TECHNICHAL REPORT

CPS 630

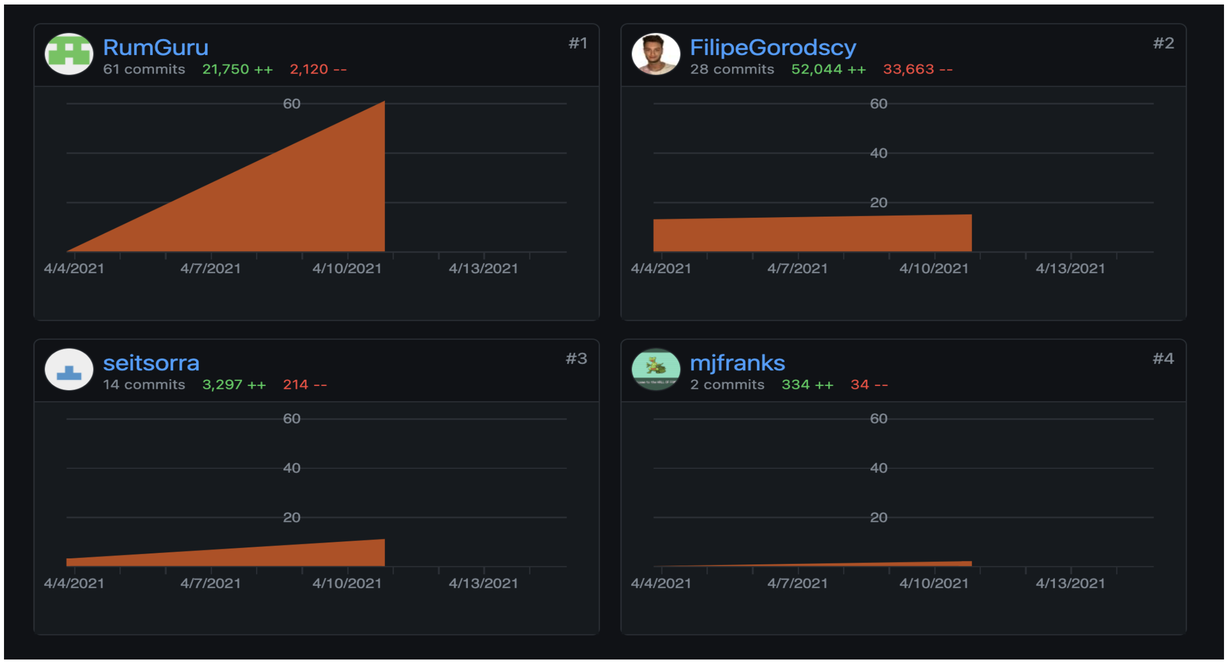
PLAN FOR SMART SERVICES (IT 3 & 4)

GROUP 11

RYERSON, 2021

Name	Work Done
Seit Sorra	Backend
Matthew Franks	Backend
Filipe Gorodscy	Frontend
Rahul Kothwal	Frontend

1) GITHUB CHART && BRIEF DESCRIPTION:



Top-Left - Rahul Kothwal: Responsible for the design and front-end implementation of the Delivery Service as well as the Grocery Delivery Service. Used Drag and Drop techniques in both pages. (25%)

Top-Right - Filipe Gorodscy: Responsible for the design and front-end implementation of all other pages, including Home, Ride Service, Ride Green Service, Contact Us, About Us, Reviews, Login and Register. Also was responsible for implementing the Database Maintain input forms where the admin could Add, Delete, Edit and Search the database using API requests. Demoed the web application to the professor. (45%)

Bottom-Left - Seit Sorra: Responsible for the complete design of the backend architecture, created PHP classes and tables using SQL commands. The PHP scripts were developed in a way that it could be called by the browser exactly like an API request. Converted the API requests to accept JSON data instead of form data. (25%)

Bottom-Right - Matthew Franks: Created one script to create and one script to read data from the reviews table in the currently designed and implemented database. (5%)

2) PROJECT OBJECTIVE && SECURITY CONCERNS:

To create a system that offers users the ability to choose their ride, deliver flowers, compare our services and deliver groceries.

