***App-etizers at SC***

***Detailed Design***

**Version 1.0**

**Prepared by:**

**Yuming Fei, Renhe Song, Xizhe Ma, Yuting Gao, Peng Gan, Marika Perlmutter**

**CSCI 201: Principles of Software Development**

**Introduction**

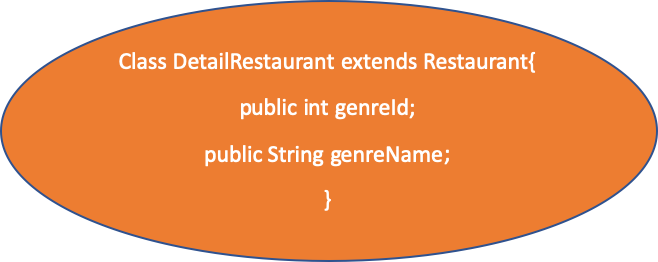
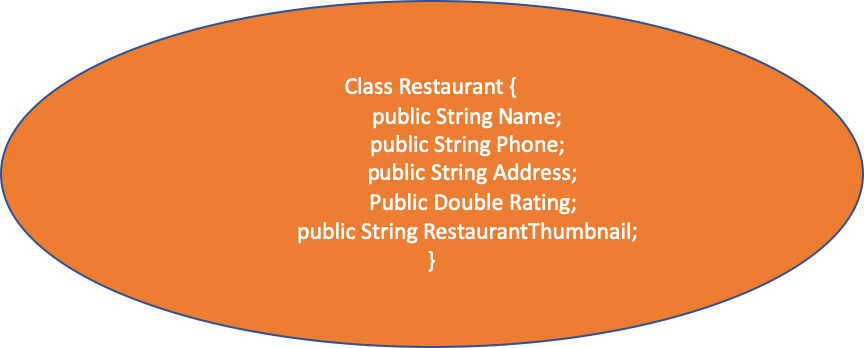
This *Detailed Design Document* for App-etizer at SC provides the complete description of the system. This document should conform to all the rules specified by the *High Level Requirement* and *Technical Specification.* All the system and document changes will be recorded in the track changes under *High Level Requirements.*

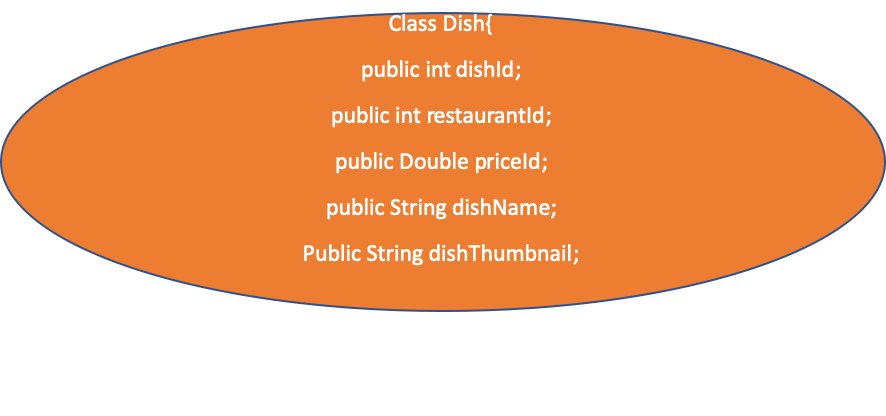
**Design Goal**

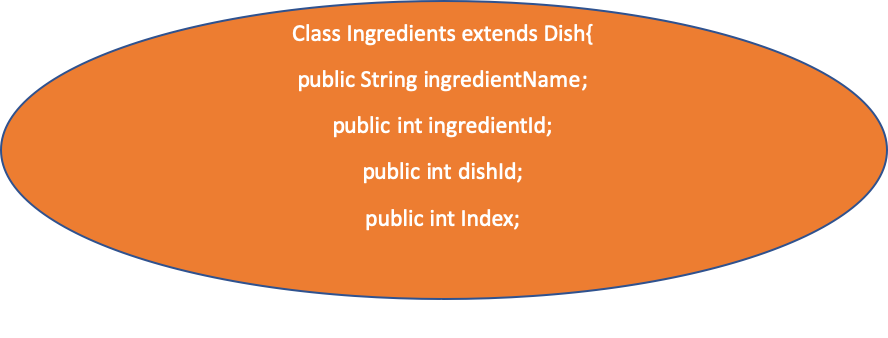
Our main purpose is to make the system bug-free and responsive to ensure a smooth user experience. Therefore, usability is our primary goal.

In addition, our web app serves as a tool for people close to campus to make dining choices. A related goal is to design the website in a way that would make people feel comfortable choosing their food. This would be accomplished by selecting appetizing images for the main and additional pages.

