**Software Requirements Specification**

**for**

**Social Networking Web Application**

Version 1.0 approved

02/29/2016

Yu Luo

Yuming Peng

Xinyu Gao

Hongtao Zhang

Hemant Pandey

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Description** | **Author** | **Comments** |
| 02/29/2016 | Version 1 | Team | First draft |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

|  |  |
| --- | --- |
| **REVISION HISTORY** |  |
| **1. INTRODUCTION** |  |
| 1.1 PURPOSE |  |
| 1.2 SCOPE |  |
| 1.3 DEFINTIONS, ACRONYMS, AND ABBREVIATIONS |  |
| 1.4 REFERENCES |  |
| 1.5 OVERIVEW |  |
| **2. GENERAL DESCRIPTION** |  |
| 2.1 PRODUCT PERSPECTIVE |  |
| 2.2 PRODUCT FUNCTIONS |  |
| 2.3 USER CHARACTERISTICS |  |
| 2.4 GENERAL CONSTRAINTS |  |
| 2.5 ASSUMPTIONS AND DEPENDENCIES |  |
| **3. SPECIFIC REQUIREMENTS** |  |
| 3.1 EXTERNAL INTERFACE REQUIREMENTS |  |
| *3.1.1 User Interfaces* |  |
| *3.1.2 Hardware Interfaces* |  |
| *3.1.3 Software Interfaces* |  |
| *3.1.4 Communications Interfaces* |  |
| 3.2 Functional Requirements |  |
| *3.2.1 Use Case* |  |
| 3.3 NON-FUNCTIONAL REQUIREMENTS |  |
| 3.3.1 Performance |  |
| 3.3.2 Reliability |  |
| 3.3.3 Availability |  |
| 3.3.4 Security |  |
| 3.3.5 Maintainability |  |
| 3.3.6 Portability |  |
| 3.4 DESIGN CONSTRAINTS |  |
| 3.5 OTHER REQUIREMENTS |  |
| **A. APPENDICES** |  |
| A.1 Appendix 1 – Login and Register Page |  |
| A.2 Appendix 2 – Home Page |  |
| A.3 Appendix 3 – Friend Search Page |  |
| A.4 Appendix 4 – Personal Profile Page |  |
| A.5 Appendix 5 – Chat Page |  |
| **Priority Table** |  |

**1. Introduction**

This section is describing the definition and all details of the social networking web application. In the document, the purpose and its functionalities are provided.

**1.1 Purpose**

The purpose of this document is going to describe the requirements for the social networking web application. All system constraints, interfaces, functionalities, performances and assumptions are listed in below contents. The document can give detailed guidance for future design and implement job.

**1.2 Scope**

The social networking application is a web-based application for messaging information between people. Also, the application can share other kind information, like image, posts and so on.

A user can create an account and make friends online. When they are connected, they can use application to send any information they want. Also, user can create a group to send message to multiple friends at same time.

**1.3 Definitions, acronyms, and abbreviations**

|  |  |
| --- | --- |
| **Term** | **Meaning** |
| User | Any individual who would be using the application for communication. |
| Client | The organization/user for whom the application is being developed. |
| Developer | The individual responsible for developing the application. |
| Tester | The individual responsible for testing the application for errors and bug fixes. |
| Existing User | A user who currently holds an account/profile in the social media website and is a returning user. |
| New User | A user who would be new to the website and has never used it before. |
| Sign In/Log In | An option that allows the user to enter an existing user into the website using his/her username and password. |
| Sign Up/Register | An option that would allow new users to create account/profile in the website by filling relevant information in the form. |
| Friend | Any individual who would be using the application for communication. |
| Add Friend | An option that adds any other users to user’s friend list, and they can start chatting with user |
| Delete Friend | An option that deletes any friend from user’s friend list, and they cannot communicate with user any more. |
| Block Friend | An option that block friend in user’s friend list, and that friend cannot see user’s post |
| Post Message/Images | An option that posts user’s images or text messages to his friends. |
| Like | An option that user shows ‘like’ to other’s post |
| Comment | An option that user comments his words on some posts. |
| Search Friend | An option that look up friends online |
| Send Message | An option that send message to the target. |
| Chat History | A window that shows user’s chat content before. |
| Post History | A window in Profile Page that shows user’s previous posts. |

**1.4 References**

[1] Pilskalns, O. Hagemeister, J. (2005). *Software Requirements Specification Template*. Retrieved from <http://www.tricity.wsu.edu/~mckinnon/cpts322/cpts322-srs-v1.doc>

[2] Geagea, S. Zhang, S. Sahlin, N. Hasibi, F. Hameed, F. Rafiyan, E. Ekberg, M. *Software Requirements Specification Amazing Lunch Indicator.* Retrieved from

<http://www.cse.chalmers.se/~feldt/courses/reqeng/examples/srs_example_2010_group2.pdf>

**1.5 Overview**

The last part of the document includes two sections, appendixes and priority.

