Software Requirements Specification for Twitter Project-Fall 2014 Group 1

Twitter Project

Patrick Burton
Arvind Nair
Lakshmi Swathi Chavvakul

Guidance Under Dr. James Hill
Version 4.0 Approved

1. Table of Contents

2. Revision History

Name	Date	Reason For Changes	Version
Iteration 1	09/23/2014	Implementing Iteration 1	V 1.0
Iteration 2	10/15/2014	Implementing Iteration 2	V 2.0
Iteration 3	11/20/2014	Implementing Iteration 3	V 3.0
Iteration 4	12/04/2014	Implementing Iteration 4	V 4.0

3. Introduction

3.1 Purpose

This software is built to satisfy the basic requirements of the twitter like service to be implemented.

The Twitter like service must satisfy the 3 basic requirements:

- User should be able to tweet from the web service
- User should be able to tweet from mobile device
- A tweet from one source(i.e., web or mobile) should show up in the other source.

This software satisfies all the 3 basic requirements.

Extra Features implemented above the 3 basic features:

- CAS Authentication.
- Ability to change username for tweeting purposes.
- Load more option to load more previous tweets.
- Time stamping of messages.

The version 4.0 of the software has been approved and deployed.

3.2 Document Conventions

This document follows the IEEE format of SRS Documentation although it has been modified slightly to suit our needs.

3.3 Intended Audience and Reading Suggestions

This document is to be read by those who are going to use the software. This document can also be read by those who wish to evaluate the software and also modify or add additional changes to this software.

3.4 Product Scope

This product is intended to satisfy the 3 basic requirements as mentioned in the purpose (4.1) in this document.

3.5 References

- The requirements document provided by Dr. James Hill for building the Twitter like service.
- Detailed Design Document which contains all the design diagrams and details for this software.

4. Overall Description

4.1 Product Perspective

The software product is intended to satisfy a messaging style in which users can send messages amongst themselves. Everyone who logs in can see the messages posted and can type and send their own messages in real time. The person must be a registered IUPUI user.

4.2 Product Functions

The users of this software can:

- Log in
- Tweet
- View previous tweets
- Change Usernames

4.3 User Classes

A user has to just login to post a tweet. So there is only one user class that is the user.

4.4 Operating Environment

This software can operate in the following environments:

- Windows XP/Vista/7/8
- Mac

- Linux
- Android and iOS

4.5 Design and Implementation Constraints

- This software can only be used by IUPUI registered users. So to use this software one must be an IUPUI student or a faculty member.
- Data is available to system administrator and it must be used with care.
- The server on which it is running should not fail, then the software will fail.
- User must be connected to the internet to use this service.
- Data must be backed up at all times in case of faults or failures in the main system.

