Joe and Dough

(team project for class)

Frameworks: PHP with my_sqli driver to connect with a mySQL database. used Twitter's Bootstrap on the front end of the project and FontAwesome to implement icons into project.

Overview of the project

project Joe and Dough is a website that serves as a coffee delivery website. In a local setting, customers can connect with deliverers who are available, and place orders to deliverers for coffee orders that they would want. We were able to create user types for deliverers and regular customers, and create an easy to use interface to set a deliverer status (saying if a deliverer is available to take orders) and an interface for customers to place an order with deliverers who would be available. We were happy that we were able to create a program that could be used on a larger scale like this, and we were happy that we were able to learn a bit more about web development in the process. With more refinement, Joe and Dough has the potential to compete with other applications on the application delivery market.

2. System Modeling and Design

2.1. Successful Major Story Cards

Note that for this portion of the project, all team members assigned to their own cards were responsible for adding their own checklist to each story card.

First Iteration:

Story 1: Create Front end and integrate, tasks: create front end, connect and display with php, connect with database

Story 2: Create Backend, tasks: configure XAMPP, create tables for backend through mySQL, be able to view data on a web page through using PHP

Story 3: As a user, I can sign up, tasks: I can sign up with email and password, I can choose to be a deliverer or a customer when signing up, I can see what my user id is when I log in

First Release:

Story 4: As a user I should be able to see a home page before logging in, tasks: create tab for signing in, create tab for making a new user, create home page

Story 5: As a user I should be able to reset my password, tasks: create page for changing password, create query for changing password, validate password change.

Story 6: As a user I should see a dashboard after I log in, tasks: create dashboard page, use get request to access ids in dashboard page, link to dashboard page from other pages in the site

Second Iteration and Release

Story 17: As a user I should be able to access Joe and Dough from a server, tasks: get server permission, get server access, get server access for all members of team labeled Story 17 since it was renamed a duplicate of a pre-existing story

Story 7: As a deliverer I can put my pickup on the database: create deliveries table in database, write query to insert into it, make php page to take necessary input then insert when submitted.

Story 8: As a user, I should be redirected after submitting delivery: create success page, add redirect functionality

Story 9: As a deliverer I want to give status as deliverer: add time frame aspects to the deliveries, let deliverers choose how long they will deliver for

Third (and Final) Iteration and Release

Story 10: As an orderee I want to be able to place an open request (without open deliverer), tasks: create requests table, create requests query and submission page, test and validate queries of orders

Story 11: As an orderer I should be able to view open deliveries: create php page, write query to get deliveries from database, take results and put them in a table

Story 12: As a user I should not be able to see certain pages after I log out: implement session functionality, insert it across all pages so users can only see them when logged in

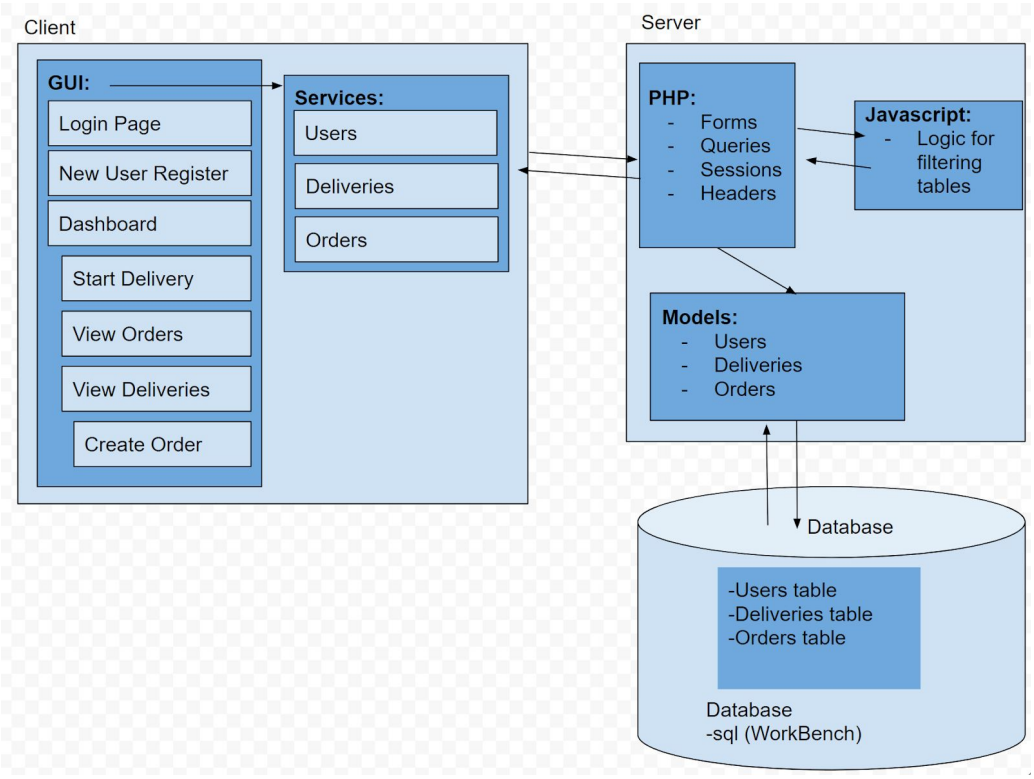
Story 13: As a user I should have better redirect pages when filling out a form incorrectly, tasks: create redirect pages, validate redirect pages, implement redirect pages

