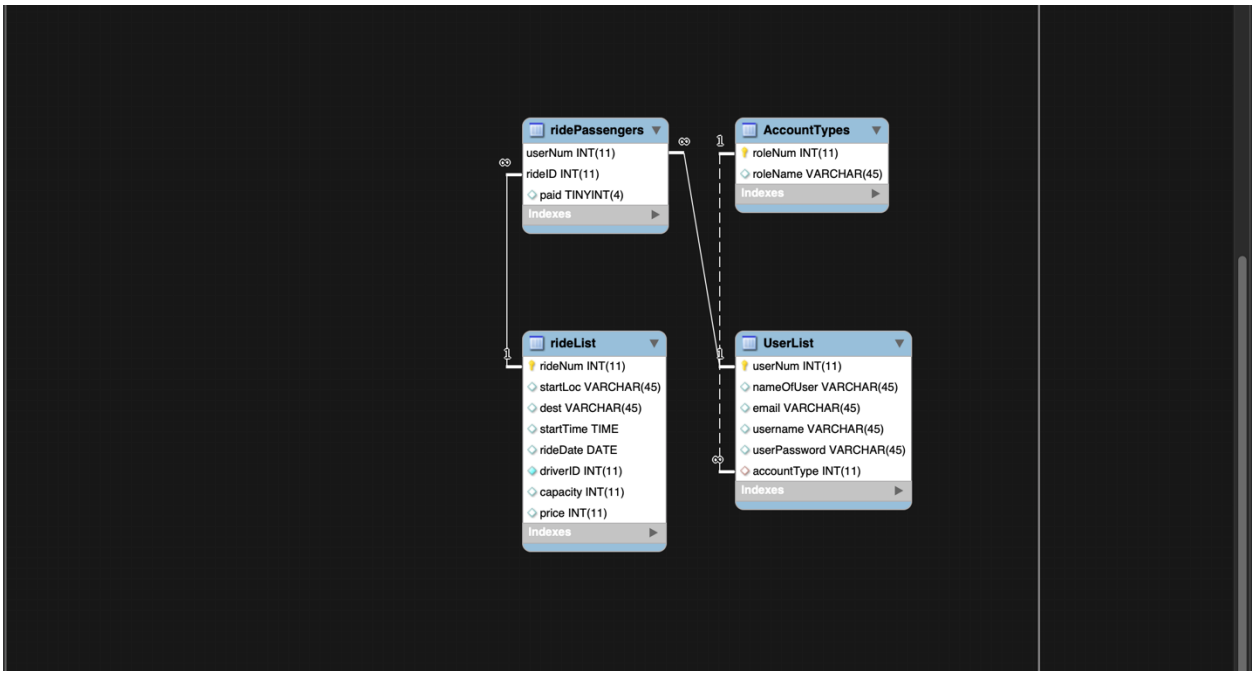
Team 8 Project Documentation
Professor Aravamudhan
ECE:5800

Team 8 Members
Ryan Price
Ben Weinberg
Nick Fallon

1. Data Dictionary



Included tables:
ridePassengers: Used to hold a rider's user number and a ride number for a ride they have signed up for. It also keeps track of if the rider has paid for the ride using a true or false in the paid field.

AccountTypes: Holds the list of possible account types that someone can have it holds a number and the name of the account type.

UserList: Holds the information about the user like name, username and password. It is a shared table for all users of different account types.

rideList: Holds information about a ride that has been created.

Other possible tables to include:
contactInfo: This table would be used to hold different kinds of contact info for a user. It would then be able to have a list of different ways to contact a user.

securityQuestions: This table would hold security questions that the user provided an answer for. This would then be used to update/ change a password.