The section two describes general information about the web application: perspectives, functions, user characteristics, constraints and assumptions. All these contents help readers understand the application.

The section three provides the requirements of user interface. The section will list functional requirements and non-functional requirements with use cases. Also, it will clearly describe the constraints and other requirements.

The appendixes list User Interface design for our five features.

The priority table list all functional requirements with different priority level. The next team should implement all high level functions. Also, the low priority functions are optional.

**2. General Description**

This section will describe the general parameters and requirements for the social networking web application. In this section, the descriptions of the assumptions, functions, user characteristics and constraints of the application are the main content. All these information could help reader understand the application clearly.

**2.1 Product Perspective**

A web container, like Tomcat, should host the social networking web application server. Then the application should listen to any messages or actions from front-end page and do the correct work.

Based on our design work, there are five pages: Home Page, Search Page, Login Page, Chat Page and Personal Profile Page, and each page could jump to other pages.

**2.2 Product Functions**

**2.2.1 Home Page**

Home Page provides several main functions: post message, review others’ posts, view friend list, process notifications and delete/block friend.

Also, there are some extra functions: jump to Search Page, jump to Personal Profile Page, jump to Chat Page, logout, zoom in images, chat request notification and etc.,

There are four main functions, and each has their functionalities:

1. Post message function could post user’s text, sticker and pictures to his friends.

2. Review others’ posts function could let user see friends’ posts. Also, use can write their comments and show ‘like’ sign below the post.

3. View friend list function could help user manage his contact list.

4. Delete/block friend function could delete or block friend from your contact list.

5. The chat/friend request notification will show on the top of the Friend List. User chooses add/view/chat action.

And, the extra functions could help user jump to other pages and watch post images clear.

**2.2.2 Personal Profile Page**

The profile page allows a user to check his own, his friend’s or a stranger’s profile. The information in profile page includes a photo, basic information, contact information, posts history and etc., If the user is on his own profile page, he can edit his profile; if user is on his friends’ profile, he can start a chatting dialog with his friends; if user is on a stranger’s profile page, he can send a friend request to the stranger.

**2.2.3 Friend Search Page**

Search Friends Page provides three main functions: search friends, friend recommendation, and add friends.

Also, there are some extra functions: jump to Home Page, jump to Personal Profile Page, Logout and etc.

Each of these three main functions has its own clear functionalities.

Search function could search friends by username, location, school, profession, etc.

Recommend friends function can recommend potential/possible friends to user based on the user’s current contacts/friend list.

Add friend can add someone into user’s friend list.

The functionalities of the extra functions could help user jump to other pages.

**2.2.4 Chat Page**

Chat Page provides four main functions: type messages, send messages, check history and add friends.

Each of these four main functions has its own clear functionalities.

Type messages could let user to type his own messages into message window, user could also add emoji to message window.

Send messages function can send messages from message window to display window.

Add friend can add someone into group chat room.

Check history can let the user to check messages several days before.

**2.2.4 Login and Register Page**

TODO: add brief introduction of your page’s function here.

**2.3 User Characteristics**

There is no limitation for users. It is an online web application, and users could register and chat with other friends.

**2.4 General Constraints**

The application is working in web container that user has to connect to Internet for using application’s functionalities.

Also, this application is only a social networking application that it cannot support for large file transaction or complex group works. The application has the similar functionalities with Facebook.

**2.5 Assumptions and Dependencies**

The application is working in a web container, like Tomcat.

The web container should work for any kind devices to access, like the browser application of computer or mobile.

**3. Specific Requirements**

This section provides a detailed introduction of the five pages of the social networking application. It guides the future design, implementation and testing with specific requirements of the application.

**3.1 External Interface Requirements**

**3.1.1 User Interfaces**

**3.1.1.1 User Interfaces – Home Page**

The user interface of Home Page should be brief for user to view. Each button or window should has a clear naming that user could understand its purpose.

