RecipEasy

Software Requirements Specification

Version 1.0

1/26/2016

Awesome Inc.

**Table of Contents**

1. Introduction 1

1.1 Purpose 1

1.2 Scope 1

1.3Overview 1

2. General Description 2

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Characteristics 2

2.4 General Constraints 2

2.5 Assumptions and Dependencies 2

3. Specific Requirements 2

3.1 External Interface Requirements 3

3.1.1 User Interfaces 3

3.1.1.1 User Login 3

3.1.1.2 Main Interface 4

3.1.1.3 Recipe Search Results Interface 5

3.1.1.4 Recommendations Interface 6

3.1.2 Hardware Interfaces 7

3.1.3 Software Interfaces 7

3.1.4 Communications Interfaces 7

3.5 Non-Functional Requirements 7

3.5.1 Performance 7

3.5.2 Reliability 7

3.5.3 Availability 7

3.5.4 Security 7

3.5.5 Maintainability 7

3.5.6 Portability 7

3.8 Logical Database Requirements 7

4. Analysis Models 8

4.1 Data Flow Diagrams (DFD) 8

5. Change Management Process 9

**1. Introduction**

**1.1 Purpose**

*This document shall layout what RecipEasy will do, could do, and how it will meet those requirements, for Awesome Inc. and all other project observers.*

**1.2 Scope**

**1.2.1**  *The software to be made by this project shall be called RecipEasy*

**1.2.2** *RecipEasy will be able to recommend recipes for users, from a database of predefined recipes, by using their past ratings on food or recipes, or by letting them look up recipes by food/type/taste*

**1.2.3 Software objectives:**

**1.2.3.1** *This software should be able to recommend many types of different foods based on past ratings from them and all other users.*

**1.2.3.2** *There shall be a large number of recipes that have been rated and new recipes shall be added on a regular basis.*

**1.3 Overview**

**1.3.1** *This document shall contain an overview of the project with generalized specifications and assumptions.*

**1.3.2** *It will follow with the detailed description of the project’s requirements and how they are to be met.*

**1.3.3** *It will then describe how we are to analysis the project.*

**1.3.4**  *It will then describe how changes would be made to this document*

# 2. General Description

*The product searches for and selects recipes based on the user’s history of liked and disliked meals.*

## 2.1 Product Perspective

*The product is similar to music streaming apps that fine-tune the selected songs based on user feedback.*

## 2.2 Product Functions

*The software will search the internet for recipes that will please the user; it will keep track of likes and dislikes.*

## 2.3 User Characteristics

*Users of the product will include anyone of cooking age, likely 14 and above. The product will need to be easy to use during cooking, which can be messy and stressful at times.*

## 2.4 General Constraints

*General constraints are internet connectivity needed to access recipes, and user interface ease of use.*

## 2.5 Assumptions and Dependencies

*The software will be dependent on the availability of a Java Virtual Machine. Since the code is compiled in Java, the operating system will not be a limiting factor so long as it can run a Java Virtual Machine.*

# 3. Specific Requirements

## 3.1 External Interface Requirements

### 3.1.1 User Interfaces

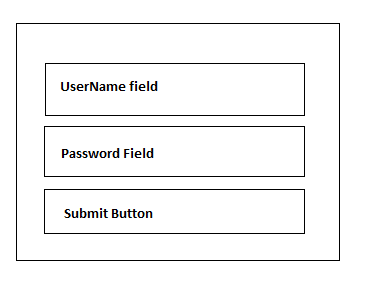
**3.1.1.1** *The application GUI shall consist of these four interfaces:*