Story 14: As a deliverer, I should be able to view orders assigned to me, tasks: create php orders page, write query to get orders based on deliverer id, take results and put them in a table

Story 15: Create automated testing to promote TDD

Story 16: Upgrade UI

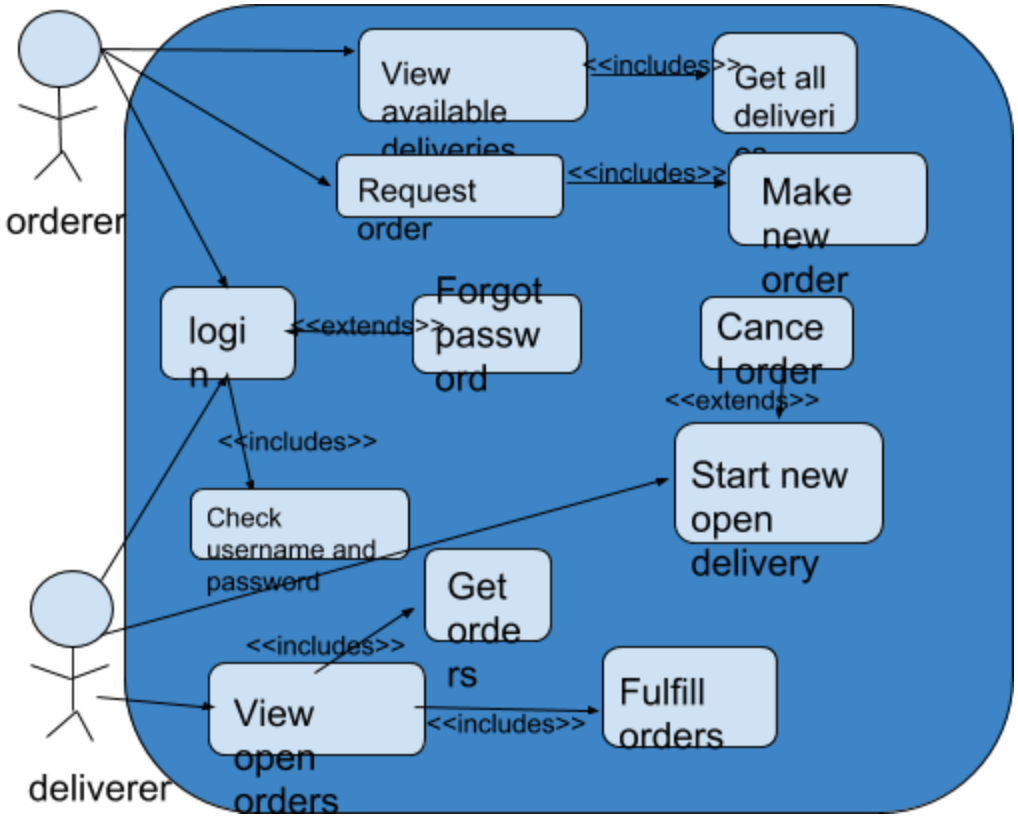
2.2 High-level design

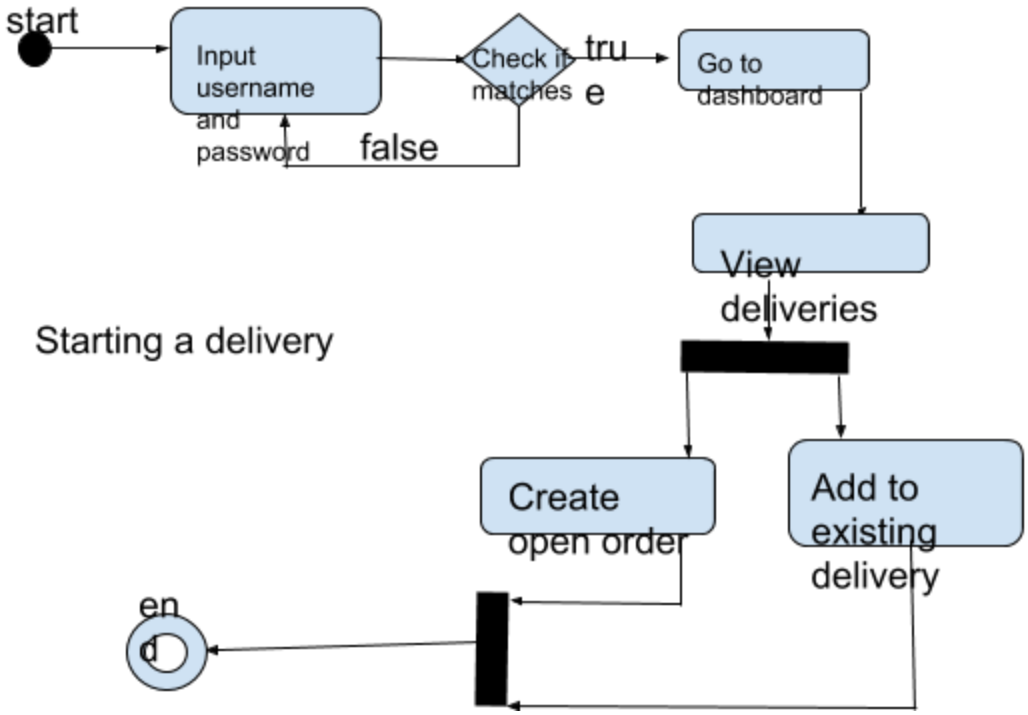


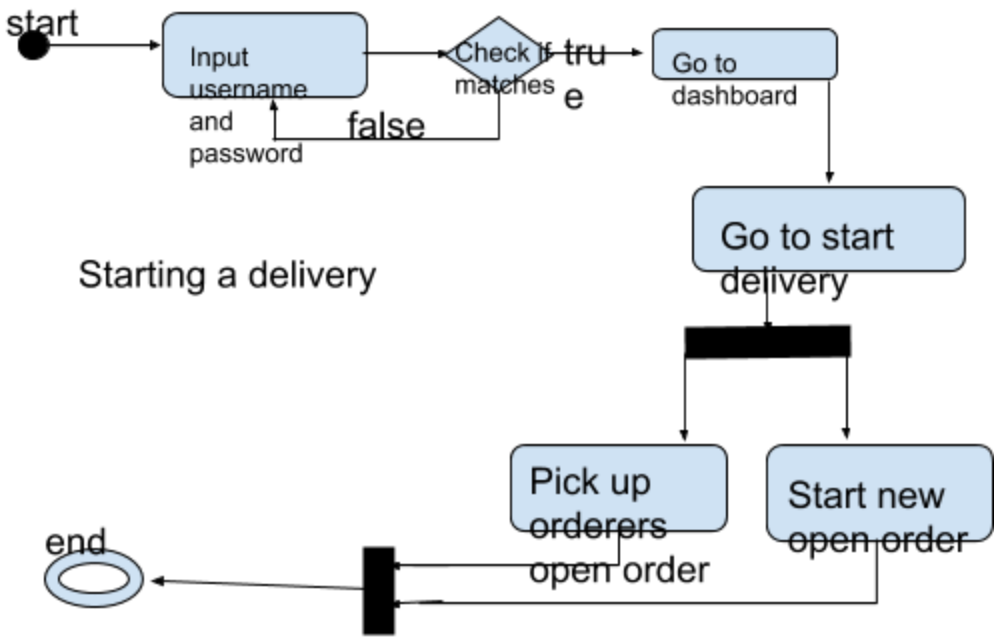
This is broken down into our client, server, and database.