Also, if user is the first-time user, he could see some guidance on screen. This tutorial could help user to be familiar with this application.

In the Home Page window, there are three parts: Friend List, Post Input Window and Post View Window.

In the Friend List, user could see all his friends’ pictures, their usernames and their simple information. Also, user could delete/block/release friend on that list. To start chatting with friend, user just left click on the friend’s name item on the Friend list, and the page will jump to chat window. Sometime, user holds its mouse on name item to see more information of his friend, or makes right click to see friend’s personal profile. At the bottom of friend list, user could click button to jump to Friend Search Page for searching friends. When a chat request or friend request is sent to user, user could see a chat request/friend request notification on top of Friend List with signal.

In the Post Input Window, user could type in text message, add stickers or insert pictures in his post. User could only add at most five pictures because of size limitation. After finishing typing message, user clicks “submit” button to post his messages.

When user posts his message or picture, he and his friends could see this post in Post Viewer Window. The posted message includes some parameters: user’s picture, messages/posted picture, date and time, ‘like’ button, comment button, ‘like’ list and comments list. Each post is separated by a line and is ordered by timestamp.

There are three buttons on the top of Home Page: Search Friend, Personal Profile, and Home Page. All these buttons could help user jump to different pages and do different actions.

At the bottom of the Home Page, there is a Logout button for user to logout and return to Login Page.

**3.1.1.2 User Interfaces – Personal Profile Page**

The user interface is a page showing the profile of a user.

On top of the page are three buttons which lead to the Home Page, Friend Search Page and Personal Profile Page.

Below three buttons displays the user’s photo, background image and name. If this is the profile page of the user own, there will be an “Edit” button in the right corner, which allows a user to edit his profile. If this is a friend’s profile page of the user, there will be a “Chat” button in the right corner, which allows him to chat with his friends. If this is a stranger’s profile, there will be an “Add” button in the right corner, which allows the user to send a friend invitation.

The second section shows details of user’s profile. There are three tabs on top of this section and clicking them respectively shows: 1) basic personal information of users; 2) contact information of users; 3) user’s own posts history.

If the user is editing his profile, there will be a “Submitting” at the bottom of this page which allows him to save his information.

**3.1.1.3 User Interfaces – Friend Search Page**

In the Friend Search Page, there are two major parts: Search area and recommendation area.

In search area, the user could input friends’ name/nickname, their location (town, city, state, and country), school (high school, college), profession, etc. to search friends. User can search by single condition or multiple conditions at the same time to filter friends. When the user click on the search button beside the search box, the application will send query to database and retrieve the result.

Once the specific friend or friends are found, the result will show up right underneath the search box. Each friend will have a profile image and brief introduction of that friend (see appendix). If the user click on one of the list, he/she will be able to see that friend’s specific profile information. There is an Add button on the right side of each friend record. When the user click on the Add button, a request will be sent to the specific friend. If that friend process the friend request, then that friend will be added to user’s friend list.

Under the search area, there is a recommendation area. User will see a list of possible or potential friends. This list is created based on the user’s current friend list. For example, if there are two friends A and B on the user’s friend list and they have a same friend C who is not on the user’s friend list. Then friend C will be recommended to the user. And the user can click on the Add button to add friend.

**3.1.1.4 User Interfaces – Chat Page**

UI-1: Basically, after we click the chat button of the friend list feature on the main page, it will directly open a new dialog frame, which is chat system's interface.

UI-2: After the system create a dialog box, it should permit complete communication and search functions using keyboard and mouse.

UI-3: if the user want to type in message, they could use keyboard to type their messages and these messages will be shown on the typing window of this dialog frame. And the user are only allowed to type 155 words one time.

UI-4: After the user type in messages, they should click send button which on the bottom right corner of the whole dialog frame.

UI-5: The window on the top of dialog frame includes all chatter's name and a button called add friends which on the top right of the frame.

UI-6: After we click the ‘add me’ button, it will achieve the same functions of the searching friends, and the system could let the users to add more friends to chat.

UI-7: After the user click the send button, then the message will be shown on the central window of the dialog frame. And the form of this window is shown below.

Name (time, day)

Message

UI-8: The system should have another window which is on top of the typing window, on the left side of the window, it should have two buttons which is emoji button and photo button, and on the right side of the window, it should have a history button.

