Development Process

In the creation of our web application, the tools we used included Express, Handlebars, Passport, Mongoose and Jest. The Express framework was used as it is an extremely powerful option for creating robust back end systems for web applications. We used it primarily for routing HTTP requests, rendering web pages and configuring middleware. Handlebars was chosen as our template system as we are most familiar with its functions and we found it to be the simplest option to create and display our application's front end. Passport is our chosen middleware for authentication. This was chosen as it allows us to easily authenticate that vendors and customers have logged in before accessing pages containing sensitive information. Originally we just used simple session storage, but switched to proper authentication with Passport to improve our application's security. Since we are using MongoDB in order to store our data, it was a simple choice when deciding to use Mongoose. This library allowed us to easily communicate with our database, find specific items within it, as well as add or edit data. Jest was chosen as our testing framework as we were more familiar with its functionality than other testing frameworks like Mocha, and found it to be an easy and effective method of testing the functionality of our requests and responses. This allowed us to properly evaluate the functionality of our back end.

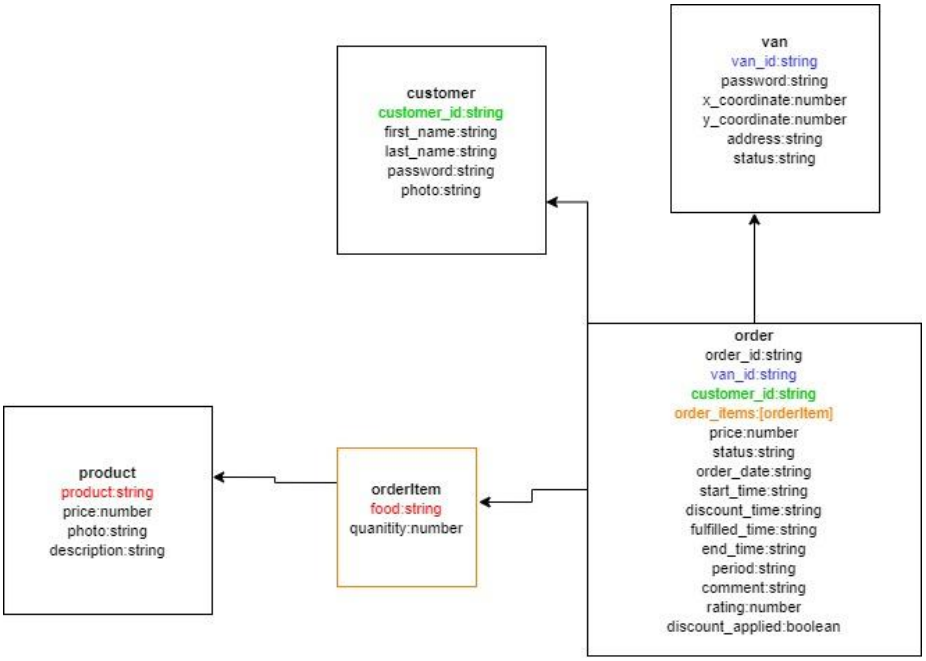
In order to assign tasks to each group member for each deliverable, we would first look at the specification for the deliverable on Canvas. A member of the group would then post the list of features required in our group Discord server, which allowed each member to pick a feature they felt they could implement effectively and ensure everyone in the group understood their role for the deliverable. This Discord server also allowed the group to easily discuss how features would be implemented as well as how they were going to be integrated in order to make the code consistent.

Both our frontend and our backend were hosted on the same GitHub repository. Files that serve different functions are divided among different sub-folders to make navigation and management easier. These include routes, controllers, models (for the schemas), views (for the frontend templates) and js (for miscellaneous javascript files). In addition, a number of markdown files are included within the repository which contain important information relevant to everybody in the group. These files include everything from how to use certain libraries and frameworks, to the current tasks in progress for each member. The main markdown file is pinned to the front of the repository and contains links to all of the other ones to ensure everyone can easily view them and navigate throughout them.

The members of the group are Rin, Bin, Khin, Eric and Declan. Due to the fact that team members would pick their desired features at the beginning of each deliverable, everyone's role changed slightly over the course of the project. However, as team members became familiar with particular parts of the system, roles tended to remain fairly consistent. Rin dealt with a number of system components, including customer orders, customer login, vendor login, as well as implementing the map function and a blog function. Bin dealt primarily with the ordering system on both the customer and vendor side of the application. This included functionality for the vendor to mark the order as ready, viewing order details, allowing the customer to view a completed order, as well as allowing the customer to rate an order. Khin helped Bin and Rin to implement the customer ordering, but also implemented or assisted on

much of the vendor side functionality including viewing outstanding orders, viewing order details and viewing order history. Eric implemented much of the cart and ordering functionality for the customer, as well as the ability for vendors to set van status. He also bolstered the application's security by improving password encryption and authentication methods. Declan dealt primarily with the customer menu functionality, including displaying a list of all snacks retrieved from the database with photos and prices. In addition, he added a customer profile page which allowed customers to change their password, as well as the functionality for vendors to search for specific orders by their order ID.

