**Detailed Design Specification of Prototype Restaurant**

John Haney

Elizabeth So

Kyle Voon

November 21, 2019

**Table of Contents**

[**1. Scope**](#_rmp84kry36j6) **2**

[1.1 System objectives](#_xnakvo7pi369) 2

[1.2 Major software requirements](#_xxmas8jfx7z6) 3

[1.3 Design constraints, limitations](#_nq8k9k2s525z) 3

[**2. Data Design**](#_ys3rsad8pgu6) **5**

[2.1 Data objects and resultant data structures](#_mad1lex7o5eo) 6

[2.2 File and database structures](#_rpt9o5x0km1g) 7

[a. External file structure](#_t0mala4iu8fu) 7

[i. Logical structure](#_tg8n88gkjyxr) 8

[ii. Logical record description](#_s84cceudapva) 8

[iii. Access method](#_iwyj3lahmgd0) 9

[b. Global data](#_m664iadfvwmn) 9

[c. File and data cross reference](#_2o9w9tdcu1sr) 10

[**3. Architectural Design**](#_rtx824bf1uby) **10**

[3.1 Review of data and control flow](#_mwrgbfponedz) 10

[3.2. Derived program structure](#_9d1a2odixqbx) 13

[**4. Interface Design**](#_58dmyk6go243) **13**

[4.1. Human-machine interface specification](#_qtyxzh74ez54) 14

[4.2. Human-machine interface design rules](#_cat2jub6u7n7) 14

[4.3. External interface design](#_q9zci18ar3p4) 18

[a. Interfaces to external data](#_oujstdi85dq1) 18

[b. Interfaces to external systems or devices](#_3kjil9pxbavu) 19

[4.4. Internal interface design rules](#_5e32uzycx4cc) 19

[**5. Procedural Design**](#_timc5pamk18r) **19**

[5.1. Module descriptions](#_9pu86e6etbst) 20

## 1. Scope

This document is designed to be read by software developers, software designers, software testers, and implementers of this website, Cafe Chang (pending name change). This document describes the website architecture for the implementation of Cafe Chang (pending name change), including the sub-architecture, associated interfaces, database schemas, and the structural and behavioral models. Both high-level and low-level designs are included in this document.

This document includes, but is not limited to the following information for the Cafe Chang website: system objectives, design constraints, design considerations, architectural strategies, system architecture, policies and tactics, and detailed procedural system design. This detailed design specification is also aimed at specifying requirements of software to be developed but it can also be applied to assist in the selection of in-house and commercial software products.

This document has an accompanying document, Requirements Analysis Document, per Requirements Analysis Document version 1.0. Any previous or later revisions of the Requirements Analysis Document require a different revision of this design document.

### 1.1 System objectives

The website is designed for the goal of the business to generate more revenue by creating more customer awareness of the business and increasing awareness and traffic within the website. In addition, the website will improve customer satisfaction by introducing our restaurant on the web for new-veteran customers to stay up-to-date on our store hours, location, menu, and any additional changes or unforeseen forecasts. Lastly, the main objective of this system website is to develop and create a website for the business owner satisfaction of presenting their business to the world.

In addition, we will automate the current workflow of the business by introducing an online takeout system. This online selection service system shall reduce the amount of time spent by both the customer and the business on taking/placing orders using the current method of answering and taking phone calls. This new system will also mitigate the amount of errors due to miscommunication from both parties, in turn, reducing the amount of food returned.

### 1.2 Major software requirements

Customer: For customers to view/access/interact with this website, the following software requirements are needed. Customers shall use a standard PC, tablet, smartphone, or laptop containing one of the following browsers: Internet Explorer 11, Microsoft Edge 18, Chrome 78, Firefox 70, and/or Safari 13.

Server Host: There shall be two options for the server host. For this design specification document, option 2 shall be selected:

1. The server host shall have the minimum hardware requirements: processor 1.6 GHz CPU, 1.75 GB RAM, HDD 1x40 GB of free space or more is recommended for the software that is listed in the software requirements (system drive).
2. GitHub shall host the following website, Cafe Chang.

Developer/Designer: For developers and designers to create this website, the following software requirements are needed with PC, Laptop, or Tablet with the following capabilities:

1. Software to develop website: Visual Studio Code (Text Editor), IntelliJ IDE
2. Internet Browser (Chrome 78, Firefox 70, Safari 13, Internet Explorer 11, Microsoft Edge 18)

