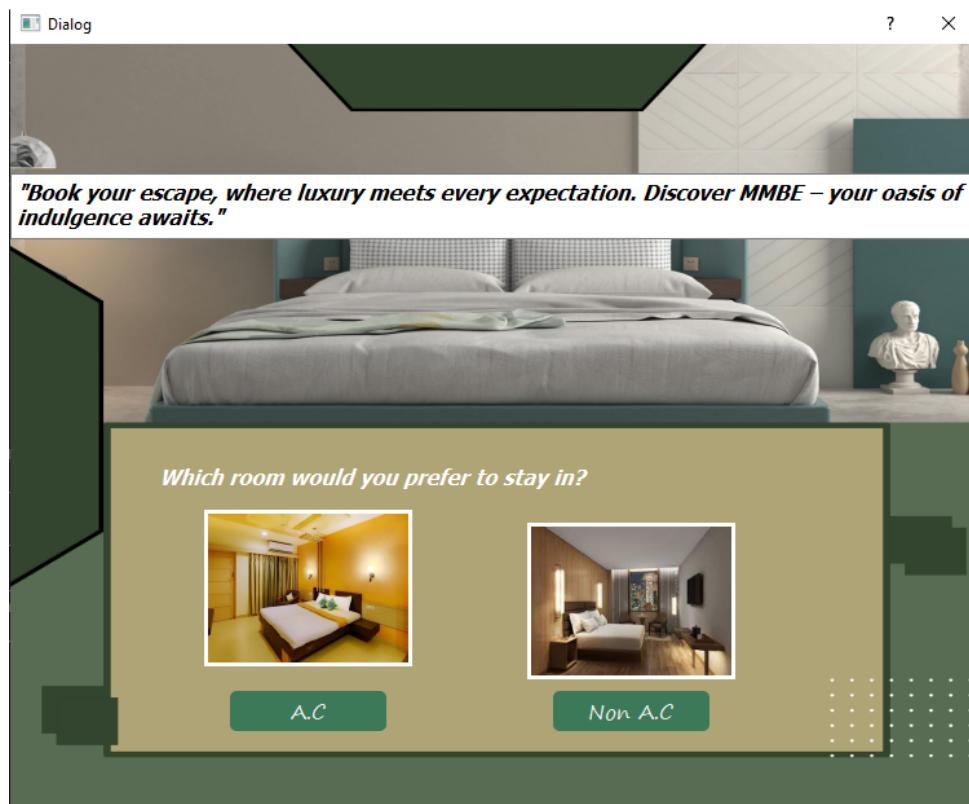


Figure 8: Ac Selection UI

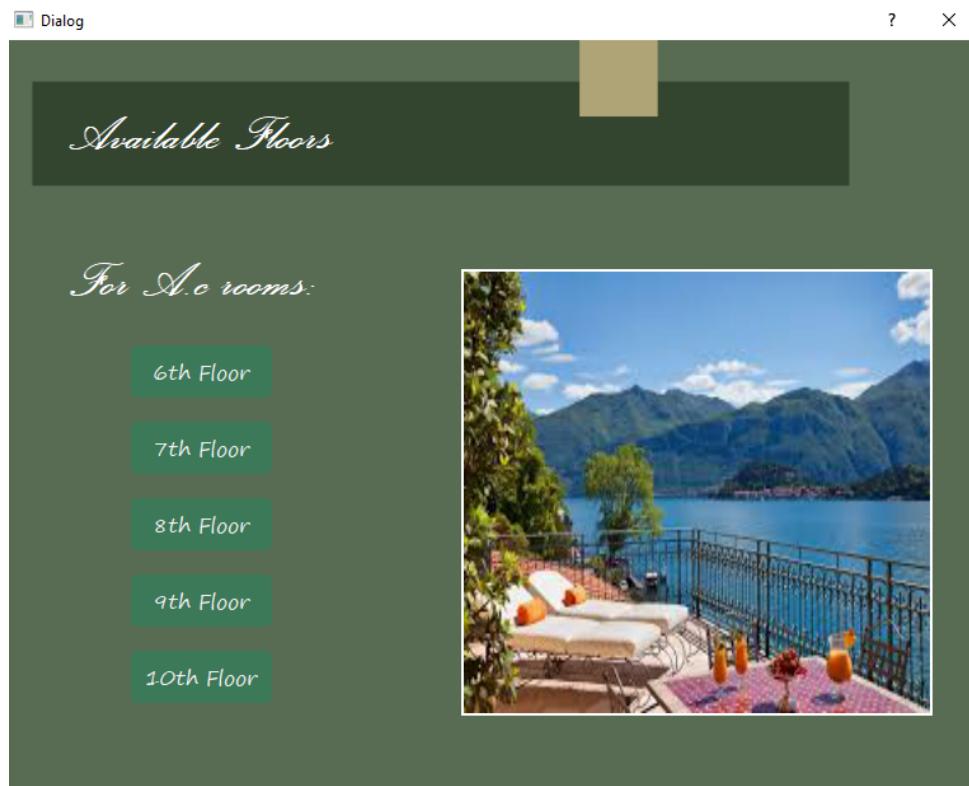


Figure 9: Floors Selection UI

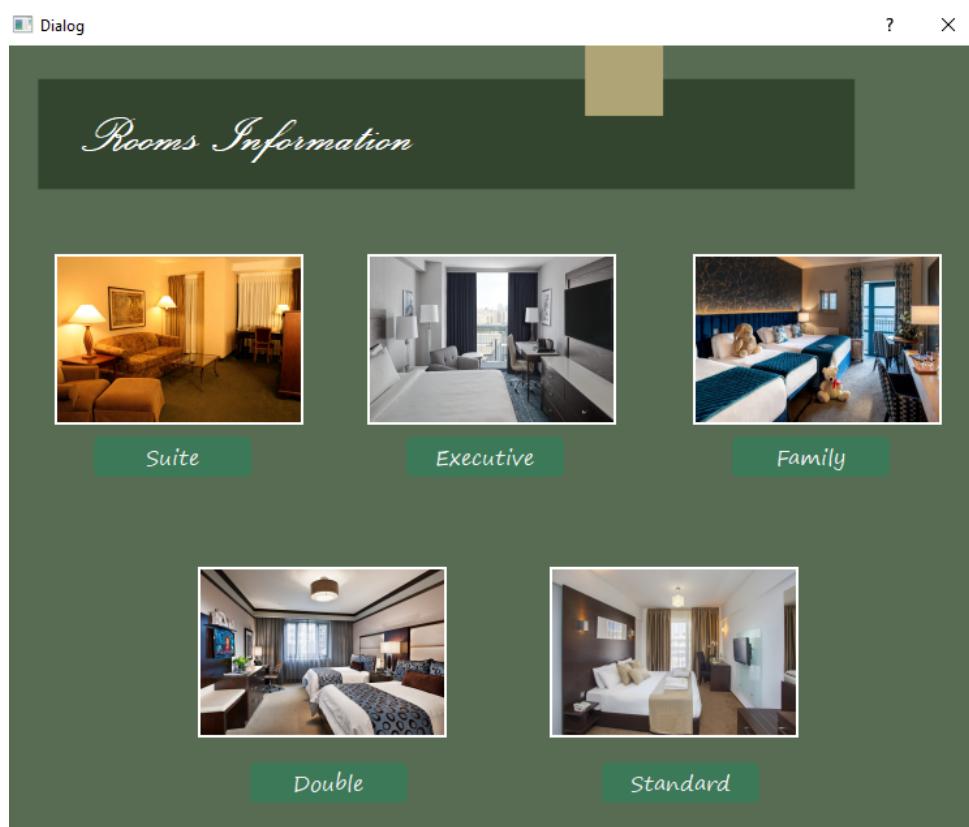


