CookieApp

Version 1.1

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 12.10.2014 | 1.0 | First Editing | Moritz G. |
| 16.10.2014 | 1.1 | Implemented Peer Review Suggestions | Christian V. |
| 17.10.2014 | 1.2 | Updated usecase diagramm, specific requirements | Christian V. |
|  |  |  |  |

Table of Contents

1. Introduction 4

1.1 Purpose 4

1.2 Scope 4

1.3 Definitions, Acronyms, and Abbreviations 4

1.4 References 4

1.5 Overview 4

2. Overall Description 5

3. Specific Requirements 6

3.1 Functionality 6

3.1.1 Register 6

3.1.2 Login 6

3.1.3 Logout 6

3.1.4 Search for recipe 6

3.1.5 Create recipe 6

3.1.6 Delete recipe 6

3.1.7 Comment on recipe 7

3.1.8 Suggestion on recipe 7

3.1.9 Create Category 7

3.1.10 Delete User 7

3.1.11 Delete Recipe 7

3.2 Usability 8

3.2.1 Fast Learning 8

3.3 Reliability 9

3.3.1 Availability 9

3.3.2 Mean Time to Repair 9

3.3.3 Accuracy 9

3.3.4 Data Defect Rate 9

3.3.5 Bugs 9

3.4 Performance 9

3.4.1 <Performance Requirement One> 9

3.5 Supportability 9

3.5.1 <Supportability Requirement One> 10

3.6 Design Constraints 10

3.6.1 <Design Constraint One> 10

3.7 On-line User Documentation and Help System Requirements 10

3.8 Purchased Components 10

3.9 Interfaces 10

3.9.1 User Interfaces 10

3.9.2 Hardware Interfaces 10

3.9.3 Software Interfaces 10

3.9.4 Communications Interfaces 10

3.10 Licensing Requirements 11

3.11 Legal, Copyright, and Other Notices 11

3.12 Applicable Standards 11

4. Supporting Information 11

5. Glossary 11

# Introduction

## Purpose

This document contains the specifications for our software system. You’ll find the descriptions of features that should be covered by our system.

## Scope

Our system is a web application, which should provide many recipes and the possibility to search for them by ingredients. Users should be able to submit their own recipes and other users will have the ability to give feedback and send suggestions for improvement. Also all submitted recipes should be divided in different categories.

## Definitions, Acronyms, and Abbreviations

See glossary.

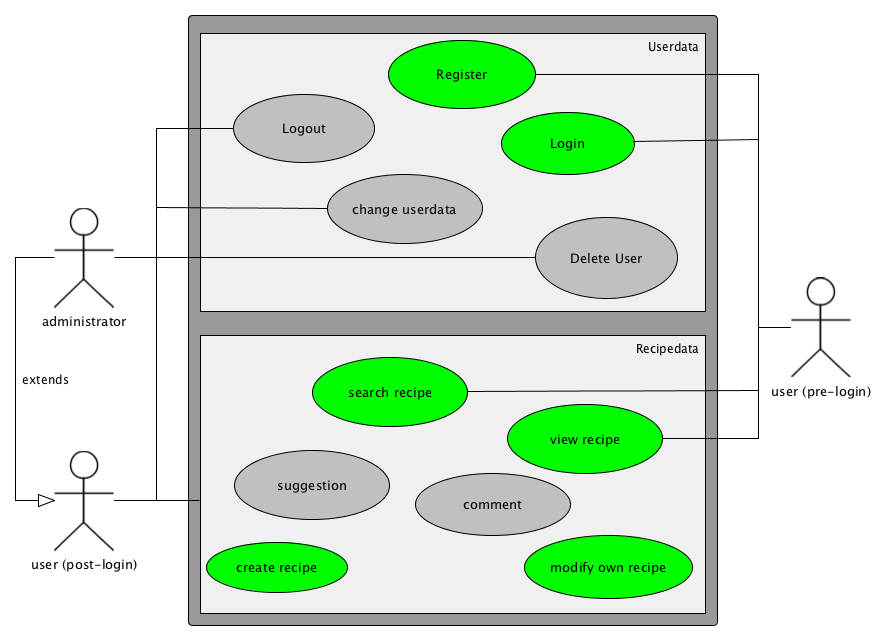
## References

To be determined.

## Overview

[This subsection describes what the rest of the **SRS** contains and explains how the document is organized.]

# Overall Description



# Specific Requirements

## Functionality

### Register

If the user is not logged in already, he has the possibility to create himself a user account.

For the registration the user has to enter a nickname, his email address and a password. Other entries are not necessary at this point and can be added in the later process.

A button to register should be available on all sites, if the user isn´t logged in.

### Login

If the user is not logged in already, he has the possibility to log into his user account.

