Penn State Food Locator

Andrew Kyffin, Jacob Smith, Luke Leiter

Software Requirements Specification Document

Version: 2.0 **Date: (12/14/2020)**

Table of Contents

1 Purpose

Penn State University is a large school that encompasses thousands of people, physical buildings, and software networks. They must improve to stay competitive among other schools. Improvements in the past 20 years have begun to lean on new software more and more. Some of this software deals with quality of life improvements for students. Specifically, the Penn State Go app that helps students discover new options for eating.

1.1 Business Background

Penn State is a diverse university that teaches tens of thousands of people each year. To stay competitive among other colleges they must produce software that helps students in their daily lives. While the university already offers on campus dining options in addition to on campus restaurants, students often want to change what they eat day in day out. They've attempted to give students access to a wider variety of food options with the Penn State Go.

Penn State Go is an app recently developed for Penn State college students for assistance around campus. Adding a feature to find nearby restaurants would be vital to incoming freshman who the app is targeted towards. Finding quality restaurants nearby that serve what you want is always difficult, we believe that adding this feature to the colleges own phone app would make everyone's lives a bit easier.

1.2 Needs

The existing Penn State Go app features a section called "Penn State Eats" this section shows campus owned dining halls and allows you to mobile order food. Limiting the Penn State Eats section to only campus owned locations heavily reduces the number of options available to its users. This system can be improved upon by adding more to the searching feature, such as searching local restaurants. Search options could include the name of the restaurant or the type of eatery one desires.

2 Scope

With this addition to the Penn State Go, we plan to achieve certain goals that allow us to find a variety of off-campus restaurants. The Penn State Food Locator will improve student quality of life. This is done adding different features that are specified under software requirements.

2.1 Goals

While the Penn State Go app allows one to mobile order from in-campus eateries, there is a clear lack of variety on most campuses. To remedy this, the Penn State Go app needs to add more off-campus options. Doing this would require the app to get a means of locating and separating restaurants based on a user's needs, such as the existence of carry out or delivery, and category of food. The objective of The Penn State Food Locator is to create an app that assists people in finding restaurants off campus to dine at. It will be available as a web application initially. There will also be a basic search feature with filters for broad restaurant searches.

2.2 Software Requirements

- Get information about nearby restaurants
    - Type of food
    - directions
    - delivery/dine in options
- Allow users to search through restaurant information
    - allow users to modify the search using in app filters

3 User characteristics

While this app is very beneficial, most people would not need this because they do not go to Penn State campuses. Therefore, we can target specifically Penn State students. More so, we can provide the most beneficial service to new residential students who are not aware of the options in from of them.

3.1 Key users

Penn State Students, specifically freshman and first year transfers. These users will be using this product to find local restaurants to eat at and help them know what kind of menu these restaurants offer to aid in the selection process. These users understand what general descriptors for restaurants are, therefore they would be rated as Novice for this product. These users have little experience with local locations, but can deduce what they sell. I would however rate them as Masters of similar technology. Due to wide range of key users we will consider multiple language options.

3.2 Secondary users

Upperclassmen, faculty. These users will be using this product to find local restaurants to eat at and also help them know what kind of menu these restaurants offer to aid in the selection process. These users understand what general descriptors for restaurants are, therefore they would be rated as Journeymen for this product. These users have significantly more experience with local locations than the key users. I would however rate them as Journeymen of similar technology having used them almost daily, but I would not however say they are as efficient than the key users. Due to wide range of key users we will consider multiple language options and font size options for those with poor eyesight

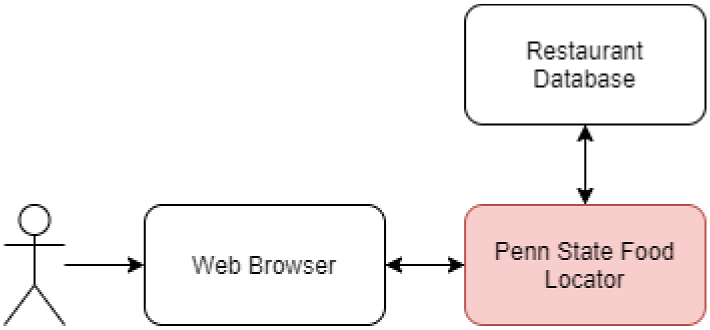
3.3 Unimportant users

Alumni and local residents These users will be using this product to find local restaurants to eat at and also help them know what kind of menu these restaurants offer to aid in the selection process. These users understand what general descriptors for restaurants are, therefore they would be rated as Master for this product. These users have plenty with local locations, and may just want an easy way to see their options. I would however rate them as Journeymen of similar technology having used them almost daily, but I would not however say they are as efficient than the key users or secondary users. Due to wide range of key users we will consider multiple language options and font size for poor eyesight.