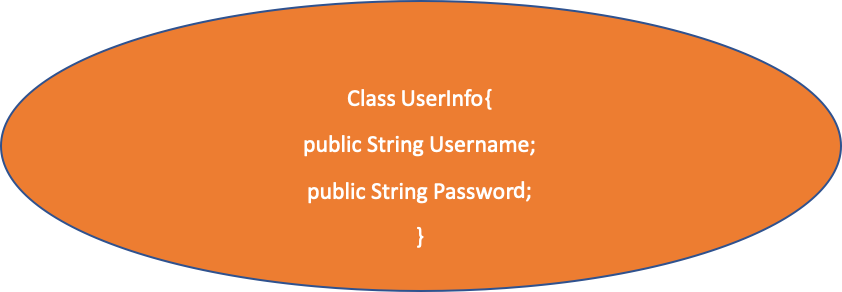
**Class Inheritance:**

* Restaurant class is the parent class.
* Restaurants with different genres such as Chinese, Mexican, bar, and fast food will inherit the parent class.
* Dish class is the parent class.
* The ingredients of the dish is the child class. For example: vegetables, beef, etc.





* User Info is the parent class.
* The logged user class extends from the User Info class, and it can see the review that the user has added.





**Hardware and Software Requirement**

* Hardware: A working PC or laptop with network connection.
* Software: Wifi-connection, chrome or any browser to open the web page.

**Search Page**

* The search page should look like this page.

****

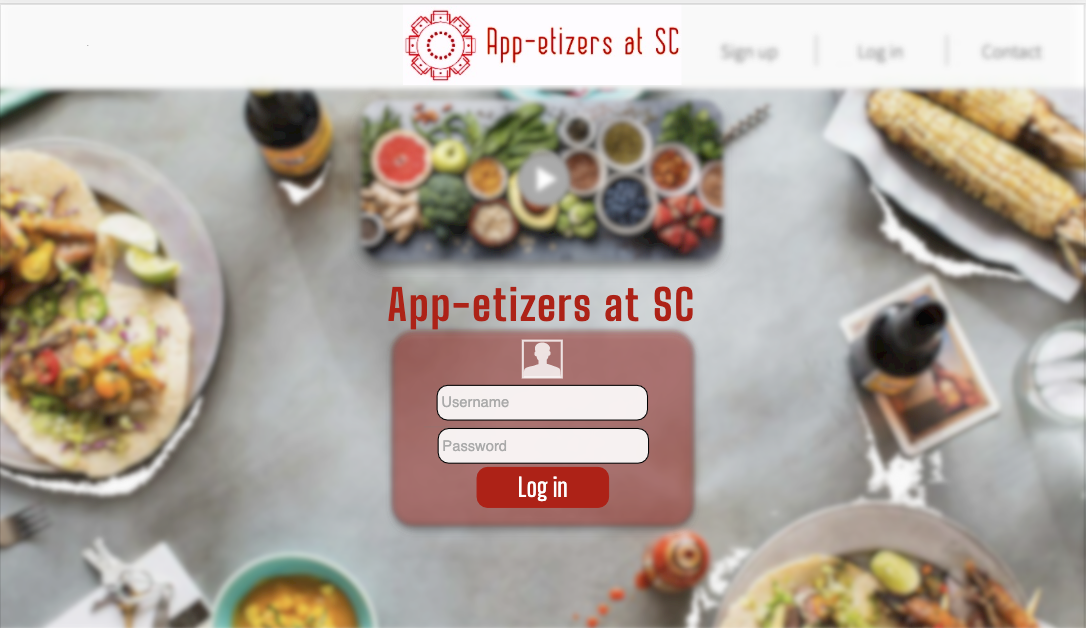
* The top section is implemented by the video section, which plays a random video about foods.
* The background will be implemented by a slide show which contains food pictures.
* When the user presses ENTER after typing their entry in the search bar, it will submit the form with value of the user’s input and send the radio button choice to the servlet called Search. The servlet will use request.getParameter(<keyword’s name>). It will then use JDBC: MySQL to access the tables Restaurant Information and Dishes Information to see options that match the keywords. If there is no result found, it will use PrintWriter to generate an error message, “No Result Found!” to the front-end. Otherwise, the servlet will use the RequestDispatcher.forward method to direct the user to the Restaurant Page or Dish Page.
* The login section is on the upper right corner of the page. On clicking the login button, the login section will pop up a small window.
* There are two default keyword buttons, Ingredient and Restaurant, below the input text field. There is also a button that is a dropdown list with keywords such as price, location, and rating, for user to choose. Once the keyword is chosen, HTML DOM will be used to generate a new button which is inline with the Ingredient and Restaurant buttons.

