

Stage II B. Project Proposal and Specifications

Leah Kazenmayer, Anthony Messana, Michael Giordano

Problem Statement

- Many hotels have so much to keep track of: schedules, inventory, staff, the infrastructure/building itself, and so forth. To keep a hotel running smoothly, all employees need to be up to date with information regarding their role in running the hotel. The hotel's main efforts are needed for the actual maintenance rather than taking lots of time retrieving the necessary information.

Description of the desired end product, and the part you will develop for this class.

- A DBMS for hotels that records information regarding the components of a hotel that are relevant to its day-to-day operations. This system will include varying levels of data permissions which correspond to staff member's roles. The DBMS's target users are hotel staff who will use the system to aid them in conducting their daily duties and for use of the hotel owner to collect statistical data that can be used to inform business location. We will develop the conceptual schema in SQL, which will dictate how the data is organized and the way said data is related. We will also develop an external schema in the form of an easy to use web-interface that will allow end-users to access and interact with data in the database.

Description of the importance and need for the module, and how it addresses the problem

- Hotels have to manage a considerable amount of data regarding customer personal information, housekeeping, employee information, inventory of goods, and employee dependents. Access to this information and the ability to modify this information in real time is a necessity in the hotel industry to provide clients with quality service.

Plan for how you will research the problem domain and obtain the data needed

- The hotel data will mostly be made up, but they will act as realistic placeholders to test and run the database so that when it is actually used in real time, it'll be satisfactory to the hotel's needs. Real raw data from hotels is impossible to obtain, considering there's privacy policies that prevent the leaks of client/hotel information.
- We will research how hotels are managed and maintained, so we'll get a good idea of what they need in databases to keep track of what they need to buy to keep a well-supplied inventory, who occupies what rooms, which clients need what services/amenities, which rooms need improvements, and so forth.
- Additionally, personal experiences in hotels will help our knowledge and understanding in the creation of databases.

Other similar systems / approaches that exist, and how your module is different or will add to the existing system

- Mainly, the module will be different based on the organization of the databases. Other systems probably will have different approaches to how they keep all the information, but ours will focus on condensing the information in the least amount of entities possible, making it easily readable. Our module puts an emphasis on staff operations and streamlining their workflow.

Possible other applications of the system (how it could be modified and reused.)

- The system could be modified and reused for other businesses that are similar to hotels. For instance, with tweaks and modifications, the system could be used for motels, cruises, inns, and so forth. They would have the same entities with similar relationships, which makes it easier to change it for those purposes.

Performance – specify how and to what extent you will address this

- We will map out all data needed to keep track of in a hotel through ER/EER diagrams, which will organize everything so that it'll be easy to retrieve, resulting in less time trying to find the data and more time focused on the upkeep of the hotel.
- Furthermore, we will make all the queries and code as short, straightforward and time efficient as possible.

Security – specify how and to what extent you will provide security features

- Have a database for usernames and passwords, which will need to be changed every year for safety measures.
 - Use a n-ary structure to prevent the client from having direct access to the database server.
 - Username is the member ID that is given to every employee
 - Password needs to be at least 8 characters long; must include one lowercase letter, one uppercase letter, one number, and one non-alphanumeric character (like *, !, ^, etc)
 - This database will only be allowed for HRMS company to access, since nobody else would need access to it

Backup and recovery – specify how and to what extent you will implement this

- Have easily readable logs of who changed what in what entity/relation of the database (The member ID, the time, the specific database, what they changed specifically, etc).
- These logs will be kept up to a certain time frame (a month, six months, a year).

Technologies and database concepts the team will need to learn, and a plan for learning these

- We learned ER/EER Diagrams and SQL basics in class, which will help set up the database in total. Additionally, we will learn python outside of class and practice it independently in order for us to use it as the host language.

A diagrammatic representation of the system boundary that specifies what data you will model and which queries you will implement

- Room (Number, # of beds, rate, Occupancy)
- Staff (Assigned rooms, salary, employee number, FName, LName, SSN, age, birthday, sex)
- Dependents (Staff, FName, LName, age, sex, ESSN)
- Inventory(type, quantity)
- Customer (Room Staying in, start date, end date, FName, Lname, age, sex, credit card number, CustomerID)
- Guests(Customer, FName, LName, age, sex)

Relationships

- Staff works Rooms
- Customers stay in Rooms
- Guests stay with Customers
- Staff manages Inventory
- Staff manages Staff

Possible Queries

- Room_Status (returns person in room or vacant)
- Room_Vacant (returns the numbers of rooms that are vacant)
- Room_Rate (returns the number of each room, and its rate)
- Inventory_Summary[one for everything, and one for each type of thing] (returns each type and quantity of each type)
- Customer_Name (returns the first and last name of all customers, and their room name)
- Guest_Name (returns the first and last name of all customers and guests, and their room numbers)
- Staff_Name (returns the first and last name of all staff)

Transactions

- Guest_Booking (change a rooms occupancy to customer and any guests, or change a rooms occupancy to vacant)
- Modify_Roster (adds or removes a staff member)
- Modify_Staff (change a value of a staff member)
- Add_Inventory (adds to the quantity of an item in the inventory)
- Remove_Inventory (removes from the quantity of an item in the inventory)
- Change_Rate (changes the rate of a room)
- Room_Change (room customer is staying in changes, and changes occupancy of old and new room)

