# Software Requirements

Specification

for

# Eatables Web Application

Prepared by

Ashwin K.V

Darshan Dinesh M.P

Delbin George

**02-02-2023**

## Table of Contents

[Table of Contents ii](#_TOC_250032)

Revision History

1. [Introduction 1](#_TOC_250031)
   1. [Purpose 1](#_TOC_250030)
   2. [Document Conventions 1](#_TOC_250029)
   3. [Intended Audience and Reading Suggestions 1](#_TOC_250028)
   4. [Product Scope 1](#_TOC_250027)
   5. [References 1](#_TOC_250026)
2. [Overall Description 2](#_TOC_250025)
   1. [Product Perspective 2](#_TOC_250024)
   2. [Product Functions 2](#_TOC_250023)
   3. [User Classes and Characteristics 2](#_TOC_250022)
   4. [Operating Environment 2](#_TOC_250021)
   5. [Assumptions and Dependencies 3](#_TOC_250018)
3. [External Interface Requirements 3](#_TOC_250017)
   1. [User Interfaces 3](#_TOC_250016)
   2. [Hardware Interfaces 3](#_TOC_250015)
   3. [Software Interfaces 3](#_TOC_250014)
   4. [Communications Interfaces 3](#_TOC_250013)
4. [System Features 4](#_TOC_250012)
   1. [System Feature 1 4](#_TOC_250011)
   2. [System Feature 2 (and so on) 4](#_TOC_250010)
5. [Other Nonfunctional Requirements 4](#_TOC_250009)
   1. [Performance Requirements 4](#_TOC_250008)
   2. [Safety Requirements 5](#_TOC_250007)
   3. [Security Requirements 5](#_TOC_250006)
   4. [Software Quality Attributes 5](#_TOC_250005)
   5. [Business Rules 5](#_TOC_250004)

## Introduction

### Purpose

The purpose of the SRS (Software Requirements Specification) document for the Eatables project is to provide a clear, complete, and detailed description of the requirements for the software. The SRS document defines what the software should do and sets the criteria for acceptance of the final product. The SRS document would specify the functional and non-functional requirements of the system, along with the performance and security requirements, user interface specifications, and any other constraints that need to be considered during the development process.

### Document Conventions

Title Page: A title page that includes the project name, date, version number, and names of stakeholders involved in the development process.

Table of Contents: A table of contents that provides an overview of the document structure and the different sections.

Introduction: An introduction that provides background information about the project, its objectives, and its scope.

Overall Description: A section that provides a high-level overview of the software, its functions, and its capabilities.

User Requirements: A section that defines the requirements for the end-users, including functional requirements, performance requirements, and usability requirements.

System Requirements: A section that defines the requirements for the system, including hardware, software, and network requirements.

Non-Functional Requirements: A section that defines the non-functional requirements, including security requirements, reliability requirements, and compatibility requirements.

Constraints: A section that defines any constraints or limitations on the software, such as resource constraints or regulatory requirements.

Assumptions and Dependencies: A section that defines any assumptions made during the requirements analysis and any dependencies on other systems or components.

Glossary: A glossary that defines any technical terms used in the document.

Revision History: A section that tracks the changes made to the SRS document and the reasons for those changes.

### Intended Audience and Reading Suggestions

Project team members: Students who are working on the project, including developers, designers, and testers.

Internal guide: The SRS will be reviewed and evaluated by the internal guide who is overseeing the project.

Peers: Other students in the class may also review the SRS to provide feedback and suggestions.

External guide: They will want to review the document to ensure that the product meets their needs and expectations.

Reading suggestions for the SRS document could include the following:

Review the document in its entirety to get a general understanding of the project and its requirements.

Focus on specific sections that are relevant to readers role on the project, such as the functional requirements or non-functional requirements.

Review the document periodically throughout the project to ensure that the requirements are being met and to make any necessary updates.

### Product Scope

Location-based restaurant search: Users can search for restaurants and food establishments in their area.

Establishment information and reviews: Users can view information about each establishment, including details of the menu items. They can also read reviews from other users and leave their own feedback.

