System Requirements Specifications

*<NAME OF WEB APP>*

**Client:**

Morgan Madeira

**Developer Team (Team 3):**

Connor Cook

Matt Law

Brent Mattox

Austin Murdock

Michael Neary

1. Introduction
   1. Purpose of This Document

This document explains the features of the <NAME> web application, a social website for people to share and review recipes. It goes into detail about the required functions of the web application, and how each component will work together. The intended audience of this document is the <NAME> developer team named on the cover page and the customer Morgan Madeira.

* 1. References
  2. Purpose of the Product

<NAME> is designed to be a web application for people to easily create, share, and rate recipes. There is a large community of home cooks/bartenders each with unique recipes and this web application will provide a platform for the people to share their recipes and ideas with each other.

* 1. Product Scope

The <NAME> web application consists of 14 use cases including: creating a new account, logging in, viewing a profile, uploading a recipe, rating a recipe, writing a comment, searching for a recipe, viewing user statistics, reciving notifications, flagging comments, recipe approval, assigning moderators, banning users, and removing comments. Refer to the diagrams below for the scope of each use case. Refer to section two for details on each use case.

<diagrams here>

1. Functional Requirements

|  |  |  |
| --- | --- | --- |
| **Number** | 6 | |
| **Name** | Create a new account. | |
| **Summary** | A user may create an account to gain the ability to post and review content. | |
| **Priority** | 5 | |
| **Preconditions** | User must access the website. | |
| **Postconditions** | The user will have an account and be logged into the site. | |
| **Primary Actor** | A user of the web site | |
| **Secondary Actors** | account/login system | |
| **Trigger** | User wants to create an account, and selects the new account button when logging in. | |
| **Main Scenario** | **Step** | **Action** |
|  | 1 | The user accesses the home page of the website. |
|  | 2 | The user clicks the button to create a new account. |
|  | 3 | The user enters their personal information, and chooses a username/password. |
|  | 4 | The system creates a new account for the user |
|  | 5 | The system logs the user into the site |
|  | 6 | The user is shown a message congratulating them on their new account, and is brought to the home page. |
| **Extensions** | **Step** | **Branching Action** |
|  | 3a | The user’s chosen username already exists in the database. |
|  | 4a | The system is unable to create a new account (improper email, name, facebook, etc.) |
| **Open Issues** | How do we handle a user being unable to create an account for any reason? | |

|  |  |  |
| --- | --- | --- |
| **Number** | 7 | |
| **Name** | Log onto the site | |
| **Summary** | User can log onto the site and access all of their content | |
| **Priority** | 5 | |
| **Preconditions** | User must have an account. | |
| **Postconditions** | User will be logged into the site | |
| **Primary Actor** | A user of the website | |
| **Secondary Actors** | account/login system | |
| **Trigger** | User selects the button to log in. | |
| **Main Scenario** | **Step** | **Action** |
|  | 1 | User directed to login screen. |
|  | 2 | User enters username and password. |
|  | 3 | Account/login system checks for a match between the username and password. |
|  | 4 | Account/login system logs the user into the site. |
| **Extensions** | **Step** | **Branching Action** |
|  | 3a | Username and password do not match |
| **Open Issues** | How do we handle a username and password that do not match?  How do we secure passwords? | |