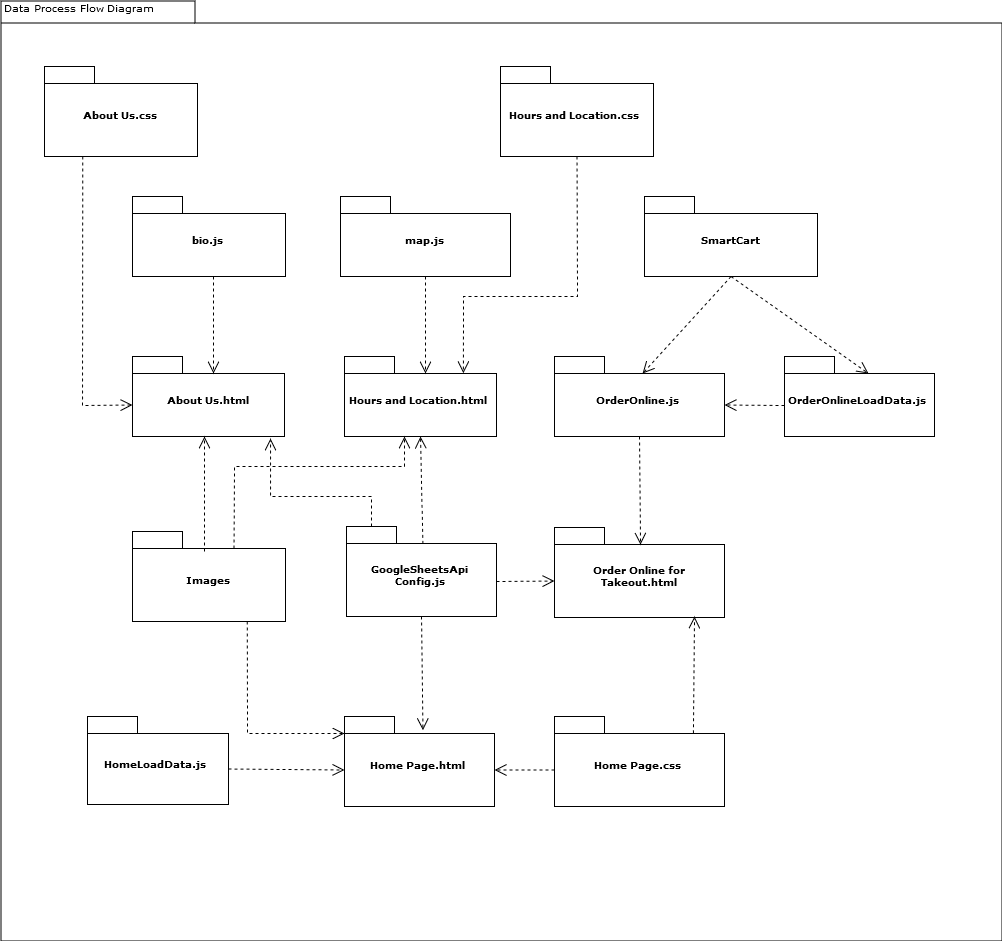
### 1.3 Design constraints, limitations

This website design has several design constraints and limitations placed on the system as follows:

* Time: Development of this website relies on a 15-week deadline as specified in COSC 625 Analysis Document - Group 1 version 1.0.
* Budget: Budget for this website is based on a $150 per hour cost multiplied by the number of hours over the 15-week deadline. The budget is expected not to exceed $100,080. A more in-depth analysis of the budget breakdown costs are outlined in COSC 625 Analysis Document - Group 1 version 1.0.
* End-user environment: Mozilla, Opera, Maxthon, Flock, SeaMonkey, Avant, K-meleon, Konqueror, Links, OmniWeb, GNOME Web, Netscape Navigator or any other web browser not specified in this document, shall not be tested for web browser compatibility with the system.
* Interface/protocol requirements: There is no HTTP protocol used or port number used or logical address of the system. There is no website domain name purchased.
* Data repository and distribution requirements: The system has not implemented data storage, such as storage in a SQL server.
* Security requirements (or other such regulations): Due to time constraints, future implementation of the following will be included: system shall use secure sockets in all transactions that include any confidential information, confirm all transactions with the customer’s web browser, not leave any cookies on the customer’s computer containing confidential information, and log out all customers after a period of inactivity.
* Performance requirements: Performance limitation of the website due to lack of budget. The following could be used to improve the performance of the website: hosting plan, HTTP requests, and caching.
* Other means of addressing quality goals: System does not support Customer login or password with password encryption. System does not have back-end servers or databases that store encrypted confidential information that is only accessible to authenticated administrators.

## 2. Data Design

The majority of the data input and processing will be associated with the Online Takeout Ordering page. The Home/Main page will involve retrieving images for display of the different food categories available on the menu and may involve retrieving data from an external data source to populate the items displayed on the menu. The Hours and Locations page will involve embedding a Google Map that will display the location of the restaurant and may involve retrieving data from an external data source to populate the hours during which the restaurant is open at the page footer. The About page will retrieve images for display in the slideshow of the chefs/owners and slide down windows with corresponding biographies. Figure 1 displays an activity/data process flow diagram for the website showing how data flows between HTML pages from JavaScript, CSS, and image files.



*Figure 1. Activity/Data Process Flow Diagram for website*

### 2.1 Data objects and resultant data structures

In the Online Takeout Ordering page, data objects will be created in JavaScript Object Notation (JSON) and manipulated using JQuery in a JavaScript file containing code specific to the page. The data objects are in key-value pairs and will represent data used for populating dropdown menus on the Online Takeout Ordering page such as the days of the week, months, and time ranges containing the open and close time of the restaurant for each day of the week. The open and close time of the restaurant should also match the hours displayed at the footer of the Hours and Locations page. Additional data objects will be created containing information about the restaurant items that can be ordered on the Online Takeout Ordering page. Similar data objects may also be used to populate the list of menu items on the Home/Main page. However, the data objects for the Home/Main page will not need to use or contain some of the properties needed for the data objects on the Online Takeout Ordering page (e.g., quantity only needs to be used for orders).

### 2.2 File and database structures

The data required for the Online Takeout Ordering page will be based on a simple data structure contained within one tab of a Google Sheets spreadsheet, to minimize the performance impact of having joins of data in multiple sheets. The data structure will include columns for the item ordered, description, price, and quantity available. If we had additional time for this project, the data could be stored in an open-source database using a similar structure. This data structure could also be used for populating the menu items displayed on the Home/Main page and for populating the hours displayed on the Hours and Location page. Images will be included on the Home/Main page and About pages in JPG format.