In this project, we have made use of the following technologies:

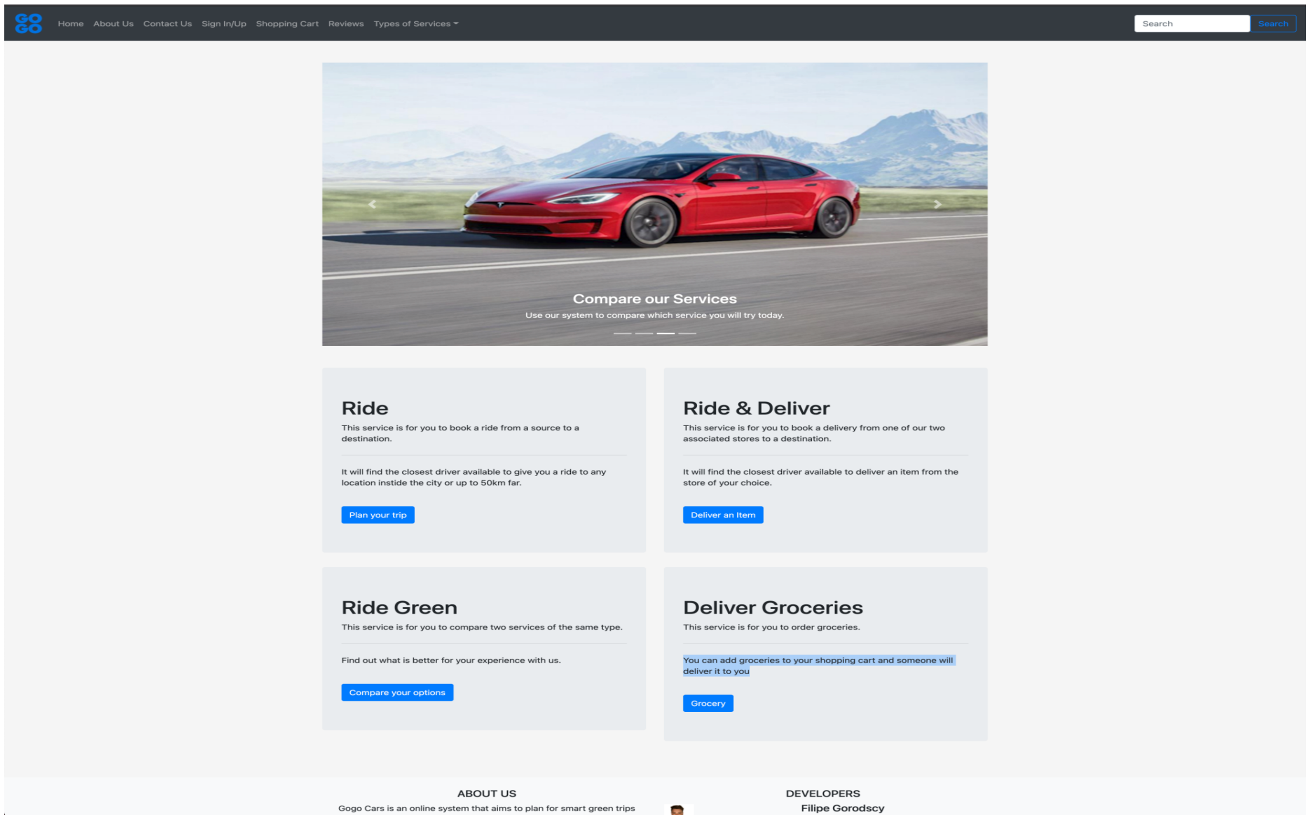
- PHP (as a server-side scripting language)
- MySql (as a database engine)
- React (in the front end) - Using Hooks and Functional Components
- React-Bootstrap CSS framework
- Google Map APIs
- Postman (as a testing tool for backend API's)

To provide security for our users, we use hashing mechanisms at security points. For example, the user password is hashed before getting saved to the database.

