

**CS-353**  
**DATABASE SYSTEMS**  
**PROJECT PROPOSAL REPORT**  
**GROUP 20**



**Erin Avllazagaj**  
**Can Bayraktar**  
**Utku Uçkun**  
**Ege Yosunkaya**

# **Table of Contents**

## **1.0 Introduction**

## **2.0 Project Description**

### **2.1. Why/How a Database is going to be used**

## **3.0 Requirements**

### **3.1 Functional Requirements**

#### **3.1.1 User**

#### **3.1.2 Driver**

#### **3.1.3 Customer Service**

#### **3.1.4 System Requirements**

### **3.2 Non Functional Requirements**

#### **3.2.1 Quick Response Time & Scalability**

#### **3.2.2 User Friendliness**

### **3.3 Pseudo Requirements**

## **4.0 Limitations**

## **5.0 E/R Diagram**

## **6.0 Conclusion**

## **7.0 Webpage**

## **1.0 Introduction**

This proposal report describes our application system which is for a Car Sharing System. The report explains the project in a detailed manner, why and how a database is going to be used for the system, its requirements limitations and the conceptual design of the database using an E/R model.

The Project Description part of this report explains the Car Sharing System in a very detailed way. Every functionality and purpose of the system will be explained. The Requirements part will explain functional, non-functional and pseudo requirements of the system which are divided into subtitles. The last part of this report will be the diagram where entities and their relations between them will be displayed as an E/R diagram.

## **2.0 Project Description**

Car Sharing System is an application where people going to the same direction share their cars. The aim of this system is to bring people together in the same platform and provide safe, cheap and effective ways to share cars. With this system drivers can create a trip saying which location they are going to go at a specific time and through which route they are going to do so. The passengers can enter a location and see all the relevant trips destined to that location. Passengers need to make a reservation to a trip to participate in that trip. The driver has the option to accept or decline the passenger. After a trip is done, the passengers can rate the drivers and other passengers and the driver can rate their passengers. Both can also write reviews about them anonymously. The payments are done beforehand and can be refunded if the driver does not show up or the passenger cannot attend due to some reasons. The system has a customer service which helps the customers to solve their problems about the trips and the refunds. The information of the customers are protected. People can only see the reviews and rates of each other before making a reservation. After the reservation is done and the passenger is accepted by the driver, their contact information becomes available to each other. There are multiple locations a passenger can get in or get out and will be charged accordingly. The driver can set up mid stops on the route and charge those stops differently. Also, the driver is responsible for estimating the trip's time and will set up a price for the trip according to the estimated time and trip length.

### **2.1 Why/How a Database is going to be used**

The aim of this project is to provide a database application for the Car Sharing System to make data management easy. As this is a complicated system, a well designed database is needed. There are many passengers, drivers, ratings, reviews, trips, vehicles, locations, routes, time and cost information to manage, therefore an automated database system is needed for this Car Sharing System. This database application is going to handle query operations, data entries and updates.

## **3.0 Requirements**

### **3.1 Functional Requirements**

#### **3.1.1 User**

- Users should be able to modify their personal information
- Users should be able to register their car and become a driver.
- Users should be able to search for trips with specific start and end location.
- Users should be able to search for trips with specific time frame.
- Users should be able to make a reservation to a available seat in a posted trip.
- Users should be able to cancel their reservation.
- Users should be able to see driver's personal information such as phone number after their registration is approved by the driver.
- Users should be able to rate drivers and other passengers if they have been in a same trip.
- Users should be able to see driver's rating before making a reservation to the trip.
- Users should be able to see trip information such as car information , stop locations , other passengers.

#### **3.1.2 Driver**

- Drivers should be able to modify the information of their car.
- Drivers should be able to post a trip with specific start and end location and start and end time.
- Drivers should be able to determine the price for their trip
- Drivers should be able to see reservation requests for their trip and approve or cancel the reservations.
- Drivers should be able to see passengers' personal informations such as phone number after approving their registration.
- Drivers should be able to see passengers' ratings if they have made a reservation request.
- Drivers should be able to rate the passengers if they have been in a same trip.

#### **3.1.3 Customer Service**

- Customer service should be able to see drivers car information.
- Customer service should be able to see user's personal information
- Customer service should be able to search for trips with specific start and end location.
- Customer service should be able to search for trips with specific time frame.
- Customer service should be able to see a trip's information

#### **3.1.4 System Requirements**

- System should be able to block reservations after the capacity of a trip is full.
- System should be able to block reservations 1 hour before the starting time of a trip.
- System should be able to verify user's' email.
- System should be able to limit the unverified users' functionality with only search for trips.

## **3.2 Non Functional Requirements**

### **3.2.1 Quick Response Time & Scalability**

- The system should be able to fast and scaleable. Since there will be searches and posting new data , system should be able to handle searches and database modifications as fast as possible.

### **3.2.2 User Friendliness**

- The system should provide a easy-to-use interface.
- The system should guide new users and show the features of the system.