**Video Section**

* The video section will be located at the top of the search page. It will use the HTML5 <video><source></source></video> tag to generate a video. The video tag will have the autoplay attribute.
* If the user’s browser does not support the HTML video, an error message, “Your browser does not support the video tag” will be displayed.
* There will also be a slideshow of different foods as the background of the search page.
* There will be different HTML <div> tags for storing different food images.
* The slide show will use Javascript. There are three functions called plusSlides(slideIndex), currentSlides(slideIndex), and showSlide(slideIndex) to switch to different <div> tags of food images. The method setTimeout(showSlides, 2000) will be used to automatically change pictures every 2 seconds.

**Login and Register Section**

* The login and register section should look like this page.

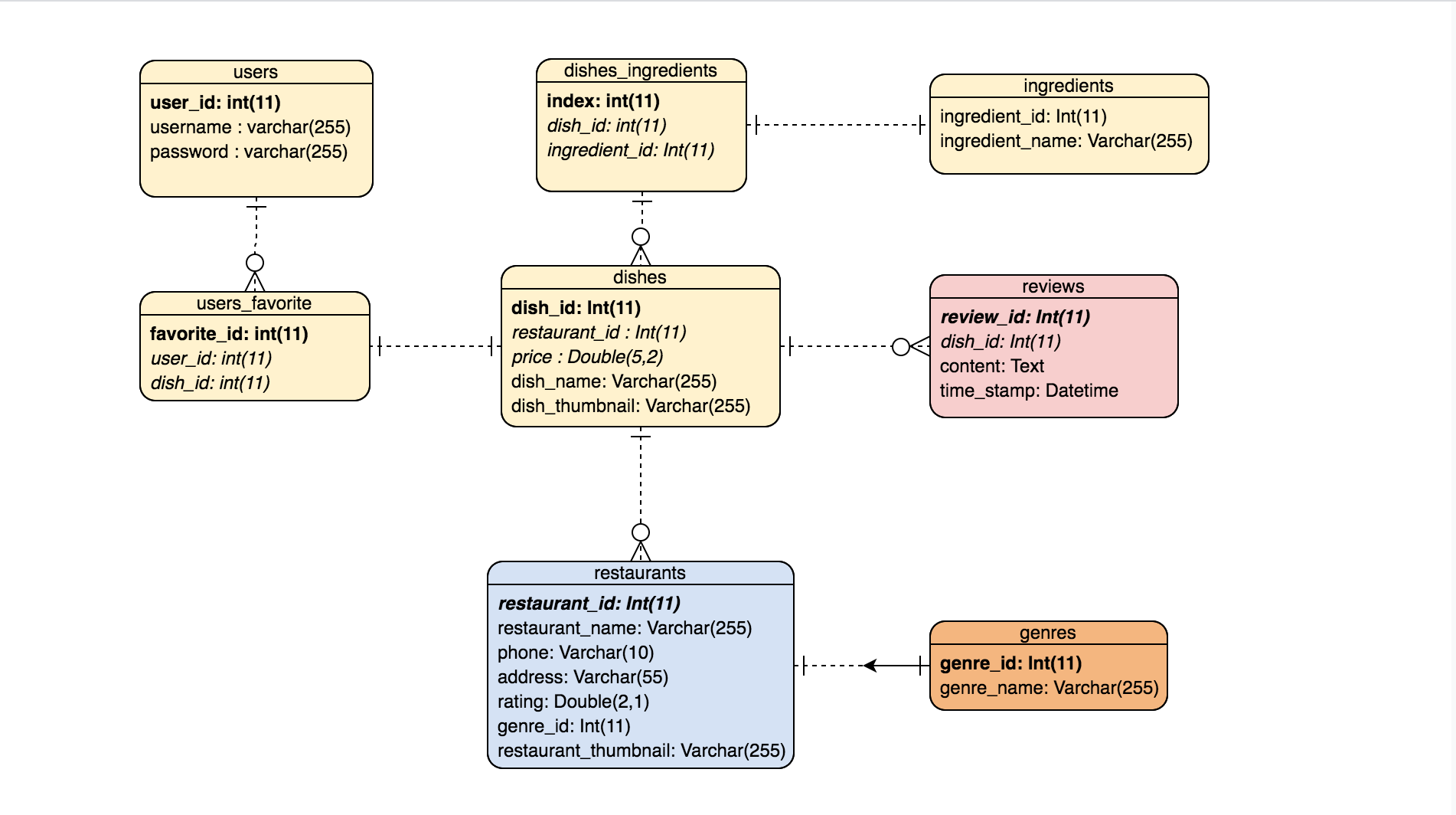


* On clicking the login button, the web page will validate the entered username and password first by using XMLHttpRequest(AJAX) to send data to the servlet called Validate. The Validate servlet will get data and use JDBC:MySQL to access Users table in the database to see if the password matches. If the username does not exist, the Validate servlet will use Printwriter to display the error message, “This username does not exist! Please Register.”
* In the register section, there will be three input text fields: Username, Password and Confirm Password.
* On clicking the register button, the web page will validate the username, password, and password confirmation by using XMLHttpRequest to send data to the servlet called ValidateRegister. The ValidateRegister servlet will get data and use JDBC:MySQL to access the Users table in the database to see if the username already exists. If the username does exist, the ValidateRegister servlet will use Printwriter to display the error message, “This username already exists! Please change to another username.” If the confirm password does not match the password, the ValidateRegister servlet will use Printwriter to display the error message, “The confirm password does not match the password! Please confirm your password.”
* Once the user successfully logs in or registers, the login and the register button will become a profile icon and sign out button, respectively. The profile icon is linked to the User Information Page. The sign out button will end the current cookie and the login and register button will show up again.



