



CS353 DATABASE SYSTEMS

PROJECT PROPOSAL REPORT

GROUP 13

İbrahim Taha Aksu
Onur Kulaksızoğlu
Berke Soysal
Gökçe Şakir Özyurt

Table of Contents

1. Introduction
2. Project Description
 - 2.1 Why we need Database Private taxi
 - 2.2 How we use database
3. Requirements
 - 3.1 Functional requirements
 - 3.2 Non-functional requirements
4. Limitations
5. E/R Diagram
6. Conclusion
7. Webpage

1.Introduction

Taxim is a private taxi portal both for customers and drivers. Taxim is a web application that connects passengers and drivers whenever needed. In our daily life, we are obliged to trust the taxi drivers that we never met before, in a locked car. What if, we had a chance to view the taxi driver's profile that contains detailed information about him and car, that has pre-passenger's review about him, and what if we had a program that provides us a fixed price between any two location, no matter the driver used which way. That would feel us more secure and comfortable in a taxi car, so that's why we built Taxim for you.

Taxim has a secure and robust database system that is useful for both sides, the passengers and drivers. For passengers, that makes their trip more trustable, they can display everything they would want to know about the driver: car model, customer rating etc. If drivers want to be more transparent, they can share their personal information (name, age,picture, educational background etc.) with customers to make them feel more comfortable. Drivers can see the customer's personal information and how many trips he had done before, that way he/she can trust the passenger. Our system will prioritize same language speakers when matching users and drivers.

2 Project Description

Taxim will be developed using the latest web technologies, supporting both mobile and desktop platforms.

In our project's client side; we will use the classical web technologies namely HTML,CSS and JavaScript. HTML for forming the skeleton of the contents, setting the structure and layout of them. CSS for describing how our contents in our application will look like, including colors, layouts, fonts as well as their precise locations padding and margin

values. JavaScript will be used to make Taxim dynamic and interactive. Also some necessary calculations, form validations, accessing or modifying DOM will be done using JavaScript without needing to access the server.

Over those mainstream front end technologies we will use additional modern web technologies in our client side. Twitter Bootstrap will be used to make our web application responsive, fast and mobile friendly. Bootstrap comes with embedded JavaScript and CSS libraries, we will also make use of those, to design more elegant and visually good looking web application. JQuery, a famous JavaScript library will be used to easily make animations, implementing Ajax and handling the events.

In server-side of the project, we will use C# .NET to implement our server system. C# is an elegant object oriented programming language that will enable us to write robust and elegant applications with ease. .NET is the framework a kind of virtual machine that the C# runs on. We chose C# .NET technology, over other web technologies since, this technology is widely used in industry thanks to its ease of use, robust structure and efficiency. Also C# .NET is around for many years, a great deal of resources and community is ready to help us out if we encounter any problems. On the other hand, Object Oriented structure of C# will be really useful since we would be able to directly map real life objects directly to our system.

We are planning to use again Microsoft's MSSQL Database Management System complementary to our C# .NET choice. There is no big difference between MYSQL and MSSQL they have similar robustness, stability, security features. Since we are planning to use Visual Studio for C# development, it will be a logical choice to stick with MSSQL for database choice. Porting a MSSQL database is fairly easy compared to other databases, additionally Visual Studio offers extra features for easily showing and modifying data for MSSQL databases that will save us a lot of time.

2.1 Why we need Database Private Taxi

We need a relational database to store the information of different entities such as user, driver and customer services. We will have a huge amount of data, maintaining it and modifying it, requires efficient, robust and safe storage system. Since that is the case, using a Database system will ease those routines. Also, concurrent accesses of all users of the program will be handled by our Database system automatically. So that, we will not have to implement explicit synchronization algorithms. Additionally, our data will not get affected from any internal or external modifications, since databases are not dependent on the core design of the application.

2.2 How we use database

We are going to use MSSQL to manage all data related to taxi. Necessary operations such as queries, data modifications, updates will be done accordingly and smoothly using Database system. Also we can make some calculations using the data contained by the database such as listing the drivers with the highest rating; most popular place to visit in a certain location and many others.