2. Estimation

Function Name	Input Type	Fields	Files	Complexity	Function Point
Log in	Input	username, password	UserList	Simple	1
Create account	Input	Name, email, username, password, comfirmed password, account type	UserList	Average	2
Add ride/route	Input	Start Ride, Destination, Start time, Ride date, car	NewRoutes	Average	2
Add new admin	Input	name, email, username, password, comfirmed password, account type	UserList	Average	2
Search for a ride	Query	Date, Start Location, Start Time, Destination	RideList	Average	2
Driver select ride	Input/output	Date, Start Location, Start Time, Destination	NewRoutes, DirverRoutes	Average	2
Add driver to ride	Input/output	Ride Number	RideList	Simple	1
				Raw Function Point	**12**

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
   The first set of screens should be created by the February due date. The first set of screens will include the homepage, login page, and new users sign up page. The first set of screens will be used as a mock up for what the rest of the project will look like.

   ii. End of March
   The end of March due date will have added functionality to the first set of screens. A user should be able to log in when they input a valid username and password. The database should be updated when a user creates a new account. The homepage should show some dynamic content from the database when a user gets to it. New screens will be added that show a list of available rides to the user, who can then accept the rides. A screen to allow drivers to select routes to drive. A screen that allows an admin to create a ride.

   iii. End of April
   The end of April due date will have a completed web app. The app will meet all the requirement defined in the definition of done, see above. Should time permit extra features will be added to the final product.

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 duration of 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 be 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.

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 exresspjs.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

e. Technology Goals –

The goals for this our technology is to have an easy to use interface that the user won't have to spend time learning to before they can sign up for a ride. We picked our stack based on its ease of use for us as the developers. The stack was easy to pick up and didn't require us to learn a whole new system because some members of our team had prior experience with the tech.

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 are stored in the database using MD5 hash

   b. Authenticate users with username and password
      -For a user to log in they will need to provide a valid username and password

   c. Allow users to sign up for a ride
      -Users can view and select rides that have a driver assigned to them

   d. Allow drivers to create their own rides
      -Drivers can create their own ride that is automatically assigned to them

   e. Admin can create rides for drivers to sign up for
      -Admin can add rides without a driver that a driver can sign up to drive

   f. Admin can add new admin accounts
      -Add a new admin when one is hired

   g. Riders can view the past rides they have been on
      -See a list of the past rides that a user has been on

   h. Drivers can view the rides they are signed up for and the riders in the car
      -See the current list of people in a ride that a driver is assigned to

   i. Drivers can delete themselves from a ride
      -Should a driver not want to do a ride they can select to have it removed

   j. Riders can remove themselves from a ride
      -Users are able to remove themselves from a ride they don’t need

5. URS (User Requirement Specification) –

   We will make a web app that allows a user to sign up for one of two different account types, rider or driver. The riders will be able to add themselves to a ride should the ride have a driver. The riders will be able to see a list of all the rides that they have been on. The drivers will be able to see the list of currently made rides that don't have a driver they can then add themselves as the driver for this ride. The drivers can also create their own ride that they will be assigned as the driver to automatically. The driver will be able to see the list of rides they are signed up to drive as well as the people who are signed up for the ride. The last account type will be an admin. The admin will be able to create a ride that doesn't have a driver for the drivers to sign up for. The admin will be able to view all past and future rides as well as the people who are signed up for them. The admin will be able to add other admin who will have the same privileges as them.

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

This test case will have two test including:

a. Adding a valid car
b. Adding an invalid car

Adding a valid car Testing Procedure:

- Load homepage
- Click link to login page
- Load login page
- Input valid admin username
- Input valid admin password
- Click log in button
- Be redirected to homepage
- Go to car adding page

- Input valid ride information
- Click publish ride button
  To consider this test passed the new car will be added to the database.

Adding an invalid car Testing Procedure:

- Load homepage
- Click link to login page
- Load login page
- Input valid admin username
- Input valid admin password
- Click log in button
- Be redirected to homepage
- Go to car adding page
- Input invalid ride information
- Click publish ride button
  To consider this test passed an error will show up on the screen explaining the type of error