In the client we have the different views and the services they interact with in order to display information and take in information. In the server we have the different items used to allow our program to run. Php is used to create forms that allow the user to input information, to insert queries in sql that allow us to access database information through the models, to implement sessions so users can only access the program when logged in, and to implement headers that can transfer information across pages when needed. In the database we have our different tables that we use to store information about our users, the deliveries being performed, and the orders that are being placed.

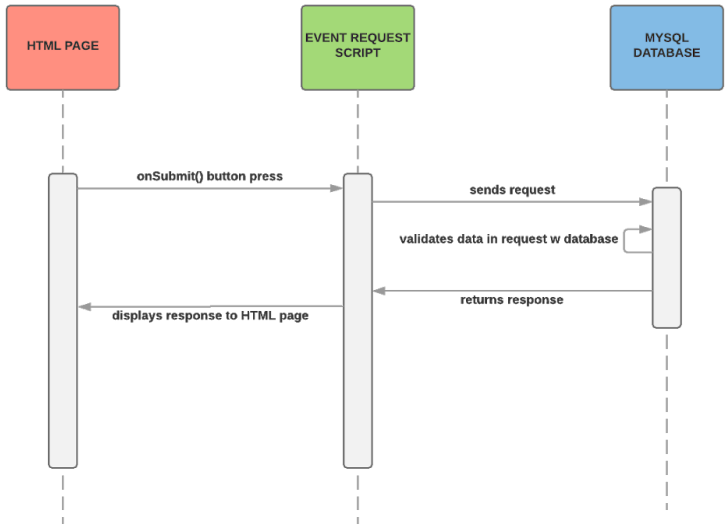
2.3. Interaction Models





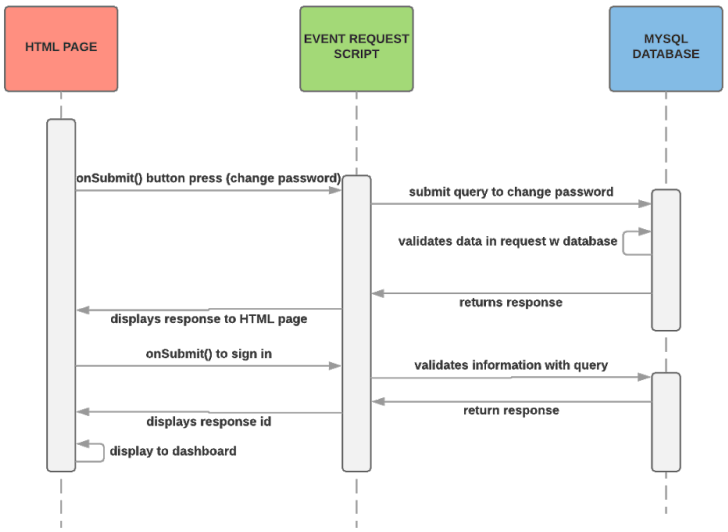


USER SIGNING IN



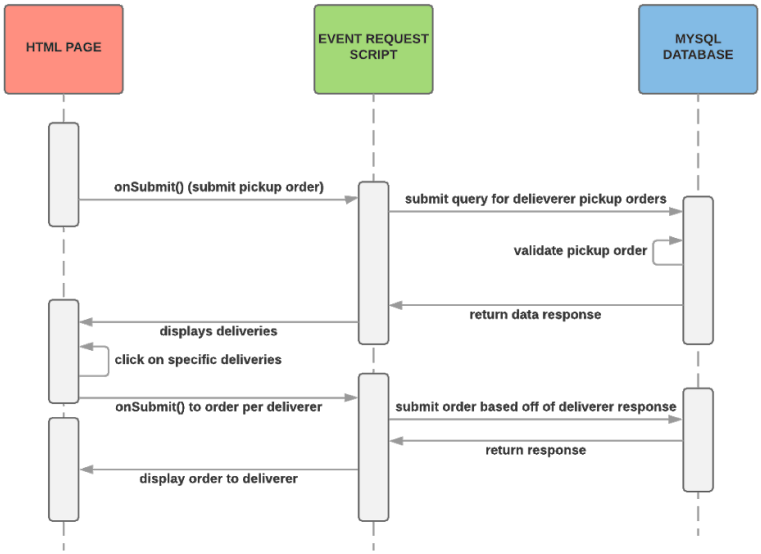
USER SIGN IN/DASHBOARD/FORGET PASSWORD

anngould | April 5, 2018

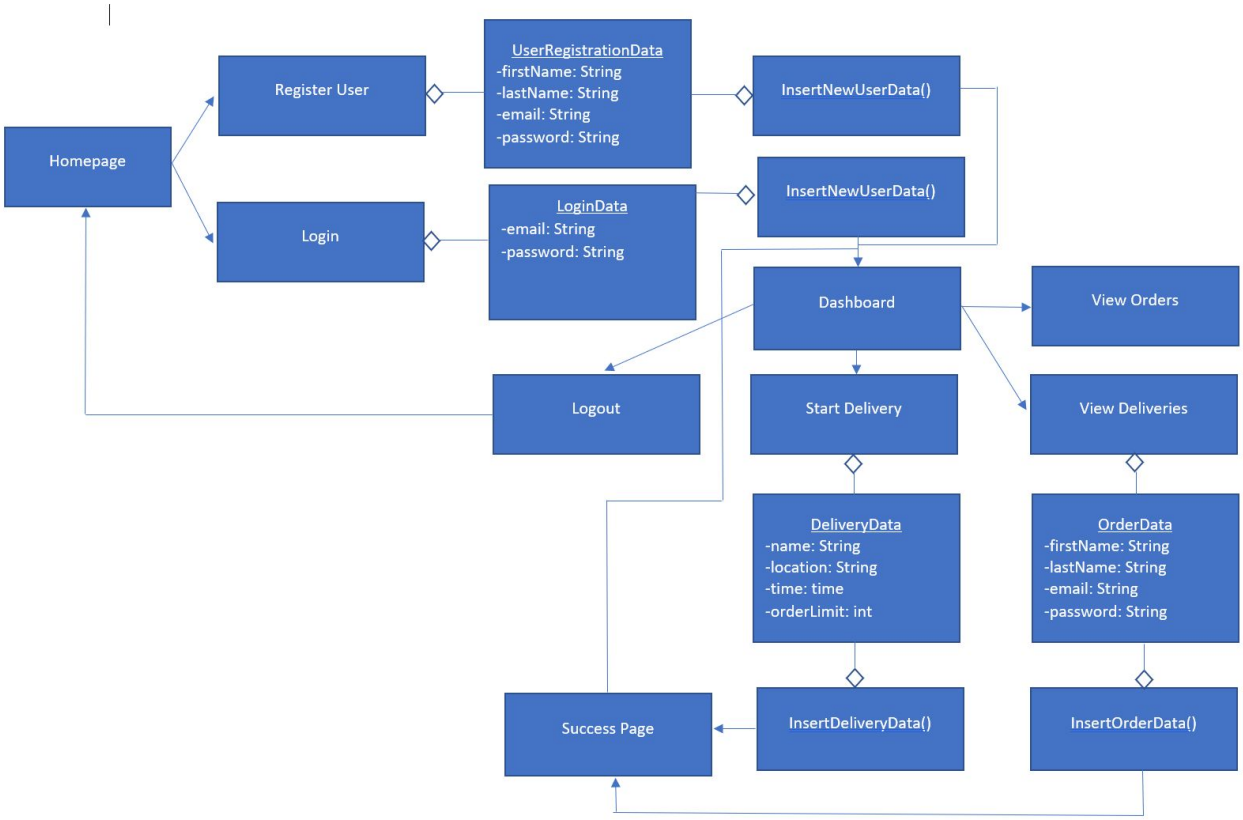


SET DELIVERER STATUS/ORDER COFFEE PER DELIVERER

anngould | April 5, 2018



2.4. Data Organization



2.5 Design Rationale

We wanted to do more with our project, but unfortunately the members of our team who were held responsible to help with the payment portion of our project dropped the course, so we didn't have time to implement that ourselves. We realized through the course