Nutritional Support: The system provides nutritional support to help users make informed choices.

Favorite places and "must-try" list: Users can save their favorite establishments and create a list of must-try restaurants.

### References

IEEE Software Requirements Specification Template:

<https://web.cs.dal.ca/~hawkey/3130/srs_template-ieee.doc>

How to Write a Software Requirements Specification (SRS Document):

<https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document>

6 Steps for Writing an SRS That Works:

https://www.uptech.team/blog/srs-document

## Overall Description

### Product Perspective

The Eatables is a new, self-contained system designed to address the need for a comprehensive solution for managing food choices and maintaining a healthy lifestyle. The platform is not a follow-on member of a product family nor is it a replacement for any existing systems.

The Eatables is a standalone system that does not have any direct interfaces with other systems. However, it does utilize data from other sources, such as restaurant information and user reviews. Additionally, the platform may integrate with other health and wellness tracking tools to provide a more comprehensive solution for managing food choices and overall health.

### Product Functions

* Restaurant search: users can search for food establishments based on their location.
* Establishment information: users can view basic information about each restaurant, such as the address and menu.
* User reviews: users can read reviews and ratings from other users, and leave their own feedback and ratings.
* Nutritional Support: The proposed system provides nutritional support to help users make informed choices.
* Mobile accessibility: The proposed system is designed to be accessible on both desktop and mobile devices, making it easy to use and accessible to a wide range of users.
* User accounts: users can create an account to save their favorite restaurants, review history, and more.
* Social sharing: users can share their experiences and recommendations with friends and followers on social media.

### User Classes and Characteristics

Casual users: casual users are individuals who use the system occasionally to search and review food establishments. Characteristics: infrequent users, less concerned about personalized recommendations, not interested in nutrition support.

Regular users: regular users are individuals who use the system frequently and have an account to save their favorite restaurants, review history, and more. Characteristics: frequent users, interested in personalized recommendations, may use nutrition support occasionally.

Health-conscious users: health-conscious users are individuals who use the system to find healthy food options and to access nutritional information. Characteristics: interested in nutritional information and healthy options.

Foodies: foodies are individuals who use the system to discover new food options, read reviews, and leave their own feedback. Characteristics: passionate about food, interested in discovering new options, active in leaving reviews.

### Operating Environment

Hardware platform: the system should be accessible on both desktop and mobile devices, including laptops, smartphones, and tablets.

Operating system: the system should be compatible with popular operating systems such as Windows, mac OS, iOS, and Android.

Browser compatibility: the system should be accessible via modern web browsers, such as Google Chrome, Mozilla Firefox, Apple Safari, and Microsoft Edge.

Other software components: the system may need to integrate with third-party software components such as mapping APIs to provide a complete and seamless user experience.

### Assumptions and Dependencies

Location data accuracy: The system relies on accurate location data to provide relevant results and recommendations. If the location data is not accurate, the system may provide incorrect results, affecting the user experience.

User reviews: The system is based on user reviews, which may not always be accurate or trustworthy. There is a risk that fake reviews could be posted, affecting the accuracy of the information provided by the system.

Dependency on network connectivity: The system may rely on network connectivity to retrieve data from the server and provide results to the user. If network connectivity is poor or unavailable, the system may not work properly.

## External Interface Requirements

### User Interfaces

The interface should be easy to navigate, with clear and consistent labeling and a logical hierarchy of information.

The system should display ratings and reviews for each restaurant, with the ability for users to leave their own reviews and ratings. The review system should be easy to use and understand, with clear guidelines for leaving reviews and feedback.

The system should include maps and directions to help users find the restaurants they are interested in. Users should be able to view maps with clear markers for each restaurant, and get turn-by-turn directions to their chosen destination.

The system should be designed to be mobile responsive, with a layout that adapts to different screen sizes and resolutions. This will ensure that users can access your system from any device, whether they are using a desktop computer, tablet, or smartphone.

### Hardware Interfaces

Supported device types: The Eatables platform should support a range of device types, including desktop and mobile devices. This will involve optimizing the UI for different screen sizes and orientations, as well as ensuring compatibility with different operating systems (e.g., Windows, macOS, iOS, Android).

