

Team 8 Project Documentation  
Professor Aravamudhan  
ECE:5800

Team 8 Members  
Ryan Price  
Ben Weinberg  
Nick Fallon

1. Data Dictionary
2. Estimation
3. Plan

- a. Summary –

We are required to make a web based Ride Share program that will allow users to select rides, drivers to select routes they want to drive, and admins to add routes, cars, and manage drivers/riders within a car.

A ride is defined by the start time, start location, drop off location, car and driver. An admin will select the start time, start and drop off location, and car, then it will be made available for a driver to select to drive it. After a ride has a driver then a rider will be able to claim a spot in it. A ride will be able to be selected by different riders until it has no more spots available in the car. Only a registered rider will be able to sign up for the rides, and only registered drivers will be allowed to sign up to drive a route. To sign up a user will need to input their name, email, a username, a password, and then select an account type, either rider or driver.

- b. Definition of Done – The basic definition of completion for this project is to have a working software with three levels of user. The three levels will include a rider, driver, and admin. For a rider they will be able to see the upcoming rides that will they will be able to take and then claim a spot in the ride, as well as see rides they are currently signed up for. A driver will be able to see the rides they are currently signed up to drive, and be able to see other rides they can sign up to drive. The admins will be able to make new rides available, add new admin accounts, and add new cars to the van pool.

- c. Schedule and Deliverables –

- i. End of February
- ii. End of March
- iii. End of April

- d. Current Technology and Resources –

- i. Human Resources –

For this project we have three team members, Ben Weinberg, Nick Fallon, and Ryan Price. We are assisted in the project by Nabeel Ahmad Khan and Raman Aravamudhan who will assist us with executive decisions for the project.

- ii. Roles and Responsibilities -

Each team member will have a specific role within the group that they are required to work on. Other group members should must be able to help with different roles as needed. Each members role will work as follows:

Ryan Price – Will be responsible for the MySQL database. Including getting the server created and hosted. Defining the tables that will be used and accessed. Defining the primary and foreign keys attributes for a table and the other necessary attributes. The tables should be at least within the Third Normal Form (3NF) to be considered done.

Nick Fallon – Will be responsible for making the basic HTML pages/ screens. Using Handlebars the pages should be able to get content to from the server and then display it on the HTML webpage. Each screen will have a specific function that the user will interact with.

Ben Weinberg – Will be responsible for making the Express Server. The server will pass information from the database to the screen and take users input and then pass it to the database. The server will be made using JavaScript with ExpressJS and NodeJS frameworks.

### iii. Technology Stack -

#### Frontend- HTML/ Handlebars

For the frontend of the web app will be made using HTML as the base, and Handlebars to add dynamic content from our database. We choose to use Handlebars because it is a logicless templating engine that can be used to add content to HTML webpages. More information about Handlebars can be found at their website [handlebarsjs.com](http://handlebarsjs.com).

#### Backend/ Server – ExpressJS

The server will be made with ExpressJS, a JavaScript server framework. Express is a NodeJS based framework that is used as the controller for the web app. As a Node based framework there are many modules that can be added, including a MySQL module, a Handlebars module, and an MD5 module. More information about ExpressJS can be found at their website [expressjs.com](http://expressjs.com).

#### Database – MySQL

Using a MySQL database allows us to store user, ride, and other necessary information for the web app. We choose MySQL because it has easy syntax to learn for the queries and is able to interact easily with ExpressJS. MySQL also provides a workbench that can be used to view, create, and

edit table easily. More information about MySQL can be found on their web site [mysql.com](http://mysql.com)

e. Technology Goals –

The goals for this our technology is for everyone to have a working knowledge of the stack and be able to help each group member with different parts of the project.

f. Appendices –

Definition of Done- Describes the minimum requirements for a completed software.

MySQL Hosting- Is done using Google Cloud Platform with a MySQLv5.7 Database.

Third Normal Form(3NF)- A relation that is in first and second normal form in which no non-primary key attribute is transitively dependent on the primary key.

4. SRS (Software Requirement Specification) –

- a. Store encrypted passwords
  - Passwords can't be stored in plain text in the database and should be encrypted using some kind of known encryption scheme.
- b. Authenticate users with username and password
  - For a user to log in they will need to provide a valid username and password.