UI-9: Once the user click the emoji button, then the system will open another window where user can select different emoji to send. Once the user click the photo button, then the system will open another window where user can select photos of certain format to send.

UI-10: Once the user click on the history button, the system will open another window which stored the history of the chatting messages. And these messages are separated by day.

UI-11: If the user have some invalid operation, then the system will announce the user what to do.

**3.1.2 Hardware Interface**

The computer has enough hardware to support this social networking application.

**3.1.3 Software Interface**

The back-end server should be implemented by Java, and the front-end UI should be implemented by using JavaScript, PHP and html. And the data should be stored by database, like MySQL.

**3.1.4 Communications Interface**

Each page should send the entire information in the transacting message to back-end server. Then back-end server could recognize the message, update the data into database, send the message to target user and reload the web page.

**3.2 Function Requirements**

**3.2.1 User Case – Login and Register**

**3.2.1.1 Functional Requirements 1**

|  |  |
| --- | --- |
| **ID** | FR-1 |
| **Title** | Sign In |
| **Description** | An existing user will be able to sign in by entering correct username and password. |
| **Pre-condition** | The user opens the web application. |
| **Post-condition** | Successful validation allows user to sign in and proceed further, whereas unsuccessful validation leads to invalid credentials page. |
| **Trigger** | The user enters username and password, and hits submit button. |
| **Exception** | No internet access, system malfunction, wrong username and password. |
| **Assumptions** | User has a web browser |

**3.2.1.2 Functional Requirements 2**

|  |  |
| --- | --- |
| **ID** | FR-2 |
| **Title** | Sign Up/Register |
| **Description** | A new user will be able to become member of the social network by entering his information, followed by some security questions and checking the “I AM NOT A ROBOT” re-captcha. |
| **Pre-condition** | The user opens the web application. |
| **Post-condition** | Successful registration allows the user to create his profile. |
| **Trigger** | The user enters registration information, and hits register button. |
| **Exception** | No internet access, system malfunction, wrong input, forgets to enter mandatory fields. |
| **Assumptions** | User has a web browser |

**3.2.1.3 Functional Requirements 3**

|  |  |
| --- | --- |
| **ID** | FR-3 |
| **Title** | Forget Password |
| **Description** | An existing user will be able to retrieve his/her password by answering security questions. Successful answering would send an email to the user containing the password. |
| **Pre-condition** | The user forgets password/username and enters wrong credentials. |
| **Post-condition** | Invalid credentials page will be shown and the user will receive an email containing the password by answering correctly to the questions. |
| **Trigger** | The user hits forgot password button. |
| **Exception** | The user is not an existing user, wrong answer to questions. |
| **Assumptions** | The user is a registered user. |

**3.2.1.4 Functional Requirements 4**

|  |  |
| --- | --- |
| **ID** | FR-4 |
| **Title** | Reset Password |
| **Description** | An existing user will be able to reset his/her password by entering his current password and then new password. |
| **Pre-condition** | The user opens login page and wishes to change password. |
| **Post-condition** | Upon entering correct password, the user can change his/her password. |
| **Trigger** | The user hits Change password button. |
| **Exception** | The user is not an existing user, wrong current password. |
| **Assumptions** | The user is a registered user. |