#### a. External file structure

Although the data will not be read from a flat file, the data will be read from a spreadsheet-like data structure that is hosted external to the location in which the website will be deployed. Data will be organized with columns representing different data and rows representing different records.

A combination of external and internal static resources in the form of JavaScript files and stylesheets will be used for providing dynamic functionality and styling for the website. Minimized/compressed external JavaScript and stylesheets will be retrieved using content delivery networks (CDNs) to improve website performance, whereas internal JavaScript and stylesheets will be hosted local to the website.

##### i. Logical structure

Different data structures will be used to store data for each of the pages. Table 1 includes a list of possible data fields to be captured for the logical structure of the data objects used on the page, excluding images.

*Table 1. Logical Structure of Data Objects for Website*

|  |  |
| --- | --- |
| **Page** | **Logical Structure for Data Object / Data Fields** |
| **Home/Main** | Menu Items   * Item name * Description * Price * Category * Food sensitivities * Spice level |
| **About** | Person Menu   * Person’s name * Initial biography * Drop down full biography |
| **Hours and Locations** | Hours   * Day of the week * Start time * End time |
| **Online Takeout Ordering** | Items Ordered   * Item name * Quantity   Item Inventory   * Item name * Quantity * Price |

##### ii. Logical record description

Table 2 contains descriptions associated with the data objects to be created for the website.

*Table 2. Logical Record Descriptions for Data Objects*

|  |  |
| --- | --- |
| **Logical Record for Data Object** | **Description** |
| Menu Items | Items available on the menu |
| Person Menu | Information associated with a person |
| Hours | Hours during which the restaurant is open |
| Items Ordered | Menu items a customer selects for an online takeout order |
| Item Inventory | Menu items the restaurant has available to be selected by the customer |

##### iii. Access method

Data stored in Google Sheets will be retrieved using the Google Sheets API (<https://developers.google.com/sheets>) in JSON format within JavaScript files prior to being displayed on the website pages. Usage of the Google Sheets API will require the use of an API key that can be obtained using the Google Developers Console (<https://console.developers.google.com>).

#### b. Global data

Use of global data objects will be minimized as much as possible to adhere to object-oriented principles of design, such as data encapsulation. Individual custom JavaScript files can be included on the HTML pages that require specific data objects to be loaded so that data can be displayed to the user.

#### c. File and data cross reference

One Google Sheet will contain different sheets or tabs representing the logical structure. Sheets corresponding to the Menu Items, Hours, Items Ordered, and Item Inventory structures will be created and contain data that can be retrieved from JavaScript.