3 Requirements

3.1 Functional Requirements

3.1.1 User

- Users should be able to modify their personal data.
- Users should be able to register with a car and become taxi drivers.
- Users should be able to list the drivers nearby when they provide the starting position of the drive.
- Users should be able to cancel the trip before it begins.
- Users should be able to see driver's personal information and a communication information such as a phone number after the trip is approved by the driver.
- Users should be able to rate drivers and leave some feedback for them.

- Users should be able to see past trip information such as car information, start and stop location and time, date, trip rate and driver information.
- Users should be able to choose and reject among different drivers who accepted the trip request. Or they can choose it to be done automatically
- Users should be able to choose multiple destination points for a trip.
- Users should be able to choose some taxi features, such as people capacity or the quality of the car.
- Users can pay with either direct cash to the driver or credit card using Taxim.
- Users can contact the customer services about the trip.
- Users should be able to plan trips for later.

3.1.2 Driver

- Drivers should be able to modify information about their car, but the quality rating of their car will be done by the app managers (whether it's luxury or not).
- Drivers should be able to update their location or automatize it.
- Drivers should be able to see trip requests from passengers and approve or cancel these.
- Drivers should be able to see passengers personal information and a communication information such as phone number after approving the request.
- Drivers should be able to see passengers ratings if they have made a reservation request.
- Drivers should be able to rate the passengers after the driving.
- Since drivers are just regular users that have cars, it should be easy to switch between driver and passenger modes.
- Drivers can contact the customer services about the trip.

3.1.3 Customer Service

- Customer service should only be able to see information about the complained trips, its driver and passenger etc.
- Customer service should be able to see drivers car information.
- Customer service should be able to see users personal information except some private information such as credit card information.
- Customer service should be able to see detailed information about the trip, its duration, cost etc.
- Customer service should be able to see the trips in a less detailed way for a given time period or a driver. Like general ratings, durations etc. for the improvement of the app.

3.1.4 System Requirements

- System should notify the user/driver in case of an approval or cancellation of the trip.
- System should be able to verify user emails.
- System should be able to restrict unverified users actions by not letting it to request any trips even though it can list the nearby drivers.