**3.2.2 User Case – Home Page**

**3.2.2.1 Functional Requirements 1**

**ID: 3.2.2.1**

**Title:** jump to Search Friend Page

**Description:** user clicks Search Friend button, and Search Friend Page is loaded

**Pre-condition:** user is on the Home Page and wants to search friends

**Post-condition:** Search Friend Page is loaded into browser.

**Trigger:** user clicks on the Search Friend Page bottom.

**Exception:** cannot load the correct Page.

**Assumption:** user is on the Home Page.

**3.2.2.2 Functional Requirements 2**

**ID: 3.2.2.2**

**Title:** stay in Home Page

**Description:** user clicks Home Page button, and Home Page is loaded

**Pre-condition:** user is on the Home Page and wants to renew page

**Post-condition:** Home Page is loaded into browser.

**Trigger:** the click on the Home Page button

**Exception:** Cannot load the correct Page.

**Assumption:** user is on the Home Page.

**3.2.2.3 Functional Requirements 3**

**ID: 3.2.2.3**

**Title:** jump to Personal Profile Page

**Description:** user clicks Personal Profile Page button, and Personal Profile Page is loaded

**Pre-condition:** user is on the Home Page and wants to update his personal profile

**Post-condition:** Personal Profile Page is loaded into browser.

**Trigger:** the click on the Personal Profile Page button

**Exception:** Cannot load the correct Page.

**Assumption:** user is on the Home Page.

**3.2.2.4 Functional Requirements 4**

**ID: 3.2.2.4**

**Title:** type message in text field of Post Input Window

**Description:** user types his message into the text field for posting

**Pre-condition:** user wants to create a text post

**Post-condition:** the message appears in the field

**Trigger:**  type in several sentences

**Exception:** Cannot type any sentences in the field; the length of text message exceeds the limitation.

**Assumption:** user is on the Home Page.

**3.2.2.5 Functional Requirements 5**

**ID: 3.2.2.5**

**Title:** add sticker into text field of Post Input Window

**Description:** user adds some stickers into the text field for posting

**Pre-condition:** user wants to add stickers into his text post

**Post-condition:** the stickers appear in the field

**Trigger:** click on ‘sticker’ bottom and choose several stickers

**Exception:** Cannot add any stickers in the field; the length of text message exceeds the limitation.

**Assumption:** user is on the Home Page.

**3.2.2.6 Functional Requirements 6**

**ID: 3.2.2.6**

**Title:** add pictures for posting in Post Input Window

**Description:** user adds at most five pictures for posting

**Pre-condition:** user wants to upload pictures into a post

**Post-condition:** the picture appears in the picture slot below text field

**Trigger:** click on the picture clot and choose pictures from local place

**Exception:** Cannot add any picture in the slot

**Assumption:** user is on the Home Page.

**3.2.2.7 Functional Requirements 7**

**ID: 3.2.2.7**

**Title:** manage pictures for posting

**Description:** user adds at most five pictures for posting, and manages these selected pictures

**Pre-condition:** user uploads pictures from local place and adds them into picture slots. Then user wants to delete and manage these pictures.

**Post-condition:** the picture appears in the picture slot below text field

**Trigger:** click on the picture slot, and decide to update or delete

**Exception:** Cannot manage pictures in the slot

**Assumption:** user is on the Home Page.

**3.2.2.8 Functional Requirements 8**

**ID: 3.2.2.8**

**Title:** submit the post

**Description:** after finish the post message, user could click the submit button to post the message into Post View Window.