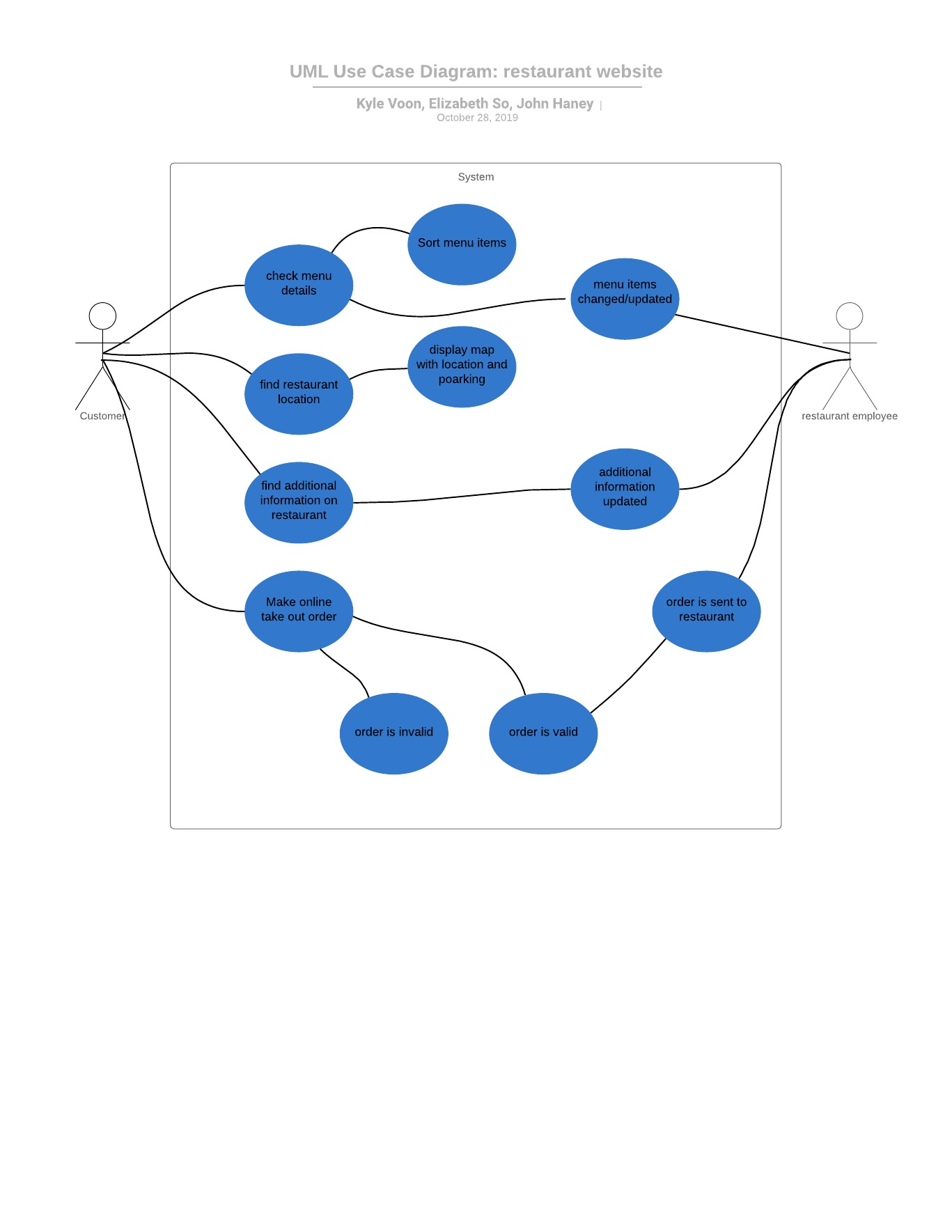
## 3. Architectural Design

### 3.1 Review of data and control flow

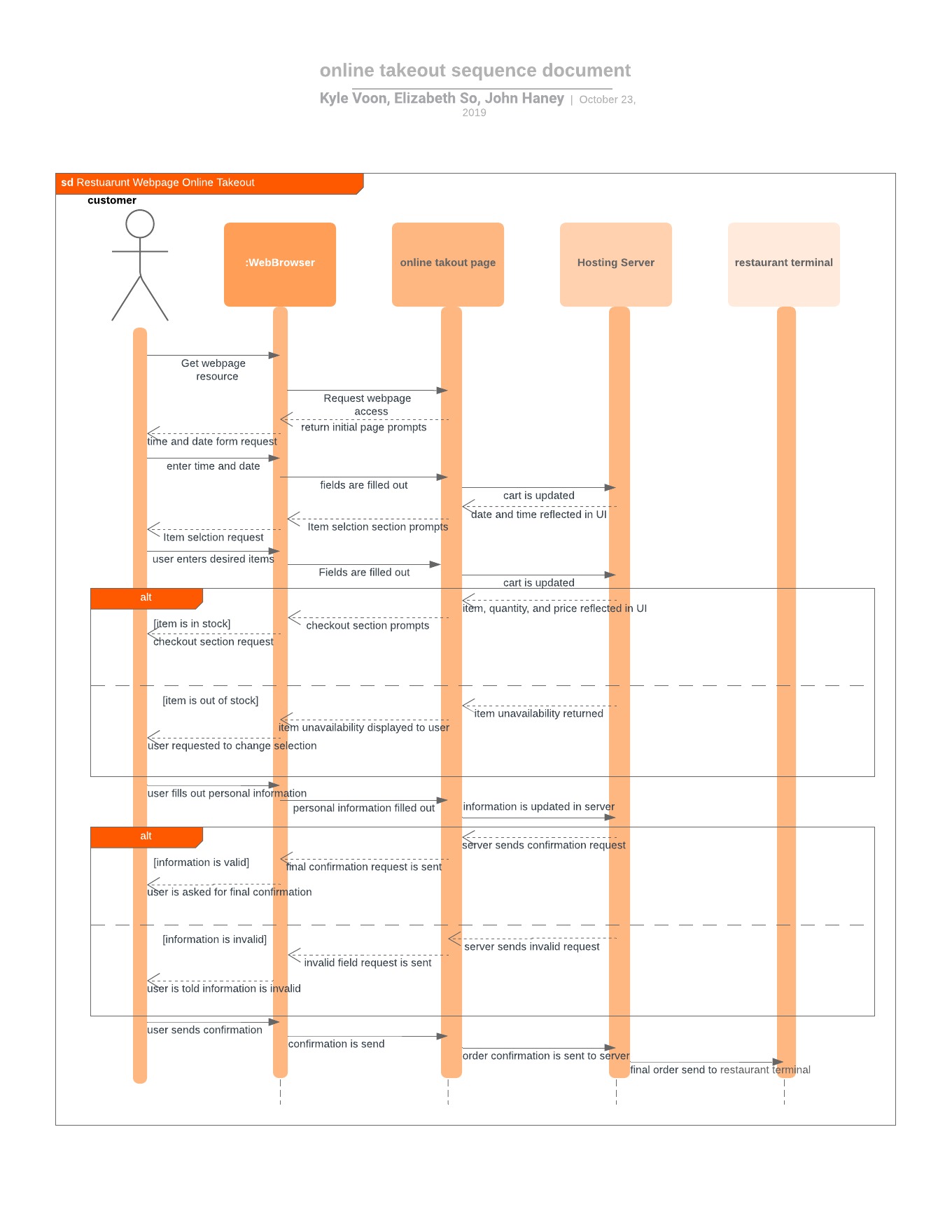
A read-only online, Google Sheets spreadsheet can be used as the data source for items displayed to the user on the Home/Main page, Hours and Locations Page, and Online Takeout Ordering page. Pulling data from the online spreadsheet will allow for ease of updating data that may change often on the website and serve as a prototype of what the structure of a database would look like so that data does not need to be hardcoded in the website. This data can include the menu items for the Home/Main page and the Online Takeout Ordering page and the hours during which the restaurant is open for the Hours and Location page.