Figure 10: Room Type Selection UI

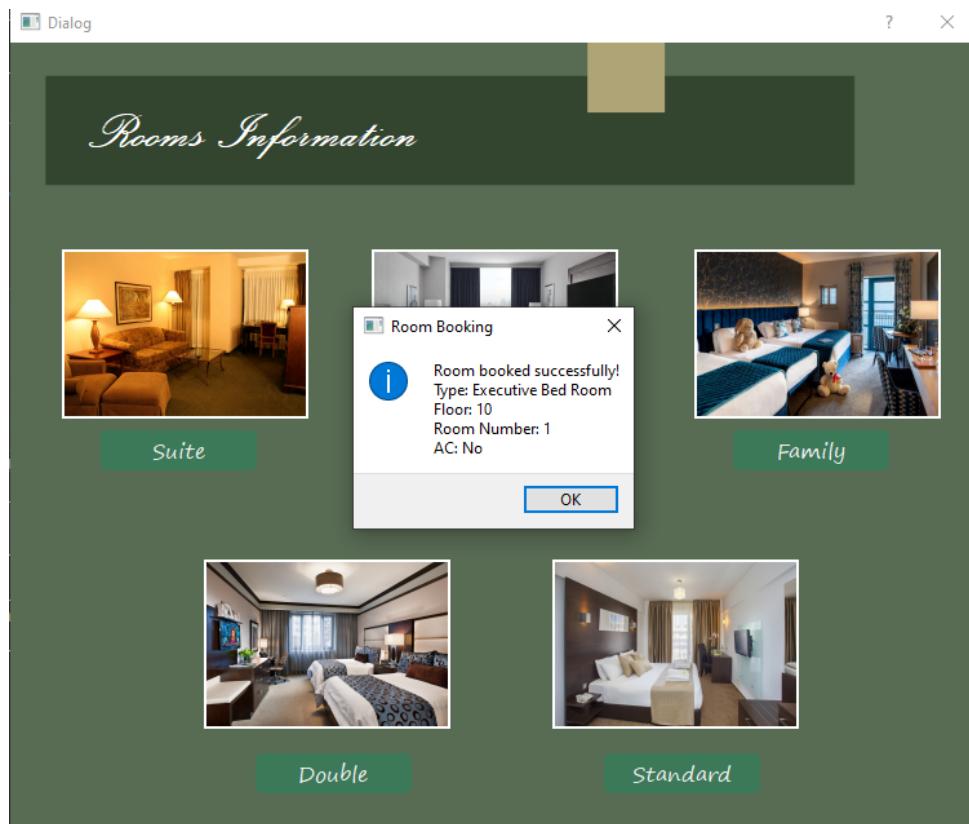


Figure 11: Booking Confirmation Message Box

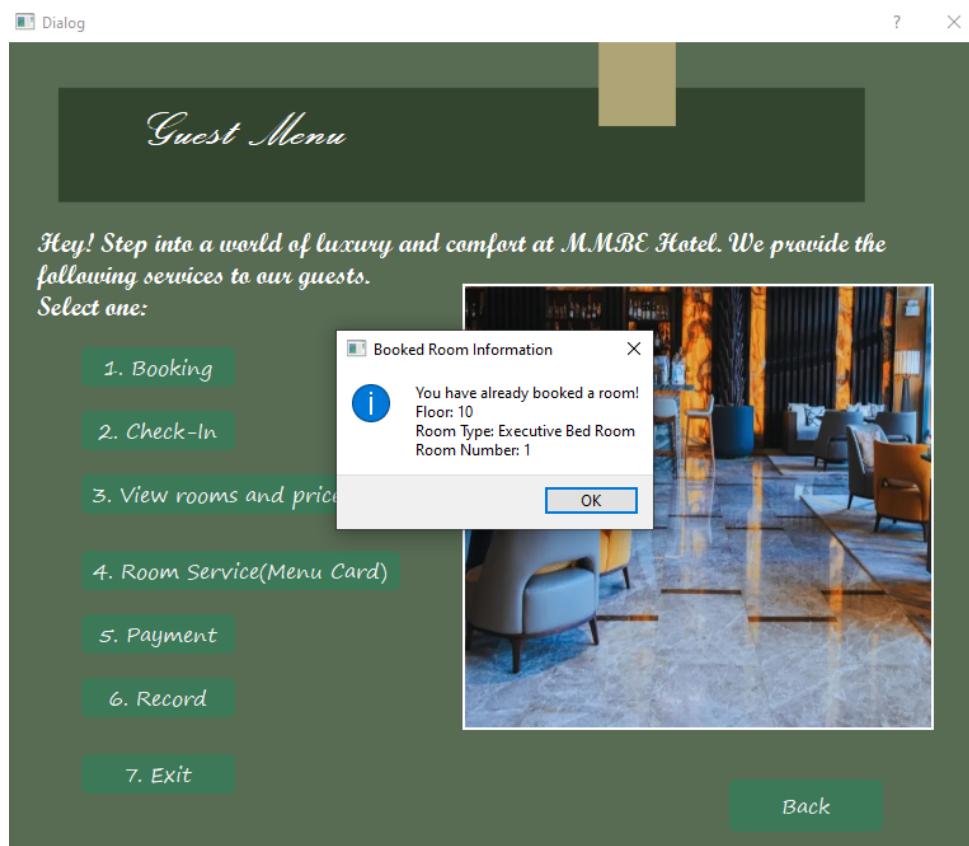


