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, receiving 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.

1. Create Account

User

Account/Login System

1. Log onto

website

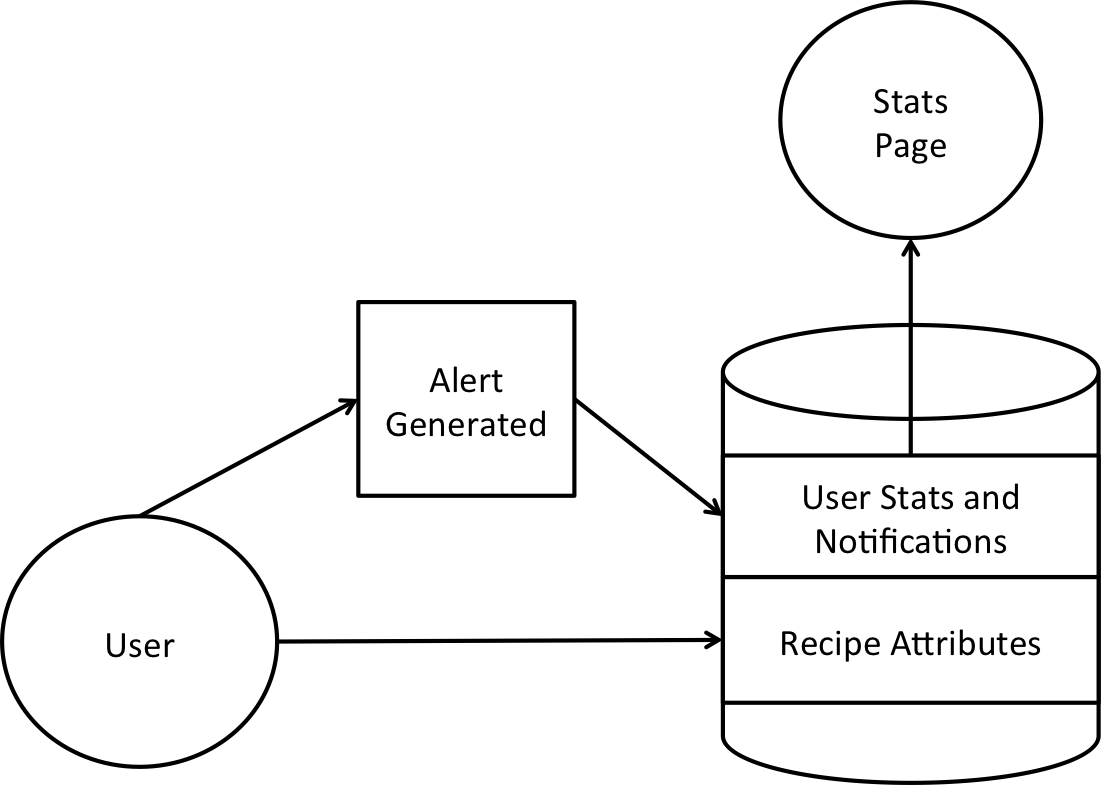
User

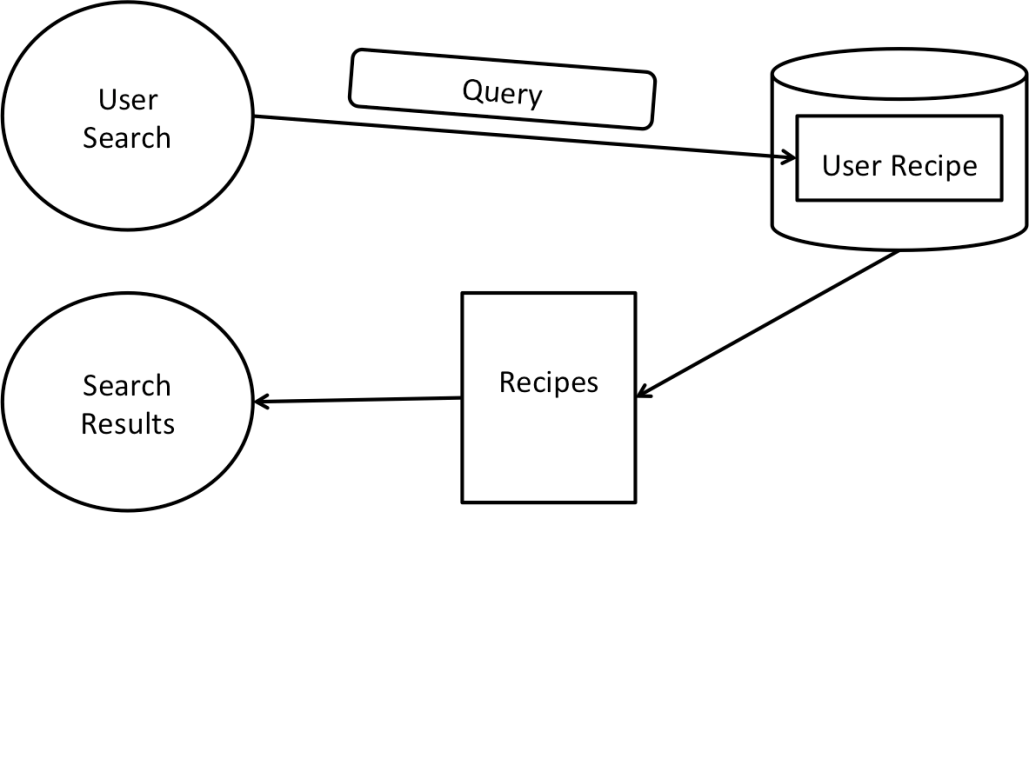
Account/Login System

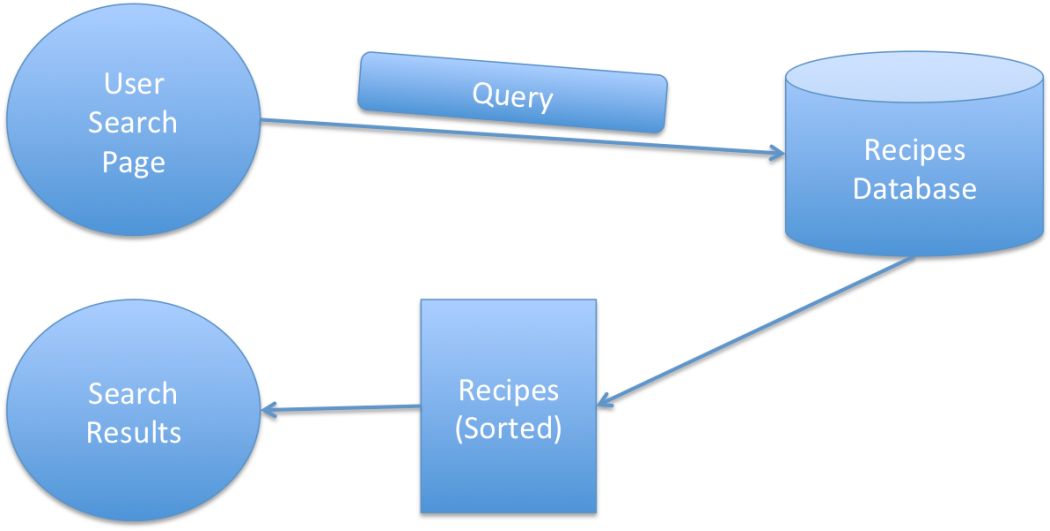
3. View a Profile

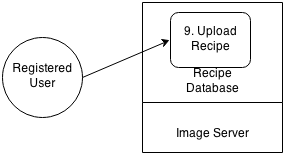
User

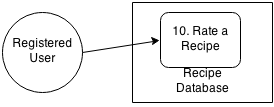
Account/Login System

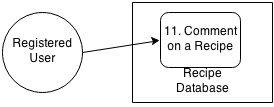












User

User Database

Notification System

12. Flag a comment

12. Flag a comment

Moderator

User database

13. Review a recipe

Administrator

14. Assign a moderator

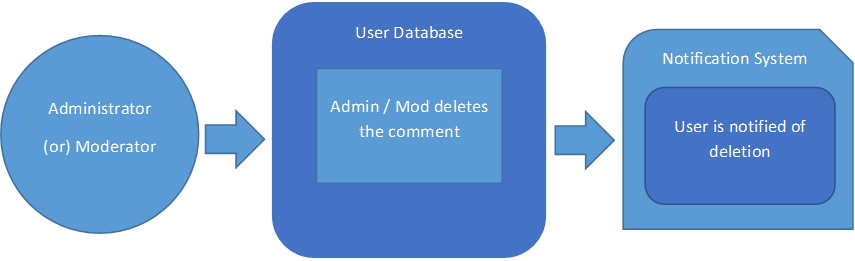
User Database

User Database

Admin flags user as banned

Notification System

User is notified of ban and whether or not they can appeal



1. Functional Requirements

|  |  |  |
| --- | --- | --- |
| **Number** | 01 | |
| **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** | 02 | |
| **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** | 03 | |
| **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** | 04 | |
| **Name** | User Will Receive Notifications | |
| **Summary** | User will receive notifications about their recipes | |
| **Priority** | 10 | |
| **Preconditions** | The user will have previously uploaded (a) recipe(s) | |
| **Postconditions** | The user will receive a notification | |
| **Primary Actor** | A user or moderator of the website | |