Database Design

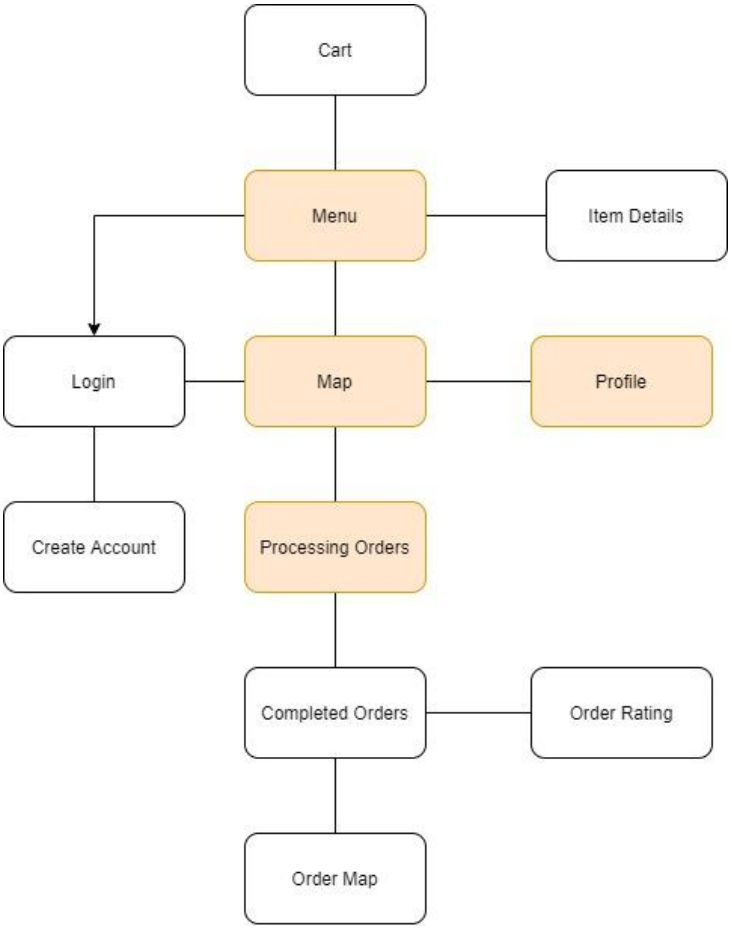


The above diagram shows the design of our database. The colour coding shows which fields are referencing which schemas. For example, the "food" field in orderItem is referencing the "product" field from the product schema and the "order_items" field in order references a list of orderItems. With this design, we can easily find the information we need about the customer and vendor who dealt with any given order, as well as find everything about the items that were included in the order.

System Architecture

Customer Application

The entry point to the customer system is through the customer login page, accessible at https://snacks-in-a-van-4399.herokuapp.com/customer/login. Once the login is authenticated, the user is taken to the map page, where they can choose which van to order from. From this point, the other pages on the application can be reached. The full architecture is shown in the following diagram:

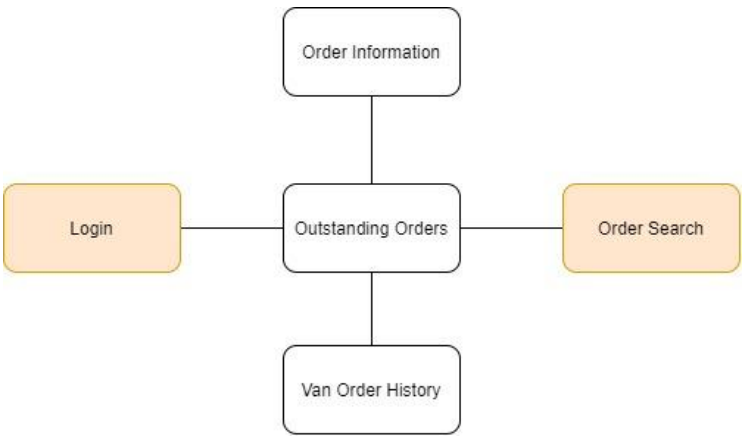


This diagram shows the flow of pages within the customer application. The pages shown in orange comprise the pages accessible from the navigation bar which is displayed on all customer pages after the user has logged in. Due to this, the menu, map, profile and orders pages are accessible globally throughout the customer application.

It can also be seen that most of the connections are bi-directional, meaning that when a user navigates from one page to another, they can also simply go back to the previous page. The one exception to this is the connection between the menu page and the login page. Upon a new login, the user is automatically brought to the map page to choose a van and then will naturally navigate to the menu to begin their order. From here, the user has the option to log out and will be brought back to the login page. From here, there is no direct way back to the menu page outside of logging back in and manually navigating to it. Hence, the single-headed arrow pointing from menu to login.

Vendor Application

The vendor, or van, system is very similar to the customer system in terms of page flow. The entry point is the login page at https://snacks-in-a-van-4399.herokuapp.com/vendor/login. From there, the vendor is taken to their outstanding orders page, where the rest of the pages are accessible. The key difference is that, since most of the functionality is available on the outstanding orders page, there aren't as many necessary pages for the vendor application. The design is shown below:



Just as with the customer application, the login and order search pages are accessible from the whole application, as they are

included in the navigation bar at the top of the application.

Once the vendor has successfully logged in, they have access to all order functionality from the outstanding order page, which contains the ability to set orders as fulfilled and picked up, as well as the ability to set the van as closed and open. It is for this reason that the vendor architecture is much simpler.