If a database was used instead of Google Sheets, connections would need to be made from within the code to retrieve data from the database. A web service could be set up using a server-side language that would be able to communicate with the database using an object-relational mapping framework that would be able to convert data from the database into objects that could be parsed within the JavaScript prior to being rendered on the webpage.

Data will only be input by the user on the Online Takeout Ordering page in the form present on the page. Once the user completes data entry on the form and submits the form, the data should be sent to an external endpoint for processing and a message should be returned indicating that the order was processed successfully or that there was an error processing the form. Figure 2 contains a use case diagram of the website and Figure 3 contains a sequence diagram for the Online Takeout Ordering system.



*Figure 2. Sequence diagram of website*



*Figure 3. Sequence diagram for Online Takeout Ordering System*

### 3.2. Derived program structure

The website is composed of four different pages: Home/Main page, About page, Hours and Location page, and Online Takeout Ordering page. The code for each page will be contained within one HTML file and any relevant dynamic functionality for each page will be contained in a separate JavaScript file and referenced in the HTML page. Similarly, stylesheets will contain styling for each page, such as the size, color, and position of elements and will be referenced in the head of the HTML page. The content of the HTML pages will be contained in the body and may include data from an external data source that is dynamically populated using JavaScript. Templates will be used for HTML code that is repeated on a page in order to maximize code reuse and allow for dynamic content to be inserted within the template. For example, HTML templates will be used on the Home/Main and the Online Takeout Ordering pages for displaying menu items.

## 4. Interface Design

Open-source technologies will be used to develop the website because they are readily available and widely used for website development. HTML and CSS will be used for the front-end technologies. Bootstrap 4 (<https://getbootstrap.com/>) will be used for the design layout, Font Awesome (<https://fontawesome.com/>) will be used for the icons, Google Fonts (<https://fonts.google.com/>) will be used for more appealing fonts and JQuery (<https://jquery.com/>) will be used to provide additional dynamic functionality to the website, JQuery UI (<https://jqueryui.com/>) will be used to provide additional user interface features on top of JQuery, and Handlebars (<https://handlebarsjs.com/>) will be used for HTML templates. The SmartCart v3.0.1 plugin (<https://github.com/techlab/SmartCart/releases/tag/v3.0.1>) will be used to provide functionality for the cart in the *Checkout* section of the Online Takeout Ordering page. All of the pages on the website will require user interaction with elements on the page for adjusting the data that is displayed on the page.

### 4.1. Human-machine interface specification

All of the pages will include a navigation bar at the top of the page with links to other pages and a footer section containing the address, hours, and phone number at the bottom of the page to help provide a consistent layout to help the user navigate to other pages and identify pertinent information necessary for contacting the restaurant. The navigation bar will remain at the top of each page, even as the user scrolls up and down on the page, acting as a sticky header so the navigation links are always present on the page and easy to locate. The middle of each page will contain content that provides information about the restaurant or elements with which the user can interact.

### 4.2. Human-machine interface design rules

*Home/Main page*

For the Home Page, an appealing simple layout shall be used. The system is broken down into eight sections: Navigation bar, Entry Page Menu icons, Appetizers, Lunch, Dinner, Cocktails, and footer. The page shall have an obligatory navigation bar and footer div that shall be uniform throughout the system. The entirety of the website will use cooler neutral colors such as whites greys, dark blues, and light reds as to create a feeling of calm and sophistication for the user. The system uses Pacifico font with a light contrast between text and images displayed throughout the Home Page. The Entry Page presents the website name with a picture of the restaurant to give a simple and elegant appeal to end-users. The menu icons enable the user to experience an interactive choice between what area of the restaurant menu they want to enter. For each of the following menu icons: Appetizer, Lunch, Dinner, and Cocktails, users can hover over to experience a hover and zoom effect as well as a clickable link directed to the section selected. Each of the menu sections: Appetizer, Lunch, Dinner, and Cocktails will have individual background images.

To enable a more readable menu, a transitional overly shall be applied to each section. Each section of the menu shall contain eight menu food items displayed within two columns and four rows. Each menu item shall have a surrounding border to be distinguishable from each other and shall include the menu item in the left corner in larger text, followed by the description below in smaller text, along with the price on the right side of the text. In addition, a chef selection label shall be added for users to distinguish what the chef has recommended. The Home Page provides an entrance for the user to experience a simple appealing layout and structure.

*Location and Hours page*

For the Location and Hours page, a simple layout will be used. The page will have the obligatory navigation bar and footer div. Additionally, there will be a facade section that will contain an image of the outside of the restaurant as a background. The main aspect of the page will be the Google Map application that will be placed in the right center of the facade section. The page will be kept simple and contain no other information than the hours and location as to ensure there is no confusion.

The map will be implemented using the Google Map API which will have its code written in a separate JavaScript file. The map itself will use a standard red pin to mark the restaurant's location and several blue “P’s” to signify the parking nearby. On the left center of the facade section will be the text divs that will show the different hours for lunch and dinner along with the parking situation. Finally, all stylization will come from either the global CSS (home page CSS), the location and hours CSS, or the open-source CSS online.

*About Us page*

For the About Us page, a href system combined with accordions will be used to make navigation easy and also to limit information crowding. Similar to the home page, href tags will be used to anchor sections of the page to reference images that the user may click on to find more information they may be interested in. In each section, there will also be “full bio” boxes that when clicked on will reveal additional text information on each person. This accordion style box will also be used in the main “about” section of the page that will reveal a more in-depth history of the restaurant.