|  |  |  |
| --- | --- | --- |
| **Number** | 8 | |
| **Name** | View a profile | |
| **Summary** | A user can view their profile or the profile of another user. | |
| **Priority** | 4 | |
| **Preconditions** | User must be logged in. | |
| **Postconditions** | User will have a profile displayed on their screen. | |
| **Primary Actor** | User of the website | |
| **Secondary Actors** | Account system | |
| **Trigger** | User clicks on a username or searches for a user. | |
| **Main Scenario** | **Step** | **Action** |
|  | 1 | User requests a profile. |
|  | 2 | Account system pulls information of user whose profile was requested. |
|  | 3 | Profile information is displayed for the user requested. |
| **Extensions** | **Step** | **Branching Action** |
|  | 2a | User requested does not exist. |
| **Open Issues** | Who is allowed to view profiles? Only users who are logged in? | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Number** | 12 | | |
| **Name** | Flag a comment | | |
| **Summary** | A user may declare (i.e. flag) a comment to be unacceptable, inappropriate, etc. | | |
| **Priority** | 4 | | |
| **Preconditions** | User must access the web site, have an account, and a comment must exist. | | |
| **Postconditions** | The comment is flagged for review (i.e. added to a review list or queue) | | |
| **Primary Actor** | A user of the web site | | |
| **Secondary Actors** | Notification system | | |
| **Trigger** | A user decides to flag a comment by selecting the flag button for that comment | | |
| **Main Scenario** | **Step** | **Action** | |
|  | 1 | The user selects the flag comment button for the comment they wish to flag | |
|  | 2 | The task of reviewing the comment is added to a review list (e.g. a queue) | |
|  | 3 | The user is thanked, and notified that the comment has been flagged for review | |
|  | 4 | System moderators and administrators are notified that a comment has been flagged for review | |
|  | 5 | A moderator or administrator reviews the content of the comment | |
|  | 6 | The comment is removed | |
|  | 7 | The comment is removed from the review list (e.g. a queue) | |
| **Extensions** | **Step** | **Branching Action** | |
|  | 5a | The moderator or administrator is unsure whether the comment should be removed, and so notifies other administrators for a vote | |
|  | 6a | The comment is allowed to remain | |
| **Open Issues** | 1 | | Regardless of the moderator’s/administrator’s decision, should the comment be automatically removed after a certain number of users flag the comment? |

|  |  |  |
| --- | --- | --- |
| **Number** | 13 | |
| **Name** | Recipe review | |
| **Summary** | Moderators will review recipes for content before allowing them to be posted | |
| **Priority** | 3 | |
| **Preconditions** | A recipe post must be awaiting review | |
| **Postconditions** | The recipe is posted or not posted | |
| **Primary Actor** | Moderator | |
| **Secondary Actors** | System maintaining the list of recipes to review | |
| **Trigger** | Moderator selects a recipe from the review list | |
| **Main Scenario** | **Step** | **Action** |
|  | 1 | Moderator selects a recipe from the recipe review list |
|  | 2 | Moderator inspects the recipe to make sure content is appropriate (e.g. accuracy, nothing offensive) |
|  | 3 | The moderator allows the recipe to be posted |
|  | 4 | The recipe is removed from the recipe review list |
| **Extensions** | **Step** | **Branching Action** |
|  | 3a | The recipe is not allowed to be posted |
| **Open Issues** | Should a user be notified if their recipe is not allowed to be posted? | |

|  |  |  |
| --- | --- | --- |
| **Number** | 14 | |
| **Name** | Assigning moderators | |
| **Summary** | Administrators may select users to become moderators | |
| **Priority** | 3 | |
| **Preconditions** | A user who wants to become a moderator must exist | |
| **Postconditions** | The user account is changed to a moderator account | |
| **Primary Actor** | Administrator | |
| **Secondary Actors** | Account management database | |
| **Trigger** | An administrator decides to make a user a moderator | |
| **Main Scenario** | **Step** | **Action** |
|  | 1 | The administrator selects a user they wish to make a moderator |
|  | 2 | The user is notified that they have been selected to become a moderator |
|  | 3 | The user accepts the account status of moderator |
|  | 4 | The account management database is updated to show that the user is now a moderator |
| **Extensions** | **Step** | **Branching Action** |
|  | 3a | The user declines the account status of moderator |
|  | 4a | The account management database is not updated because the user declined moderator account status |
| **Open Issues** |  | |

1. Non-Functional Requirements

|  |  |
| --- | --- |
| Items | Priority (1-5 lowest-highest) |
| Must follow Spiral development process. | 5 |
| Must use a PHP framework. | 1 |
| Must have bi-weekly updates with customer. | 4 |
| Must utilize Github to store the project. | 5 |
| All code will be well documented. | 4 |
| Must adhere to group set coding standards. | 4 |
| Application will use MVC design. | 3 |
| Site will be secure. | 5 |
| Utilize only publicly available technologies. | 5 |
|  |  |

1. UI
2. Deliverables
3. Open Issues