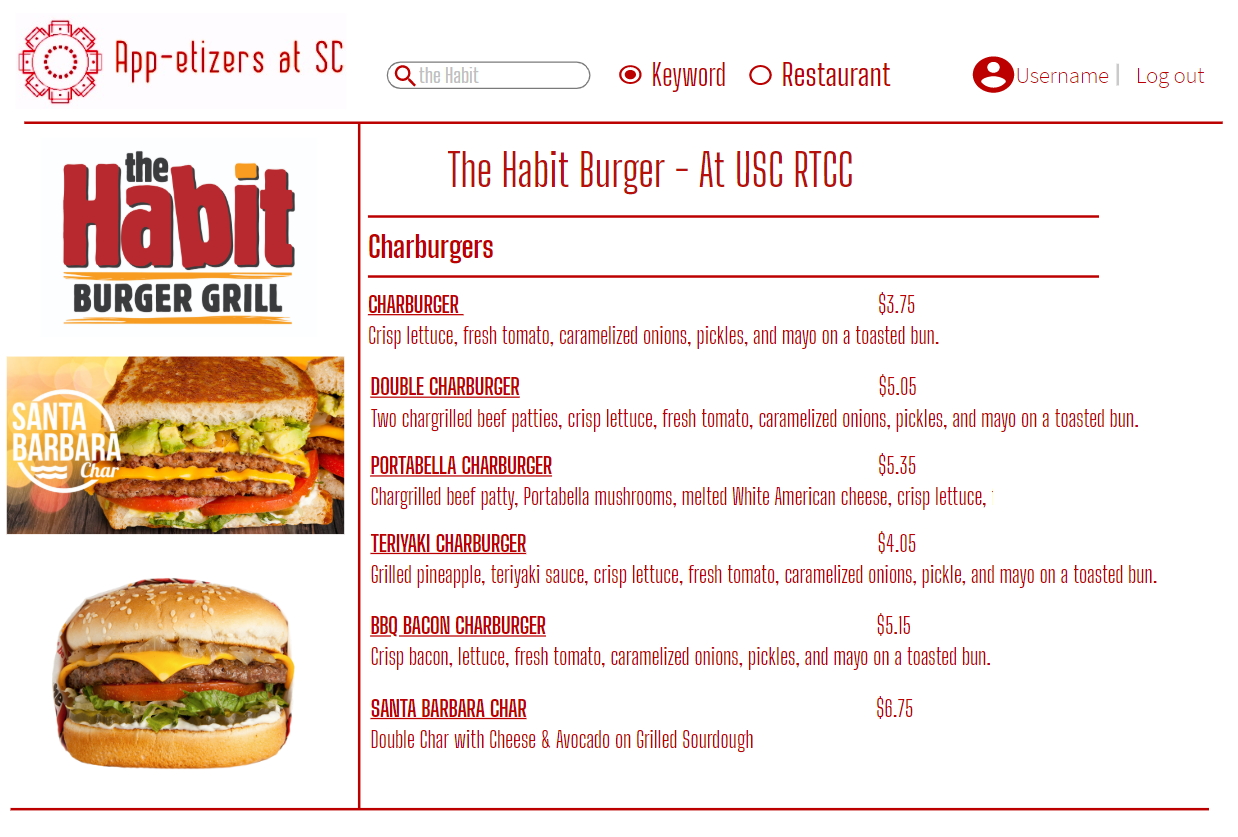
**Database Schema**

* Bolded font stands for primary key
* Italicized font stands for foreign key

****

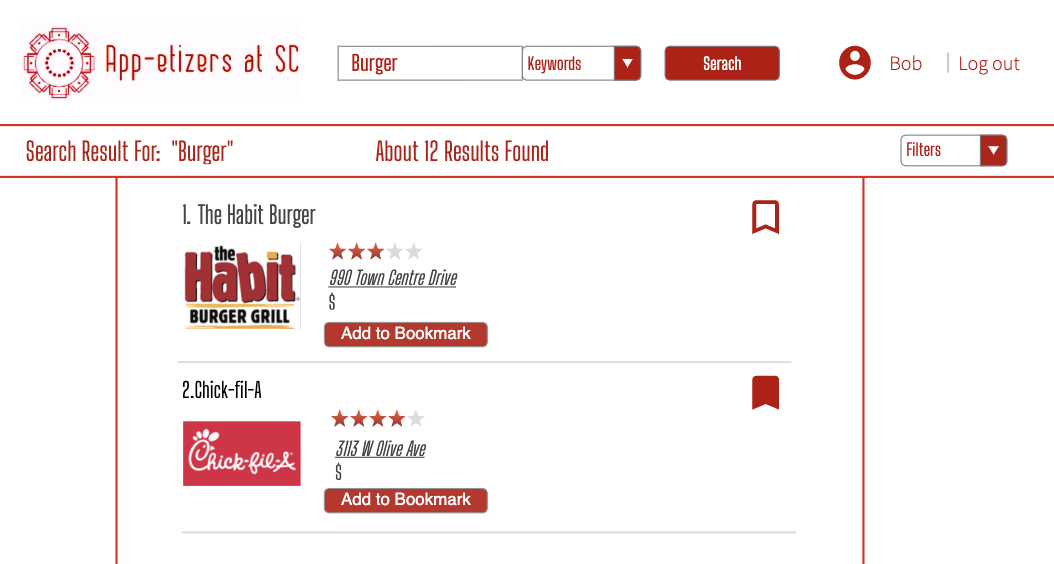
* This design is influenced by our initial plan for a program to search based on ingredients of culinary dishes. Therefore, every dish is an entity that users can search for. This entity includes prices, restaurant name, other ingredients in the dish and review section, etc. Also, there is a functionality for users to add that dish to their Favorites in order to view them later when signed in. However, since users need the restaurant information to locate and enjoy the dishes they are searching for, the majority of the data that users will see is stored in the restaurant table for the simplicity of our database.

