

# TECHNICAL REPORT

CPS 630

PLAN FOR SMART SERVICES (IT 3 & 4)

GROUP 11

RYERSON, 2021

Name	Work Done
Seit Sorra	Backend
Matthew Franks	Backend
<add your names>	

## **2) PROJECT OBJECTIVE:**

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)

## **3) PROJECT DESIGN:**

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
  - o Ride – a service where user can request a ride
  - o Ride and Deliver – a service where the user can request a delivery
  - o Compare our Services - a service where the user can compare the other services
  - o 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
  - o Add → Used to Insert data into the database
  - o Delete → Used to delete data into the database
  - o Edit → Used to update the database
  - o 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.

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.

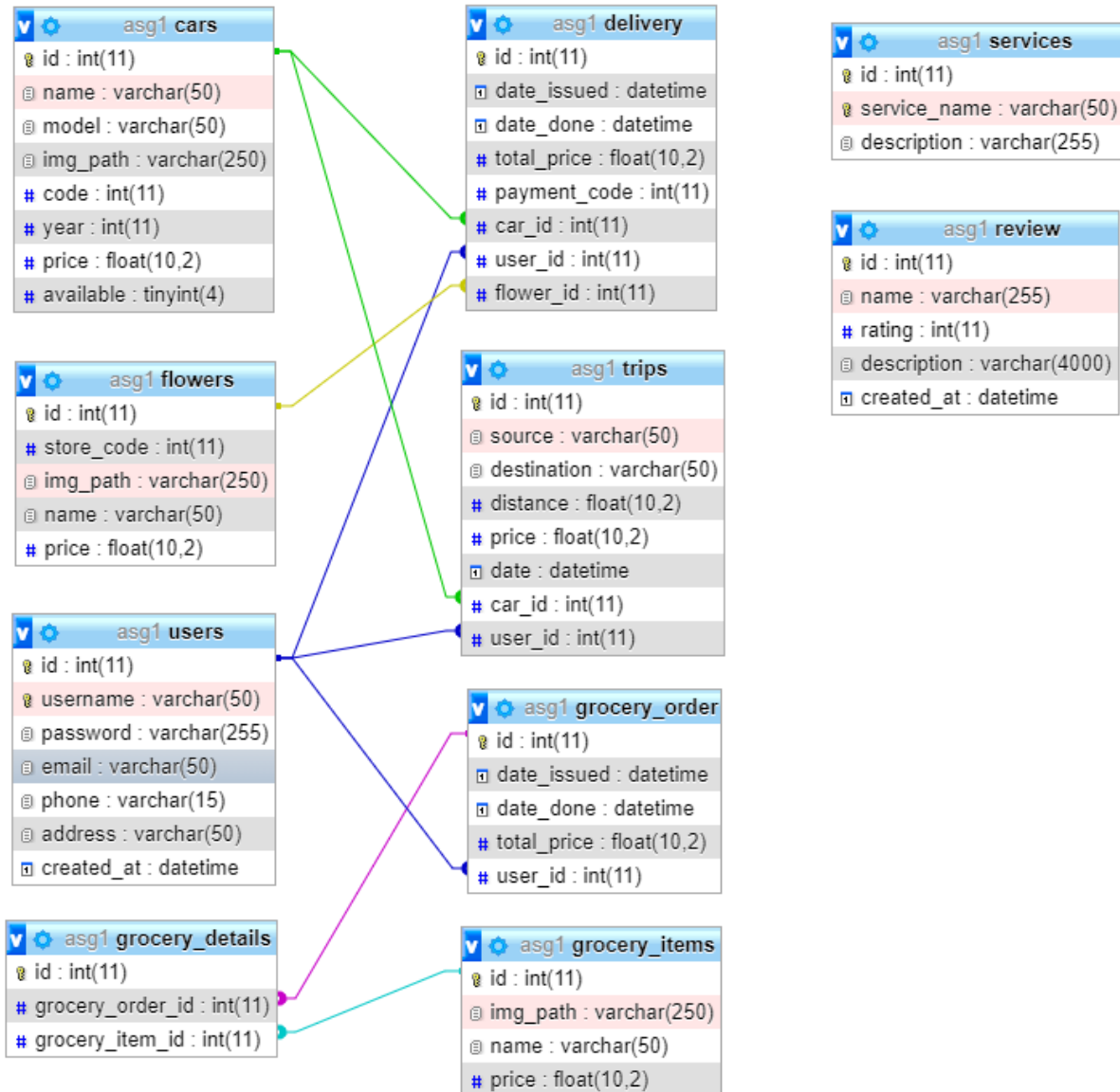
For example *cars*, *flowers*, *delivery*, *trips* are folders containing the API endpoints for the respective tables.

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

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.

#### 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.