Bilkent University
Department of Computer Engineering

CS353 Database Systems

Group 13 Project Proposal
Car Rental System
20.10.2021

Group Members:
Akmuhammet Ashyralyyev - 21801347
Berke Ceran - 21703920
Hakan Gülcü - 21702275
Sila Saraoglu - 21803313

Instructor: Özgür Ulusoy
Teaching Assistant: Mustafa Can Çavdar

Table Of Contents

1. Introduction

This report is the report of the car rental system project of the database systems course. In the content of the report, it gives information about the project's objectives, functions, limitations, database E/R model, requirements, and usage and improvements of the project.

Chapter 2 contains the description of the project. This section explains what the project is, why the project needs a database, and how to use the database in the project.

Chapter 3 contains the requirements of the project. This topic is examined under 2 separate headings: Functional requirements and Non-functional requirements. In the functional requirements section, it is explained how different types of users will use the application and how the system should/will respond to incoming requests. Non-functional requirements are divided into two: Quality Requirements and Constraints. Quality Requirements talk about system features other than functional ones, eg security, performance etc. Constraints, on the other hand, talks about the technologies that will be used to develop the project.

Section 4 contains limitations. In the Limitations section, the limits set by the system and the actions that users can and cannot do are shown.

Chapter 5 contains the Entity Relationship Diagram. In this section, the Entity Relationship Diagram for the Car Rental System is shown and the database design of the project will be done in accordance with this diagram.

Section 6 contains website information. From the link added here, the uploaded proposal, design, final reports for the project and viewing on GitHub options can be seen.

2. Project Description

This project is an online Car Rental System application that functions over the web and database. In this project, users can rent various vehicles over the web according to their driving experience (license information). Users have the chance to rent the vehicles they want in a city and deliver them to the city they go to. In addition, users can transfer the vehicles they want from other cities to branches in their own cities when they want and fulfill the conditions. Users own the vehicles for a certain period of time after renting their vehicles on the website for the criteria they want, such as the color, model, vehicle status and the year of the vehicle, etc. In this rental agreement, users sign for a certain period of time and must deliver their vehicles to any branch of the company within this period. If they do not deliver their vehicles within the agreed time, they may be subject to various penalties set by the company. After the delivery of the vehicle, the employee who receives the vehicle in that branch checks whether there are any defects, such as scratches or damages in various parts of the vehicle, and if there is a defect on the vehicle, an additional fee is charged from the users. In addition to the car rental process, users can give various feedback to the company, various branches of the company and the vehicles they use, and other users may rent or not rent the vehicles according to these feedbacks. Thus, users get rid of various prejudices while renting vehicles. According to these feedbacks, the company may remove various vehicles from the rental list, punish employees who do their job badly or hire new employees in unsatisfactory situations.

2.1 Why Database Needed

In our daily life, file systems or database systems are used to store various data in a computer environment. When we consider our project, many users will access various data at the same time, change that data in various ways and save it to the system again. Doing these operations over file systems may cause various problems, delays, and errors. Also, simultaneous access of multiple users to file systems will cause various problems, as file systems cannot perform various read or write operations fast enough, unlike database systems. In addition, database systems can be easily managed in terms of various access and restriction permissions, and these systems allow us to store a lot of complex data in a more efficient and orderly manner and to use them later in a feasible way.

2.2 How Database Will Be Used

The database system will be used for various purposes in our project. The first of these goals is to store, retrieve, and represent various complex and important data in an easy and efficient way. For example, in our system we aim to store multiple branches, multiple employees and cars in those branches, details of cars, and user information. In addition, the addresses of various data kept in the database enable users to find the data with different characteristics they are looking for more efficiently through various queries. For example, when users filter several characteristics of cars, specific cars will be returned from the database system in queries to show them to the user. In addition, updates, removals, and insertions are made in a simple way for any entry, and management of various tables is easy. For example, managers hire employees and buy new cars for their branches and they will be easily stored in the database system.

3. Requirements

3.1 Functional Requirements

3.1.1. Customer

- The customer should register to the system before using the rental application. He/she must provide the personal information, which are name, surname, birth year, national id, gender, email address, password, contact number. This information is asked to the user since they are all required in renting the car.
- The customer must verify his/her email before entering the system. In the Customer entity, we had defined an attribute called account_status which will store either ‘verified’ or ‘unverified’.
- The customer should be able to see the cars available at the specific date which is specified by the customer himself/herself.
- Customer should be able to request to change his/her national id. This is implemented since there might be a case if the user is a foreigner and changes its nationality, resulting in change of national id.
- The customer should be able to request to change his/her date_of_birth and gender by contacting the administrator. These cases are considered since in the real world, gender can be changed as well as date_of_birth by the will of the customer. He/she might not want to create a new account in order to keep older renting records etc.
- The customer should be able to edit his/her information such as email address, password, contact number. But for email address and contact number, the verification will be asked by sending the code to their email or phone.