**Detail Page**

****

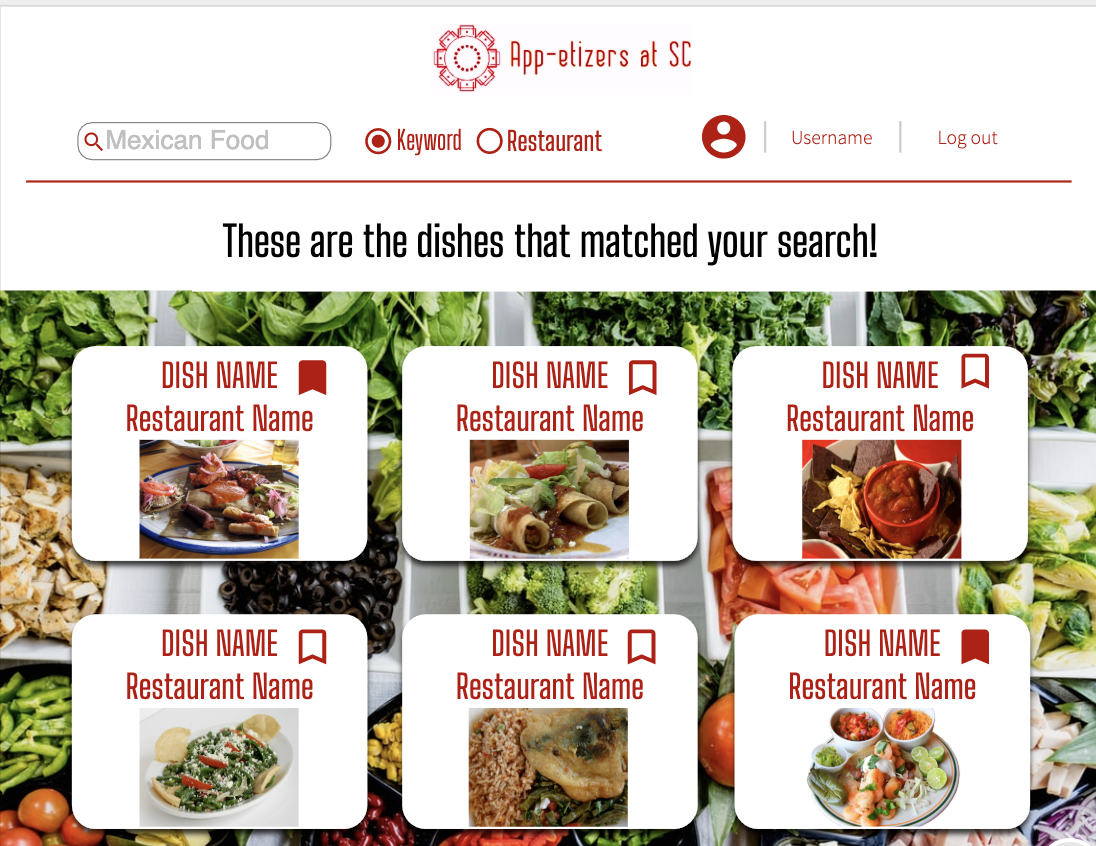
* The menu of the restaurants shows when the user clicks on the selected restaurant, the entire menu will be shown, and the user might have to scroll down to review more options available
* There will be the dish name and a detailed description of the dish, and the price of it. Both guest users or login user could see the information presented in the menu
* The top part of the website is still our significant icon and the search bar, plus it shows whether the user had logged in or not
* There will be few significant pictures of the dishes shown on the left, and if users are interested in the comments and reviews of the dishes, they could click the dishes, and it will prompt them to other pages to read the comments.

