

**704 On the Go**

Group 7

Zelaya A., Sonu B., Rhea K., Joshua R., Melody S.

1

**Table of Contents**

**1 Project Overview 2**

**2 Architecture Overview 2** 2.1 Subsystem Architecture 3 2.2 Deployment Architecture 4 2.3 Persistent Data Storage 4 2.4 Global Control Flow 5

**3 Detailed System Design 5** 3.1 Static view 6 3.2 Dynamic view 7

2

**1. Project Overview**

*704 On the Go!* Is an information and solutions website focused on providing users with the tools to explore the Charlotte area to maximize their experience while visiting. We want users to be able to uncover the diamonds among our many restaurant locales and entertainment hotspots that fit what they are looking for. Its main function is to help people get to know Charlotte better and be familiar with what the city has to offer. Additionally, it will provide information on everything that goes into traveling in Charlotte, including transportation routes, parking, and, as mentioned, the businesses. Our hope is that *704 On the Go!* is able to alleviate this space where other websites and applications seem to fall short by providing a more full experience in local information. There are two stakeholders in this system. The first stakeholders are the customers who could use the system if they are ever in the Charlotte area and need somewhere to eat, shop or tour. The second stakeholders are businesses such as shop or restaurant owners who could pull in new customers in the area. If a business is listed on the site, people are likely to visit the location, increasing the amount of visitors. The system can also provide more exposure to small local businesses in the area.

**2. Architectural Overview**

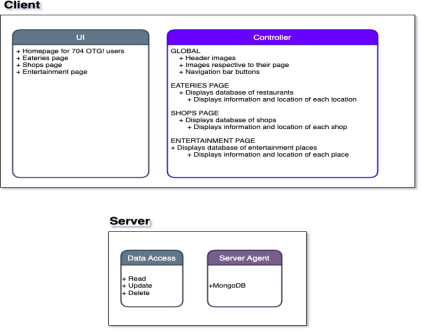
Since the main purpose of the website is to include shops, eateries, transportation, and entertainment in one page, we will be creating a webpage with everything. We believe that navigating through a web page would be more efficient than an app. We plan to have a landing page where users can learn more about Charlotte and who we are. All the pages will have a navigation bar on top that will include all the pages the user can go to. As our web page requires information about the different eateries, shops, transportation, and entertainment, we will use

3

MongoDB as our database to store all the information we would need for each location. The website will also use the MVC architecture, JavaScript, CSS, and HTML. We will be using Node.js to create a dynamic website. Initially, we did think about adding a login/sign up page but after further evaluation, our group decided there was no need. The central idea of the system is to view different locations where logging in/ signing up are not required.

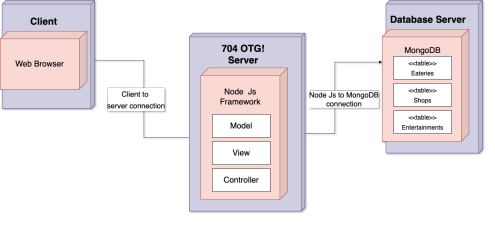
**2.1 Subsystem Architecture**

Our application has two layers: client and server. The client layer includes all the pages seen in the website such as the home page, eateries page, shops page, and entertainment page. While the server layer showcases a lengthy list of restaurants, shops, and entertainment places, along with the details of each place. Databases were first created in Google Sheets, then converted and populated using MongoDB. As node.js is a single threaded software system, our site will be hosted locally.



4

**2.2 Deployment Architecture**

****The client connects to the 704 On the Go! server using the HTTPS on their web browser through the internet. Node Js hosts the whole website server with the MVC (model, view, controller) architecture. It is then connected to the database server through MongoDB. With the MVC architecture, the model component sets up the data to be used throughout the site such as the name, location, description, ratings, price range, and url for the different pages. The view component holds the front-end side of the system such as the different pages. Last but not least, the controller component defines what to do with the data from the model to the view.

**2.3 Persistent Data Storage**

Our group first individually collected data about each shop, eateries and entertainment places we have on the website. The data was collected by going to each business’s website and gathering their description, rating and address. The data was stored in Google Sheets for the group’s convenience. Once all the description was collected, the data was exported as JSON. We then

5

used MongoDB to store the data as documents. At this point, we are not going to store any user data such as login/sign up information.

**2.4 Global Control Flow**

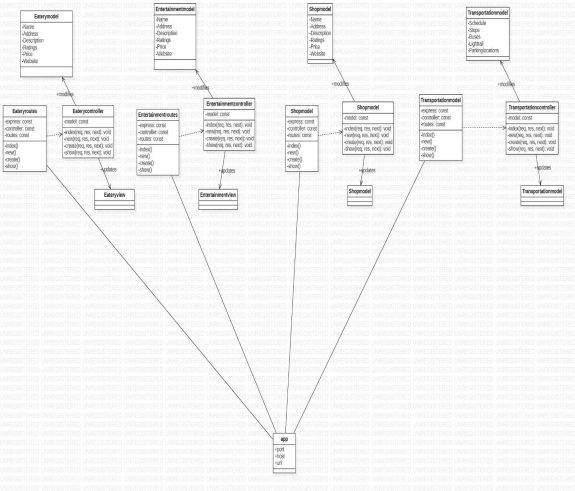
Our system is procedure-driven, meaning that it executes in a linear fashion, where every user has to go through the same steps every time. When going on the website, the user will land on the home page. The user can choose if they want to view the eateries page, shops, or entertainment page. In each page, the format is the same where they can search where they want to visit, shop or eat at. Once they have the destination set, they can proceed to the transportation page where they are able to explore the options they have such as using buses, Blue Lynx line light rail, or parking.

**3. Detailed System Design**

We used MongoDB for our model as shown below. It has four different models to present as Eatery, Entertainment, Shops and Transportation. They each have different controllers and the controllers modify the models. The controllers are used to manage the flow of the data which will be displayed in view. The routes depend or rely on the controllers. Then, the controllers would update the view of it. The view would demonstrate the buttons and the information or updates that the user will be able to see.

6

**3.1 Static View**

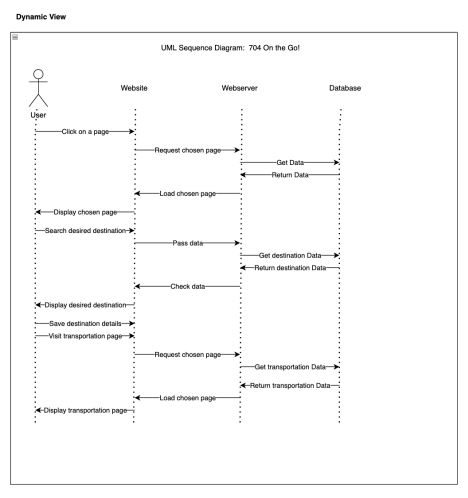
****

The Eatery model, entertainment model and shops model hold the Name,Address, Description, Ratings, Price and Website. Each controller holds the functions such as index, new, create and

7

show. The routes depend on the controllers so it has similar functions. The view is updated by the controllers as well. The transportation model has a bit different attributes such as Schedule, Stops, Buses, Light Rail and Parking locations. Lastly, the app is associated with the routes classes.

**3.2 Dynamic View**

****

8

When the user first goes on the website, they will land on the home page where they can learn a bit about Charlotte. From the home page, there is a navigation bar on top that allows users to visit different pages such as the shops page, eateries page, and the entertainment page. Since we have all the data saved about countless shops, eateries and entertainment places, the database would get the data and return it to the web server. After the user has determined their destination and saved it, they can visit the transportation page to explore how they want to get to their destination.