# **The honeycomb**

# **Software Requirements Specifications**

## Version 1.0

## 23.02.2023

## Salma Eyadi

CONTENTS

[**Software Requirements Specifications**](#_mkue1smajfar)

[**Introduction** 3](#_75shq08jri5b)

[**Requirements** 4](#_be8c7p8p9uuv)

[**Functional Requirements 4**](#_kjae0hiyfdhk)

[Recipe Management](#_95x80nhia5n) [4](#_5zfkg31qcgcm)

[Review Management 4](#_6y7mnn1wuo7v)

[**Non-functional Requirements 5**](#_jjp8mmibazmg)

[Performance 5](#_rufebtycchq6)

[time 5](#_5zfkg31qcgcm)

[Security](#_ei51shxi0ic0)

[**System Design 6**](#_eaycf8un2mge)

[Architecture 6](#_sgogsfpj3ejg)

[User Interface](#_ss55akpooowj)

[**Database Design**](#_y47ith1gkm94)

[**Conclusion** 7](#_fp0t55rppkjf)

[**Glossary 8**](#_9guzeq43lty2)

## Introduction

#### This Software Requirements Specification (SRS) document outlines the functional and non-functional requirements for a recipe management web application. The recipe management web application is a web-based platform that allows users to upload and store their favorite recipes, view and search other users’ recipes, and leave reviews on the recipes. The application will be developed using the Flask web framework in Python, and SQLAlchemy will be used as the object-relational mapping (ORM) tool to interact with the database.

## Requirements

### **Functional Requirements**

#### Recipe Management

#### The application must allow users to upload recipes with information such as recipe name, description, ingredients, instructions, publisher, publish date, and image (optional).

#### The application must provide the ability for users to edit and delete their uploaded recipes.

#### The application must allow users to view recipes uploaded by other users.

#### The application must provide a search functionality to search for recipes based on the name.

#### The application must allow users to view a recipe's details such as recipe name, description, ingredients, instructions, publisher, publish date, and image (if available).

#### Review Management

#### The application must allow users to leave reviews on recipes.

#### The application must allow users to rate recipes while leaving a review.

#### The application must allow users to view reviews left by other users on a recipe.

### 

### 

### **Non-functional Requirements**

#### Performance

#### The application must handle concurrent users without slowdowns or crashes.

#### The application must respond to user actions within a reasonable amount of time.

#### Security

#### The application must use secure protocols to transmit and store user data.

#### The application must not allow unauthorized access to data.

#### Usability

#### The application must have a user-friendly interface that is easy to navigate.

#### The application must have clear and concise error messages when errors occur.

## System Design

### **Architecture**

#### The application will be developed using the Flask web framework in Python.

#### SQLAlchemy will be used as the object-relational mapping (ORM) tool to interact with the database.

#### The application will use an SQLite database to store user data and recipe information.

### **User Interface**

#### The application will have a user-friendly interface that is easy to navigate.

#### The interface will allow users to upload recipes, edit recipes, delete recipes, leave reviews, and view recipes.

#### 

### **Database Design**

#### The SQLite database will have two tables: Recipes and Reviews.

#### The Recipes table will have columns for id, recipe\_name, description, ingredients, instructions, publisher, publish\_date, and filename.

#### The Reviews table will have columns for id, recipe\_id, name, review\_text, and rating.

## Conclusion

#### This SRS document outlines the functional and non-functional requirements for a recipe management web application. The application will allow users to upload and store their favorite recipes, view and search other users’ recipes, and leave reviews on the recipes. The application will be developed using the Flask web framework in Python, and SQLAlchemy will be used as the object-relational mapping (ORM) tool. The application will use an SQLite database to store user data and recipe information

#### 

#### Database Design

#### The SQLite database will have two tables: Recipes and Reviews.

#### The Recipes table will have columns for id, recipe\_name, description, ingredients, instructions, and publisher

# Glossary

# Flask web framework: Flask is a micro web framework written in Python used for developing web applications. It provides flexibility and is easy to use.

1. SQLAlchemy: SQLAlchemy is an Object-Relational Mapping (ORM) library that provides a set of high-level APIs for interacting with databases. It allows developers to interact with databases using Python code instead of SQL.
2. SQLite database: SQLite is a lightweight database management system that is self-contained and serverless. It is widely used in small to medium-scale web applications and mobile applications.
3. Object-Relational Mapping (ORM): ORM is a technique that maps object-oriented programming (OOP) concepts to relational databases. It provides a bridge between object-oriented programming and relational database management system.
4. User Interface (UI): User Interface (UI) is the graphical layout and design of an application. It includes the buttons, text boxes, and other elements that users interact with on the screen.
5. User-friendly interface: A user-friendly interface is an interface that is easy to navigate, understand and use for users with little or no training or guidance.
6. Secure protocols: Secure protocols are communication protocols that provide a secure way of transmitting data between applications. Examples include HTTPS, SSL, and SSH.
7. Unauthorized access: Unauthorized access is the act of accessing an application or data without proper authorization or permission. It is considered a breach of security.
8. Error messages: Error messages are messages that are displayed on the screen when an error occurs in the application. They help users to understand the cause of the error and how to resolve it.
9. Flask: A micro web framework written in Python used for developing web applications
10. Responsive Design: A design approach that ensures the application's UI is optimized for different devices and screen sizes.
11. Recipe: A set of instructions for preparing a particular dish or meal.
12. Review: A written evaluation or critique of a recipe left by a user who has tried it.
13. Rating: A numerical evaluation of a recipe, typically on a scale of 1 to 5, left by a user who has tried it.
14. Search functionality: The ability of the application to allow users to search for specific content based on certain criteria.
15. Upload: The process of transferring files or data from a user's computer to the application's server.
16. Edit: The process of modifying existing content in the application.
17. Delete The process of removing existing content from the application.
18. Publisher: The person or entity responsible for publishing the recipe.
19. Publish date: The date on which the recipe was published.