Data and control interactions: The data and control interactions between the software and the hardware will primarily involve the software accessing and displaying information from the device's location services, as well as storing and retrieving user data.

### Software Interfaces

Databases: The Eatables platform will require a database to store and retrieve user data, such as reviews, ratings, and restaurant information. The database could be a SQL or NoSQL database, such as MySQL, MongoDB, or Cassandra, depending on the specific requirements of the platform.

Operating systems: The Eatables platform should be compatible with a range of operating systems, including Windows, macOS, iOS, and Android.

Tools and libraries: The Eatables platform may use a range of tools and libraries to support its development and deployment, such as web frameworks (e.g., Ruby on Rails, Django), front-end frameworks (e.g., React, Vue), and libraries for working with databases and location data (e.g., Geolocation API).

Data items and messages: The data items and messages coming into the system will primarily include user-generated data, such as reviews, ratings, and calorie tracking information, as well as location data from the user's device. Data items and messages going out of the system will primarily include restaurant information, recommendations, and search results.

Services needed: The Eatables platform will require a range of services to support its functionality, such as geolocation services, data storage and retrieval services, and user authentication services.

Communications: The nature of communications between the software components will depend on the specific components being used, but may include HTTP and HTTPS requests and responses, direct device-to-device communication, and database queries.

Shared data: Data that will be shared across software components will primarily include user data, such as reviews, ratings, and calorie tracking information, as well as restaurant information. The data sharing mechanism will likely involve database queries and API requests and responses.

### Communications Interfaces

Web browser: The Eatables platform will primarily be accessed through a web browser on desktop and mobile devices. It should support modern browsers such as Google Chrome, Mozilla Firefox, and Safari.

Network server communications protocols: The Eatables platform may use HTTP and HTTPS protocols for communication between the client-side web application and the server-side APIs.

Electronic forms: The Eatables platform may use electronic forms for user authentication, such as sign up and login, and for collecting user-generated data, such as reviews and ratings.

Communication standards: The Eatables platform may use HTTP or HTTPS for communication between the client and server. It may also use the Geolocation API for retrieving location data from the user's device.

Communication security and encryption: To ensure the security of sensitive user data, such as login credentials, the Eatables platform should implement encryption for communication between the client and server using SSL/TLS (Secure Socket Layer/Transport Layer Security).

## System Features

## Geolocation:

## Description:

## This feature uses the device's location to provide users with a list of nearby restaurants. When a user opens the system, it should request permission to access the user's location. If the user grants permission, the system should display a list of restaurants near the user's current location.

## Stimulus/Response Sequence:

User opens the app -> App requests permission to access the user's location -> User grants permission -> App displays a list of nearby restaurants.

## Functional Requirements:

The system should use an accurate and reliable geolocation service to determine the user's location. The system should also allow users to search for restaurants in other locations, such as by entering a city or zip code.

## Restaurant information:

## Description:

This feature displays detailed information about each restaurant, including its name, address, contact information, menu items, and ratings and reviews.

## Stimulus/Response Sequence:

User selects a restaurant -> App displays detailed information about the restaurant.

## Functional Requirements:

The system should display accurate and up-to-date information about each restaurant. The information should be easy to read and understand, with clear labeling and organization.

## Ratings and reviews:

## Description:

This feature allows users to leave ratings and reviews for each restaurant and food items. Users should be able to rate restaurants on various criteria, such as food quality, service, and atmosphere.

## Stimulus/Response Sequence:

User selects a restaurant -> User leaves a rating or review -> App displays the rating or review.

## Functional Requirements:

The system should have a clear and easy-to-use rating and review system.

## Other Nonfunctional Requirements

### Performance Requirements

Speed: The app should be fast and responsive, with pages that load quickly and smoothly. Users expect web pages to load within a few seconds.

### Security Requirements

The web application should validate user input to prevent SQL injection attacks and cross-site scripting (XSS) attacks.

### Software Quality Attributes

Reliability: the app should be available and functional at all times, with minimal downtime or errors. It should also be able to recover quickly in the event of a failure or disruption.