Figure 12: Already Booked Message Box

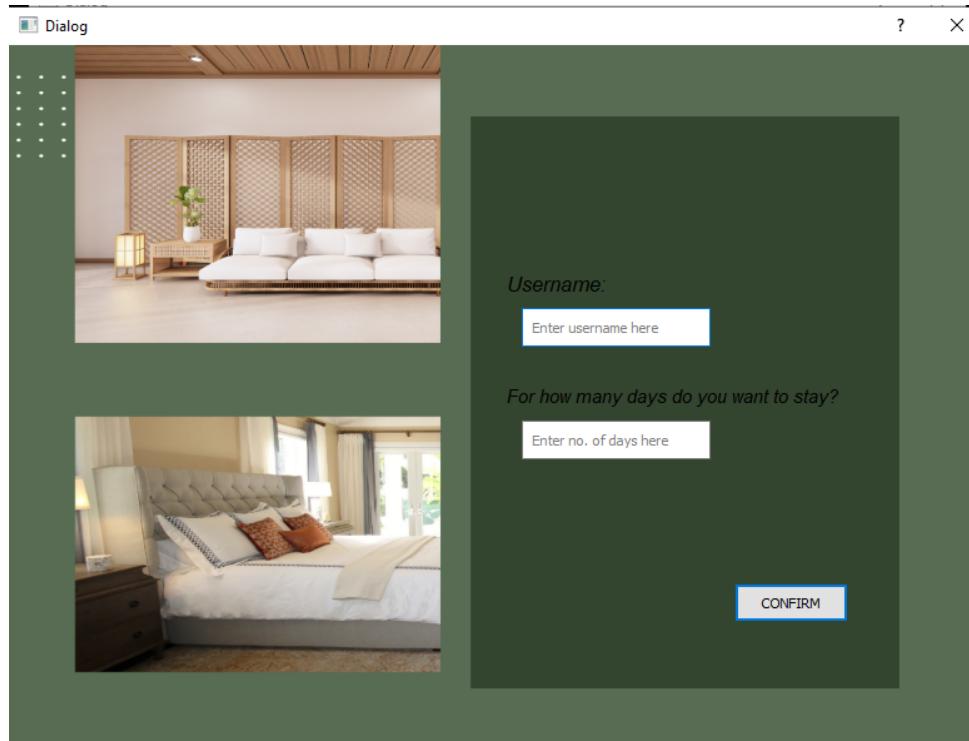


Figure 13: Check In UI

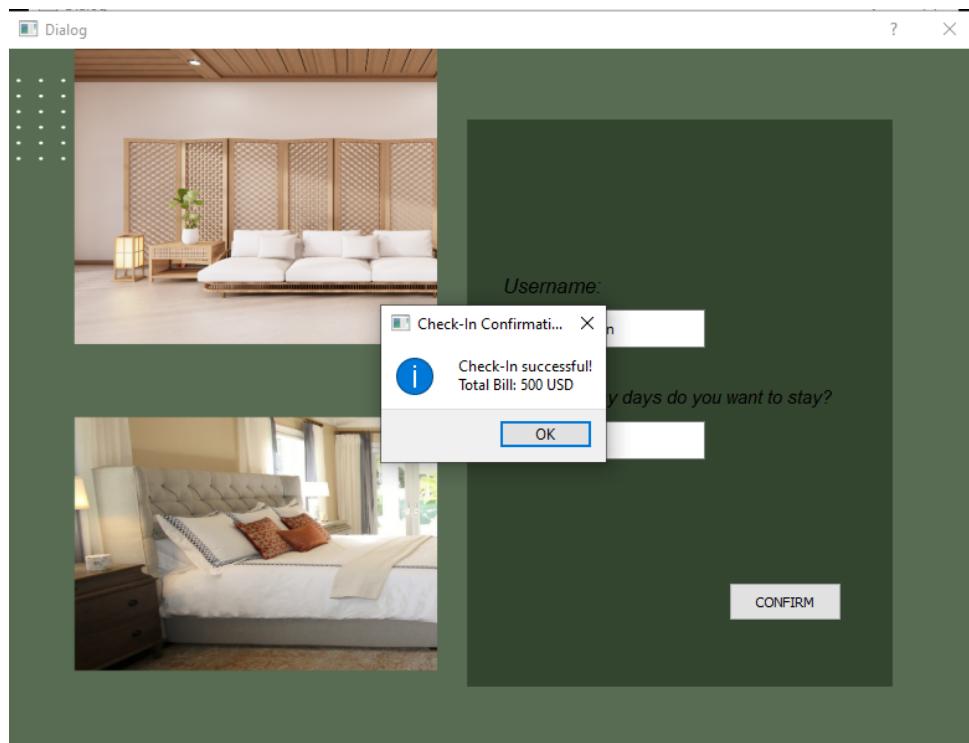


Figure 14: Staying Bill Message Box