- The customer should be able to set the profile image
- The customer should be able to reserve the car if it is available. However, he/she must introduce his/her license earlier. Also, the license type should allow him/her to drive the reserved vehicle.
- Customer can become a Premium Customer only if he/she meets the total number of renting which is determined by the company. After he/she returns the car, the system will check the count and if it meets the limit, then he/she will automatically be upgraded to it.
- If a customer and employee are the same person, which means that employee wants to rent a car, then he/she cannot rent himself/herself a car.
- Users can rent a car from every branch but once at a time. If there is ongoing renting, the system will not allow him/her to rent a new car.
- The customer will have to pay a proportion of the deposit while reserving the car. Also, he/she will not be refunded if on cancelation of the reservation.

3.1.2. Premium Customer

- Premium customer will have a starting date of his/her premium membership.
- Premium customer will be offered by campaign or special discount offered only to her by the manager.
- Premium customer will be able to rent sports cars.
- Premium customer will not be refunded if he/she cancels the car reservation.

3.1.3. Employee

- The account for the employee will be created by the manager of the branch where he/she applied.
- The employee will be able to start renting protocol or decline the request.

- The employee will be able to see all information related to cars, customers and rental scheduling.
- The employee will be assigned to the department and this information will be changed only by his/her manager.
- After the approval of the assignment request to another branch by an employee, the change in branch will be made by the administrator.
- The employee will have some number of leave predefined in contract. In order to use them, he will request it from his/her manager. In case of approval, this number will decrease accordingly. Renewal of this number will be annually, and remaining numbers will be summed up with new ones. Manager will have to decline this request if there is a lot of work that needs to be done but if there remains no adequate time to use them for the employee, then the manager must approve (if the employee uses this right at the end of the renewal time).
- The employee can be fired only by the manager.

3.1.4. Manager

- The manager can hire people as well as fire them.
- The manager can also start the renting protocol.
- The manager will be able to see all information related to cars, customers, rental scheduling and employees.
- Manager has similar functionalities and rights such as start renting protocols. On the other hand, even if the number of allowed leave is limited by the manager, he/she does not need to take approval from any authority.
- Manager cannot approve a rent applied by himself/herself. Other employees in the company will have to approve it.

- Employees can change their status to manager only by administrator.
- Manager can request cars from other branches.

3.1.5. Car

- Cars can be transferred to various branches.
- Cars can be requested from customers.
- Cars have various details such as fuel type, current-km-value, color etc.
- Cars can be insuranced during rental time by customers.

3.1.6. Company

- The company can be described as a singleton object. No other company will be allowed to the database.
- The company will have branches, hence, refer to the branch table.
- The name and description can only be changed by the administrator. The request for them will be done by managers.

3.1.7. Branch

- Each branch will have one manager
- Branch can be created only by the administrator.
- Branch will have a city name, branch name and address, which can also be changed by the administrator. The request for these changes will be done by the manager of that branch.
- Each branch will own the number of cars and cars will be possessed only by them. The customer will be able to rent those cars only from their branch.

3.1.8. Administrator

- The administrator can create a company.

- The administrator can create branches and their managers.

3.2 Non-Functional Requirements

3.2.1. Quality Requirements

3.2.1.1. Usability

- To ease navigation between pages, a navigation bar will be present at the top of the page and even if the page changes, this navigation bar will never change to enable users to go to different pages easily.
- To help users who are new in the system, there will be a help page which explains what to do in several situations. There will be an “Contact Us” part whenever there is a problem with the system, users can directly communicate with the us.
- Specific buttons for several actions will be colored according to the action’s category. For example, when deleting a request, the button will be colored red to give the user the message that it is a warning. Moreover, specific actions in the system for example renting a car, the system will pop up a modal to ask whether the user is sure or not for the action. In this way, users will make less mistakes in the system.

3.2.1.2. Performance

- The system which is a web application needs to support at least 1000 users at a time.
- The system’s response for every instruction in the system must be done in at most 5 seconds.

- The loading time of a page needs to be done in at most 2-3 seconds.
- The system should handle the process of storing large data because images of the cars before and after rent needs to be in the database.
- The system should have a high performance rate during payment processes.

3.2.1.3. Scalability

- The system must be scalable because the number of customers which use our system can always increase or when a manager hires a new employee, the number of the employees who are using our system will increase.

3.2.1.4. Security

- The system should protect the visibility of passwords at any time.
- The system should not create a user until he/she verifies his account with his/her password.
- The system should protect the visibility of the confidential information of the users such as nationality id.

3.2.1.5. Availability

- The system should be available 7/24 so that users can use the system whenever they need.

3.2.1.6. Portability

- Users can reach the system on their phones and computers. The system will be also compatible with Chrome, Safari, Firefox browsers.

3.2.1.7. Capacity

- The application will be built into a database system. Therefore, the system should have enough capacity to store the company's branches, customers, and employees etc. The aim at the beginning is to store 1000 user capacity in the system.

