Requirements of UW Foodie

1 Purpose of the product

Studying and working at the University of Waterloo consumes vast amount of energy. It's necessary and sometimes urgent for students and faculty to find the nearest tasty food that's available in an effective and efficient way. A web app that collects the food store info on campus and provides the search function can make eating at the University of Waterloo more enjoyable.

2 Customers and Stakeholders

The users of UW Foodie are mainly the students, faculty and staff of the University of Waterloo. Anyone who visits the campus and wants to have a taste of UW is also our temporal user. The web app will be free for the users.

Food stores and food service providers and even the University of Waterloo are the stakeholders of this product. Reviews of a specific food store may impact its popularity in the future and a specific meal can be the top choice, which could possibly influence how the food service providers operate. After seeing the feedback from the students, faculty and staff, the University of Waterloo might have a new perspective to choose the food suppliers or make some adjustment based on these comments.

3 Mandated Constraints

3.1 Solution constraints

The web app is only intended for users to search for the food stores within the campus. Due to lack of data accessibility, other food stores near the campus would currently not be available. All food services information is taken from the uWaterloo food services API, if the food store has updated its menu or changed its open hours which are not synced to this API, then the info users get might be outdated.

In addition, the search function can search for the food store based on its building and opening hours. A geographical distance calculation based on GPS coordinates would not be performed.

3.2 Implementation and User Environment

The front-end of this web app will be implemented using Sublime with html5, CSS and JavaScript. The back-end will be built on springboot.

As UW foodie is a web app, users can get access to our product via Internet with any web browser on any device.

3.3 Off the shelf packages

The information of food stores is available from uWaterloo food services API. Data extracted from this API is used to build the database for our product.

Restaurants search and review app like Yelp and OpenTable provided model templates for the interface and webpage design. They also showed possible ways of interaction between users and a food store search app.

3.4 The length of the project

The project must be completed within two months. The deadline to deliver this product is **March 28.**

4 Scope of the Product

Our product is only intended for use within the University of Waterloo campus. The food stores that appear on the search results would be only those are available from the UW food services API. Also, the search function our product provides is based on the pre-defined related buildings. It would not track the GPS coordinates of the user and calculates the geographically distance between the food store and the user. Its main purpose is to provide a tool for the user to check the availability of the food stores nearby and have a quick screen for them to decide what to eat. The product aims to focus on satisfying the users' core needs instead of the variety of needs that a user may come up.

As this tool runs on a webpage, it's not necessary to deal with the specific platform specification, whether it be iOS or Android. This would greatly simplify and facilitate our development process.

5 Functional and Data Requirements

5.1 Functional Requirements

1) User Story 1: Search. As a user, I would like to search for the available food store at the current time or any specified time. I want to see a list of food services on campus near my selected building.
2) User Story 2: Sort. As a user, I would like to sort the search result, which is a food store list, based on ratings and building location.
3) User Story 3: Check store detail info. As a user, I am willing to check the menu and other detailed info of a food store.
4) User Story 4: Review. As a user, I definitely want to give ratings to and comment on the food store. Also, other users should be able to see these reviews.
5) User Story 5: View Map. As a user, I would like to check the food store locations.

5.2 Data Requirement

The functions our product provide depend on the data we have stored in the database. To achieve the functional requirements, we should include store information in the following categories.

1) Name of the food store
2) Location of the store, specifically the building where the store resides
3) Open hours, including seven days a week and special holidays
4) Menu

The data can be accessed from the uWaterloo food services API. The data used in our product would be stored in MySQL format on the backend and transmitted to the front end in json files.

6 Look and Feel Requirements

6.1 The interface

The interface facing the user should include a list of food stores on campus and dropdown menus for the users to specific the time they want to eat and where they want to find the food store. Labels should be placed on top of the dropdown menus so that the users could easily figure out what they are used for. The dropdown menus would be at the center of the interface and their sizes should be larger than 50*300 px. Pictures used in this product should be not less than 150 ppi. The background images used in any webpages should have resolutions larger than 600*500 px.

6.2 The style of the product

As a general rule of thumb, the web app is intended for finding an available food store nearby efficiently, any redundant design elements should be avoided. In terms of color design, the general style should match the typical color used in the uWaterloo websites, namely yellow, purple and black.