**Restaurant Page**



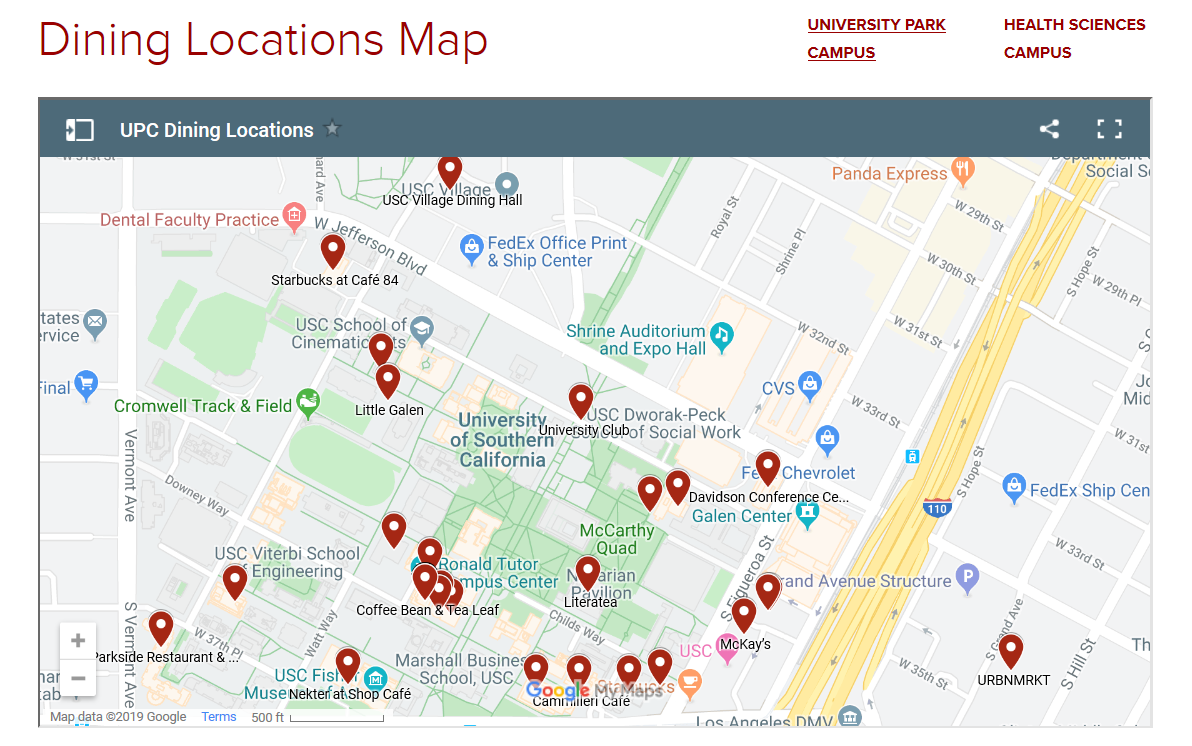
* The restaurant page is the result of keywords input from the search page, if the users are searching by restaurants rather than by dishes..
* On the top of the page, there is an input text field which allows users to type their preference of restaurants.
* The “Add to Bookmark” button will allow user to add the specific restaurant to its favorite list without loading the page. On clicking the button, the page will submit the information of the restaurant to the servlet called Favorite, and the Favorite servlet will use JDBC: mySQL to access the table users\_favorite. Then, it will use insert query to put the information of the restaurant into the table. The page will also use one javascript function called checkcookie() to see whether the user is logged in. If the user’s session expires, it will use PrintWriter to prompt the user “You have to log in first”. Once the restaurant is successfully added to the table, the page will use HTML DOM to prompt the user “You successfully added it !”
* The button will display “Remove from the favorite” if this restaurant is already in the user’s favorite list. And clicking this button uses JDBC:mySQL remove query to remove the information from the table.
* The button next to the restaurant’s image will be generated using javascript function Button(). It will first use AJAX to check whether the restaurant already exists, and then it will use HTML DOM to set value and onclick attributes.
* The image is clickable and directing users to the Detail pages correspond to the Detail Page section above.
* There will also be ratings next to each restaurant for users to reference, the small dollar sign also indicates whether it is a more expensive or regular price restaurant.
* The top part of the program resonates our logo, and also has the status of users logged in or not.
* Filters are able to sort the result restaurant lists by user preference. For example, it will be sorted based on user rating by default. On the dropdown menu, there will be four options. 1. From lowest $ to highest. 2. From highest $ to lowest. 3. Highest Rating. 4. A-Z