3.2 Non-functional Requirements

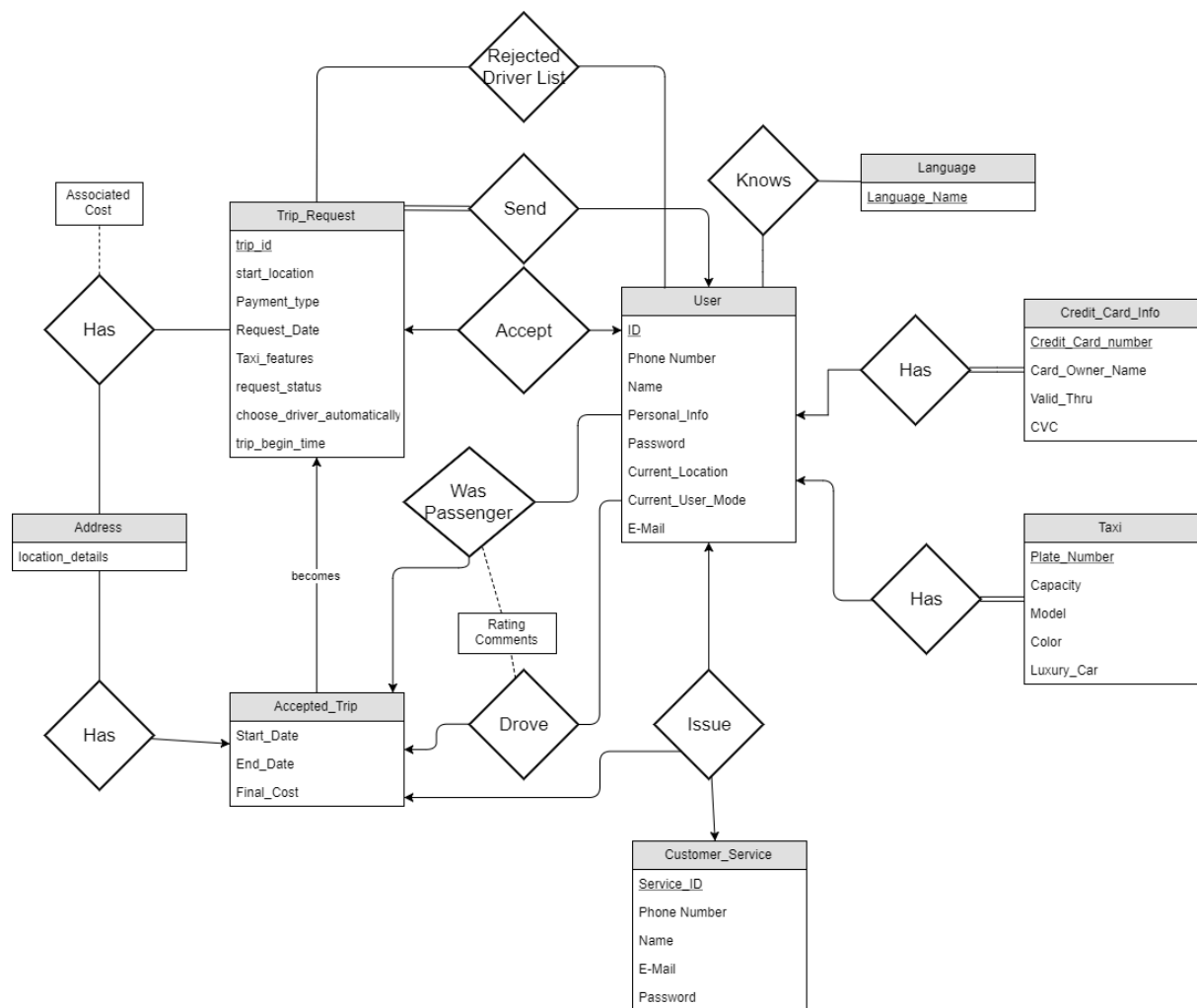
- **Accessibility:** Taxim would provide features for disabled person e.g. bigger buttons, voice control.
- **Response Time:** Taxim should respond fast to user inputs.
- **Auditability:** Taxim should track of the list of past events: Trips, failed trips, chat logs etc.
- **Availability:** Taxim should be available 7/24 to customers.
- **Sufficient network bandwidth:** Network bandwidth should be sufficient to handle large amount of users and data online.

- **Recoverability:** Database of taxim should be backed up daily in case of an unexpected failure, so that data will be recovered.
- **Scalability:** The Taxim database should handle large amount of data and query requests.
- **Capacity:** Server capacity (CPU, Network, Memory, Disk) should be optimized and upgraded when the capacity exceeded.
- **Maintainability:** The source code of the system should be readable so that, it can be modified easily.
- **Securability:** Personal and billing information should not be reached from outside of the application by malicious people
- **Usability:** Taxim should be easily used by any person with any educational background.
- **Reliability:** Taxim should always give the right information to the user.
- **Interoperability:** The SQL system and Js front-end should communicate and work cooperatively.

4 Limitations

- Guest can search for nearby drivers but they can not participate in a trip unless they have logged in.
- The registered car has to be a valid one. This will be checked within the traffic database.
- User has to specify the starting and ending point of the trip before the trip begins.
- A trip after it begun can only be cancelled if both driver and user cancels the trip.
- Passengers can not rate drivers they have not participated in a trip with.
- Users which could also be drivers should have different ratings for their account as a passenger and as a driver.
- When a driver is deleted from the database his/her car gets also deleted.

5 E/R Diagram



6 Conclusion

Taxim is a web based application for anyone willing to travel and work as a taxi driver. The system keeps track of all the data related to passengers, drivers, locations, time and cost of the trips and can filter the data for a relevant user when needed.

This report aims to explain what the system will be about, its target audience and how is it going to be implemented. What environments are going to be used in each and every part of the system and why we are using specifically those are also explained throughout the report. The functionalities for each type of user is also given as a separate part in the report. Non-functional requirements that would be vital for the application are as

well listed. The limitations between tables are stated in another part in the report. Last but not least the entity relation diagram of the database system is included in the report.

7 Web Page

<https://github.com/cuthalionn/Taxim>