The sections of the page will be displayed in the following order: navigation bar, About section with background image and history text, more information menu with picture hrefs to click on, founder section with image of founder and full bio box, head chef with full bio box, and finally obligatory footer section.

*Online Takeout Ordering page*

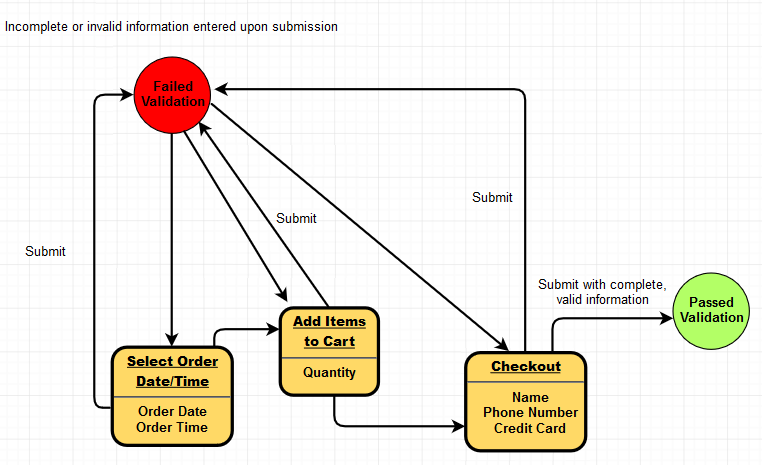
For the Online Takeout Ordering page, an accordion style layout will be used to reduce the amount of scrolling required on the page and allow the user to focus on specific sections of the takeout page. Only one section will be open at a time, with the first section open by default so that the user knows where to start the ordering process on the page. The sections to be included on the page are: 1) *Order Date/Time* where the user can select the date and time when they want to place an order, 2) *Add Items to Cart* where the user can add items from the menu to the online takeout cart, and 3) *Checkout* where the user can enter customer contact and payment information and submit the order. The sections will also be numbered so the user will understand that the first section should be completed before the subsequent sections.

In the *Order Date/Time* section, the user can select the date and time when the user wants the order to be ready from two different dropdown menus. The dropdown menu containing the date will include the day of the week, month, date, and year for up to seven days during which the user can place the order, which are ordered chronologically, with the current day listed first. The dropdown menu containing the time will include times in intervals of fifteen minutes, ordered chronologically starting with the time at which the restaurant opens and ending with the time at which the restaurant closes. If the current day is selected, a time close to the current time will be the first available time the user can select to order. The times present in the time dropdown menu will change based on the date that is selected from the date dropdown menu.

In the *Add Items to Cart* section, the user can enter the quantity for menu items to be added to the cart and press a blue “Add to Cart” button to add items to the cart. The cart will display a summary of the items added to the cart and a subtotal for all of the items added to the cart. The *Checkout* button in the *Checkout* section will not be clickable until items have been added to the cart. The user can click the *Checkout* button to navigate to the *Checkout* section or the *Reset* button to clear items from the cart.

In the *Checkout* section, the user must complete all required fields before the form can be submitted. If there are errors during form validation, a red outline will be placed around the fields with errors and an error message will be displayed under the form element with an error because red is typically used to indicate an error. For example, if a 10-digit phone number is not entered, a message stating “Please enter a 10-digit phone number” will appear under the *Phone Number* input field. Error messages are also displayed if a credit card number is not entered in the correct format or a name is not entered for the *Name* input field.

Similar to the behavior of the *Checkout* button in the *Checkout* section, the *Submit* button will not be clickable until all required fields on the form have been selected. Enabling buttons only once the required preconditions have been satisfied will serve as a constraint to signify to the user of the intended functionality of elements on the form. The color of a button can also change when the user hovers over the button to serve as an indicator that a button is clickable. The data entered in the form will be validated upon submission. Figure 4 displays a state diagram associated with the Online Takeout Ordering page, in which the possible states are *Passed Validation* or *Failed Validation* depending on the form input at the time the user presses the *Submit* button.



*Figure 4. State diagram associated with the Online Takeout Ordering page*

### 4.3. External interface design

The website can include interfaces used to retrieve or submit data to and from external systems or devices that exist within JavaScript functions.

#### a. Interfaces to external data

JavaScript functions will use the Google Sheets API in order to retrieve data from a Google Sheets document containing data to be populated on pages in the website. Data can be retrieved from different tabs or sheets and returned as a response to a request within a JavaScript function.

