**Design and Implementation of a UML-Based Hotel Management System in Python**

Ahmed Abdelaziz Al Qahtani

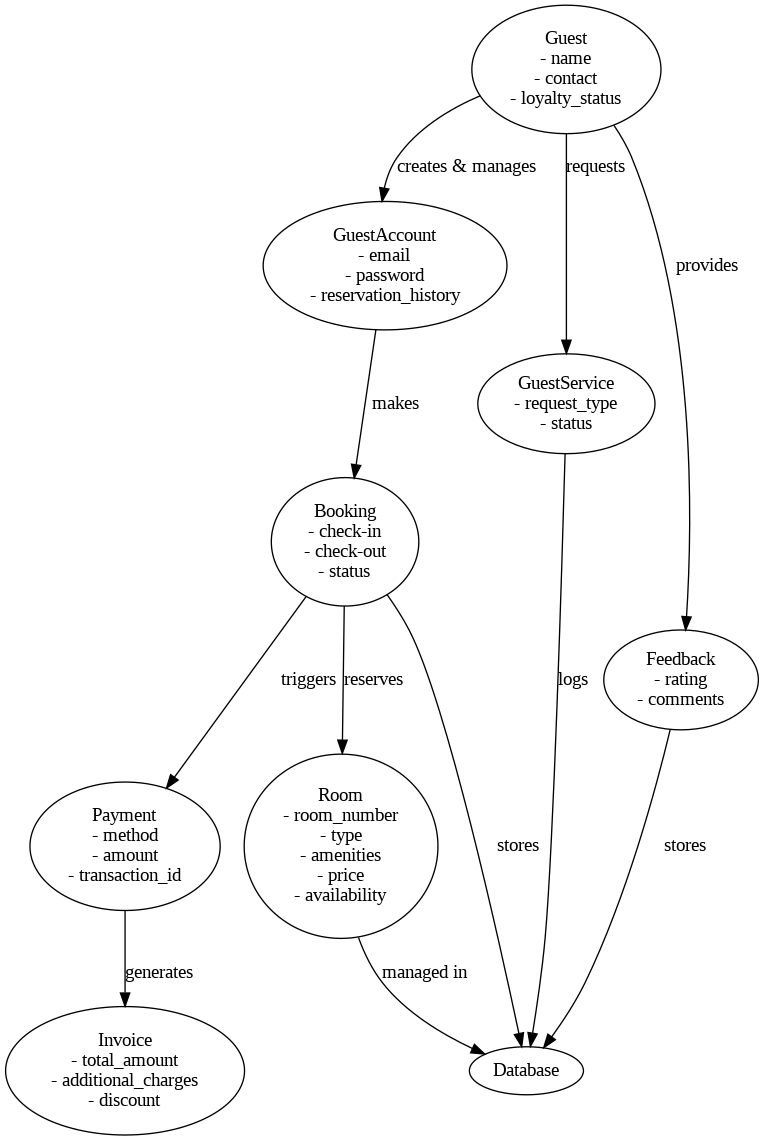
student id 202206351

Program Fund

18/03/2025

Design and Implementation of a UML-Based Hotel Management System in Python

**UML Class Diagram**

****

The Royal Stay Hotel Management System is designed using Object-Oriented Programming (OOP) principles, ensuring modularity and scalability. The key classes are:

**Guest** - Stores guest details such as name, and contact.

**GuestAccount** - Manages guest login and reservation history.

**Room -** Contains room attributes like number, type, price, and availability.

**Booking** - Handles reservation details, including dates and status.

**Payment** - Processes transactions for instance method, amount, and reference number.

**Invoice -** Generates billing details for bookings.

**Relationships**

**Composition** - GuestAccount owns multiple Booking objects.

**Association** - Booking is linked to both Guest and Room.

**Aggregation** - Invoice aggregates Booking details.

The Royal Stay Hotel Management System allowed me to further learn the Object-Oriented Analysis and Design (OOAD) principles. I structured the system with Unified Modeling Language (UML) to represent real-world hotel operations. The UML class diagram represented the interaction between different system components as a part of the critical step for efficient design without coding.

As for the Python implementation, it was built with modularity in mind, making each class a separate file. I used OOP concepts of inheritance, encapsulation, private attributes, getters, setters, and composition for maintainability. The method \_\_str\_\_() improved readability for the class, in addition, structured doc strings made it clear. The project was completely dependent on testing. I wrote code that uses unit testing and validated functionalities such as guest account creation, room reservations, invoice generation, payment processing, and booking cancellations. Writing test cases was also useful in locating potential errors as early as possible, resulting in a robust, and free of error, system.

Version control with GitHub also helped in project management because it allows for incremental progress to be tracked as well. Sharing the repository encourages collaboration and together, we can improve further. This project helped me gain a better grasp of correctly designing software so that one can use OOP best practices, and have the ability to create real world applications using modular code and structured documentation. The experience helped in bridging the gap between theoretical UML concepts and how they can be practically implemented using Python.