4 Product perspective

4.1 System Context

Define the system's relationship to users or other related systems. If the system is an element of a larger system, then identify the interfaces between the system covered by this SRS and the larger system. A block diagram showing the major elements of the larger system, interconnections, and external interfaces can be helpful. Below are two examples block diagrams (with the system to be developed highlighted):



4.2 User interfaces

1) Web Browser
    a. It is required that the system can access the internet.
2) Penn State Food Locator
    a. It is required that the system provides a restaurant's location, open hours, and serving options.
    b. It is required that the system allows for the user to narrow the list of available restaurants based on desired options.
3) Restaurant Database
    a. It is required that the system contains information about a restaurant's serving options, opening hours, and location.

4.3 Software interfaces

- For each required software product, if possible, specify: a) Software name; b) Software specification number; c) Software version number; d) Source.
- Any Updated Web Browser
    - Google Chrome
    - Microsoft Edge

- Internet Explorer

4.4 Hardware interfaces and Memory constraints

N/A

4.5 Deployment requirements

N/A

5 Assumptions and Dependencies

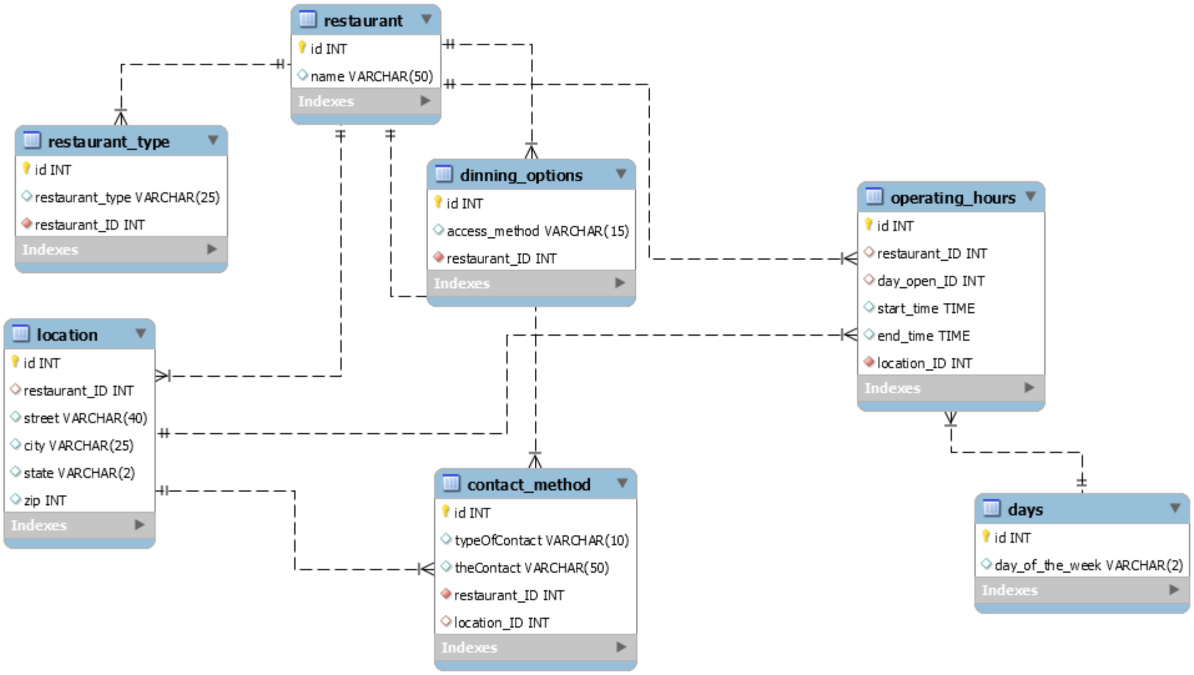
The project is dependent on web browsers and how they run JavaScript.

6 Specific requirements

6.1 System Functional Requirements

1. Sort data into objects created by the application
2. Present the data in a visually appealing way

6.2 Logical Database Requirements



6.3 Software System Attributes

6.3.1 Usability

A computer able to run a web browser is required for the user to run the application. The better the computer can run the web browser, the better the user experience.

6.3.2 Performance

The system shall be able to handle 1 user at once.

90% percent of searches shall be completed in less than 10 seconds.

6.3.3 Reliability/Dependability

Specify the factors required to establish the required reliability/dependability of the software system at time of delivery.

- Internet Access

6.3.4 Security

Specify the requirements to protect the software from accidental or malicious access, modification, or destruction. Specific requirements in this area could include the need to:

1) The application will be used through Penn State Go; they can only access it with their Penn State account which is protected by 2 Factor Authentication.

6.3.5 Maintainability

Specify attributes of software that relate to the ease of maintenance of the software itself. These may include requirements for certain modularity, interfaces, or complexity limitation. Requirements should not be placed here just because they are thought to be good design practices.

- All data will be stored in a database to maintain integrity
- Implement code using only Python and JavaScript languages.