CookieApp

Version 1.1

[Note: The following template is provided for use with the Rational Unified Process. Text enclosed in square brackets and displayed in blue italics (style=InfoBlue) is included to provide guidance to the author and should be deleted before publishing the document. A paragraph entered following this style will automatically be set to normal (style=Body Text).]

[To customize automatic fields in Microsoft Word (which display a gray background when selected), select File>Properties and replace the Title, Subject and Company fields with the appropriate information for this document. After closing the dialog, automatic fields may be updated throughout the document by selecting Edit>Select All (or Ctrl-A) and pressing F9, or simply click on the field and press F9. This must be done separately for Headers and Footers. Alt-F9 will toggle between displaying the field names and the field contents. See Word help for more information on working with fields.]

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, reliablities, functionality | 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 7

3.1.7 Comment on recipe 7

3.2 Usability 7

3.2.1 Fast Learning 7

3.3 Reliability 8

3.3.1 Availability 8

3.3.2 Mean Time to Repair 8

3.3.3 Accuracy 8

3.3.4 Data Defect Rate 8

3.3.5 Bugs 8

3.4 Performance 8

3.4.1 <Performance Requirement One> 8

3.5 Supportability 8

3.5.1 <Supportability Requirement One> 9

3.6 Design Constraints 9

3.6.1 <Design Constraint One> 9

3.7 On-line User Documentation and Help System Requirements 9

3.8 Purchased Components 9

3.9 Interfaces 9

3.9.1 User Interfaces 9

3.9.2 Hardware Interfaces 9

3.9.3 Software Interfaces 9

3.9.4 Communications Interfaces 9

3.10 Licensing Requirements 10

3.11 Legal, Copyright, and Other Notices 10

3.12 Applicable Standards 10

4. Supporting Information 10

5. Glossary 10

# 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

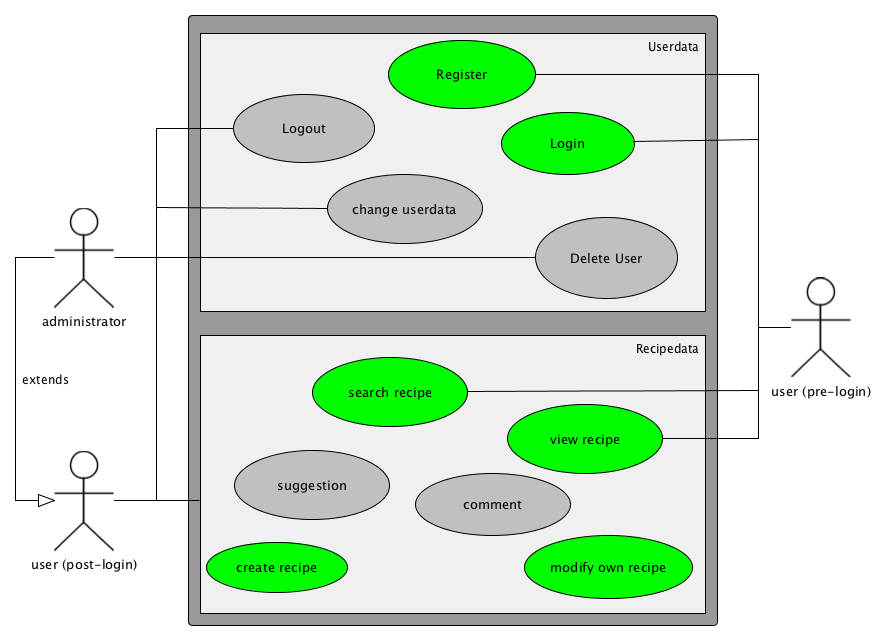
To be determined.

[This subsection provides a complete list of all documents referenced elsewhere in the **SRS**. Identify each document by title, report number if applicable, date, and publishing organization. Specify the sources from which the references can be obtained. This information may be provided by reference to an appendix or to another document.]

## 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.

### Suggestion on recipe

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

### Create Category

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

### Delete User

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

### Delete Recipe

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

## 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.

[This section includes all those requirements that affect usability. For example,

* specify the required training time for a normal users and a power user to become productive at particular operations
* specify measurable task times for typical tasks or base the new system’s usability requirements on other systems that the users know and like
* specify requirement to conform to common usability standards, such as IBM’s CUA standards Microsoft’s GUI standards]

### 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

To be determined (we need the specifications of our server)

[The system’s performance characteristics are outlined in this section. Include specific response times. Where applicable, reference related Use Cases by name.

* Response time for a transaction (average, maximum)
* Throughput, for example, transactions per second
* Capacity, for example, the number of customers or transactions the system can accommodate
* Degradation modes (what is the acceptable mode of operation when the system has been degraded in some manner)
* Resource utilization, such as memory, disk, communications, and so forth.

### <Performance Requirement One>

[The requirement description goes here.]

## Supportability

Our system should work on every popular webbrowser.

### <Supportability Requirement One>

[The requirement description goes here.]

## Design Constraints

To be determined.

[This section indicates any design constraints on the system being built. Design constraints represent design decisions that have been mandated and must be adhered to. Examples include software languages, software process requirements, prescribed use of developmental tools, architectural and design constraints, purchased components, class libraries, and so on.]

### <Design Constraint One>

[The requirement description goes here.]

## On-line User Documentation and Help System Requirements

To be determined.

[Describes the requirements, if any, for o-line user documentation, help systems, help about notices, and so forth.]

## Purchased Components

To be determined (maybe premium users)

[This section describes any purchased components to be used with the system, any applicable licensing or usage restrictions, and any associated compatibility and interoperability or interface standards.]

## Interfaces

To be determined.

[This section defines the interfaces that must be supported by the application. It should contain adequate specificity, protocols, ports and logical addresses, and the like, so that the software can be developed and verified against the interface requirements.]

### User Interfaces

[Describe the user interfaces that are to be implemented by the software.]

### Hardware Interfaces

[This section defines any hardware interfaces that are to be supported by the software, including logical structure, physical addresses, expected behavior, and so on.]

### Software Interfaces

[This section describes software interfaces to other components of the software system. These may be purchased components, components reused from another application or components being developed for subsystems outside of the scope of this **SRS** but with which this software application must interact.]

### Communications Interfaces

[Describe any communications interfaces to other systems or devices such as local area networks, remote serial devices, and so forth.]

## Licensing Requirements

To be determined.

[Defines any licensing enforcement requirements or other usage restriction requirements that are to be exhibited by the software.]

## Legal, Copyright, and Other Notices

[This section describes any necessary legal disclaimers, warranties, copyright notices, patent notices, wordmark, trademark, or logo compliance issues for the software.]

## Applicable Standards

[This section describes by reference any applicable standard and the specific sections of any such standards which apply to the system being described. For example, this could include legal, quality and regulatory standards, industry standards for usability, interoperability, internationalization, operating system compliance, and so forth.]

# Supporting Information

[The supporting information makes the **SRS** easier to use. It includes:

* Table of contents
* Index
* Appendices

These may include use-case storyboards or user-interface prototypes. When appendices are included, the **SRS** should explicitly state whether or not the appendices are to be considered part of the requirements.]

# Glossary