Sprint Review — 4

Sprint 4 took place between Sprint 3 and the submission of the project.

Sprint goals

Complete frontend-backend communication

- Deployment of Spring-boot backend on linux server
- Deployment of React built Front end
- Sending request to and from React front-end to and from Spring backend
- Be able to handle HTTP request on Spring backend

Search functionality

- From user perspective being able to search for certain cars
- Be able to search and alter entities in the database by HTTP request methods

Other:

- We decided to use email/password to log-in to the website instead of username/password
- The two mandatory users to log-in:
    - Admin user: Email: dave@mail.com password: Dangerous2024
    - Normal user: Email: chuck@mail.com password: Nunchuks2024

Dto (Data transfer object):

- For all our entities we created simple duplicate classes so that we could separate the entities from the data we transfer from the database.
- Reason: We implemented this when because when getting certain requests from the database, because of the relationship between the entities caused some getrequests to get a nested loop of the same request which caused stackoverflowerrror.
- DTO implementation helped us to avoid the nested loops when getting .Json from the controller classes by separating the data we sent from the data in the database.

Who did what	
Jeremy	Was in charge of the backend: - Deployment of backend on linux server - Database managment - Controller classes to handle HTTP requests - Service classes to handle logic for controllers - Repository classes for saving entities in the database
Oscar	Was in charge of the frontend: - React implementation on frontend - Sending HTTP request to backend server - Deployment of front end - Making the website resemble the wireframe

Review

We accomplished most of our goals by deploying the spring-boot backend on the linux server and having a functional React front end that can communicate together.
We were able to handle searching through the database and getting the data from the database.

There are some safety concerns we were unable to implement when it came to backend and password encryption. The backend has no https secure connection as we struggled to give access of the Openssl key to nginx. So the the backend is currently operating on Http connection. We also intended to properly encrypt password when they are sent back and forth from the the back end and the front end.