



CS 353 - Database Systems

Hotel Database Management System Project Proposal

Group 38

Maryam Shahid	21801344
Bora Çün	21802838
Çağrı Eren	21801831
Hammad Khan Musakhel	21801175

Contents

1. Introduction	3
2. Project Description	3
2.1. Why do we need a database for the hotel management system?	3
2.2. How do we use a database as a part of the system?	4
3. Functional Requirements	5
3.1. Guests	5
3.2. Managers	5
3.3. Housekeepers	6
3.4. Security Staff	6
3.5. Recruiters	6
3.6. Job Candidates	7
3.7. Receptionist	7
4. Non-functional Requirements	7
4.1. Usability	7
4.2. Security	7
4.3. Performance	7
4.4. Portability	7
4.5. Availability	8
4.6. Maintainability	8
5. Limitations	8
6. E/R Diagram	9
7. Conclusion	10
8. Web Page	10

1. Introduction

This report consists of a proposal for a hotel database management system. It explains the basic functionality and extent of the project. The project description describes the application system, why a database is necessary, and how it can be integrated into the system. Next, the requirements are analyzed. Functional requirements thoroughly discuss the functionalities and scope of the users and capabilities of the project. The non-functional requirements make up the system design goals, such as usability, security, maintainability, and high-performance of the system. Next, the pseudo constraints and limitations are discussed, which address the limitations of the application system. It is then followed by an E/R diagram which forms the basis of the database system.

2. Project Description

The hotel database management system is a web-based project designed to allow users access to a hotel. The users consist of guests, employees, and candidates. Guests are able to book a room in whichever building and location they prefer. They are also able to rate and make comments on their reservations. Moreover, the system allows them to order food from various restaurants depending on the availability of food and buy tickets for a plethora of events conducted at the hotel, such as activities, group tours, pool parties, etc. Similarly, employees consist of managers, housekeepers, security staff, and recruiters. The system allows them to access their role-specific features. Managers are able to assign tasks to other employees - food orders for delivery and cleaning duties to housekeepers and buildings for security walks to security staff. They can also initiate events - guest activities, group tours, training programs - in various locations of the hotel. The system also gives them the authority to accept or reject applications for training programs and leave requests. Similarly, housekeepers are assigned cleaning duties to various rooms and deliver food to guests. Recruiters are able to view candidate applications, just as candidates are allowed to apply for available positions in the hotel. All the employees are allowed to fill a leave request form by checking their annual leave.

2.1. Why do we need a database for the hotel management system?

A hotel management system contains a plethora of data about the participants of the system. The entire system of the hotel has a vast array of information that needs to be

stored and updated via a database management system for efficient access to the information or queries requested. The hotel system has different functionalities that enable managers of the hotel to track food orders, assign duties to staff, and manage events, housekeepers to be assigned food orders for delivery, guests to reserve rooms and comment on their stay at the hotel, and much more. The database management system devised covers all the functionalities of the hotel management system in a precise and efficient manner.

Database management systems are the advanced format of conventional register systems where paperwork was done for management systems; database management systems are more efficient as they only need to be updated with time to keep it relevant as once designed, it has the relevant information at a click away. It can be viewed that the hotel management system has so many participants and relations (shown in the ER diagram) that it needs a database management system for better functioning: fulfilling its Engineering prospect by making the system easier to manage, use, and understand.

2.2. How do we use a database as a part of the system?

The database is going to provide all the information to manage all data related to the hotel management system. The database would be available to be used by guests and Employees overall. The user would be able to reserve different types of rooms based on availability in different buildings in other locations; the employees would also have access to the database: managers would be able to track food orders given by guests and also be able to assign the food orders to housekeepers for delivery, to create different types of events in different locations of the hotel, to accept or deny training program applications or short-leave requests, and be able to assign duties to other staff members; housekeepers would be able to receive food orders by the managers via the database system; a recruiter would be able to hire candidates as employees based on applications provided through a systematic manner. All of the functionalities listed above are only possible when the database is updated with relevant information, in essence, requires maintenance. There are numerous relations between different participants and elements of the database system with their own attributes, keys, and entity statuses which need to be updated accordingly. Consequently, a smooth management system is at disposal where the groundwork of the database management system allows the users to interact with one other in an engaging way.

