**Unnamed Private Cookbook Site**

**Software Requirements Specification**

**For Base Site**

**Version 1.0.0**

**Table of Contents**

[1.          Introduction](http://sce.uhcl.edu/helm/rationalunifiedprocess/webtmpl/templates/req/rup_srsuc.htm#1.%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20Introduction)

[1.1      Purpose](http://sce.uhcl.edu/helm/rationalunifiedprocess/webtmpl/templates/req/rup_srsuc.htm#1.1%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20Purpose)

[1.2      Scope](http://sce.uhcl.edu/helm/rationalunifiedprocess/webtmpl/templates/req/rup_srsuc.htm#1.2%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20Scope)

[1.3      Definitions, Acronyms and Abbreviations](http://sce.uhcl.edu/helm/rationalunifiedprocess/webtmpl/templates/req/rup_srsuc.htm#1.3%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20Definitions,%20Acronyms%20and%20Abbreviations)

[1.4      References](http://sce.uhcl.edu/helm/rationalunifiedprocess/webtmpl/templates/req/rup_srsuc.htm#1.4%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20References)

[1.5      Overview](http://sce.uhcl.edu/helm/rationalunifiedprocess/webtmpl/templates/req/rup_srsuc.htm#1.5%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20Overview)

[2.          Overall Description](http://sce.uhcl.edu/helm/rationalunifiedprocess/webtmpl/templates/req/rup_srsuc.htm#2.%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20Overall%20Description)

[2.1      Use-Case Model Survey](http://sce.uhcl.edu/helm/rationalunifiedprocess/webtmpl/templates/req/rup_srsuc.htm#2.1%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20Use-Case%20Model%20Survey)

[2.2      Assumptions and Dependencies](http://sce.uhcl.edu/helm/rationalunifiedprocess/webtmpl/templates/req/rup_srsuc.htm#2.2%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20Assumptions%20and%20Dependencies)

[3.          Specific Requirements](http://sce.uhcl.edu/helm/rationalunifiedprocess/webtmpl/templates/req/rup_srsuc.htm#3.%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20Specific%20Requirements)

[3.1      Use-Case Reports](http://sce.uhcl.edu/helm/rationalunifiedprocess/webtmpl/templates/req/rup_srsuc.htm#3.1%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20Use-Case%20Reports)

[3.2      Supplementary Requirements](http://sce.uhcl.edu/helm/rationalunifiedprocess/webtmpl/templates/req/rup_srsuc.htm#3.2%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20Supplementary%20Requirements)

[4.          Supporting Information](http://sce.uhcl.edu/helm/rationalunifiedprocess/webtmpl/templates/req/rup_srsuc.htm#4.%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20Supporting%20Information)

**Software Requirements Specification**

**1.**                  **Introduction**

Documentation of the requirements for the Unnamed Private Cookbook Site.

**1.1**               **Purpose**

This document describes the breakdown and description of the Unnamed Cookbook Site/application.

**1.2**               **Scope**

The Unnamed Cookbook Site is an application/website that is used to store user recipes and to link up other users recipes. The user will have the ability to share their recipes with only known and approved users.

**1.3**               **Definitions, Acronyms and Abbreviations**

None at this time.

**1.4**               **References**

None at this time.

**1.5**               **Overview**

The purpose for the Cookbook site is to allow users the flexibility to store their recipes and to share them with known and authorized friends and family. It will also give them access to the recipes through the internet. Recipes will can be added and viewed by all family members, but not those outside of the online “family”.

**2.**                  **Overall Description**

1. **Front End:** Built using HTML, CSS, and Bootstrap

1. Sign in:
   1. Website should be built allowing a user to sign up or log into system. The homepage or landing page should have a contact, about and description of services.
   2. Have a sign in ability for Administration.
2. Sign up:
   1. The site or application will require a new user to supply a email address and password to sign up.
3. View Cookbook:
   1. Once logged into the system it will take you to the cookbook page that will have a list of all the recipes that the user has loaded into the system. It will also have a link to add recipes.
4. Add Recipes:
   1. The add recipe link will take the user to the add recipe page. Recipe page will have the ability to upload photo of the food and to input the Title, ingredients, and directions. It will also have a place to input tags to help with searches.

2. **Back End:** Built using either .NET, SQL, or PHP.

1. Sign in Authentication
   1. Will verify the user with in the website/application based on User and Administrators.
2. Database
   1. It will store the recipes based on the user input.
   2. It will also have the ability to link to other users based on “Family/Friend” acceptance.
   3. It will have a search functionality that will allow the user to search by title, ingredient or type of food.

**2.1**               **Use-Case Model Survey**

None at this time.

**2.2**               **Assumptions and Dependencies**

1. Sign in Authentication will need to be researched further to see if it is feasible to do within our time constraints.

**3.**                  **Specific Requirements**

User Story. 1 Title: Login Username/Password

Description: Have the ability to login to the website with a username and password. No unauthorized viewing of the website. Have multiple logins for different family members.

