TALLINN UNIVERSITY OF TECHNOLOGY

School of Information Technologies

Your Department

Tavo Annus   186060IAIB

WEB SYSTEM FOR SELF BUILD CAR DASHBOARD

Distributed Systems project

Supervisor

Andres Käver

BsC

Tallinn 2021

Author's declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Tavo Annus

                                                    ....
                                                        (signature)

Date: Month Day, Year

List of abbreviations and terms

API	Application Programming Interface
ERD	Entity Relations Diagram
CAN	Controller Area Network

Table of Contents

1. Introduction

There are many free Apps for both PC and Android to diagnose your car via OBD2 port. These Apps give access to data such as fault codes, current rpm, oil temperature and many more. Since older cars don't have fancy touch screens it would make sense to have some app running and reading this data to show out on screen.

To have more general statistics for a driver it would make sense to sync them up to some database so that you could access the data also outside your car. This would also give the possibility to see stats containing many cars and compare different car marks / models stats to one another.

1.1 Project scope

The scope of this project is to build web application and database system capable of storing and querying data from multiple users at once. Building an app capable of reading desired data from a car is not in the scope of this project and will be done at a later time. Although there will be no actual client to upload data, proper API endpoints shall still be made. During the course, possibly android app made last year will be used to get mock data.
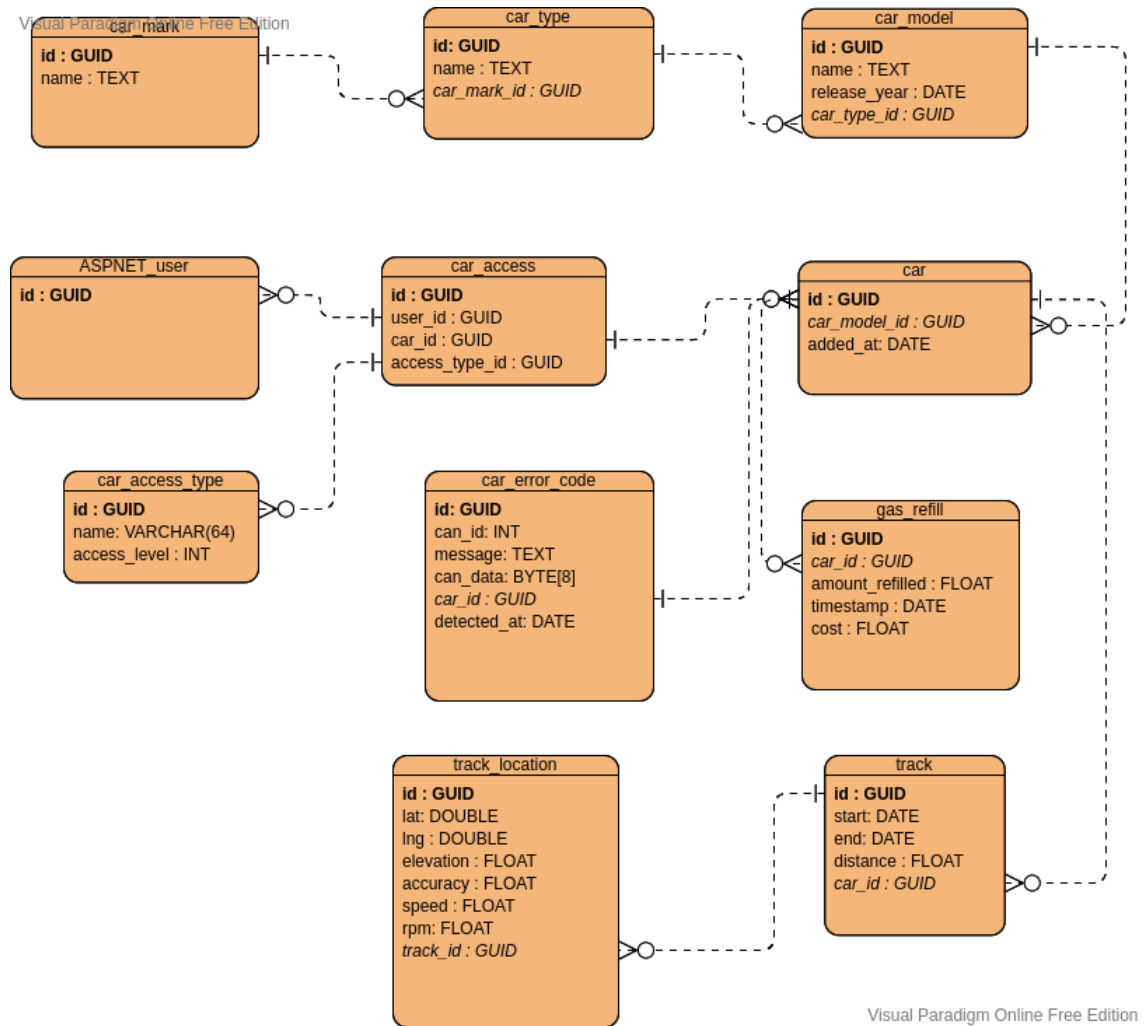
Must haves:

