**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**

R0: Users will be able to create a login

--Each login will have unique email and vice-versa

--when the user creates a password, a password strength tester will appear onscreen.

R1: Users can search for hashtags

R2: Users will be able to create posts with a maximum of 200 characters

--hastags are part of the character count

--hashtags themselves have no limit

--all posts are visible to everyone in the system

--posts will only consist of ASCII characters and hashtags

R3: Users will be able to follow posts/ people

R4: Users will be able to block other users

--Even if blocked, a user will be able to search for content that the person who blocked them posted

R5: Posts will be sorted by chronological order

--When created, a post will automatically have a timestamp

R6: Users will be able to like posts

--Unliking will also be supported

R7: While composing a post, the user should see on onscreen character count

R8: Users should be notified (through email) when someone follows them

R9: Passwords should be recoverable through email

R10: After x failed login attempts, the account is locked out for y minutes

R11: Hashtags should stand out from plaintext

--Each post should be able to have multiple hashtags

R12: When a user logs in, they should go to a home screen

--The home screen should show what you’ve liked in one pane

--another part of the screen should show the posts you’ve made

R13: When looking at a post, no one will be able to see who liked the post

--Everyone will be able to see how many people liked the post

R14: Users will be able to search by username

--Also searching by First Name or Last Name is supported

R15: When a user gets blocked, they should receive a notification

R16: Users will be able to comment on other posts.

**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.

**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.

**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. (Kevin TODO: update)

PHP: Create classes for relevant objects. (Ryan TODO: update)

Javascript/jQuery: needs to wait for html to be written? (TODO: update)

mySQL: create tables based around what is being done in PHP (TODO: update)

**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.