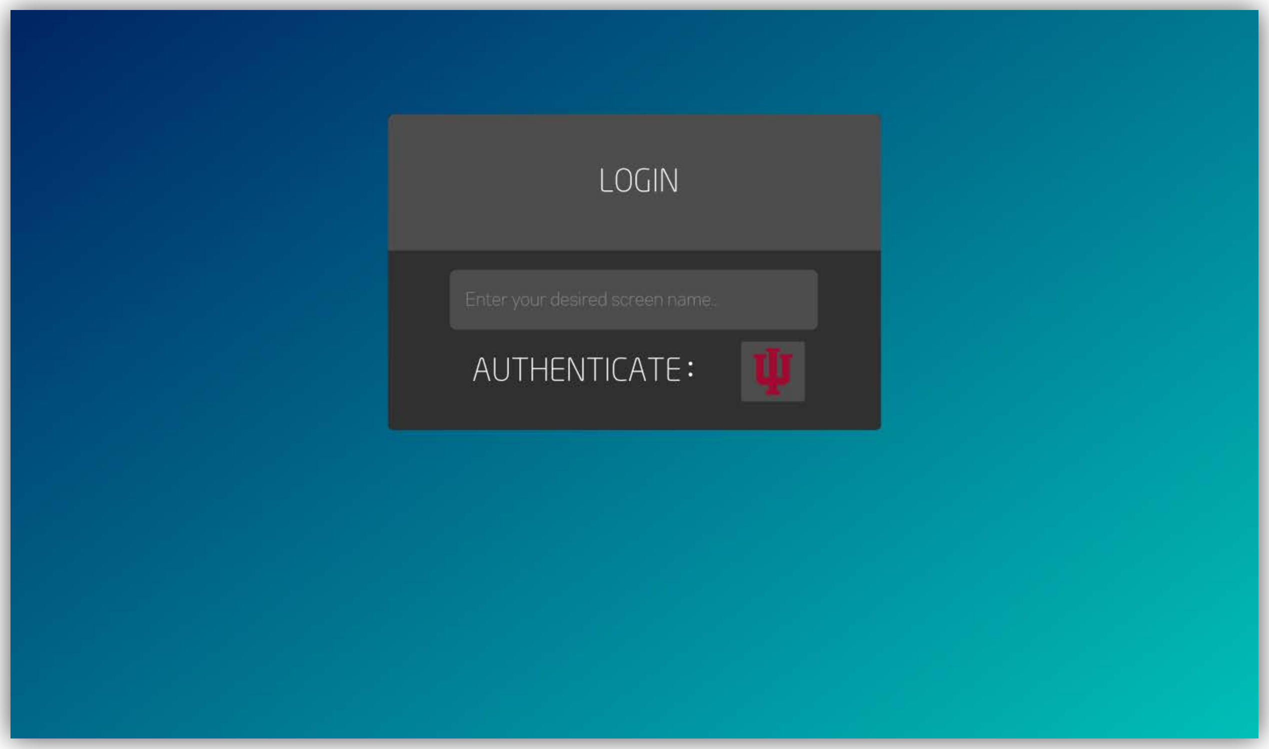
4.6 Assumptions and Dependencies

- To use this software the user must already be a registered IUPUI user.
- The main server is always running properly and there are no hardware, software, power or connection faults.
- The user is connected through the internet to use this software.
- The mobile devices the user uses are smart phones.
- The software is available at all times.
- There must be a web browser installed on both the PC or Desktop and mobile devices to load the website.

4.7 Screenshots

4.7.1 Login Page

The image below shows the login page screenshot.



LOGIN

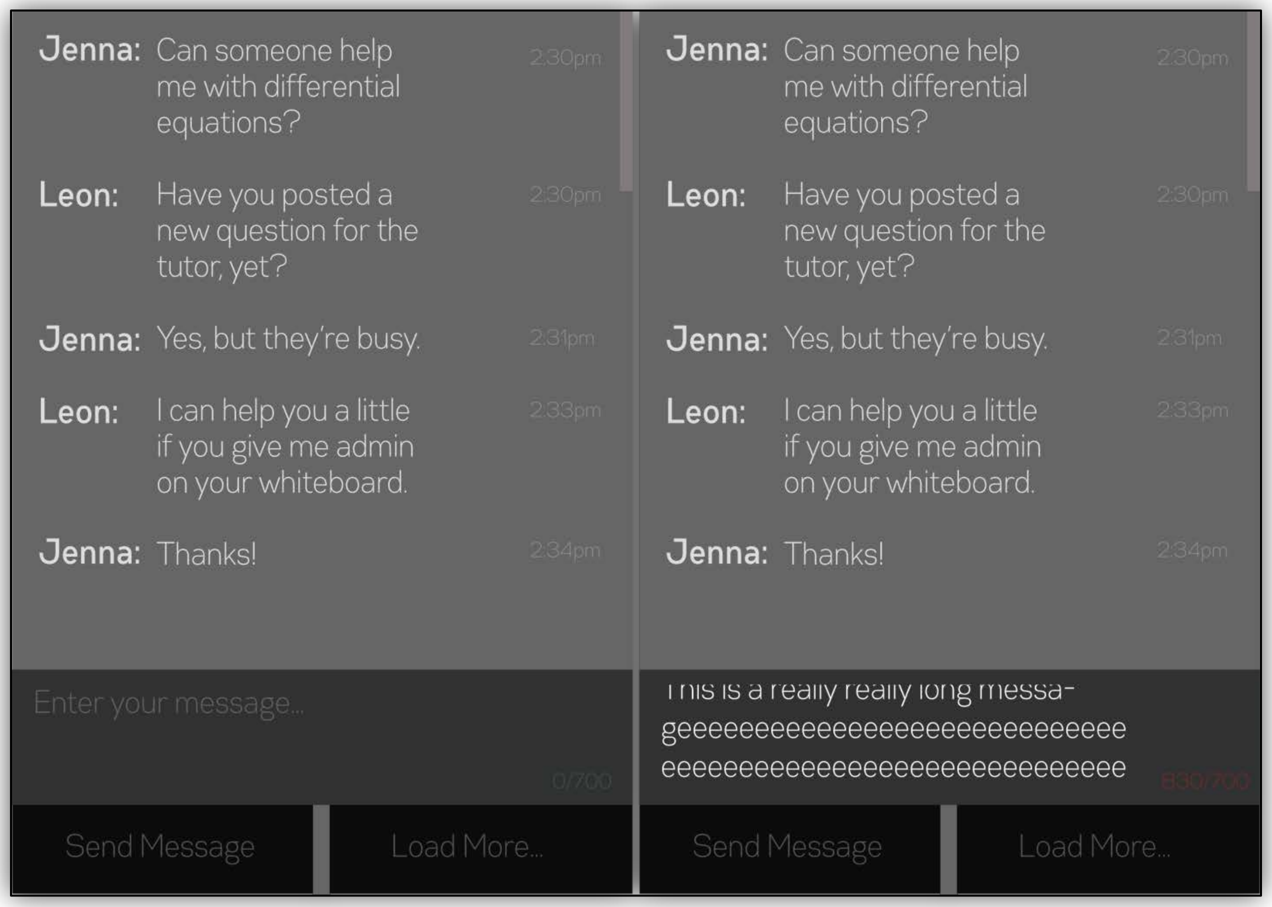
Enter your desired screen name..