**Pre-condition:** user finished his post and wants to submit it.

**Post-condition:** the post appears in the Post View Window

**Trigger:** click on the ‘submit’ bottom

**Exception:** the posted message is missing; friends cannot see the post.

**Assumption:** user is on the Home Page.

**3.2.2.9 Functional Requirements 9**

**ID: 3.2.2.9**

**Title:** show ‘like’ on the post

**Description:** user could show his ‘like’ on other’s post

**Pre-condition:** user wants to show his ‘like’ on other’s post

**Post-condition:** user’s name is added into the like list below the post

**Trigger:** click the ‘like’ button

**Exception:** the ‘like’ list cannot show the user’s name

**Assumption:** user is on the Home Page.

**3.2.2.10 Functional Requirements 10**

**ID: 3.2.2.10**

**Title:** add comments on the post

**Description:** user could add his comments on other’s post

**Pre-condition:** user wants to show his comments on someone’s post

**Post-condition:** user’s name with his comment is added into the comment list below the like list

**Trigger:** click the ‘comment’ button and type the text in text field

**Exception:** the comment list cannot show the user’s name and his comment

**Assumption:** user is on the Home Page.

**3.2.2.11 Functional Requirements 11**

**ID: 3.2.2.11**

**Title:** view picture of the post

**Description:** user could view picture of other’s post

**Pre-condition:** the picture on the post is small and unclear

**Post-condition:** user could see a clear and larger picture on the screen

**Trigger:** click the picture

**Exception:** the picture cannot be zoomed in; the picture window cannot be closed

**Assumption:** user is on the Home Page.

**3.2.2.12 Functional Requirements 12**

**ID: 3.2.2.12**

**Title:** see friend’s brief information

**Description:** user could see brief information of friend in Friend List Window

**Pre-condition:** user wants to see friend’s brief information

**Post-condition:** friend’s brief profile information will display near the mouse

**Trigger:** hold mouse on the friend item for few seconds

**Exception:** extra information window doesn’t appear; the information is incorrect.

**Assumption:** user is on the Home Page, and the friend is listed in the friend list

**3.2.2.13 Functional Requirements 13**

**ID: 3.2.2.13**

**Title:** see friend’s profile page

**Description:** user could see personal profile of friend

**Pre-condition:** user wants to see the friend’s entire information

**Post-condition:** the page is changed to the friend’s profile page

**Trigger:** make right click and choose the “profile” item

**Exception:** the friend’s profile page doesn’t load; load the wrong page

**Assumption:** user is on the Home Page, and friend is listed in the friend list

**3.2.2.14 Functional Requirements 14**

**ID: 3.2.2.14**

**Title:** delete friend

**Description:** user could delete friend

**Pre-condition:** user wants to delete a friend from his friend list.

**Post-condition:** the friend’s item is removed from Friend List

**Trigger:** make right click and choose the “delete” item

**Exception:** cannot delete friend from the Friend List

**Assumption:** user is on the Home Page, and friend is listed in the friend list

**3.2.2.15 Functional Requirements 15**

**ID: 3.2.2.15**

**Title:** block friend

**Description:** user could block friend

**Pre-condition:** user doesn’t want a friend see his post

**Post-condition:** the friend’s item is still in the Friend List, but there is a “block” signal on it. The Post View Window won’t show any user’s post to that selected friend.

**Trigger:** make right click and choose the “block” item

**Exception:** cannot block friend

**Assumption:** user is on the Home Page, and friend is listed in the friend list

**3.2.2.16 Functional Requirements 16**

**ID: 3.2.2.16**

**Title:** release friend

**Description:** user could release friend from blocking

**Pre-condition:** user wants his locked friend could see his post

**Post-condition:** the friend’s item is still in the Friend List, but the ‘block’ signal is gone. The Post View Window starts to show any post from that selected friend.

**Trigger:** make right click and choose the “release” item

**Exception:** cannot release friend from blocking

**Assumption:** user is on the Home Page, and his friend is locked before.

**3.2.2.17 Functional Requirements 17**

**ID: 3.2.2.17**

**Title:** start to chat with friend

**Description:** user could start to chat with friend

**Pre-condition:** user wants to start chatting with one friend.

**Post-condition:** the Chat Page is load into browser

**Trigger:** make left click on friend’s item

**Exception:** cannot load the Chat Page

**Assumption:** user is on the Home Page, and friend is listed in the friend list

**3.2.2.18 Functional Requirements 18**

**ID: 3.2.2.18**

**Title:** add friend

**Description:** user could add friend

**Pre-condition:** user wants to add friends

**Post-condition:** the Search Friend Page is load into browser

**Trigger:** click on ‘add friend’ button at the bottom of Friend List

**Exception:** cannot load the Search Friend Page

**Assumption:** user is on the Home Page.

**3.2.2.19 Functional Requirements 19**

**ID: 3.2.2.19**

**Title:** logout

**Description:** user could logout the application

**Pre-condition:** user wants to logout from the web application

**Post-condition:** the Login Page is load into browser, and user cannot return to the previous page

**Trigger:** make click on ‘logout’ button at the bottom of Home Page

**Exception:** cannot load the Login Page; user could return to previous page

**Assumption:** user is on the Home Page.

**3.2.3 User Case – Personal Profile**

**3.2.3.1 Functional Requirements 1**

|  |  |
| --- | --- |
| **ID** | FR-1 |
| **Title** | Show Basic Personal Information |
| **Description** | Showing a user’s basic personal information. |
| **Pre-condition** | The user opens the web application. |
| **Post-condition** | The webpage shows the user’s own or others’ basic personal information. |
| **Trigger** | The user clicks the “profile” button on his own homepage or other users’.  The user clicks the “Basic Personal Information” tab in the page. |
| **Exception** | No internet access, system malfunction. |
| **Assumptions** | User has a web browser |

**3.2.3.2 Functional Requirements 2**

|  |  |
| --- | --- |
| **ID** | FR-2 |
| **Title** | Show Contact Information |
| **Description** | Showing a user’s contact information. |
| **Pre-condition** | The user opens the web application. |
| **Post-condition** | The webpage shows the user’s own or others’ contact information. |
| **Trigger** | The user clicks the “profile” button on his own homepage or other users’.  The user clicks the “Contact Information” tab in the page. |
| **Exception** | No internet access, system malfunction. |
| **Assumptions** | User has a web browser.  If the profile page is not the users, the one whose profile is being checked is a friend of the user. |