that our design wasn't that modular, so changing one part of the project would impact many other parts of the project, which was a poor design. Since we created our project to be easy to use, it was easy to find and change these fixes though, but from later on out we will use a more, better structured stack of frameworks to complete our project.

3. System Implementation

3.1 Implementation Outline

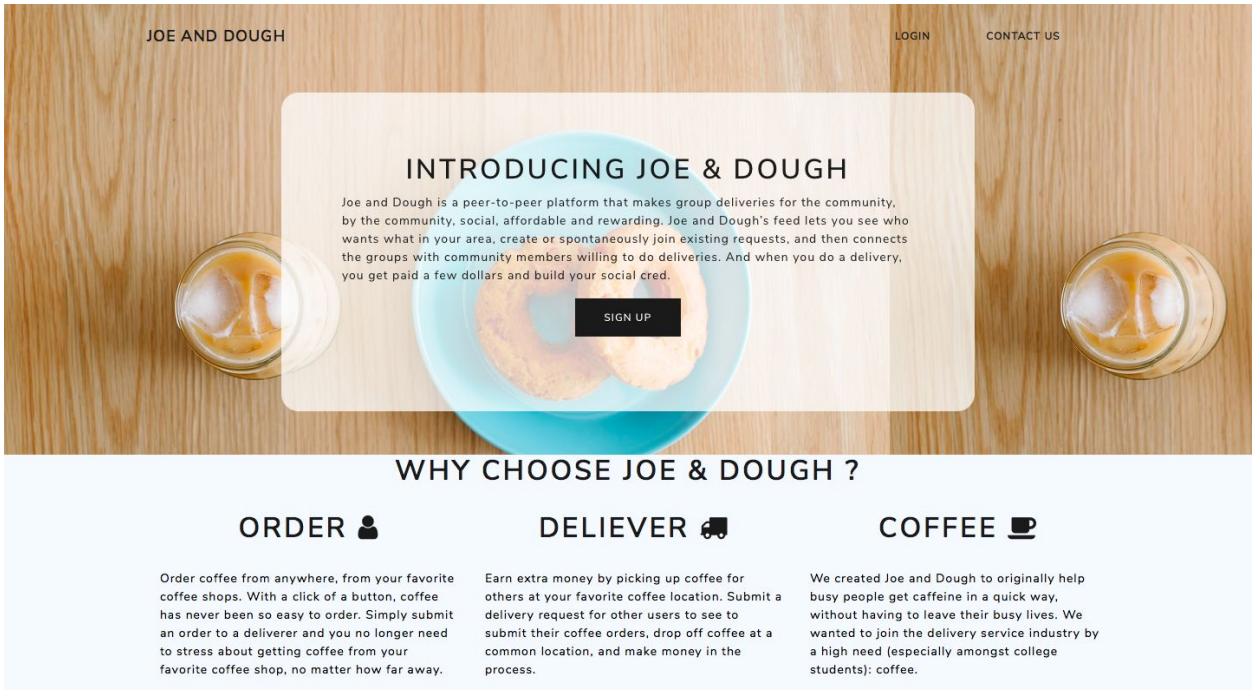
Our project was developed as a mobile website, but should be responsive for use on any platform because of Bootstrap. This application ideally would be used on phones, but could be used on other platforms like computers as well.