| **Secondary Actors** | The recipe interface and recipe database system | |
| **Trigger** | User leaves a comment for User’s recipe or a moderator removes a recipe | |
| **Main Scenario** | **Step** | **Action** |
|  | 1 | A user’s comment is stored in the database or user’s recipe is removed from the database |
|  | 2 | A notification is generated indicating the action taken in step 1 |
|  | 3 | Users ‘statistics’ page is updated under notifications with the message generated in step 2 |
| **Extensions** | **Step** | **Branching Action** |
|  | 3a | The system is unable to update the users statistics page and attempts to notify an administrator |
| **Open Issues** | What do we want to do if the system is unable to notify an administrator? | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Number** | 05 | | |
| **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** | 06 | |
| 1. **Name** | Comment on Recipes | |
| **Summary** | A user can comment on a recipe that has been posted on the site | |
| **Priority** | 3 | |
| **Preconditions** | The user will have an account and be logged into the site, viewing a recipe. | |
| **Postconditions** | The user's comment will be appended to the recipe | |
| **Primary Actor** | A user of the website | |
| **Secondary Actors** | The recipe interface and recipe database system | |
| **Trigger** | User clicks on the comment option for a recipe | |
| **Main Scenario** | **Step** | **Action** |
|  | 1 | System prompts user for comment input |
|  | 2 | User enters comment and hits submit |
|  | 3 | System adds comment for given recipe in database |
|  | 4 | System thanks user if successful |
| **Extensions** | **Step** | **Branching Action** |
|  | 4a | System informs user there was an error and returns to recipe |
| **Open Issues** | Do we want to enforce a character length limit on comments? Should we have profanity filters? | |

|  |  |  |
| --- | --- | --- |
| **Number** | 07 | |
| **Name** | Searching | |
| **Summary** | User will be able enter a search queries and receive a list of recipes based on relevance | |
| **Priority** | 3 | |
| **Preconditions** | The user will enter ascii text | |
| **Postconditions** | The user will see a list of related recipes | |
| **Primary Actor** | The User | |
| **Secondary Actors** | The search page and the recipes database | |
| **Trigger** | The user clicks on their user account link | |
| **Main Scenario** | **Step** | **Action** |
|  | 1 | User clicks on search link |
|  | 2 | User enters a search query |
|  | 3 | Database is searched and matching results are pulled |
|  | 4 | Results are sorted by relevance |
|  | 5 | Results are displayed to the user |
| **Extensions** | **Step** | **Branching Action** |
|  | 2a | If the user does not enter ascii text, a message will be displayed to the user |
|  | 3a | If not results are found, a message will be displayed to the user |
| **Open Issues** | How we will determine ‘relevance’ | |