We also provided security with regards to the database endpoints, where the server blocks cross region calls.

Concerning security, only logged people would have access to our services and only the admin with their own username and password have access to the Database Maintain mode.

3) PROJECT DESIGN && ARCHITECTURE:



Compare our Services

Use our system to compare which service you will try today.

Ride

This service is for you to book a ride from a source to a destination.

It will find the closest driver available to give you a ride to any location inside the city or up to 50km far.

Ride & Deliver

This service is for you to book a delivery from one of our two associated stores to a destination.

It will find the closest driver available to deliver an item from the store of your choice.

Ride Green

This service is for you to compare two services of the same type.

Find out what is better for your experience with us.

Deliver Groceries

This service is for you to order groceries.

You can add groceries to your shopping cart and someone will deliver it to you

ABOUT US

Gogo Cars is an online system that aims to plan for smart green trips

DEVELOPERS

Filipe Gorodscy

The user is requested to login in order to use this application.

The main page of the application offers users to search for an available car in the system.

The menu offers the following options to the user:

- Home – a button that takes the users to the homepage
- Types of Services – drop-down menu with the types of services
  - Ride – a service where user can request a ride
  - Ride and Deliver – a service where the user can request a delivery
  - Compare our Services - a service where the user can compare the other services
  - Grocery Shop - A service where the user can order groceries.
- About Us – a button that takes to the about us page
- Contact Us – a button that takes to contact us page
- Reviews – a button that takes to the reviews page
- Database Maintain – a drop-down menu with the following options
  - Add → Used to Insert data into the database
  - Delete → Used to delete data into the database

        - Edit → Used to update the database
        - Search → Used to query the database
    - Shopping Cart – a button that takes the user to the shopping cart
    - Sign In/Up - A button that takes the user to the Login page or the Register page if needed.

BACKEND

This application offers API endpoints for communication with the backend services (found in the backend folder). All the backend APIs return JSON format for the responses.

The backend folder structure is as follows:

tables folder which contains PHP files implementing PHP classes for each table in the database

Then for each table in the database, there is a folder that offers its API endpoints such as, CREATE, READ, UPDATE and/or DELETE.

For example *cars, flowers, delivery, trips* are folders containing the API endpoints for the respective tables.

The database handling is done through the PHP database class at backend*/config/database.php*

COMPATIBILITY

All of the well knwon browsers are supported in our web application. We developed our project respecting the differences between Chrome, Firefox and Safari. Unfortunately, we could not test our website using IE.

SERVICES

In this project the user can choose between four services:

- **Ride**: This service is for you to book a ride from a source to a destination.
- **Ride and Delivery**:This service is for you to book a delivery from one of our two associated stores to a destination.
- **Ride Green**:This service is for you to compare two services of the same type.

- **Grocery Delivery**:You can add groceries to your shopping cart and we will find the best match to deliver it to you.

We designed and implemented the fourth service based on our second service (Ride&Deliver). We used drag and drop techniques to make it easier and fun for the customer. The user can select multiple items of different types and also multiple items of the same type.

