



Car Rental System Project Proposal

CS 353 Database Systems

Project Proposal

Car Rental System

Group No 17

Emre Caniklioğlu - 21803577

Celal Berke Can - 21702886

Ege Demirkıran - 21802482

Berk Saltuk Yılmaz - 21903419

Table of Contents

1. Introduction	2
1.1 Overview	2
1.2 Description	2
1.3 Objective	2
1.4 Why & How	3
1.4.1 Why a database system will be used?	3
1.4.2 How a database system will be used?	3
2. Requirements	4
2.1 Functional Requirements	4
2.1.1 For Customers	4
2.1.2 For Managers	4
2.1.3 For Employees	4
2.2 Non-Functional Requirements	5
2.2.1 Reliability	5
2.2.2 Maintainability	5
2.2.3 Performance	5
2.2.4 Authentication	5
3. Limitations	5
4. Constraints	6
4.1 Backend	6
4.1.1 Java	6
4.1.2 PostgreSQL	6
4.2 Frontend	6
4.3 Project Reports	6
5. ER Diagram	7

1. Introduction

1.1 Overview

This proposal is for a database management system project which is about implementing a web-based Car Rental System. This proposal will contain a brief description of the application, establish functional and non-functional requirements, mention limitations, restrictions the technologies which we will be using while implementing the application. At the end of the proposal we will propose an E/R model for the database system. This system will be used by both the company's employees and their customers.

1.2 Description

This application will be a web-based system for customers to rent a car according to their driving experience. The system will be used by customers, employees, and managers. There will be a password for each user to log in to the system. For customers, we will require a driving license at the registration stage in order to control reservation requests suitable to customers' driving experience. After signing in or signing up, a customer can see the cars and can reserve the cars in the database with a smooth and easy-to-use UI. Moreover, a customer can make a request to have the car transferred to any city. After rental time ends the customer will be able to return the car to any branch. After the return, the customer will do a review about the rental experience and these reviews will be published to give an opinion to other users. Moreover, employees can also log in to the system to handle the transfer request and complete a reservation, employees will be able to see if there is a need for a penalty due to any damage to the car or if the deadline is passed. Finally, managers will be able to use the system and they will be able to hire new employees and buy new cars and then add these employees or cars to the database.

1.3 Objective

In the database system project, we are going to implement a web-based Car Rental System which will imitate a real life car rental system. While implementing the database and the application layer we intend to use the best practices followed by many other developers around the world. Implemented system will be used by both

the company's employees and the customers. The main goals of this web-based system are as follows:

1. To offer customers an easy to use car rental service where they can choose from a wide range of options.
2. To offer customers different branches that they can rent cars suitable for their taste and driving experience.
3. To offer customers an easy renting experience with an easy to use UI, online payment with flexible return and pick-up locations.
4. To improve the system and customers experience, a review system where each customer can give feedback about their rental experience.
5. To offer employees an easy to use management platform where they can manage their daily tasks.
6. To offer an easy monitoring system for reservations such as monitoring ending and starting dates, and charge a penalty for excess time and for damage of cars if needed.
7. To help managers to hire new employees from the job offerings section, and buy new cars by adding them to the database.
8. To provide an admin panel where managers and employees can access the database.

1.4 Why & How

1.4.1 Why a database system will be used?

To keep the data of various elements of the car rental system such as cars, customers, reservations, employees and managers, we are in need of a database system. We are using a database system instead of a file system because with the usage of database systems it is much easier to keep data and access to the data later, and also it is faster. Moreover, by having a database we are able to have limitations on the data, and we can visualize the data organization by using relational model and entity-relationship model.

1.4.2 How a database system will be used?

Relational schema will be used to keep track of main tables, and to determine relation between tables, foreign keys, the Entity-Relationship model diagram will be used that can be found in section five of this paper. Using this relational schema and structured query language, PostgreSQL database will be created. Finally, we will manage, modify, and keep track of the application's data using the car rental system database.