AUTHENTICATE :

Figure 1 : Login Page Screenshot

4.7.2 Chatbox

The image below shows the chatbox.



Jenna: Can someone help me with differential equations? 2:30pm

Leon: Have you posted a new question for the tutor, yet? 2:30pm

Jenna: Yes, but they're busy. 2:31pm

Leon: I can help you a little if you give me admin on your whiteboard. 2:33pm

Jenna: Thanks! 2:34pm

Jenna: Can someone help me with differential equations? 2:30pm

Leon: Have you posted a new question for the tutor, yet? 2:30pm

Jenna: Yes, but they're busy. 2:31pm

Leon: I can help you a little if you give me admin on your whiteboard. 2:33pm

Jenna: Thanks! 2:34pm

This is a really really long messa-
geeeeeeeeeeeeeeeeeeeeeeeeeeee
eeeeeeeeeeeeeeeeeeeeeeeeeeeeee

Figure 2 : Chatbox Screenshot

5. External Interface Requirements

5.1 User Interfaces

Add Screenshots.

5.2 Hardware Interfaces

- A PC installed with Windows XP/Vista/7/8, Mac or Linux.
- A smart phone with Android or iOS operating system.

The devices must be connected to the internet either through the LAN cable or through wireless network.

5.3 Software Languages used

5.3.1 Front End:

TypeScript, HTML5 (JavaScript, CSS and HTML).

5.3.2 Back End:

PHP, MySQL, Apache and Ubuntu (Apache server running on Ubuntu).

5.4 Communication Interfaces

The software uses HTTPS protocol to communicate over the internet.

6. Functional Requirements

The requirements below are arranged as per their respective iterations:

6.1 Iteration 1

- ID 1: User should be able to locate and load the website.
- ID 2: Use HTML5 for proper accessibility on both desktop and mobile devices.
- ID 3: User should be able to enter username into username field.
- ID 4: User should be able to enter password into password field.
- ID 5: User should login on clicking the login button.
- ID 6: User should be able to type in a message into the text field.
- ID 7: The message should be displayed alongside other messages on the screen.
- ID 8: The user must be able to log out of the system.
- ID 9: Another user's tweet must be visible on the user's screen.
- ID 10: User's name should be displayed alongside his tweet.
- ID 11: Use file to store data
- ID 12: Data is not visible to unauthorized people.
- ID 13: The system must be able to validate the user's identity.
- ID 14: User must perform the appropriate functions its role has been assigned to.

- ID 15: The system must be available to all and not down due to vulnerable attacks.

6.2 Iteration 2

- ID 16: Using sessions for duration of login.
- ID 17: Using CAS logout functionality.
- ID 18: Using HTTPS.
- ID 19: User can retry login only certain number of times(as per IU authentication specification).

6.3 Iteration 3

- ID 20: Use mySQL for storing user data.
- ID 21: Use load more option which will load more previous chats.
- ID 22: Add maximum length of characters to be sent as a tweet.
- ID 23: Send button is present for clicking and sending tweets.
- ID 24: Send box highlights on clicking it.

6.4 Iteration 4

- ID 25: Deploy on AWS (Amazon Web Services) the website.
- ID 26: Website must be accessible to users on all devices.
- ID 27: Ability to change usernames (we decoupled screen name from authentication).
- ID 28: Recording of Timestamp of tweets for unique identification of usernames tracked.
- ID 29: Tweets stored in database as real User ID.
- ID 30: Ability to keep track of custom names.
- ID 31: Improving aesthetics or the look and feel of the website.

7. Non-Functional Requirements

7.1 Performance Requirements

- The performance requirements are not much since the software does not occupy much memory. Only the tweet relevant information and the user specific details like who tweeted the messages and timestamps are stored.
- The system should work on most of the devices even on low memory as the user needs to only load the web page on his browser.
- However the user must have a good internet connection to load the website. If it is too slow it may take long to load the website or if the internet disconnects the website will not work on the particular device.
- If on mobile the internet while accessing the website disconnects then the website will be difficult to access and will not work.
- The server must be continuously running to accept requests anytime and as we have deployed it on AWS it should be have a good uptime.
- There must be no faults in hardware either on the user's devices or in the server.

7.2 Security Requirements

- The database must be kept confidential by the system administrators as if they share the information with unauthorized people then it would be a violation of privacy, integrity and confidentiality of the system.
- The Usernames and passwords are maintained by the IUPUI CAS server so there is no storage at the system so the security is very strong.
- The users must keep their passwords with themselves and not share with others as it would then lead to access by other users via impersonation.

8. References

1. *Software Engineering Modern Approaches Second Edition* Eric J. Braude and Michael E. Bernstein, Wiley Publications.

2. B. W. Boehm, J. R. Brown, M. Lipow, *Quantitative Evaluation of Software Quality*, TRW Systems and Energy Group, (1976).

3. SRS template is taken from (some parts) http://www.cs.gmu.edu/~kdobolyi/cs421/srs_template.doc Copyright © 1999 by Karl E. Wiegers. Permission is granted to use, modify, and distribute this document.