**3.1.1.1.1** User Login

*Username field*

*Password field*

*Submit button*



**3.1.1.1.2** Main Interface

*Search field*

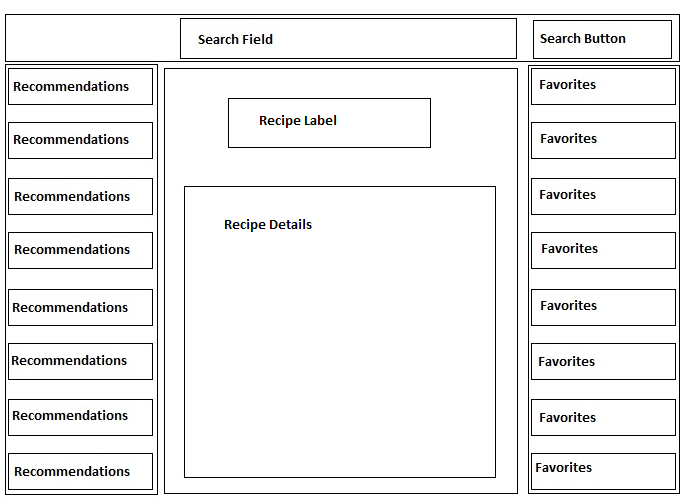
*Search submit button*

*Current recipe label*

*Recipe step details labels*

*Sidebar with favorites buttons*

*Sidebar with recommendation buttons*



**3.1.1.1.3** Recipe Search Results Interface

*Search again field*

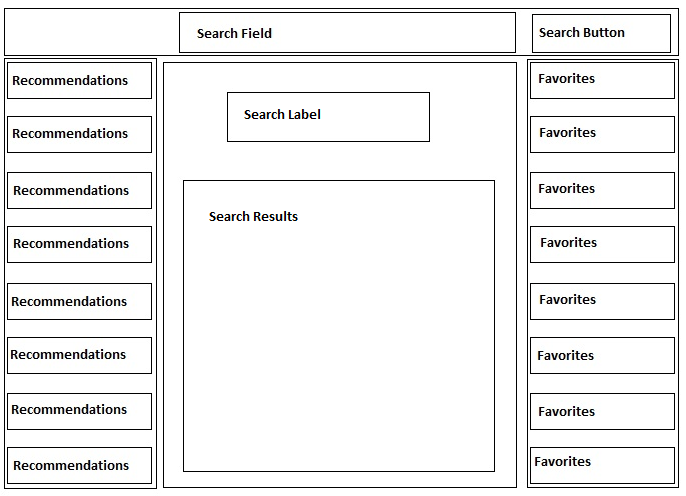
*Search again button*

*Search label*

*Search results buttons*

*Sidebar with favorites buttons*

*Sidebar with recommendation buttons*



**3.1.1.1.4** Recommendations Interface

*Recommendations label*

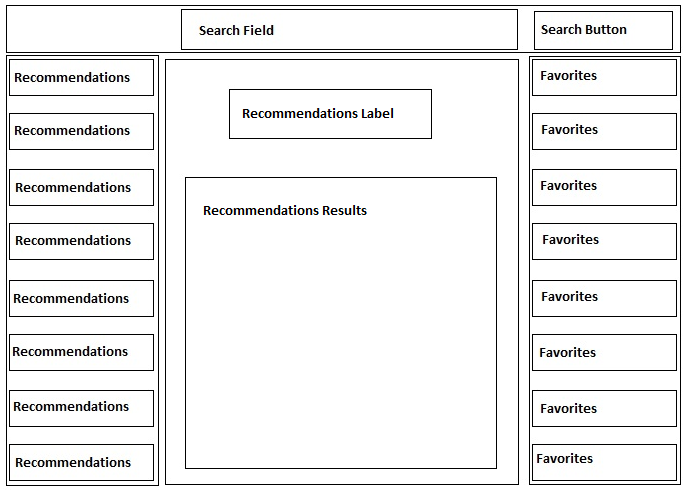
*Recommendation results buttons*

*Search field*

*Search submit button*

*Sidebar with favorites buttons*

*Sidebar with recommendation buttons*



### 3.1.2 Hardware Interfaces

The software requires a personal computer with at least:

Intel Pentium Processor with 2.0 Ghz or AMD Athlon 64 2800

1 GB RAM

128 MB VRAM

### 3.1.3 Communications Interfaces

The software requires an internet connection to communicate with an external database.

## 3.2 Non-Functional Requirements

Non-functional requirements may exist for the following attributes. Often these requirements must be achieved at a system-wide level rather than at a unit level. State the requirements in the following sections in measurable terms (e.g., 95% of transaction shall be processed in less than a second, system downtime may not exceed 1 minute per day, > 30 day MTBF value, etc).

### 3.2.1 Performance

95% of queries shall take no longer than 5 seconds

### 3.2.2 Reliability

The software shall have fewer than 5 bus per 1000 lines of code.

### 3.2.3 Availability

The software downtime shall not exceed 20 minutes per day.

### 3.2.4 Security

The data stored within the connected database shall be accessible only to those authorized.

### 3.2.5 Maintainability

The software shall be evaluated on a weekly basis to determine whether updating is neccessarry.

### 3.2.6 Portability

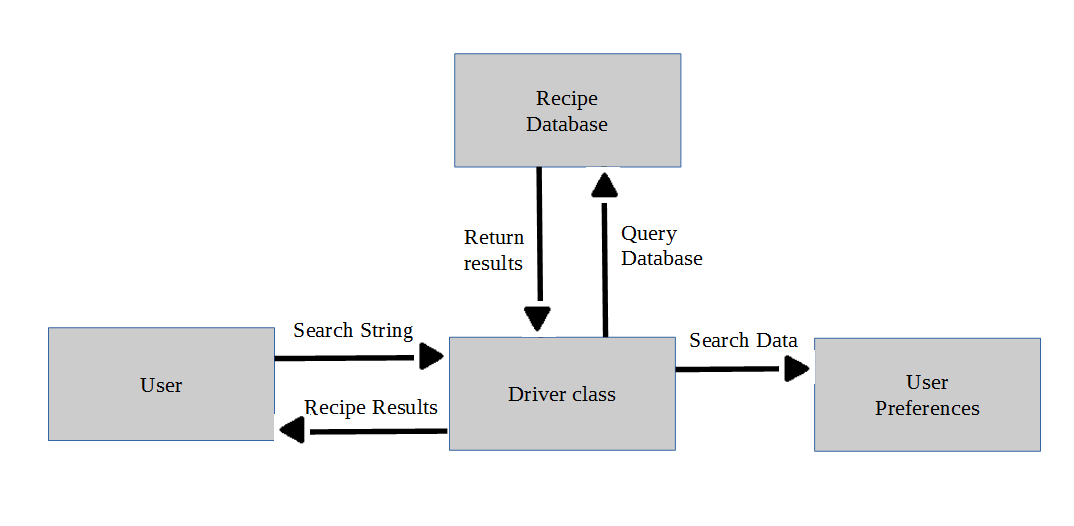
The software shall function on all systems meeting the hardware requirements, and are using the Windows 7 operating system or later.

## 3.3 Logical Database Requirements

A database using SQL is required by the application. The database shall take up no more than 1 GB of hard drive space on the host server. Data shall be backed up daily.

# 4. Analysis Models

## 4.1 Data Flow Diagrams (DFD)



# 5. Change Management Process

## *If changes are needed to be made to the SRS document when the scope or requirements change, detailed list of the current version of the document and any edited changes will be sent to and approved by the team leader. This can be sent via email or an in person meeting.*