# 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

The Software Requirements Specification (SRS) document is a crucial artifact in the software development process. It serves as a comprehensive guide that captures the functional and non-functional requirements of a software system. The SRS document outlines the purpose, scope, and architecture of the software, providing a clear understanding of what needs to be developed and how it should behave. It serves as a reference for stakeholders, including developers, designers, testers, and clients, ensuring a common understanding of the project goals and facilitating effective communication and collaboration throughout the software development lifecycle. The SRS document acts as a contract between the development team and the stakeholders, guiding the development process and serving as a benchmark for validation and verification of the final product.

### Purpose

The purpose of the project is to develop a comprehensive and user-friendly food discovery platform that allows users to explore, review, and interact with restaurants in their vicinity. The project aims to simplify the process of finding new dining options, providing users with valuable information, user reviews, and the ability to save favorites. By creating a platform that enhances the dining experience and facilitates informed decision-making, the project seeks to improve user satisfaction and promote the culinary exploration of different food options.

### 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

### The platform aims to provide users with a convenient and user-friendly interface to discover nearby restaurants, view essential information such as contact details and location, and read reviews from other users. Users can also leave their own reviews, add food items or restaurants to their favorites list, and upload food vlog videos through the Drops feature. Additionally, the project includes an admin panel for restaurant management, allowing the admin to add, edit, or delete restaurants, manage menu items, and review user requests for adding new restaurants. The project focuses on delivering the core functionalities mentioned while ensuring a seamless user experience and providing a comprehensive solution for food discovery and engagement.

### 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 food discovery platform is designed to operate as a standalone system, independent of any other software or external systems. It serves as a centralized platform for users to explore and engage with restaurants in their vicinity. While the platform interacts with external services such as location services for restaurant discovery, it does not rely on or integrate with any specific external systems for its core functionality.

The platform is developed as a web-based application accessible through web browsers on various devices, including desktops, laptops, tablets, and smartphones. It does not require any specific hardware components or dependencies beyond standard web-browsing capabilities.

As an independent system, EATABLES aims to provide a comprehensive solution for users to discover, review, and interact with restaurants. It does not seek to replace or integrate with existing restaurant management systems or third-party services. However, future integrations with external systems such as online ordering and delivery services can be considered to enhance the user experience and expand the platform's capabilities.

### Product Functions

* User Registration and Authentication: Users can create accounts and securely log in to access the platform's features and personalize their experience.
* Restaurant Discovery: The platform utilizes location services to identify nearby restaurants and provides users with essential information, such as contact details, location, and reviews from other users.
* User Reviews: Users can leave detailed reviews for restaurants they have visited, sharing their experiences and opinions to assist other users in making informed dining decisions.
* Favorites Management: Users can create a personalized list of favorite food items or restaurants, allowing for quick and easy access to their preferred options.
* Drops (Food Vlog Upload): Users have the ability to upload and share food vlog videos, creating an engaging platform for users to showcase their culinary experiences and interact with others.
* Price Filtering: Users can filter search results based on their desired price range, helping them find dining options that align with their budget or affordability preferences.
* Admin Panel: The admin panel provides administrative privileges, allowing the admin to manage restaurants, add new restaurants, edit existing ones, and review user requests for adding new restaurants.

### User Classes and Characteristics

General Users: These are individuals who use the platform to discover and explore restaurants in their vicinity. They may have varying levels of technological expertise and preferences when it comes to dining experiences. General users rely on the platform to provide accurate and relevant restaurant information, user reviews, and an intuitive interface for seamless navigation.

Administrators: Administrators have elevated privileges and are responsible for overseeing the overall functioning of the platform. They have access to the admin panel to review user requests for adding new restaurants, moderate content, and maintain the integrity of the platform. Administrators possess advanced system management skills and a comprehensive understanding of the platform's functionalities.

### 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 to help users find the restaurants they are interested in. Users should be able to view maps with clear markers for each restaurant.

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).

### 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,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.

Shared data: Data that will be shared across software components will primarily include user data, such as reviews, ratings, 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.

## System Features

User Registration and Authentication:

Description:

This feature allows users to create an account and login securely to access the system.

Stimulus/Response Sequence:

User clicks on the "Sign Up" button on the login page

System prompts the user to enter their name, email address, and password

User fills in the details and clicks on the "Submit" button

System verifies the information and creates the user account

User enters their credentials and clicks on the "Login" button

System authenticates the user and grants access to the platform

Functional Requirements:

User account creation

Secure login and authentication process

## 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.

## Restaurant information:

## Description:

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

## Stimulus/Response Sequence:

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

## Functional Requirements:

Location-based restaurant identification

Display of essential restaurant information

Filtering options based on price

## Reviews:

## Description:

This feature allows users to leave reviews for each food items and read reviews posted by others.

## Stimulus/Response Sequence:

User selects a restaurant -> User selects an item -->User leaves a review -> App displays the review.

## Functional Requirements:

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

Favorites Management:

Description: This feature allows users to create a personalized list of favorite food items or restaurants and quickly access them for future reference.

Stimulus/Response Sequence:

User selects a restaurant -> User selects an item --> user add item to their favorites list

User can access their list of favorites from their account page

Functional Requirements:

Option for users to create a personalized list of favorite food items

Quick access to saved favorites for future reference

Drops (Food Vlog Upload):

Description: This feature allows users to upload and share their food vlog videos and interact with other users' videos.

Stimulus/Response Sequence:

User creates a food vlog video -->User uploads the video to the system -->

System displays the video on the Drops page for other users to view and interact with.

Other users can like, comment, and share the video

Functional Requirements:

Users can upload and share their food vlog videos

Like and comment functionalities on uploaded videos

Price Filtering:

Description: This feature allows users to filter search results based on their desired price range.

Stimulus/Response Sequence:

User selects a price range filter option -->System retrieves and displays the list of restaurants that fall within the selected price range

Functional Requirements:

Filtering search results based on desired price range

Helping users find dining options within their budget

Restaurant Management (Admin Panel):

Description: This feature allows the admin to manage restaurants and their information, including adding new restaurants to the platform, editing or deleting existing restaurant details, and reviewing and approving user requests for adding new restaurants.

Stimulus/Response Sequence:

Admin logs into the admin panel using their credentials-->

System authenticates the admin and grants access to the

admin functionalities-->Admin manages the restaurant

Functional Requirements:

Admin login and authentication

Adding new restaurants to the platform

Editing and updating existing restaurant details

Deleting restaurants from the platform

Reviewing and approving user requests for adding new restaurants

## 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.