For the login the user has to enter his nickname or his email address and his password.

A button to login should be available on all sites, if the user isn´t logged in.

### Logout

If the user is logged in, he has the possibility to log himself out of his user account.

For the logout the user should only need to press one button.

A button to login should be available on all sites, if the user is logged in.

### Search for recipe

The user should have the possibility to search for recipes in every state (logged in, logged off).

To search for a recipe the user needs to enter the name of the recipe or a part of the name. The user also has the possibility to search for ingredients or a special category. After pressing the “search” button a list should appear, showing the search results.

A button to search for a recipe should be available on all sites.

### Create recipe

The user should have the possibility to create a new recipe if he is logged in.

In order to create a recipe the user has to enter a name for the recipe, a short description, the approximate time to prepare, the ingredients with their amount and a description how to put them together.

A button to create a recipe should be available on all sites, if the user is logged in.

### Delete recipe

The user should have the possibility to delete a recipe he submitted if he is logged in.

In order to delete a recipe the user has to open the recipe he submitted. On the upper section of the recipe there should be a button to delete the recipe.

A button to delete a recipe should be available on the recipe site it self, if the user is logged in and has submitted the recipe.

### 

### Comment on recipe

Any user who is logged in should have the possibility to leave a comment on a recipe.

In order to do that, he has do open the recipe he want´s to comment. By clicking the leave comment button the user can leave a comment. The user has to enter plain text as the comment. After clicking the send comment button the comment will be saved and now be shown on the button area of the recipe page.

The leave comment button will only be shown on the recipe site when logged in.

### Suggestion on recipe

Any user who is logged in should have the possibility to suggest an improvement of the recipe to the one, who has uploaded it.

In order to do that, he has do open the recipe he want´s to leave a suggestion for. By clicking the suggest improvement button the user can leave a improvement suggestion. The user has to enter plain text. After clicking the send suggestion button the suggestion will be saved and now be shown to the submitter of the recipe as a message.

The suggest an improvement button will only be shown on the recipe site when logged in.

### Create Category

The administrator should have the possibility to add an ingredient to a category. So the recipes will generate their category by them self.  
The approach for this has to be determined in future.

### Delete User

The administrator can delete user accounts. To delete a user, the administrator has a management UI on the Webpage and can delete a user.

The approach for this has to be determined in future.

### Delete Recipe

The administrator should have the possibility to delete a recipe.

In order to delete a recipe the administrator has to open the recipe he wants to delete. On the upper section of the recipe there should be a button to delete the recipe.

A button to delete a recipe should be available on the recipe site it self, if the user is administrator.

## Usability

It’s our goal to create a user friendly GUI which is easy to understand and easy to use. Therefore we try to keep our functions simple and self-explanatory.

The essential features of this application should be easy accessible and should require least number of clicks.

### Fast Learning

Users should take five to ten minutes to get along with the web application and know how to use a large part of our functions

## Reliability

### Availability

We are aiming at a high accuracy with our system outputs and therefore we’ll be especially looking at our search algorithms.

### Mean Time to Repair

We are aiming at a high accuracy with our system outputs and therefore we’ll be especially looking at our search algorithms.

### Accuracy

We are aiming at a high accuracy with our system outputs and therefore we’ll be especially looking at our search algorithms.

### Data Defect Rate

Some loss of data errors will probably happen but we’re aiming at a minimum loss of data in our system.

### Bugs

Do to testing we are aiming a very small number of bugs in the system.

## Performance

### Response Time

The response from the server shouldn´t be delayed more than 200ms. The average response time should be around 150ms

### Capacity

In the early stages of development, the server should at least be able to have 10 different users logged in and working.

### Resource Utilization

The usage of CPU time, memory and disk space for the database is of no concern in early stages.

## Supportability

### Supported Operation Systems

The webpage should run in every operating system with a web browser

## Design Constraints

### Language

The Project will be programmed in Java. The documentation is in English.

### Development Tools

The Eclipse IDE will be used as Development environment.

The Eclipse RAP Tooling plugin will be used in Eclipse to build the webpage.

GitHub will be used for versioning of the project and its documentation.

### Frameworks

The project takes use of the Eclipse RAP framework for the website UI.

We will be using Eclipse CDO to implement a database.

## On-line User Documentation and Help System Requirements

There will be no Help Section accessible.

## Purchased Components

There will be no purchased components or content accessible.

## Interfaces

To be determined.

### User Interfaces

To be determined.

### Hardware Interfaces

To be determined.

### Software Interfaces

### 

To be determined.

### Communications Interfaces

To be determined.

## Licensing Requirements

To be determined.

## Legal, Copyright, and Other Notices

To be determined.

## Applicable Standards

# Supporting Information

# Glossary