**3.2.3.3 Functional Requirements 3**

|  |  |
| --- | --- |
| **ID** | FR-3 |
| **Title** | Show Posts History |
| **Description** | Showing a user’s posts history. |
| **Pre-condition** | The user opens the web application. |
| **Post-condition** | The webpage shows the user’s own or others’ posts history. |
| **Trigger** | The user clicks the “profile” button on his own homepage or other users’.  The user clicks the “Post History” tab in the page. |
| **Exception** | No internet access, system malfunction. |
| **Assumptions** | User has a web browser.  If the profile page is not the users, the one whose profile is being checked is a friend of the user. |

**3.2.3.4 Functional Requirements 4**

|  |  |
| --- | --- |
| **ID** | FR-4 |
| **Title** | Edit&Save Profile |
| **Description** | A user edits his profile and saves the change. |
| **Pre-condition** | The user opens the web application. |
| **Post-condition** | The change of user’s profile will be saved and updated. |
| **Trigger** | The user clicks the “Edit” button on his profile page.  After finishing editing, the user clicks the “Submit” button. |
| **Exception** | No internet access, system malfunction. |
| **Assumptions** | User has a web browser |

**3.2.3.5 Functional Requirements 5**

|  |  |
| --- | --- |
| **ID** | FR-5 |
| **Title** | Starting chatting |
| **Description** | The user starts a chatting dialog with a friend whose profile is being checked by the user. |
| **Pre-condition** | The user is checking a friend’s profile. |
| **Post-condition** | Leading to a chatting page with this friend. |
| **Trigger** | The user clicks the “Chat” button on this friend’s profile page. |
| **Exception** | No internet access, system malfunction. |
| **Assumptions** | User has a web browser |

**3.2.3.6 Functional Requirements 6**

|  |  |
| --- | --- |
| **ID** | FR-6 |
| **Title** | Adding new friends |
| **Description** | A user sends a friend request to the one whose profile is being checked. |
| **Pre-condition** | The user opens the web application.  The user is checking a stranger’s profile. |
| **Post-condition** | The server will send a friend invitation to the one who is invited. |
| **Trigger** | The user clicks the “Add” button on the other’s profile page. |
| **Exception** | No internet access, system malfunction. |
| **Assumptions** | User has a web browser |

**3.2.3.7 Functional Requirements 7**

|  |  |
| --- | --- |
| **ID** | FR-7 |
| **Title** | Back to homepage |
| **Description** | A user goes back to his homepage. |
| **Pre-condition** | The user opens the web application. |
| **Post-condition** | The user will go back to his homepage. |
| **Trigger** | The user clicks the “Home Page” button on his profile page. |
| **Exception** | No internet access, system malfunction. |
| **Assumptions** | User has a web browser |

**3.2.4 User Case – Chat Page**

**3.2.4.1 Functional Requirements 1**

|  |  |
| --- | --- |
| **ID** | FR-1 |
| **Title** | Open Dialog Box |
| **Description** | After the user click the chat button of the friend list for a certain friend, then the system will open a new dialog box to achieve chatting feature |
| **Pre-condition** | There exists a chat button in the friend list |
| **Post-condition** | a new dialog box is loaded into browser |
| **Trigger** | The click on the chat button of a certain friend on friend list. |
| **Exception** | Cannot open a correct dialog box. |
| **Assumptions** | User has a web browser, server receives the request |

|  |  |
| --- | --- |
| **ID** | FR-2 |
| **Title** | shown the name of chatters |
| **Description** | After a new dialog box is loaded into browser, then all chatters' name must be shown correctly on the top left of the box. |
| **Pre-condition** | There exist a new dialog box after click chat button |
| **Post-condition** | a new dialog box is loaded into browser and the name is shown on the correct place |
| **Trigger** | the click on the chat button of a certain friend on friend list |
| **Exception** | Chatters' name are not shown correctly or not on the correct place. |
| **Assumptions** | User has a web browser, server receives the request |

|  |  |
| --- | --- |
| **ID** | FR-3 |
| **Title** | add more friends |
| **Description** | After we click the add friend button which on the top right of dialog box, then the system could add one or more friends from friend list to chat together. |
| **Pre-condition** | There exists a add friend button in the dialog box, and there exists a dialog box. |
| **Post-condition** | New users' name will be added on the top left of dialog box. |
| **Trigger** | The click on the add friend button which on the top right of dialog box. |
| **Exception** | New users are not correctly added into chatting room or their name are not shown on the top left of the system. |
| **Assumptions** | User has a web browser, server receives the request |