User Story. 2 Title: Add Recipe manually

Description: Have the ability to add a new recipe by typing it into the website

User Story. 3 Title: Add Recipe from a Picture

Description: Have the ability to add a new recipe by using a picture.

User Story. 4 Title: Title

Description: The recipe should have a title for the recipe and you should be able to edit.

User Story. 5 Title: Ingredients

Description: The recipe should have the ability to have ingredients or multiple ingredients. The ingredients should specify the many different types of measurements

User Story. 6 Title: Cooking Instructions

Description: Should be a paragraph or listed instructions.

User Story. 7 Title: Comments

Description: Should be able to make comments on the recipe from users.

User Story. 8 Title: Delete

Description: Admin should be able to delete recipes.

User Story. 9 Title: Picture

Description: Add photos of the food/recipe.

User Story. 10 Title: Search by Title

Description: Able to search for the recipe by title.

User Story. 11 Title: Search by Ingredient

Description: Able to search for the recipe by an ingredient.

User Story 12 Title: Admin User

Description: Admin users would have authority over deleting and adding new members.

**3.1**               **Use-Case Reports**

Login:

Pre-condition: none

Non functional requirements: all users must have individual password, but username remains the same for the family

Post condition: user selects their individual profile from list of family members

Add recipe manually:

Pre- condition: select type of recipe (breakfast, lunch, dinner)

Non functional requirements: none

Post-condition: recipe saved to family database

Comments

Pre condition: recipe must exist

Delete

Pre-condition: recipe must exist

Post condition: recipe is removed from family database and comments are deleted

Search by Title:

Precondition: recipes must be in the family database

Included use cases: title, ingredients, cooking instructions

**3.2**               **Supplementary Requirements**

None at this time.

**4.**                  **Supporting Information**

None at this time.

GUIDELINES FOR REQUIREMENTS

|  |  |  |
| --- | --- | --- |
| Checkpoints:  Software Requirements Specification  * The following basic issues should be addressed:   + *Functionality*: What is the software supposed to do?   + *External interfaces*: How does the software interact with people, the system's hardware, other hardware, and other software?   + *Performance*: What is the speed, availability, response time, recovery time of various software functions, etc.?   + *Attributes*: What are the portability, correctness, maintainability, security, etc. considerations?   + *Design constraints imposed on an implementation*: Are there any required standards in effect, implementation language, policies for database integrity, resource limits, operating environments, etc.? * Are any requirements specified that are outside the bounds of the SRS? This means the SRS   + Should correctly define all of the software requirements,   + Should not describe any design or implementation details,   + Should not impose additional constraints on the software. * Does the SRS properly limit the range of valid designs without specifying any particular design? * Does the SRS exhibit the following characteristics?   + *Correct*: Is every requirement stated in the SRS one that the software should meet?   + *Unambiguous*     - Does each requirement have one, and only one, interpretation?     - Has the customer's language been used?     - Have diagrams been used to augment the natural language descriptions?   + *Complete*     - Does the SRS include all significant requirements, whether related to functionality, performance design constraints, attributes, or external interfaces?     - Have the expected ranges of input values in all possible scenarios been identified and addressed?     - Have responses been included to both valid and invalid input values?     - Do all figures, tables and diagrams include full labels and references and definitions of all terms and units of measure?     - Have all TBDs been resolved or addressed?   + *Consistent*     - Does this SRS agree with the Vision document, the use-case model and the Supplementary Specifications?     - Does it agree with any other higher level specifications?     - Is it internally consistent, with no subset of individual requirements described in it in conflict?   + *Ability to Rank Requirements*     - Has each requirement been tagged with an identifier to indicate either the importance or stability of that particular requirement?     - Have other significant attributes for properly determining priority been identified?   + *Verifiable*     - Is every requirement stated in the SRS verifiable?     - Does there exist some finite cost-effective process with which a person or machine can check that the software product meets the requirement?   + *Modifiable*     - Are the structure and style of the SRS such that any changes to the requirements can be made easily, completely, and consistently while retaining the structure and style?     - Has redundancy been identified, minimized and cross-referenced?   + *Traceable*     - Does each requirement have a clear identifier?     - Is the origin of each requirement clear?     - Is backward traceability maintained by explicitly referencing earlier artifacts?     - Is a reasonable amount of forward traceability maintained to artifacts spawned by the SRS?   Reference: [[IEEE93](http://sce.uhcl.edu/helm/rationalunifiedprocess/process/referenc.htm#IEEE93)]  [Copyright  © 1987 - 2001 Rational Software Corporation](http://sce.uhcl.edu/helm/rationalunifiedprocess/copyrite/copyrite.htm) |  | [Display Rational Unified Process using frames](about:blank) |

|  |
| --- |
| Rational Unified Process   http://sce.uhcl.edu/helm/rationalunifiedprocess/_borders/rupversion.gif |