3.2.2. Pseudo-Requirements (Constraints)

3.2.2.1. Implementation

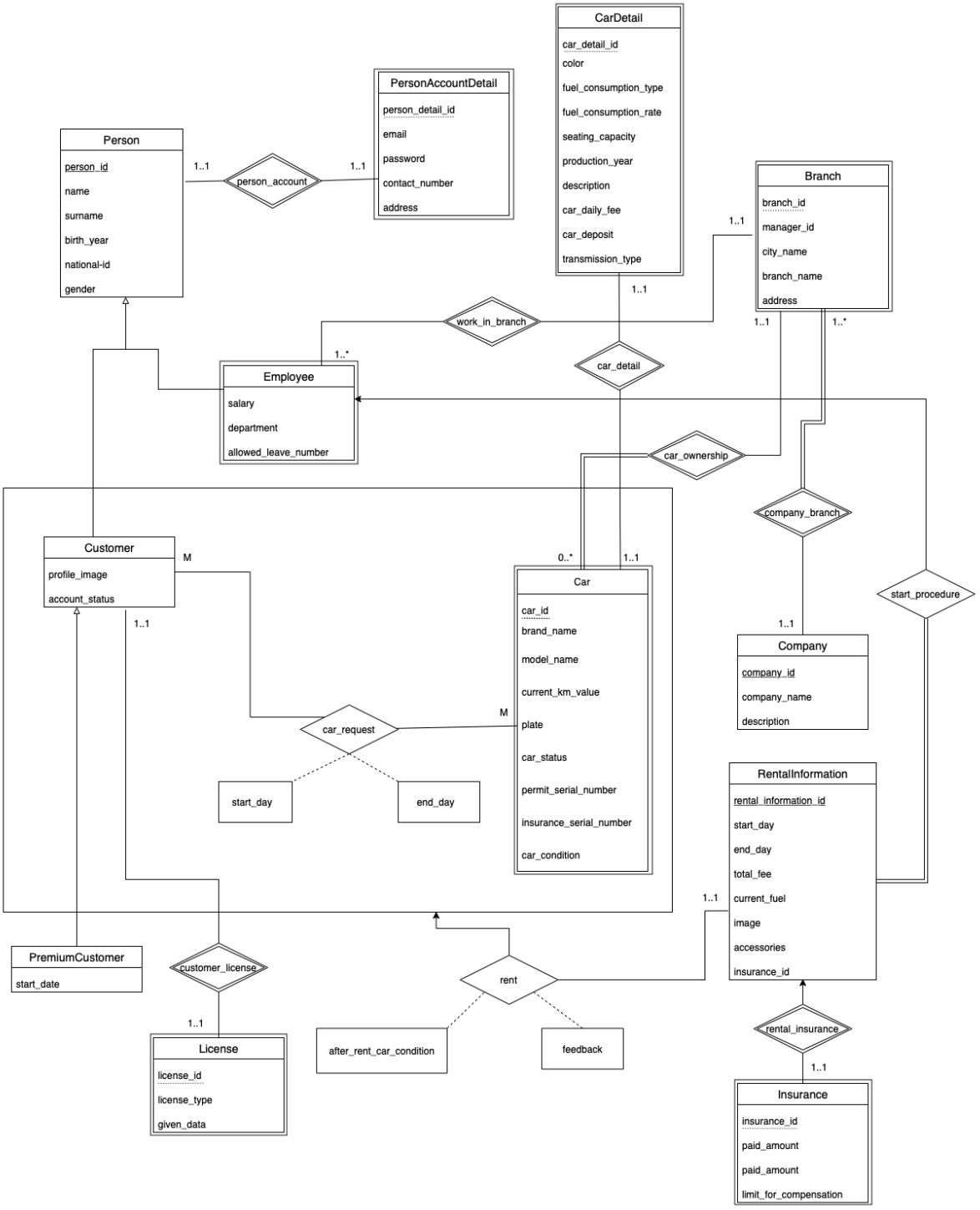
- MySQL will be used for the database management system.
- PHP will be used to develop the back-end of the application.
- React -which is a Javascript library- will be used for the front-end of the system.

4. Limitations

- A user has to have an email or username and password to login.
- Passwords have to contain at least 8 characters with at least one uppercase character, one number and one symbol.
- Username has to contain at least 8 and at most 32 characters.
- A maximum of one account can be opened with the same email and username.
- An user can only have one user role.
- A customer can request multiple cars however an employee can approve at most one request within the same time period.
- A customer can rent a maximum of one car at a time.

- Customers who obtained the driving license less than 2 years ago or who have not a driving license cannot rent a car.
- Customers who have been a member for less than 6 months can rent a car for a maximum of 15 days.
- Customers registered for less than 12 months cannot become premium members.
- Customers who have not rented a car at least 3 times cannot benefit from discounts and premium features.
- Customers can only comment on cars they have previously rented.
- Customers who prepay but do not receive the car on the day of delivery cannot get a refund if they are not a premium customer.
- Only employees can reply to incoming comments.
- Penalties are paid when delivery is delayed by more than 3 hours.
- Penalty paid when a car accessory is lost or car gets damaged during rental period if it is not insuranced.
- If the insurance limit exceeds during an accident, customers have to pay the rest fee.
- A company must have at least one branch and in this branch there should be at least one employee.

5. Entity Relationship Diagram



6. Website

https://berkeceran.github.io/CarRentalSystem/

You can access the reports by clicking these links.