|  |  |  |
| --- | --- | --- |
| **Number** | 08 | |
| **Name** | User Statistics | |
| **Summary** | User will be able to view statistics about their account | |
| **Priority** | 4 | |
| **Preconditions** | The User has made an account and is logged in | |
| **Postconditions** | The users statistics page will be populated with information about the user from the database. | |
| **Primary Actor** | The User | |
| **Secondary Actors** | The view (statistics page) and user database system | |
| **Trigger** | The user clicks on their user account link | |
| **Main Scenario** | **Step** | **Action** |
|  | 1 | User click on their view account link at top of page |
|  | 2 | Users statistics are pulled from the database |
|  | 3 | User is redirected to their statistics page |
|  | 4 | Statistics page is populated with User stats |
| **Extensions** | **Step** | **Branching Action** |
|  | 2a | The system is unable grab a statistic about the user from the database and an error message is displayed. If the system is unable to find the user in the database, an error is displayed and a notification is sent to an administrator. |
| **Open Issues** | Should other users be able to visit another users statistics page via URL? (add step about checking login) | |

|  |  |  |
| --- | --- | --- |
| **Number** | 09 | |
| 1. **Name** | Upload a Recipe | |
| **Summary** | A registered user can upload a recipe to the site | |
| **Priority** | 5 | |
| **Preconditions** | The user will have an account and be logged into the site | |
| **Postconditions** | The user's recipe will be uploaded for review by the moderators | |
| **Primary Actor** | A user of the website | |
| **Secondary Actors** | The user interface, file server, and recipe database system | |
| **Trigger** | User clicks on the option in his/her profile to upload a recipe | |
| **Main Scenario** | **Step** | **Action** |
|  | 1 | System prompts user for recipe name and description |
|  | 2 | User enters recipe name and description |
|  | 3 | System prompts user for recipe ingredients |
|  | 4 | User enters recipe ingredients |
|  | 5 | System prompts user for recipe steps and step pictures |
|  | 6 | User enters recipe steps |
|  | 7 | System prompts user for main recipe picture |
|  | 8 | User browses file prompt to select the recipe picture |
|  | 9 | User hits submit |
|  | 10 | Picture files are uploaded to the image server |
|  | 11 | Interface system sends request to database system to add recipe to recipe table with all entered information and image file information and pending flag set |
|  | 12 | Interface system notifies user if upload was successful |
| **Extensions** | **Step** | **Branching Action** |
|  | 6a | User selects to upload a picture for a step |
|  | 6b | System opens file browser to allow user to select file to upload |
|  | 6c | User selects file to upload and clicks upload |
|  | 6d | System caches file information to upload file when user hits submit |
|  | 8a | System informs user that he/she has selected an invalid file and reprompts |
|  | 9a | System reprompts user for missing required data (main picture, ingredients, description, steps) |
| **Open Issues** | Do we want a separate table for pending recipes or should we put them in the same table with a flag set for review? Should we have a limit on the number of images or file type/size? Should we allow videos? Should image file uploads begin asynchronously before hitting submit? How can we secure the uploading system?  Note: many of these prompts may take place at the same time but in linear succession on a single view. | |

|  |  |  |
| --- | --- | --- |
| **Number** | 10 | |
| 1. **Name** | Rate Recipes | |
| **Summary** | A user can rate a recipe posted on the site out of 5 stars | |
| **Priority** | 3 | |
| **Preconditions** | The user will have an account and be logged into the site, viewing a recipe. | |
| **Postconditions** | The recipe's 5 star rating will be adjusted according to the user's rating and the user will not be able to rate it again. | |
| **Primary Actor** | A user of the website | |
| **Secondary Actors** | The recipe interface/display and recipe database system | |
| **Trigger** | User clicks on the rate button near recipe's rating | |
| **Main Scenario** | **Step** | **Action** |
|  | 1 | System checks if user has already rated this recipe |
|  | 2 | Interface prompts user for rating out of 5 |
|  | 3 | User enters rating and confirms entry |
|  | 4 | System updates rating in database |
|  | 5 | System thanks user if successful |
| **Extensions** | **Step** | **Branching Action** |
|  | 1a | System informs user that he/she has already rated this recipe and returns to recipe |
|  | 5a | System informs user there was an error and returns to recipe |
| **Open Issues** | Do we want to prevent users from rating a recipe more than once? | |