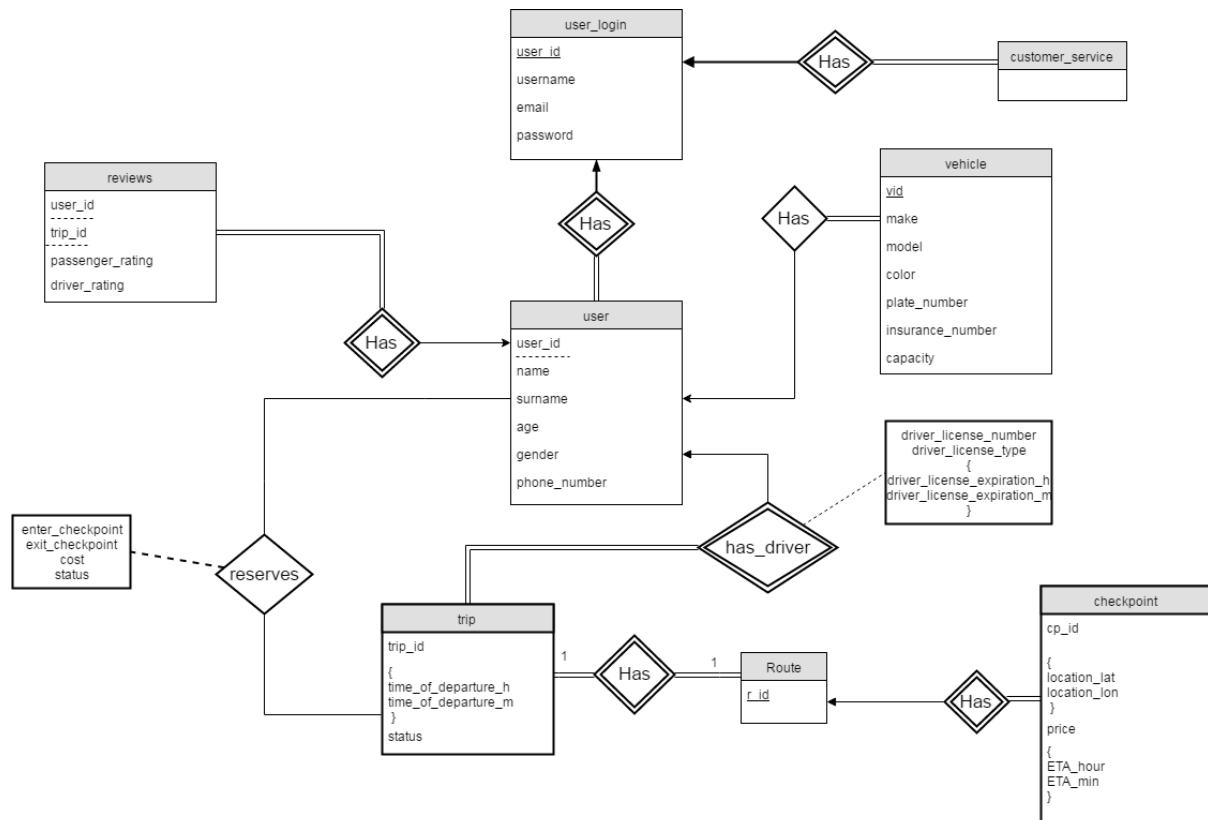
## **3.3 Pseudo Requirements**

- MySQL will be used for database.
- PHP will be used for backend.
- HTML ,CSS, JavaScript will be used for web page.

## **4.0 Limitations**

- People can search for trips but they need to login to their user account in order to reserve a trip or host one.
- Only the users with a validated car, a car that is registered on their name, can host a trip. This information will be collected from traffic database.
- A driver needs to accept a passenger's reservation in order to confirm the reservation.
- A driver needs to specify where he is going and where he is willing to stop (to pick up people) when creating a new trip.
- When a vehicle's capacity is filled users cannot reserve place for that trip.
- Drivers can only write reviews about passengers they drove, Passenger can write review about the driver and the passengers they took a trip with.
- Users will have different reviews for their driver and passenger behaviours.
- When a user is deleted his/her vehicle will also be deleted.

## 5.0 E/R Diagram



## 6.0 Conclusion

This car sharing application is a web based system for people who need a ride from A to B and there is a willing driver who also be visiting A and B. Our system keeps track of all the data from drivers, passengers, trips, locations, time and cost and show the relevant information to relevant user when required.

In this report we explained what this system will be, how we are going to implement the system. How each part of the system is going to be important for the system's functionality and the importance of using an database implementation. Later we gave the functionalities for each type of user in specific cases, scenarios. We also included non-functional and pseudo requirements that are mostly relevant for us the developers. We also presented the limitations of relations between tables. Last of all we included the entity relation diagram of the system. The web link to the system is included.

## 7.0 Web Page

<https://github.com/Albocoder/CS353-CarSharingSystem>