**Hotel Reservation System**

Sean Kuo, Siyuan Sun, Sakshi Singh

**Documentation:**

**Strategy/approach to the project:**

Our final project entailed creating a "Hotel Reservation System" designed to streamline the process of booking hotel rooms. This system allows "users" to manage the platform by executing CRUD operations on the room details database via an interface. users have the capability to input and update reservation details including dates, and room IDs. Guests can browse through a list of rooms—curated and updated by the user—and select their preferred room for booking. Additionally, guests have the option to update their profiles and modify their booking details as needed.

The database has three tables includes:

1. “Booking” table (with columns “Booking-ID”, “Room-ID”, “Start-Date”, “End-Date”)
2. “Room” table (with columns “Room-ID, “Room-Name”, “Floor” “Beds”, “Room-Type, “Room-Number”)
3. “User” table (with columns “User-ID, First-Name, Last-name, Username)

A diagram of a computer

Description automatically generated

The three classes of user, room, and booking are first initiated into the HotelReservation.sqlite database through the respectively named files. Those files utilize CSV reading methods to initialize any data from the user, room, and booking into the database.

The system comprises several classes tailored to handle different aspects of a hotel reservation system, each equipped with a set of specific methods. The User class is designed to manage user data, providing functionalities such as creating, updating, retrieving, and deleting user profiles, along with a method to reset the user database. The Room class includes methods for room management, such as adding, updating, and deleting room entries, fetching room details by ID, and resetting the room database. A subclass of Room, the Booking class, extends the functionality to handle reservations, featuring methods to check room availability, add, fetch, and delete bookings, and find booked dates. An interactive menu is integrated to facilitate user interaction with the system, allowing seamless navigation and execution of various operations across user, room, and booking management. This structured approach ensures a comprehensive and efficient management system for both the administrative backend and the user frontend of the hotel reservation platform.

The user class has the following methods:

* Reset\_db: Resets the user database by deleting and recreating the user table, effectively clearing all user data.
* Create\_user: Adds a new user to the database.
* Update\_user: Updates exciting user details in the database.
* Retrieve\_all\_users: Retrieves details of all users in the database
* Delete\_user: Deletes a user from the database

The room class has the following methods:

* reset\_db: Resets the user database by deleting and recreating the room table, effectively clearing all user data.
* Create Room: Creates new room to database
* Update Room: updates room details to room id
* Fetch room by Id: fetch room information by room id
* delete room: Deletes room from DB

The booking class has the following methods:

* Reset DB: Resets the user database by deleting and recreating the room table, effectively clearing all user data.
* Fetch Booking Dates: finds all the booked dates given a room id
* Find Booked Dates: finds all the dates given a selected start and end date
* Check Availability: Checks if the selected start and end date fall within any previously booked dates
* Add Booking: adds a booking to a room id and checks that the room exists and that the dates are available before booking
* Fetch All Bookings: retrieves all bookings in the booking table
* Fetch Booking by Booking ID: retrieves booking information by a booking id
* Delete Booking: Deletes a booking by the booking ID

After all the methods are developed, an interactive menu is ran that allows for options to run the CRUD methods through the python console. Actions such as creating bookings, updating users, or retrieving rooms can all be done through the interactive menu.

**Ethics & critical thinking**

When developing a Python project like a hotel booking system, incorporating ethical considerations, and employing critical thinking are essential to creating a reliable and user-friendly platform. Here are some key points to keep in mind:

**Ethics**

* Data Privacy and Security: One of the primary ethical concerns involves protecting the personal information of users. Ensure that your system adheres to data protection laws such as GDPR or CCPA, depending on the geographical operation. Implement strong encryption for data storage and transmission and ensure that access to sensitive information is restricted to authorized personnel only.
* Transparency: Maintain transparency with users about how their data is being used. Include clear, accessible privacy policies and user agreements that outline what data is collected, how it is used, and who it is shared with.
* Fairness: Ensure that the system does not discriminate against any user or group. This includes implementing features that cater to the needs of people with disabilities, such as accessibility options in the user interface.
* Accountability: Develop an audit trail for administrative actions to track who made changes to the system, what changes were made, and when. This is crucial for maintaining accountability and troubleshooting issues.

**Critical Thinking:**

* Problem Identification: Use critical thinking to identify potential problems or shortcomings in the existing system. For instance, consider scenarios such as overbooking or double bookings and devise strategies to mitigate such issues.
* Decision Making: When faced with decisions, such as choosing between multiple ways to implement a feature, use critical thinking to evaluate the pros and cons of each option. Consider factors like user impact, cost, implementation time, and future scalability.
* Innovative Solutions: Encourage innovation by thinking beyond conventional solutions. For example, instead of just booking rooms, think about integrating additional services like room customization options (e.g., selecting pillow type or room fragrance upon booking).
* Feedback and Improvement: Utilize critical thinking to analyze user feedback and system performance data. This analysis can help identify areas for improvement, optimize processes, and enhance user satisfaction.
* Ethical Dilemmas: Prepare to face ethical dilemmas, such as handling user data when law enforcement requests information. Develop a clear protocol that balances legal compliance with user privacy rights.