**Dish Page**

****

* The dish page is the result of a keyword input from the search page, if the user is searching by dishes rather than by restaurants.
* The header is the same as other pages within the program, with the ability to see the user’s profile or log out (if logged in), select if the user is searching by keyword or restaurant, and input another keyword to search.
* The dish page displays a selection of dishes that match the search, along with the restaurants that offer them. The name of the dish is above each section, with the name of the restaurant (which is a link to its page) beneath it. Below is a picture of the dish, if available.
* When visiting a dish page, users can select the name of a restaurant result to be taken to its specific detail page. They can also search for a different keyword in the top search bar.

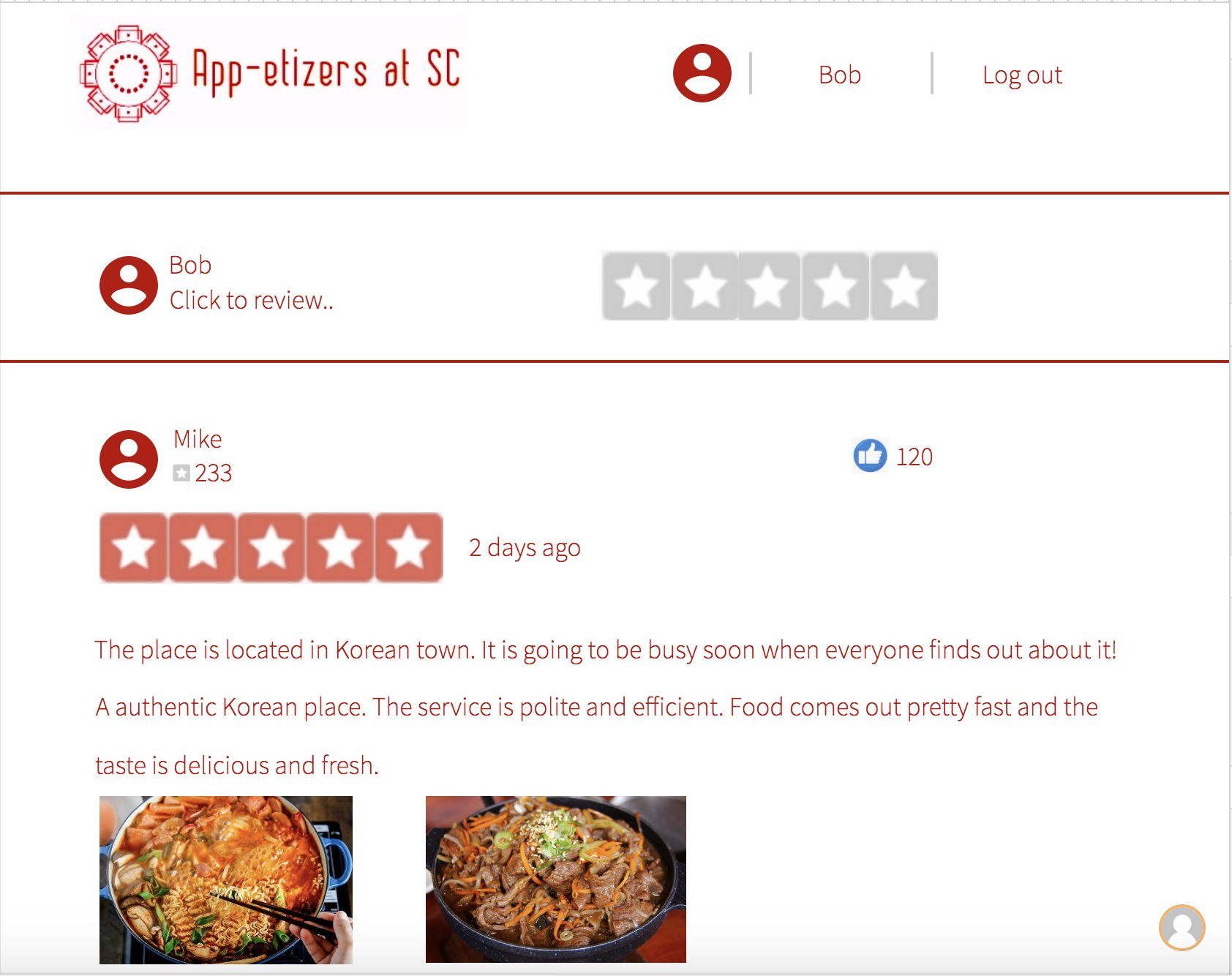
**Map Section**



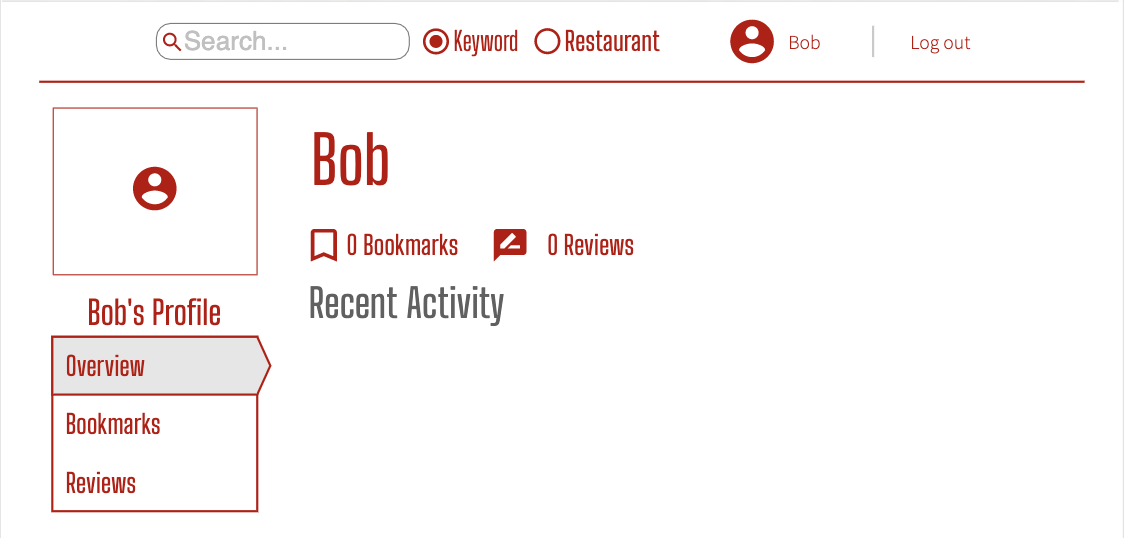
* The map will display a static background image of USC campus map. The map lists various major streets near USC and displays visual landmarks (e.g. USC building outlines) to help users better understand its layout.
* Restaurant location icons will be displayed in an overlay on the map at their respective approximate positions. These icons each list the name of the restaurant below them.
* This combined image of the USC map and its local restaurants will be displayed using a combination of HTML and CSS styling and positioning. The map will take up the majority of the page, both to facilitate readability and simplicity of design.
* When clicked, an icon will redirect the user to the detail page of that associated restaurant.

**Comment and Review section:**

* The review page shows the user reviews about the restaurant with/without pictures. The reviews are stored in the Reviews database, which includes timestamps of the review, which dish(es) the review relates to, and the content of the review.
* The page displays the program’s logo and the content in the same style as the other pages, using HTML and CSS.
* Upon clicking on the ratings stars, the page will display a textbox and a place for the user to input images using Ajax so that the page is not reloaded. Upon submitting, the page would add the comment to both the database and the web page. The page displays other users' comments, with the timestamp at which they were submitted. At the top-right corner, the user can “like” a comment, and the page will add one “like” to the review using innerHTML.

****

**User information page**

* The user information page should look like the following screenshot.****
* The username will be shown at the center of the page.
* The total number of comments and reviews of the user will be presented on the page below the username.
* The user can choose to see an overview of their recent activities, their bookmarks of collected restaurants or dishes, and past reviews they commented, by clicking on the list on the left.
* The user can still type in the search bar on the top to search for food or restaurants.