2. Requirements

2.1 Functional Requirements

2.1.1 For Customers

1. Customers will be able to reserve cars.
2. Customers will need to upload their driving license.
3. Customers will need to have a certain level of experience to reserve a car which will change according to their driving license.
4. Customers will be able to make requests for the transportation of the car between branches.
5. Customers will be able to drop off the car at any branch they want.
6. Customers will be able to make payments at any branch they want.
7. Customers will be penalized if the deadline for the drop off is passed.
8. Customers will be able to give reviews about their renting experience after they dropped off the car.
9. Customers will have access to the admin panel.
10. Customers will need to sign a contract while renting a car.
11. Customers will have an extra option for insurance.

2.1.2 For Managers

1. Managers will be able to hire new employees to the branches they work in.
2. Managers will be able to buy new cars to the branches they work in.
3. Managers will have access to the admin panel.
4. Managers will be able to do every task that an employee can do.

2.1.3 For Employees

1. Employees will be able to approve or decline the customer's reservation.
2. Employees will be able to handle customer's request for a car in another city.
3. Employees will be able to take payments from the customer.
4. Employees will be able to check the car for damages when it is returned and penalize the customer accordingly.
5. Employees will be able to sign contracts with the customers.
6. Employees will be able to get a variety of reports about past data to examine them just giving parameters to the system like start date and end date.

2.2 Non-Functional Requirements

2.2.1 Reliability

- The database system will be designed in a way that it ensures coherence during data modifications.
- The database system will also ensure that there will be no data lost during any kind of system failure or malfunctions.

2.2.2 Maintainability

- Creational, structural or behavioral objects oriented programming patterns will be used to improve the application's modularity.
- 'SOLID' software practices will be used for SQL databases to make the system extensible, but not modifiable.
- SQL queries will be encapsulated and modularized to improve scalability of the system.

2.2.3 Performance

- Application layer will use caching for faster response times.
- For better performance, SQL queries will be optimized.
- The spring entities will be used for entity lifecycle management to improve the performance of the system.

2.2.4 Authentication

- PostgreSQL users with different access levels will be used to give access to the admin panel.
- Role based authentication will be used for different endpoints.

3. Limitations

- There cannot be more than one user with the same email address.
- Passwords must be longer than six characters, it must contain at least one number and it can't contain a sequence of numbers longer than three.
- Customer reviews cannot exceed 250 characters.
- A customer can request his/her desired cars to be transferred to the branches in his/her city and can rent the car from there, but there is no delivery to home option.
- A customer will have a cost penalty if he/she hands over the car after the reservation deadline, damages the car in the reservation period, or abandons the approved transfer request.
- A customer cannot reserve a car, if he/she does not have a compatible driver license.
- Each reservation is associated with only one customer, but can be checked by more than one employee.

- Each transfer request is associated with only one customer, but can be handled by more than one employee.
- A customer will receive a penalty, if the reservation deadline passes.
- Each branch must have at least one manager.
- A manager can be associated with only one branch.
- A manager can hire employees and purchase new cars for only his/her own branch..
- A customer cannot cancel his/her reservation if there is less than one hour to the reservation starting time.
- Customers cannot give a reservation review before the reservation deadline.
- Managers/employees cannot modify customer reviews.

4. Constraints

4.1 Backend

4.1.1 Java

Java will be used as a main language while implementing a web server that will be responsible for querying the required data and serving static HTML pages to the user. To achieve this goal in addition to other libraries we will mainly be using the following packets from the spring framework.

- Spring Web for creating a web server
- Spring Security for setting up authentication of certain URL endpoints
- Spring Data JDBC for executing SQL queries
- PostgreSQL driver for connecting web server to database
- Thymeleaf as Java templating engine

4.1.2 PostgreSQL

PostgreSQL will be used as the structured query language for the creation and management of the car rental database system.

4.2 Frontend

Javascript, HTML, CSS, and their related libraries will be used for building frontend of the project.

4.3 Project Reports

Project implementation and reports can be reached via following website and repository:

<https://edemirkirkan.github.io/Car-Rental-System/>

<https://github.com/edemirkirkan/Car-Rental-System>

5. ER Diagram

