**Software Requirements Specification**

**for**

Munchmaps.io

**Brendon Linthurst**

**Shane Spangenberg**

**CSUF**

**8/28/2019**

**Table of Contents**

**Table of Contents 2**

**Revision History 2**

**1. Introduction 3**

1.1 Purpose 3

1.2 Document Conventions 3

1.3 Intended Audience and Reading Suggestions 3

1.4 Projected Time Frame for Development 3

**2. Overall Description 3**

2.1 Product Perspective 3

2.2 Product Functions 4

2.3 Development Environment 4

2.4 Operating Environment 5

**3. External Interface Requirements 5**

3.1 User Interfaces 5

3.2 Software Interfaces 6

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# **Introduction**

## **Purpose**

*The software requirements for Munchmaps.io are specified in this document. This specification is designed to describe the function of the web application.*

## **Intended Audience and Reading Suggestions**

*This document is intended for developers, project managers, and software testers. We suggest developers and testers read the entirety of the document and that project managers and end users focus attention on core features.*

## **Product Scope**

*The vision of Munchmaps.io is to connect the user to local restaurants by utilizing a familiar swipe mechanic found in popular dating applications. Users are presented with a random “stack” of food images sourced from Yelp user submitted photos and encouraged to swipe until they find their perfect match.*

## Projected Time Frame for Development

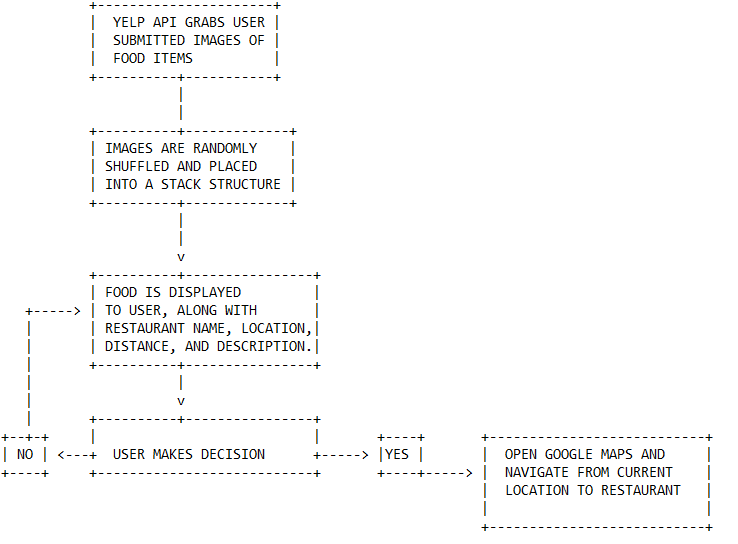
*We expect Munchmaps to be completed by mid December 2019.*

# **Overall Description**

## **Product Perspective**

*Munchmaps.io is a stand alone and self-contained product. It is possible that follow up applications may narrow the specificity of the food, such as a dessert only or bar only app, however it currently is designed to pull random images of all user submitted food photos to serve to the user regardless of meal type.*

## **Product Functions**

**

*Munchmaps.io follows the simple flowchart outlined above.*

1. *Collect user submitted images from an assortment of restaurants listed on Yelp.*
2. *Images are shuffled and inserted into a food object stack of N length.*
3. *Display the item at the top of the stack to the user.*
4. *If user swipes left or clicks NO, pop the top item of food stack and return to step 3.*
5. *When user finally selects YES, use google maps integration to navigate to restaurant.*