- ERD schema.
- Controllers for desired CRUD operations with entities.
- Views for Refills, tracks and statistics.
- Detailed view for a track.
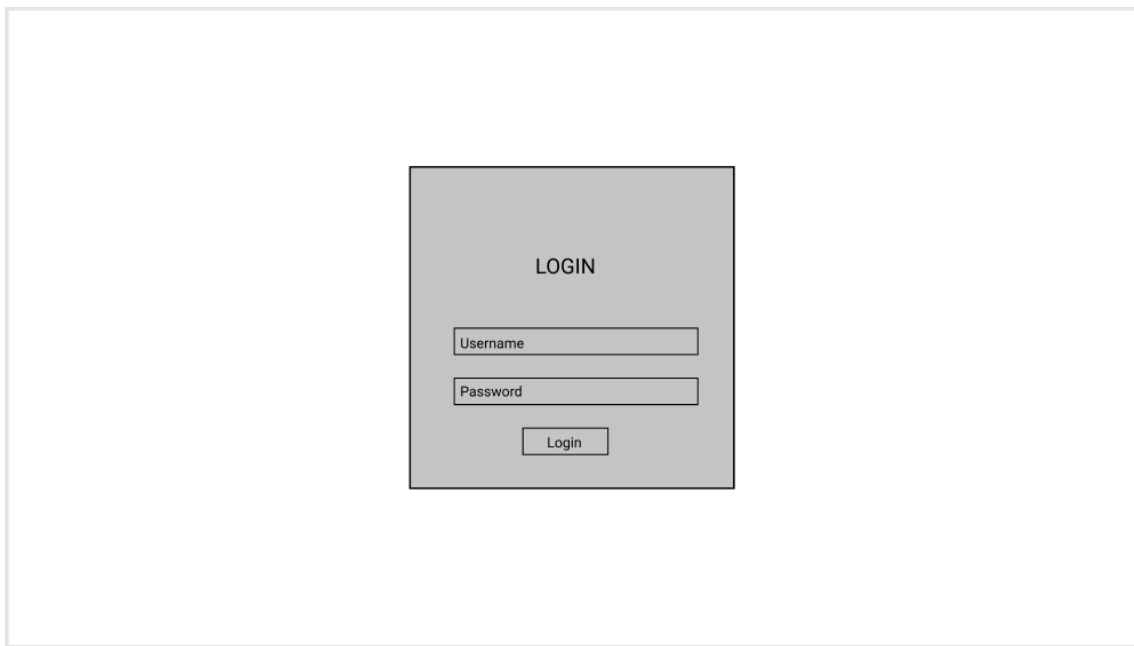- Server must be scalable to allow numerous clients.

Nice to haves:

- Comparison mode for statistics (car vs type average etc.).
- Live data for tracks.

2. ERD schema

car_mark
id : GUID
name : TEXT

car_type
id: GUID
name : TEXT
car_mark_id : GUID

car_model
id : GUID
name : TEXT
release_year : DATE
car_type_id : GUID

ASPNET_user
id : GUID

car_access
id : GUID
user_id : GUID
car_id : GUID
access_type_id : GUID

car
id : GUID
car_model_id : GUID
added_at: DATE

car_access_type
id : GUID
name : VARCHAR(64)
access_level : INT

car_error_code
id: GUID
can_id: INT
message: TEXT
can_data: BYTE[8]
car_id : GUID
detected_at: DATE

gas_refill
id : GUID
car_id : GUID
amount_refilled : FLOAT
timestamp : DATE
cost : FLOAT

track_location
id : GUID
lat: DOUBLE
lng : DOUBLE
elevation : FLOAT
accuracy : FLOAT
speed : FLOAT
rpm: FLOAT
track_id : GUID

track
id : GUID
start: DATE
end: DATE
distance : FLOAT
car_id : GUID

3. Client flow screens

3.1 Login page

User can log in from this page to access his/her vehicles data.

3.2 Refills page

This page shows all gas refills in last month (or possibly custom period). Data is showed in table and total amounts are in a header.

3.3 Tracks page

This page shows all tracks in last month. Total statistics are showed in header and rest of the data is in a table. Clicking in table row will take you to track details page where you can see more detailed statistics and data about the track.

3.4 Track detail page

This page visualizes single track data. There will be a map that shows path traveled (possibly color coded for speed). Underneath will be graphics that show rpm, speed and maybe some other useful data

3.5 Statistics page

This page will show general statistics for single user. Data shown here highly depends on what kind of graphs can be made without having to generate huge amounts of mock data. Possibly heatmap for drive speed / fuel consumption. Simple area / line graphs for average speed, average rpm and gas consumption.

4. Summary

Appendices