SINGLE-PAGE APPLICATION ARCHITECTURE

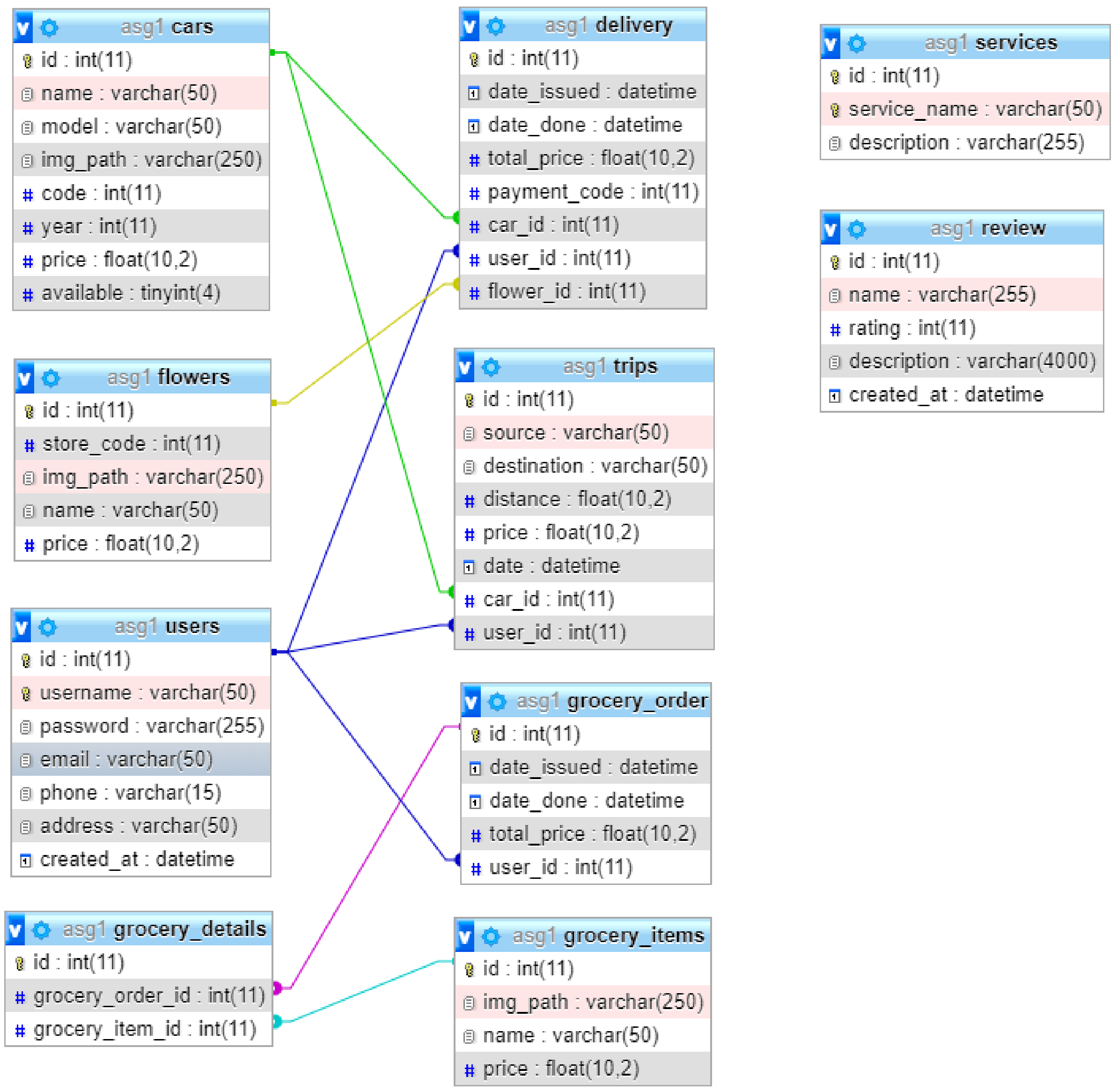
For the **SPA Architecture** we used **React Router**. Our web application has several components and would have around 10 pages if we had chosen to follow the multipple page application approach. Using React, we could achieve:

- A Single Page Application, with different views
- Specific rendering to the screen (only modified components will rerender)
- Less latency for the users (no need to reload the CSS files when changing views)

As shown below in the diagrams for both the backend and frontend, our architecture follows the MVC pattern by having the model layer implemented using PHP and SQL and, for the frontend,we make use of React and React Router for the View and Controller.

