**CSE361 Requirements Specification**

Team Maverick Applications Corporation:

Ryan Carlson, Stephen Pandorf, Jeremy Wagner, Kevin Rock

October 3, 2013

**1.0 Introduction**

The product we are developing is a “twitter-like” web application for use by employees of the company HAL to facilitate communication of employees regardless of geographical location. The information provided by the application will be built from information stored in a database. The information will be 200 character maximum “tweets” that will hold productive information for employees to share.

* 1. **Purpose**

The purpose of this project is to help employees of HAL collaborate by sharing “expert knowledge” on the subjects of which they are knowledgeable. Specifically, a coder may be searching for an algorithm to use in his work and would like to search within the company to find another coder with a solution. In our implementation, the coder in question can search for posts based on hashtags that indication subject matter. The information will be gathered though the posts, stored on the database. The posts will then be searchable though our interface by searching hashtags or reading a particular user’s posts.

* 1. **Document Conventions**

To aid the readability of this document, certain conventions will be used. The main purpose of the document itself is to convey the requirements, so emphasis is placed on the segmentation of requirements and their descriptions. Each requirement will be begin with R# where R indicates a requirement and # will be a unique number assigned to that requirement. Any technical or otherwise ambiguous terms will be defined in section 4.

* 1. **Intended Audience**

This document is intended for the client, and members of the software team that will be building the application. It is a contract between the client and builders of the application that states exactly what they would like to have built and what will be built. Some aspects of the application in terms of requirements will be explained at a high level, such as functionality. Other aspects will be described at a low level, as to emphasize the feasibility of the functions to be carried out. This document will aid the testers, as a clear understanding of the requirements will lend to the way in which it can be tested.

* 1. **Product Scope**

The scope of this product is limited to developing a web application to be rendered on either a desktop computer screen or mobile computer screen. The application will capture user input through posts that will be stored in the database for retrieval in a variety of ways that are unique to each user of the application. The functionality of the application will tailor the view every user gets by: following users and hashtags, or liking posts. Sections 2.2 describes these functionalities in greater detail. The information displayed will be dynamic, as every X minutes the page will automatically refresh to show the most current information. The application will be interactive, users will be able to sort posts that they see. Sorting is also described in Section 2.2.

**1.5 References**

[1] New Oxford American Dictionary

[2] twitter.com/about

**2.1. Product Perspective**

This software will consist of a web application which the company will use as a social media interface to communicate with employees.

**2.2 Product Functions**

**R1: User login**

R1.1 Username/password necessary to login.

R1.2 If username/ password incorrect, popup ‘username or password are incorrect’.

R1.3 Password recovery is done through email, the expected password is sent to the email linked to the user’s account.

R1.4 The user will be locked out (account can’t be accessed) after 7 failed attempts, this will last for 15 minutes. After that the use has 7 attempts to log in again before another lockout.

R1.5 After 7 minutes of not using the application (no user input), the application logs the user out.

**R2: Account creation**

R2.1 Anyone with an email address that is not already linked to an account can create an account.

R2.2 To create an account, user enters First Name, Last Name, email, unique username, and password (twice—make sure it matches)

R2.3 Upon creation, before login, the user must confirm (verify) email. The process takes place as follows:

1. A code is sent to their email
2. Upon attempted (first) login, until they enter the correct email verification code, they can’t finish the login.

**R3: Home Page**

R3.1 Upon successful login, go to home page

R3.2 The home page consists of two main parts:

1. Posts you made, and posts from people/ hashtags you follow
2. Posts you like

R3.3 A logout button is visible in the top right corner that returns the user to the login page.

R3.4 The Home page is accessible from anywhere in the site once logged in.

**R4: Searching**

R4.1 Available only on the homepage.

R4.2 Search requires specification of the search term (name, username, or hashtag), this is done using a dropdown menu.

R4.3 Hashtags, usernames, First/ Last Names are searchable (Plain text is not)

R4.4 If searching with an empty string, every user is listed out in alphabetical order by username.

R4.5 Search always covers entire site.

**R5: Sorting**

R5.1 Default sorting for posts is chronological order.

R5.2 Default sorting for hashtags and users is alphabetical sorting.

**R6: Following**

R6.1 A user can follow users and hashtags.

R6.2 When clicking on a username, the application will go to a page where their posts are displayed. A button next to their username called follow will allow them to follow the user.

R6.3 If you are on another user’s page and are already following them, the follow button will instead say unfollow, and will make it so their posts no longer show up on the main page.

R6.4 You cannot follow yourself. If viewing your own page the follow button does not appear.

R6.5 Following users makes their posts show on the main page by most recent post.

R6.6 Following hashtags makes posts with that hashtag show in the main page by most recent post.

R6.7 If another user is following your user account, you can see who is following you by username.

**R7: Liking**

R7.1 A user can like posts only.

R7.2 Liked posts show up on the home page in the favorites section.

R7.3 Every post will show how many likes it has as a non negative integer.

R7.4 A user can unlike a post after they like it. This is done by changing the like button to an unlike button.

**R8: Posting**

R8.1 Users can write ASCII text into a text box up to 200 characters.

R8.2 A post button will accept the input and store it on the database as one of their posts.

R8.3 Links are allowed to be posted but will not be converted into hyperlinks.