3. Functional Requirements

Every user type can sign up to the system using their personal information. Later, they can log in using this information. However, if the user is not registered to the system, they can still see some general information about the hotel. The user-specific functional requirements are as follows:

3.1. Guests

- Guests can see all available rooms.
- Guests can reserve a room.
- Guests can see information about their room, such as location, reservation status, etc.
- Guests can view and change their information.
- Guests can see details about their stay such as room information, the extra payment amount, etc.
- Guests can comment and give a rating on their reservations.
- Guests can ask questions from the reception.
- Guests can see their questions and the receptionist's answer.
- Guests can see a list of restaurants.
- Guests can see the menu of each restaurant.
- Guests can order the food they want from available restaurants.
- Guests can see the status of their food order.
- Guests can see a list of activities.
- Guests can buy tickets for activities and group tours.
- Guests can see a schedule of activities they have signed up for.

3.2. Managers

- Managers can see the food orders of the guests.
- Managers can assign food orders to housekeepers for delivery.
- Managers can track the food deliveries.
- Managers can assign cleaning duties to housekeepers.
- Managers can track the cleaning duties.
- Managers can assign security staff for security walks.
- Managers can see which security staff is at which location.
- Managers can create events. An event could be a training program, an activity for guests, or a group tour.

- Managers can see all events with their time and locations.
- Managers can choose and schedule places to visit in a group tour.
- Managers can accept or reject applications to training programs.
- Managers can accept or reject leave request forms.
- Managers can see statistics about the employees, guests, events, the economy of the hotel, etc.
- Managers can fire other employees.
- Managers can add new locations or buildings and remove the existing ones.

3.3. Housekeepers

- Housekeepers can see what assignments they need to accomplish at what times.
- Housekeepers can mark their delivery and cleaning duties as done.
- Housekeepers can apply to training programs.
- Housekeepers can fill leave request forms.
- Housekeepers can view their applications and forms.

3.4. Security Staff

- Security staff can see where and when they have security walks.
- Security staff can apply to training programs.
- Security staff can see their active training programs.
- Security staff can fill leave request forms.
- Security staff can view their applications and forms.
- Security staff can report suspicious activity.

3.5. Recruiters

- Recruiters can see the job candidates who applied for different positions.
- Recruiters can view the cover letter of each candidate.
- Recruiters can accept or reject job applications. If the application is accepted, the candidate becomes an employee.
- Recruiters can apply to training programs.
- Recruiters can fill leave request forms.
- Recruiters can view their applications and forms.

3.6. Job Candidates

- Job candidates can apply for jobs in the hotel.
- Job candidates can choose the position they are applying for (housekeeper, security, staff, manager, recruiter).
- Job candidates can submit a cover letter for their job applications.
- Job candidates can view their applications and its status.

3.7. Receptionists

- Receptionists can view the questions of guests.
- Receptionists can answer the questions of guests.
- Receptionists can apply to training programs.
- Receptionists can fill leave request forms.
- Receptionists can view their applications and forms.

4. Non-functional Requirements

4.1. Usability

- The site will have a simple UI that is not complicated with unnecessary details.
- Pages will be easy to navigate.
- Each page will only have features that are related to it.
- User sessions will be created so users will not have to re-login to the system if he/she leaves.

4.2. Security

- Each user type will have its own level of access to pages and functionalities.
- Preconditions of the wanted action will be checked before the action occurs.
- Important information like passwords will be hashed then stored in the database.

4.3. Performance

- Software engineering concepts will be used to enhance the efficiency of system code.

4.4. Portability

- The application can be used on computers, phones, tablets, and other devices.

4.5. Availability

- The system will be available at any time and continue its functionality.