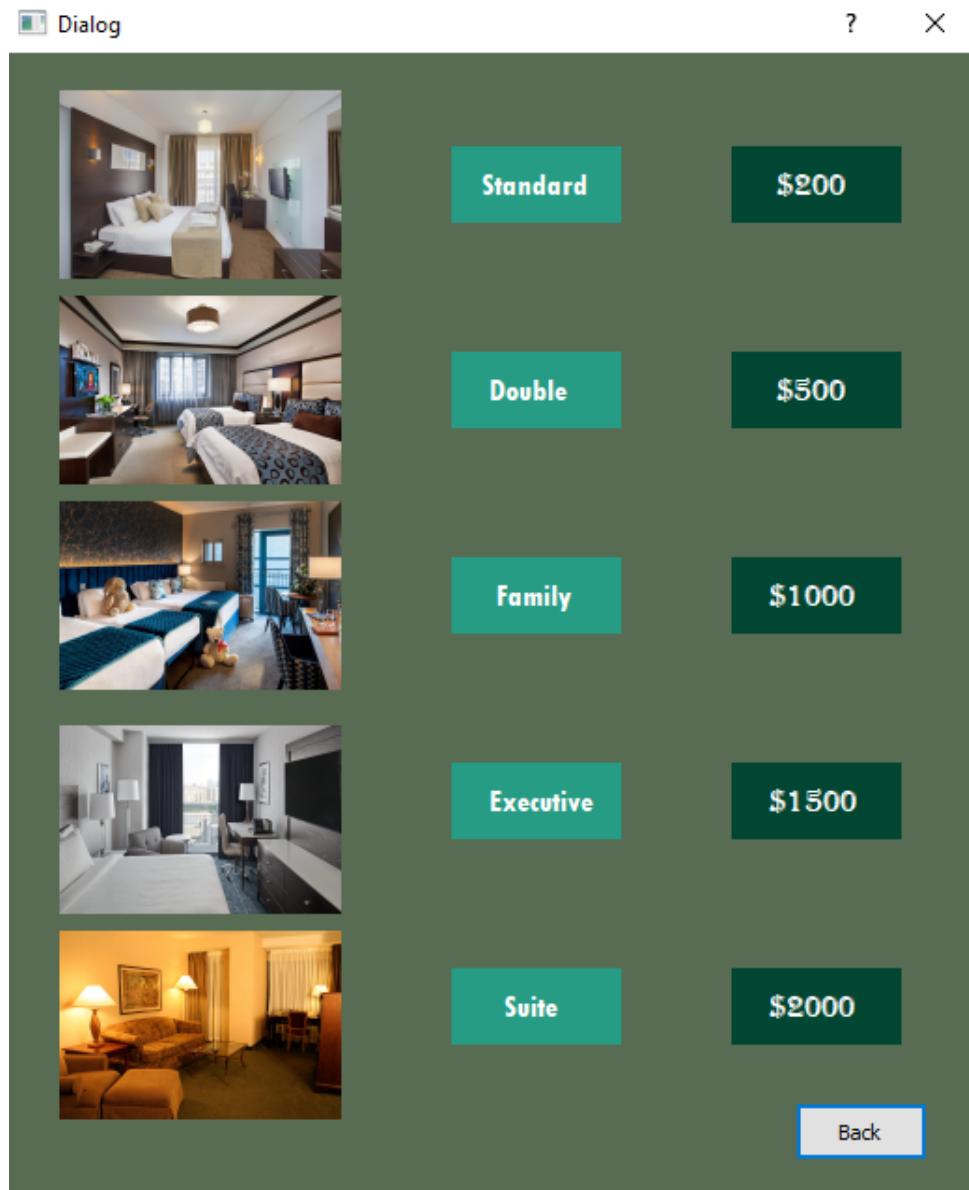


Figure 15: View Rooms and their prices UI

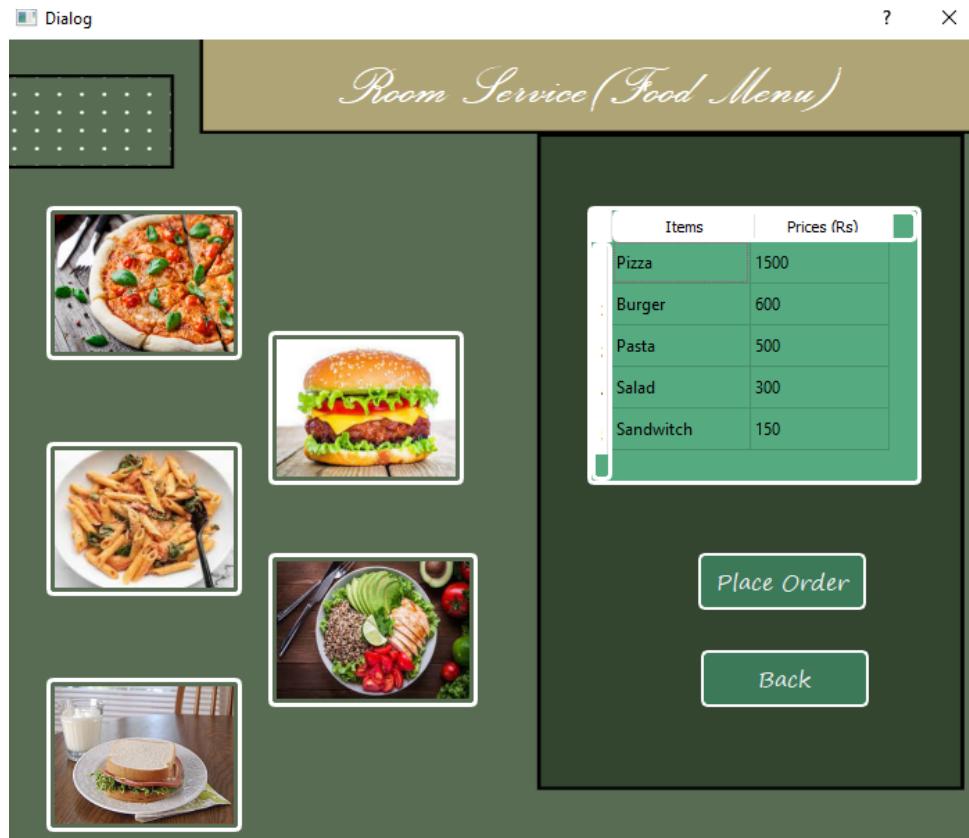


Figure 16: Food Menu prices UI