#### b. Interfaces to external systems or devices

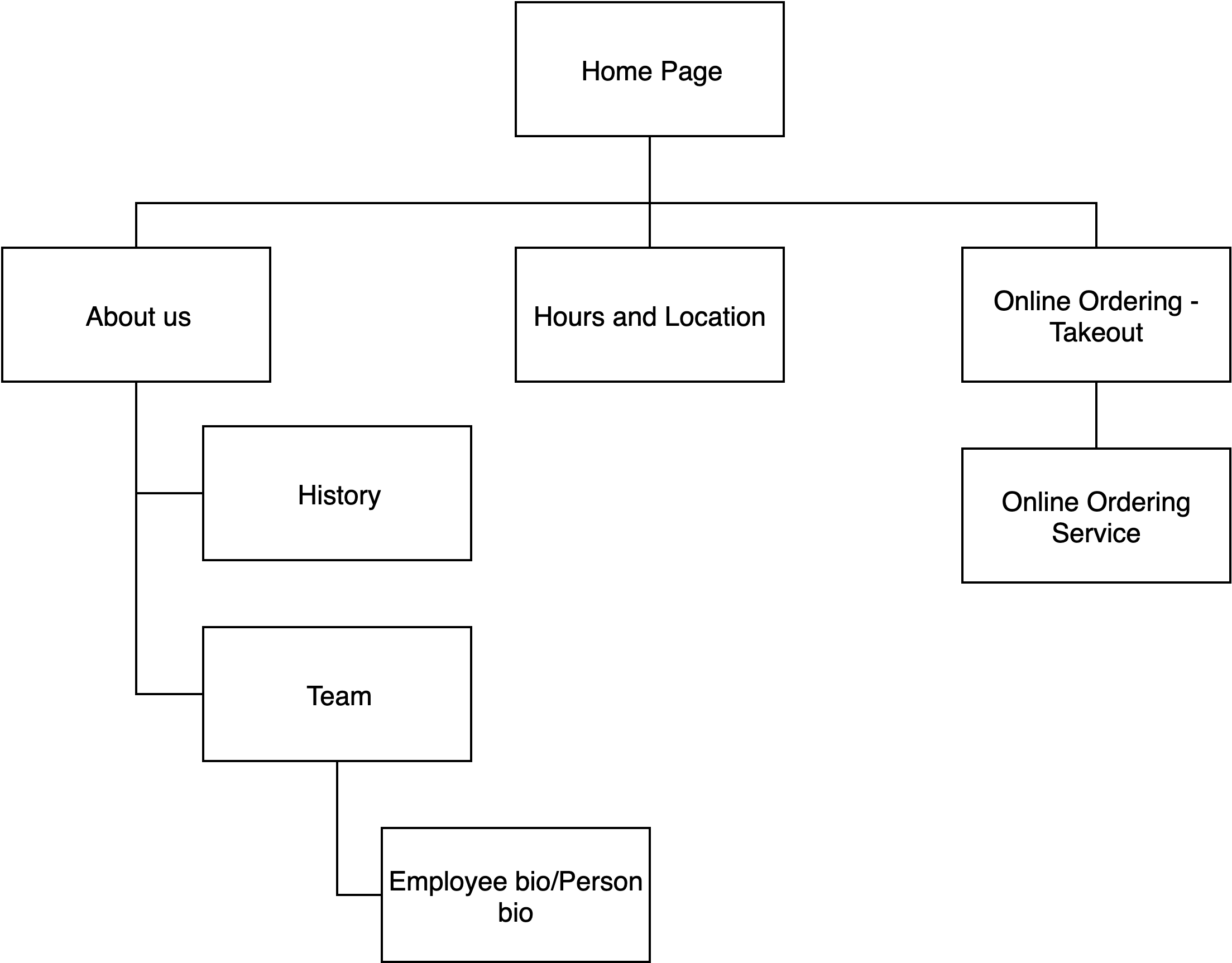
The Hours and Location page can display a Google Map of the restaurant’s location when connected to an external system that responds to requests while using the Google Maps API. The user can also enter data in the form on the Online Takeout Ordering page, which will be converted to JSON and then submitted to an external data system, which will be able to process the data. For this phase of the project, only the website will be developed, not the system external to the website that would be responsible for processing and saving the data, and actually notifying the restaurant staff about the order. The website and an external system receiving the data input by the user should be properly secured so that communications are encrypted, especially since personal information such as the customer name, phone number, and credit card number would be transmitted from the form to the external system.

### 4.4. Internal interface design rules

Internally, navigation links on the navigation bar allow a user to navigate to different pages on the website. Anchors on a page can be used to allow a user to navigate to different sections on a page. For example, on the Home/Main page, when a user clicks on anchor links associated with the different menu categories, Appetizers, Lunch, Dinner, and Cocktails, the user is directed to the section on the page containing the particular menu items. In addition, JavaScript functions are called when a user interacts with elements on the form. For example, on the Online Takeout Ordering page, when a user selects a date from the date dropdown in the *Order Date/Time* section, a JavaScript function is called that populates the times in the time dropdown.

## 5. Procedural Design

The Procedural design shall be in reference to the site map structure of Figure 5.

[](https://www.draw.io/?page-id=IunPgF-ctpZHmw9idTP0&scale=auto#G1Y2wVPo2Ch36SgMAceHqT-9FlS7XtK9aU)

*Figure 5: Site Map of Cafe Chang*

### 5.1. Module descriptions

A module is specified for each of the four pages used in the website. No abstract classes are used in the module descriptions below due to the implementation of website pages for the project. A basic framework consisting of HTML and CSS using open-source technologies is used across all pages. Figure 6 contains a UML diagram representing the components within the HTML pages, JavaScript files, and CSS files used in the website.