The overall design of the product should be clean and tidy so that the users could find the information they want within 10 seconds. Also, the information loaded on each webpage should be organized in order. The search results and reviews displayed to the users should not contain more than 10 items in one page.

7 Usability Requirement

1) The product should provide the users with the requested info within at most four clicks/inputs from any page.
2) The search, sort and other command buttons should be labelled or displayed in a specific shape that any user could easily understand.
3) No error should occur due to a user’s normal operation within the product’s capacity load.

4) For any user, it should take less than 3 minutes (preferably 1 minute) to get familiar of how to use this product.

8 Performance Requirement

8.1 Speed requirements

All web pages should load up within 1 second within the capacity. The search and sorting function should return and display result within 2 seconds.

8.2 Precision requirement

The info displayed on the web pages should be in accordance with the latest food services available and thus should be updated on a weekly basis. In addition to ensuring the database store the accurate info, this product should be able to return the requested info based on the inputs provided by the users.

8.3 Reliability and Availability requirements

Users would effectively and efficiently find the info they request. No errors should occur with normal user operations within capacity. The service should be available ideally 24/7. Taking maintenance into account, the product would be unavailable for at most 3 hours per week.

8.4 Capacity requirements

The current capacity of the product is determined at most 200 requests per minute at peak hours. Depending on the popularity of this product after delivery, modification to the product design may be implemented to satisfy a possible increasing capacity.

9 Security Requirements

The data security is vital to the meet the usability and performance requirement. Our product should protect the database from being modified by any unauthorized attempts. The interface and webpage design should also be protected from modification by any unauthorized users.
In addition, the identity of the reviewers should not be revealed to any other user or food store to ensure the privacy of the users.

10 Procedure and Development phases

Evolutionary model will be employed to guide the development process of this product. Development phases will be divided by the three major functional requirements, which may be modified during the development process. The general procedure is shown below.

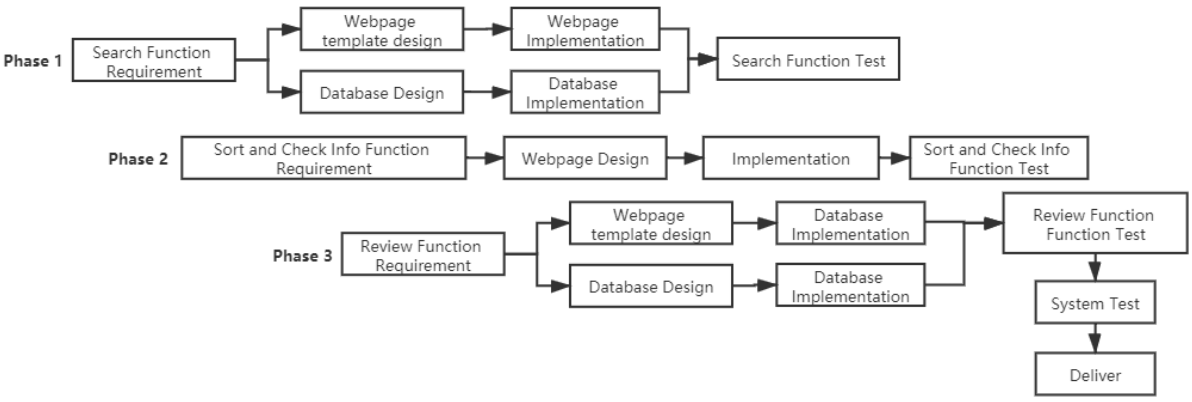

Figure 1. Development Model

11 Risks

Risks related to our products exist in every step of our development process.

1) Due to inadequate survey of our potential users, we might end up building the unnecessary functions or not being able to satisfy their specific needs.
2) An improper way of system breakdown might lead to difficulty of integrating the modules as a whole. The interaction between the user and the interface should be analyzed in a detailed and comprehensive manner. Therefore, the function of each module could be defined clearly.
3) Technical issues may pop up when implementation starts. Hopefully deadline would result in the ultimate productivity to overcome the difficulties.
4) Test suite may not be well designed to cover all the user cases. This may end up in a malfunctional system with not detected bugs.