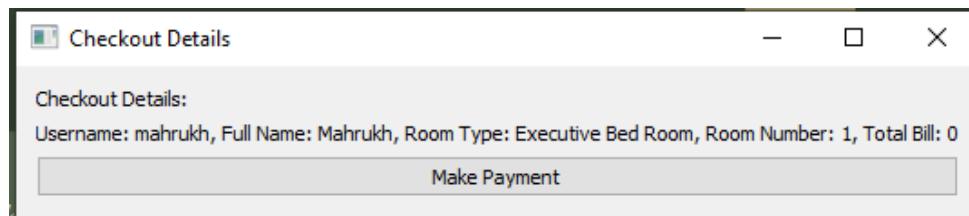


Figure 17: CheckOut UI

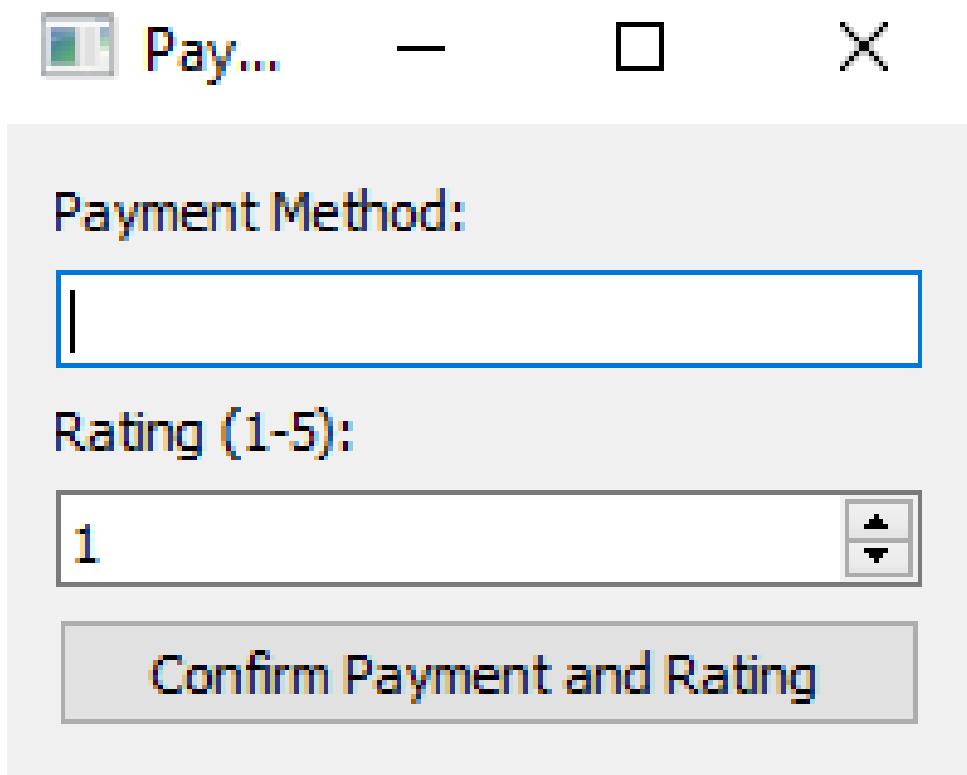


Figure 18: Payment and rating UI

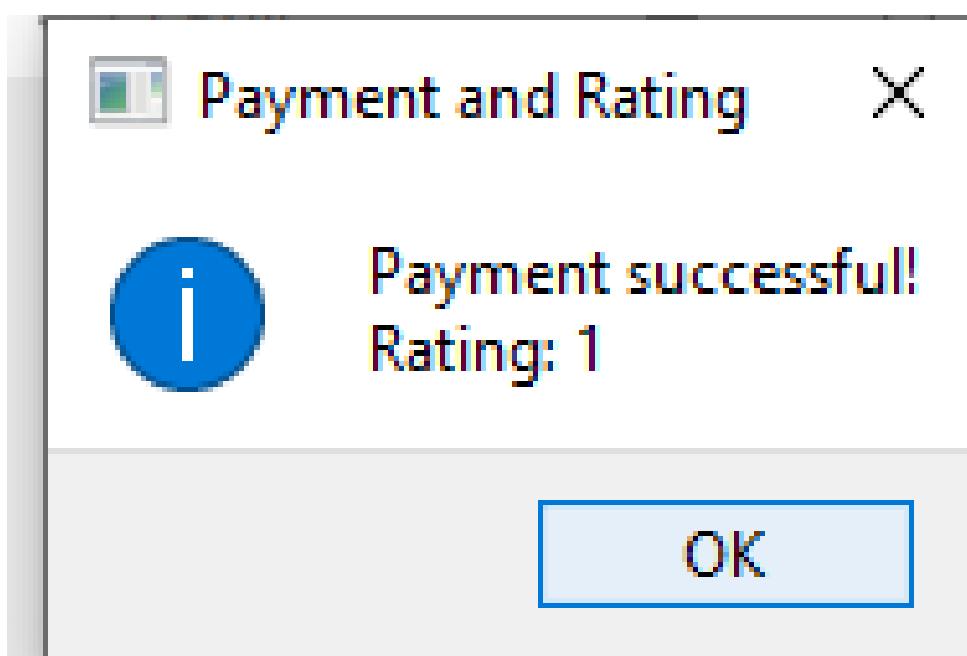