5. URS (User Requirement Specification) -

6. Testing – For testing automated methods will be used to verify various test cases. The test cases are as follows:

a. Logging in

i. Valid user information

This test will be used to ensure that a user is correctly logged in after they enter a valid username and password.

Testing Procedures:

- Load homepage
- Click link to login page
- Load login page

- Fill in a valid username
- Fill in a valid password
- Click log in button on page

To consider the test passed the user will need to be redirected back to the home page.

ii. Invalid user information

This test will be used to ensure that a user is not logged in after they enter invalid user information

There will be four test cases for this test including:

- a. Invalid username with a valid password
- b. Invalid password with a valid username
- c. Invalid username and password
- d. No user input

Invalid username and password testing procedures:

- Load homepage
- Click link to login page
- Load login page
- Input invalid username
- Input invalid password
- Click log in button

To consider the test passed the user should stay on the same login page to refill out login information.

Invalid username and valid password testing procedures:

- Load homepage
- Click link to login page
- Load login page
- Input invalid username
- Input valid password
- Click log in button

To consider the test passed the user should stay on the same login page to refill out login information.

Invalid password and valid username testing procedures:

- Load homepage
- Click link to login page
- Load login page
- Input valid username

- Input invalid password
- Click log in button

To consider the test passed the user should stay on the same login page to refill out login information.

No user input testing procedures:

- Load homepage
- Click link to login page
- Load login page
- Click log in button

To consider the test passed the user should stay on the same login page to refill out login information.

b. Logging out

This test will be used to ensure that after a user's sessions has ended information about the session is be erased.

c. Signing up for a ride

This test will be used to ensure that a rides information is updated when a user signs up for the ride.

There will be two test cases for this test including:

- a. Signing up for a ride with extra spaces left
- b. Signing up for the last available spot on a ride

Signing up for a ride with extra spaces

- Load homepage
- Click link to login page
- Input valid username
- Input valid password
- Load login page
- Click log in button
- Be redirected to homepage
- Find available ride
- Claim spot in ride
- Find ride in upcoming rides tab
- Repeat steps with different rider account

To consider this test passed the ride's information should be updated in the database and the first user should be able to see the ride in upcoming rides. The second account should be able to sign up for the ride.

Signing up for last available spot

- Load homepage
- Click link to login page
- Input valid username
- Input valid password
- Load login page
- Click log in button
- Be redirected to homepage
- Find available ride
- Claim spot in ride
- Find ride in upcoming rides tab
- Repeat steps with different rider account

To consider this test passed the ride's information should be updated in the database and the first users should be able to see the ride in upcoming rides. The second account should not be able to sign up for the ride.

d. Signing up to drive a route

This test will be used to ensure sure that a driver can select a route they would like to drive and then update the route information.

Testing Procedures:

- Load homepage
- Click link to login page
- Input valid username
- Input valid password
- Load login page
- Click log in button
- Be redirected to homepage
- Find available route to drive
- Claim driving spot
- Find drives in upcoming drives tab
- Repeat steps with different driver account

To consider this test passed the routes information should be updated in the database. The second account shouldn't be able to sign up to drive the route.

e. Adding a route

This test will be used to ensure an admin is able to add a new route and the information is stored correctly in the database.

There will be two test cases for this test including:

- a. Adding route with past date
- b. Adding a valid route

Adding a route with a past date:

Past date is a day that has already happened during the year and can't have a ride scheduled to it.

- Load homepage
- Click link to login page
- Input valid username
- Input valid password
- Load login page
- Click log in button
- Be redirected to homepage
- Go to route adding page
- Input ride information including the invalid date
- Click publish ride button

To consider this test passed the ride should not be added to the database and the user should be told to use a different day.

Adding a valid route:

- Load homepage
- Click link to login page
- Input valid username
- Input valid password
- Load login page
- Click log in button
- Be redirected to homepage
- Go to route adding page
- Input valid ride information
- Click publish ride button

To consider this test passed the ride should be added to the database and the user should see it on published rides.



f. Adding a car

This test will be used to ensure that an admin can add a new car and update the available car database