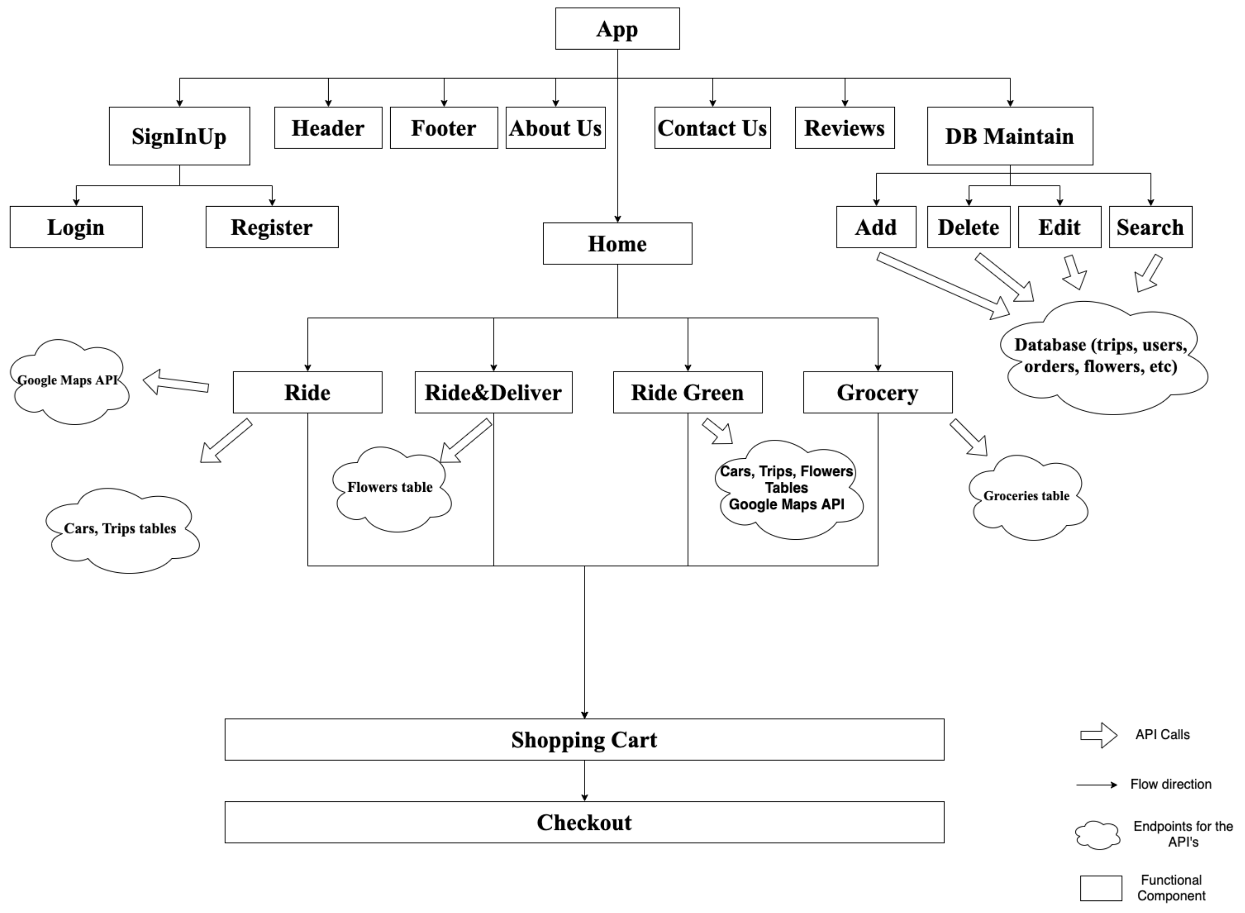
4) DATABASE STRUCTURE: MODEL LAYER.

An image representing the relation of tables of the *asg1 database* used for this iteration:



5) FRONT-END STRUCTURE: VIEW-CONTROLLER LAYER.

The image below shows the front-end structure of our web application, where we decided to use React as the chosen framework.


App
SignInUp
Header
Footer
About Us
Contact Us
Reviews
DB Maintain
Login
Register
Home
Add
Delete
Edit
Search
Database (trips, users, orders, flowers, etc)
Google Maps API
Ride
Ride&Deliver
Ride Green
Grocery
Flowers table
Cars, Trips, Flowers Tables Google Maps API
Groceries table
Cars, Trips tables
Shopping Cart
Checkout
API Calls
Flow direction
Endpoints for the API's
Functional Component