Figure 19: Successful payment and rating UI

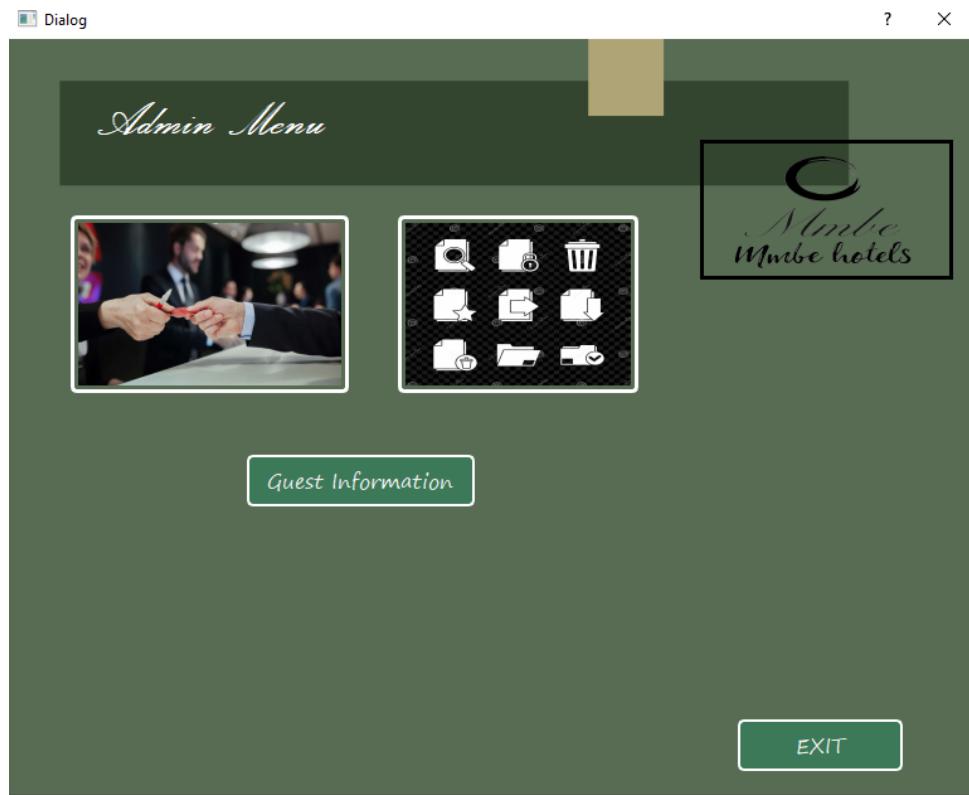


Figure 20: Admin UI

The screenshot shows a "Dialog" window with a search bar at the top containing the placeholder "Enter username" and a "search" button. Below the search bar are three buttons: "Remove", "Modify", and "Sort". To the right of "Sort" is a "Back" button. Below these buttons is a table with the following data:

	username	password hash	full name	age	city	contact num
1	mahrukh	147431c5761...	Mahrukh	19	Lahore	09876543211

Figure 21: Admin Page UI