We used HTML, CSS, and Bootstrap to organize the front end of our project. We used PHP and mySQL for the backend of our project. We used Creately to create most of our UML diagrams for the class. Our group used Sublime Text editor and Visual Studio Code as the IDE for our project as well.

3.2 Testing

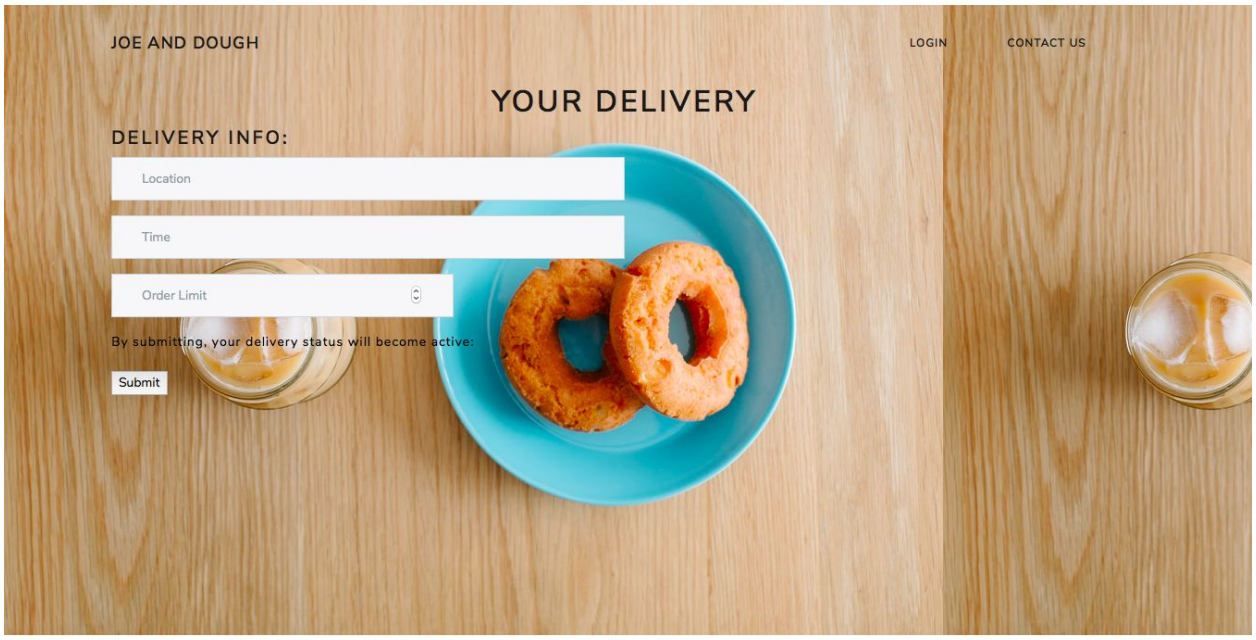
In testing the project, we decided to use the JavaScript library of Nightmare.js, which allows us to simulate user actions within the websites UI. This allows us to test many different essential cases that a user might execute. The Nightmare.js library allows us to keep track of the inputs given by the user to the websites UI and thus allows us to fluently create test cases that otherwise would have stacked data into the actual server. An example of this is creating multiple new users, if we were to create multiple new users within the actual website itself then unavailing data will be stored to the server which we don't require. Nightmare allowed us to effectively test our UI without having to douse ourselves with too much code.

3.3 UI description with Screenshots



Above is the homepage for our application. This is where the user is introduced to our product and can then either login or become a new user.

Above is the dashboard for our application. This is where the user can navigate to all the portions of our product and use all its features.



Above is the start delivery page for our application. This is where a deliverer would input their delivery information. From there, the info is put into the database and displayed for all users so they can begin to submit orders to that deliverer.

Above is the view orders page. This is where a deliverer would go in the application to see the orders that have been submitted to them so that they can purchase the orders and deliver them to the customers.

Above is the view deliveries page. This is where a customer would go if they wanted to order a coffee or doughnut. Here they can choose which coffee shop they would like to order from. The customer can even filter their options via the filter text input box towards the top of the page. Once the customer decides on a coffee shop, they can select the deliverer they want, and submit an order to them.