## Development Environment

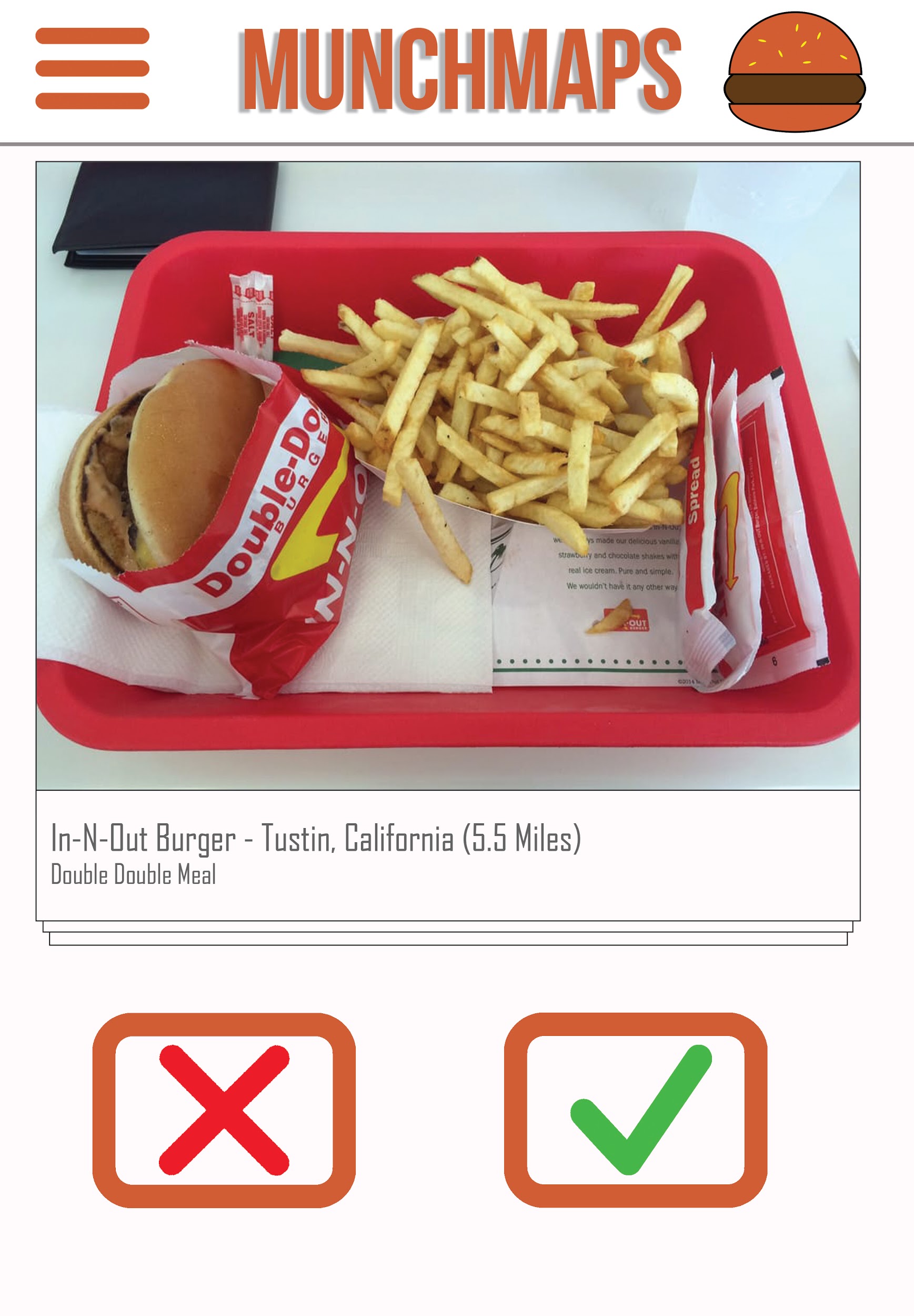
*Various text editors such as, Atom, Visual Studio Code, Git. Development operating systems will include Mac OS X and Linux.*

## **Operating Environment**

*Munchmaps.io is supported by the following browsers, Chrome, Firefox, Safari, Microsoft Edge, Internet Explorer 11. Other browsers may work, but may not be able to take advantage of all the features available. Any operating system able to run one of these browsers will be supported including Windows 7 and up, Mac OS X 10.10 and up, and Linux.*

# **External Interface Requirements**

## **User Interfaces**

**

*The user will be able to interact with a hamburger button to specify the maximum distance they would like restaurants to appear. The main interface will consists of two buttons, an accept button and a reject button. When clicking the accept button the browser will open a new tab mapping to the restaurant location through Google Maps. When clicking the reject button the restaurant currently displayed will be flushed out of the user’s sight and will cycle in a photo of a different restaurant. The user may also click and drag the entire photo to the left end of the screen, which invokes the reject method. Similarly, if they click and drag the photo to the right end of the screen the program will invoke the accept method.*

## **Software Interfaces**

*The programming languages to be used in Munchmaps.io will be JavaScript, Node.js, HTML, and CSS. The program will pull its data from the Yelp Fusion API in order to show photos, distance, and ratings of the restaurants. The Google Maps JavaScript API will also be used in order to create a mapped route from the user’s location to the restaurant.*