**Module Name:** Home/Main Page

* Abstract: N/A
* Implementation: Home/Main Page (Home Page.html) shall contain all information including restaurant text, headings, menu items, and associated links. Styling in Home Page.css shall be applied to the Home/Main Page to create a more appealing design and layout for the client and client’s customers. In addition, styling shall be assisted by Bootstrap for an appealing navigation bar.
* Exports: N/A
* Imports: Data with menu item information from Google spreadsheet
* Pre Conditions: N/A
* Post Conditions: N/A
* Input: N/A
* Output: N/A
* Subparts: Entry page/landing page shall be broken up into several module sections include: Restaurant Name and Home picture, menu icons, Appetizers, Lunch, Dinner, and Drinks.
* Side Effects: None
* Miscellany: N/A

**Module Name:** About Us Page

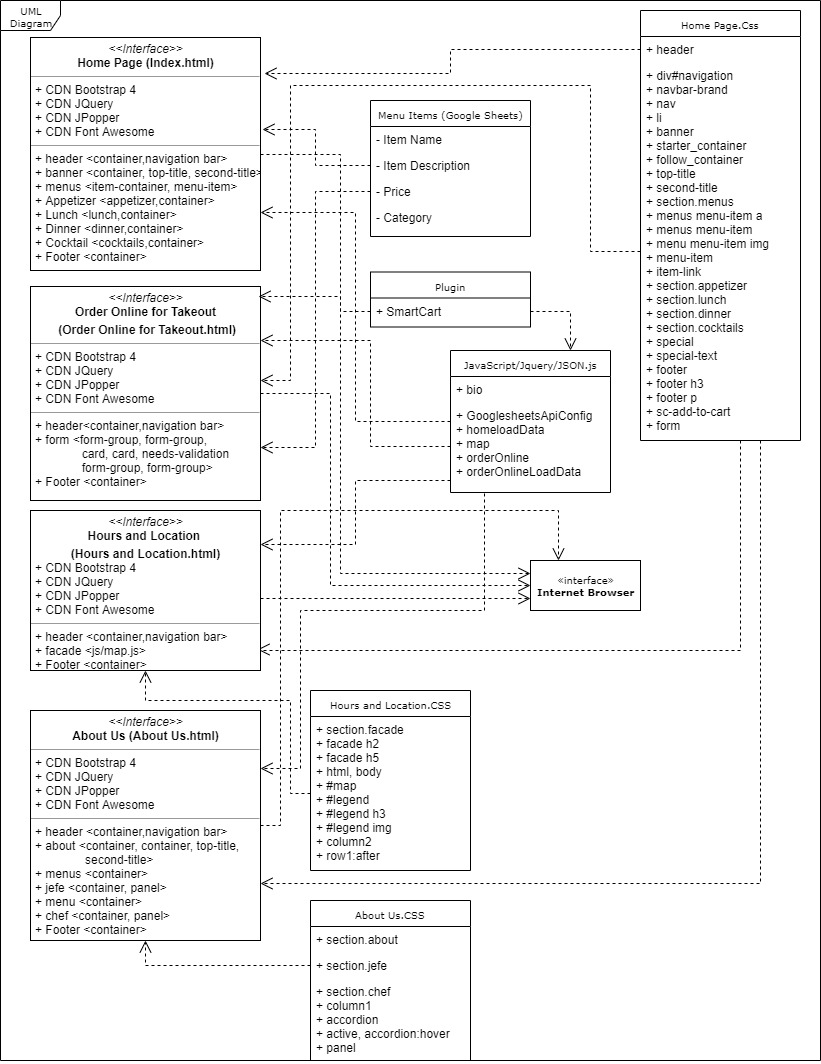
* Abstract: N/A
* Implementation: The code is contained in one HTML page to display layout utilizing CSS pages to help with stylisation and dynamic responsiveness. There will also be a JavaScript file in use to instantiate the “bio” accordion buttons.
* Exports: N/A
* Imports: N/A
* Pre Conditions: N/A
* Post Conditions: N/A
* Input: N/A
* Output: N/A
* Subparts: The subparts include navigation bar, the opening about us section, the bio navigation bar, founder section, chef section, and bottom footer with standard logo, location, and hours.
* Error Handling: N/A
* Side Effects: None
* Miscellany: N/A

**Module Name:** Hours and Locations Page

* Abstract: N/A
* Implementation: The code is contained in one HTML page to display the layout utilizing CSS pages to help with stylisation and dynamic responsiveness. There will also be a JavaScript file in use to instantiate the Google Map and populate it with the appropriate icons.
* Exports: Navigational coordinates of locations sent to Google Maps
* Imports: Google Map API for instantiating map on page
* Pre Conditions: N/A
* Post Conditions: N/A
* Input: N/A
* Output: N/A
* Subparts: Sections of the location and hours page include: restaurant name and facade picture, map application, and bottom footer with standard logo, location, and hours.
* Error Handling: N/A
* Side Effects: N/A
* Miscellany: N/A

**Module Name:** Online Takeout Ordering Page

* Abstract: N/A
* Implementation: The code is contained in one HTML page to display the layout and one JavaScript file containing code to dynamically update the layout on the page, such as populating the dropdown and data on the page. A separate CSS file is used to style the elements on the page.
* Exports: Data submitted from form to an external service (not included in this website)
* Imports: Data with menu item information from Google spreadsheet
* Pre Conditions: The user has completed all required fields of the online takeout form.
* Post Conditions: The data entered in the online takeout form is submitted to an external service and the user is redirected to the Home/Main page.
* Input: The user enters values for the fields in the three sections of the form, including order date and time, items and quantity added to the takeout order, name, phone number, and credit card number.
* Output: The user receives a response on the online takeout page indicating that the form data was submitted successfully or there was an error processing the form.
* Subparts: Subsections of the form include: Order Date/Time, Add Items to Cart, and Checkout.
* Error Handling: Validation of the elements will be performed on the online takeout ordering form, including checking that all required elements are completed prior to form submission, values are input based on the expected format (e.g., number of digits), and menu items are added to the cart so that an order can be placed.
* Side Effects: None
* Miscellany: N/A



*Figure 6. UML diagram for website components*