R8.4 A character counter will be shown next to the text box that indicates the number of characters remaining before the user meets the 200 character limit.

R8.5 When the user has 200 characters in the text box and attempts to type more, the application will not allow more characters to be added, the user can delete.

**R9: Hashtags**

R9.1 Any consecutive string of characters except space after the character “#” will be considered a hashtag.

Ex: #ethnic\_cuisine (hashtag is “ethnic\_cuisine”)

#ethnic cuisine (hastag is “ethnic”)

R9.2 There can be multiple hashtags per post.

R9.3 When displaying posts, the hashtag will be shown in bold.

**R10: Blocking**

R10.1 A block is the action of having another user unfollow you.

R10.2 A block is a one time action; the user can refollow you again.

R10.3 A user can block another user by navigating to their account settings page and clicking a block button next to the user who is indicating as following them.

**R11: Account Settings**

R11.1 A user will be able to view their account information. This information will include their username, email address, and a list of who is following them.

**R12: Replying**

R12.1 A user can reply to a post. This will show up as a comment below the original post.

R12.2 All replies will show up in most recent order below the original post.

R13.2 Only the original post can be liked.

**2.3 User Classes and Characteristics**

Users of this system will fall into two categories: administrators and employees. Administrators will handle user errors and maintenance of the application. Employees will post and read messages other employees have posted.

**2.4. Operating Environment**

The web application will run on Firefox, Chrome, and mobile devices. The backend of the application will be a MYSQL database on the CSE server. The application will work on Windows, Mac, and Linux operating systems.

**2.5. Design and Implementation Constraints**

The application is limited by a few constraints. First, storage is limited by the space on the CSE server. Second, the application must scale for a number of screen sizes and must be usable on mobile devices. Third, the application must refresh the message feed at regular intervals. Lastly, the application must maintain

**2.6. Assumptions and Dependencies**

The application will be dependent upon a database stored on the CSE server. It will not function outside this environment without further development. The users of the website will use up to date web browsers. Functionality may suffer on older browsers.

**2.7 Errors/Edge Cases**

The system will log user errors and display a message based on the error encountered. General edge cases will be tested against. If an edge case results in an error it will be logged by the system.

**3.1 User Interfaces**

The user interface will be a dashboard page that can lead your navigation in the application. Features needed include the Main or Dashboard page, a page for new users without an account to create an account, a Forgotten password page which will allow you to reset your password, a profile page with a profile picture specific to that user's profile, and an account settings page to change your password or profile picture. See last page for mockup for user interface.

**3.2 Hardware Interfaces**

The software will require a MySQL database for information tied to users including their tweets and profile picture.

**3.3 Software Interfaces**

This project will use a Java Applet to connect to the MySQL server from JavaScript. <Certain software will be used for the dashboard portion of the product and will be determined by the implementing team and this is a placeholder for that information.>

**3.4 Communication Interfaces**

The project will use communication between the MySQL database and the browser by using a Java Applet to connect between the server-side MySQL and client-side JavaScript.

**4.0 Glossary of Terms**

* **Hashtag** – A sequence of characters beginning with “#”. Indicates the subject matter of a post.
* **Post** – A 200 maximum sequence of characters that a user shares information through.
* **Twitter** – An online service that allows users to post messages and follow other users [Twitter.com].
* **Database** – A structured set of data held in a computer, esp. one that is accessible in various ways [New Oxford American Dictionary].
* **Follow** – A function a user can perform on another user or hashtag to make their posts appear on their main page.
* **Like** - A function a user can perform on a post to make the post appear in their favorites.
* **ASCII –** A set of digital codes representing letters, numerals, and other symbols, widely used as a standard format in the transfer of text between computers [New Oxford American Dictionary].
* **Hyperlink** - A link from a hypertext file or document to another location or file, typically activated by clicking on a highlighted word or image on the screen [New Oxford American Dictionary].

**5.0 Appendix A**

**Milestone 1 (10/10):**

HTML: Start creating basis of what the site will look like.

PHP: Create classes for relevant objects.

Javascript/jQuery: needs to wait for html to be written?

mySQL: create tables based around what is being done in PHP

**Milestone 2 (10/17):**

HTML: Continue creating HTML.

PHP: Begin creating methods as per functional requirements.

Javascript/Jquery: begin linking relevant HTML pages together with javascript.

mySQL: create detailed tables with relations between records in tables.

**Milestone 3 (10/24):**

HTML: Finish creating majority of HTML.

PHP: Continue creating methods as per function requirements.

Javascript/Jquery: continue linking relevant HTML pages together with javascript.

mySQL: finish creating majority of mySQL tables and relations.

**Milestone 4 (10/31):**

PHP: Start writing code to write/retrieve data to/from database.

Javascript/Jquery: start writing scripts to envoke appropriate php code from the buttons in html

**Milestone 5 (11/7):**

PHP: continue writing code to write/retrieve data to/from database.

Javascript/Jquery: continue writing scripts to envoke appropriate php code from the buttons in html

**Milestone 6 (11/14):**

PHP: finish writing/retrieval code

Javascript/Jquery: finish writing scripts to envoke appropriate php code

**Milestone7 (11/21):**

Testing: create test cases and test application.