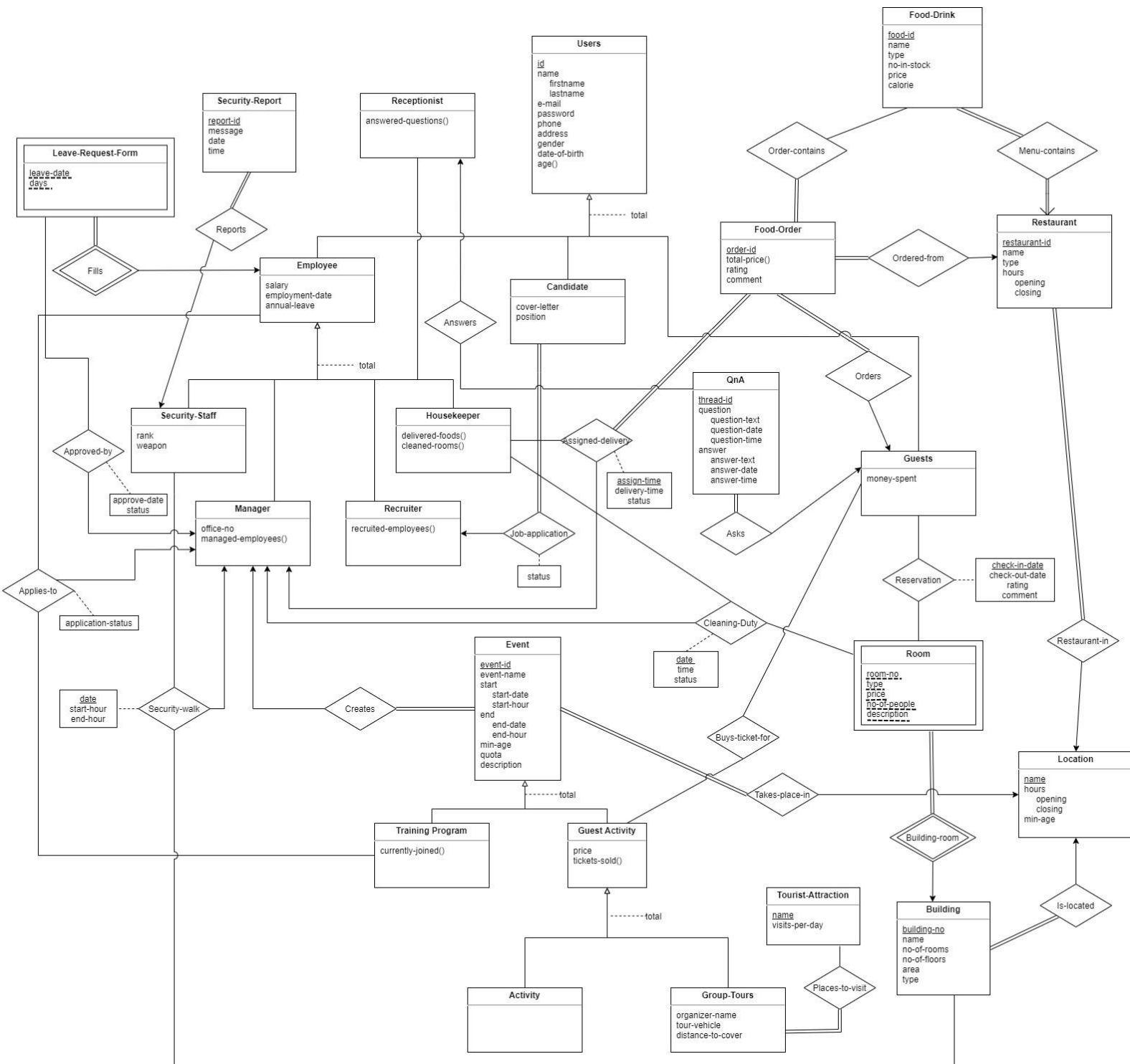
4.6. Maintainability

- The system will be implemented using software engineering concepts that will make it easier to implement further needed changes.

5. Limitations

- No new restaurants can be added to the system, and the existing ones cannot be removed.
- The restaurants cannot edit their respective menus.
- There is no limit on how many employees can be employed at a time.
- A particular assignment cannot be assigned by more than one manager.
- A particular application cannot be accepted or rejected by more than one manager.
- A registered user must be either a guest, an employee, or a job candidate.
- An event must be either a training program, an activity, or a group tour.
- There cannot be multiple locations with the same name.
- A room can be cleaned at most once per day by a particular housekeeper.
- A guest's question can have at most one answer.

6. E/R Diagram



7. Conclusion

The hotel database management system is a web-based application for managing a hotel system. It is designed to be used by guests and employees primarily with its functionalities assisting in the management of an enormous system; its enormity comes along with its demand for a vast number of functionalities and data management of different types of users. The database system designed here covers all the requirements of the hotel management system. In this report, we laid the foundation for our database management system by introducing and describing the project description. Afterward, we listed the functional and non-functional requirements of the system, its limitations, and lastly also presented the ER diagram of the database management system in our proposal.

8. Web Page

The reports of this project will be published on the following web page:

<https://cs353-group38.github.io/>