|  |  |  |
| --- | --- | --- |
| **Number** | 11 | |
| **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** | 12 | |
| **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** |  | |

|  |  |  |
| --- | --- | --- |
| **Number** | 13 | |
| **Name** | Ban a user | |
| **Summary** | An administrator may ban a user, preventing them from using their account for anything besides appealing the ban. (Cannot post, edit, review, comment, etc) | |
| **Priority** | 3 | |
| **Preconditions** | The user must have an account. | |
| **Postconditions** | The user is flagged as “banned” and may no longer submit content, comment on content, review content, or otherwise participate in the site. | |
| **Primary Actor** | Administrator | |
| **Secondary Actors** | User | |
| **Trigger** | Administrator clicks a “ban user” link found on the offending users profile page, visible only to the Administrator. | |
| **Main Scenario** | **Step** | **Action** |
|  | 1 | Administrator clicks the “ban user” link. |
|  | 2 | The administrator is prompted to confirm their decision, and clicks “yes”. |
|  | 3 | The administrator is given the option of allowing the offending user to appeal their decision. |
|  | 4 | The user is flagged as banned and barred from further participation in the site. |
| **Extensions** | **Step** | **Branching Action** |
|  | 3a | If the administrator allows the user to appeal the decision, the user is notified about the ban and is given the option to appeal it. Pending a review, the user remains banned from the site. |
|  | 3b | The administrator who banned the user is notified of the appeal and given the chance to reexamine their decision. |
|  | 4a | The ban is upheld by the Administrator and the user remains banned. |
|  | 4b | The ban is dropped by the Administrator, the user is unflagged as “banned” and can once again participate in activities on the site. |
| **Open Issues** | Should a ban make the users profile page empty, unlisted, otherwise publicly indicated as “banned”?  Should there be tiers to the bans to strip certain privileges (like just posting comments), or to enact a site-wide ban that prevents them from even logging in?  Should ban notices be delivered as a Notification to the User, or as an email to the users provided email address? | |

|  |  |  |
| --- | --- | --- |
| **Number** | 14 | |
| **Name** | Remove a comment | |
| **Summary** | Administrators and moderators can remove comments. | |
| **Priority** | 4 | |
| **Preconditions** | The comment exists and the user removing the content is either an administrator or a moderator. | |
| **Postconditions** | The comment is removed from the database. | |
| **Primary Actor** | Administrator / Moderator | |
| **Secondary Actors** | Comment, User who posted the Comment | |
| **Trigger** | Administrator / Moderator clicks on the “remove comment” icon next to a comment. | |
| **Main Scenario** | **Step** | **Action** |
|  | 1 | “Remove Comment” icon is clicked. |
|  | 2 | The administrator / moderator is prompted to confirm their decision. (“Are you sure you would like to delete this comment?”) |
|  | 3 | The comment is deleted |
|  | 4 | The user who posted the comment is notified. |
| **Extensions** | **Step** | **Branching Action** |
|  | 2a | The administrator / moderator decides not to delete the comment. |
| **Open Issues** | Should the user be given the option to appeal the deletion of the comment? Should the moderator or administrator be able to reverse their decision? | |

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

See User Interface Design Document for <NAME>.

1. Deliverables

Our customer will receive hard copies of:

* System Requirements Specifications
* System Design Document
* User Interface Design Document
* User Manual
* Administrator Manual

Our customer will receive an electronic copy of:

* The web application
* All aforementioned hard copies of documents
* All source code
* Software required for the website to function properly

1. Open Issues

none?