|  |  |
| --- | --- |
| **ID** | FR-4 |
| **Title** | Typing message into message window. |
| **Description** | After the user move mouse into message window, then the user could typing message using keyboard into message window. And user are not allowed to type more than 155 words one time. |
| **Pre-condition** | There exists a dialog box which has a message window. |
| **Post-condition** | These messages are shown on the message window. |
| **Trigger** | The click on the message window, several sentences. |
| **Exception** | Messages are not shown correctly on the message window, or the length of message exceeds the limitations. |
| **Assumptions** | User has a web browser, User has a keyboard |

|  |  |
| --- | --- |
| **ID** | FR-5 |
| **Title** | Send message to display window. |
| **Description** | After the user click the send button, then the messages stored in message window will be shown on display window. |
| **Pre-condition** | There exists a dialog box which has a display window. |
| **Post-condition** | These messages are shown on the display window. |
| **Trigger** | The click on the send button which on the button right of the dialog box. |
| **Exception** | Messages are not shown correctly on the display window, or the length of message exceeds the limitations. |
| **Assumptions** | User has a web browser, User has a displayer |

|  |  |
| --- | --- |
| **ID** | FR-6 |
| **Title** | Add name and time to display window. |
| **Description** | Once a message is about to shown on display window, then the system will add the time and name on display window before each message. |
| **Pre-condition** | There exists a dialog box which has a display window. |
| **Post-condition** | Name and time are shown before each message on the display window. |
| **Trigger** | The click on the send button which on the button right of the dialog box. |
| **Exception** | Name and time are not shown correctly on the display window or they are shown after displaying each message. |
| **Assumptions** | User has a web browser, User has a displayer |

|  |  |
| --- | --- |
| **ID** | FR-7 |
| **Title** | Add emoji to message window. |
| **Description** | Once we type the emoji button of the dialog box, then the user can choose emoji from a new window, and after click each emoji, it will directly show on message window. |
| **Pre-condition** | There exists a dialog box which has a message window and emoji button. |
| **Post-condition** | Emoji is shown on message window. |
| **Trigger** | the click on the emoji button, and the click to a certain emoji. |
| **Exception** | Emoji is not shown on message window, or not shown the correct emoji on message window. |
| **Assumptions** | User has a web browser, User has a displayer |

|  |  |
| --- | --- |
| **ID** | FR-8 |
| **Title** | Add photo to message window. |
| **Description** | Once we type the photo button of the dialog box, then the user can choose photo from local files, and after click each photo, it will directly show on display window. |
| **Pre-condition** | There exists a dialog box which has a message window and photo button. |
| **Post-condition** | Photo is shown on display window. |
| **Trigger** | The click on the photo button, and the click to a certain photo. |
| **Exception** | Photo is not shown on display window, or not shown the correct photo on display window. |
| **Assumptions** | User has a web browser, User has a displayer |

|  |  |
| --- | --- |
| **ID** | FR-9 |
| **Title** | check history of message |
| **Description** | Once we click on history button, then system will open a new window which stored the history of the messages. |
| **Pre-condition** | There exists a dialog box which has a history button. |
| **Post-condition** | a new window with history of the messages |
| **Trigger** | The click on the history button. |
| **Exception** | Window is not shown correctly, or the history of the messages are not correctly. |
| **Assumptions** | User has a web browser, User has a displayer, and User has a Database. |

**3.2.5 User Case – Friend Search Page**

**3.2.5.1 Functional Requirements 1**

|  |  |
| --- | --- |
| **ID** | FR-1 |
| **Title** | Jump to Home Page |
| **Description** | User clicks Home Page link, and Search Friend Page will load |
| **Pre-condition** | The user is on Search Friend page. |
| **Post-condition** | Home Page is loaded into browser |
| **Trigger** | The click on the Home Page link |
| **Exception** | Cannot load the correct Page |
| **Assumptions** | User has a web browser and Internet connection |

**2.2.1.2 Functional Requirements 2**

|  |  |
| --- | --- |
| **ID** | FR-1 |
| **Title** | Jump to Personal Profile Page |
| **Description** | User clicks Personal Profile Page link, and Personal Profile Page will load |
| **Pre-condition** | The user is on Search Friend page. |
| **Post-condition** | Personal Profile Page is loaded into browser |
| **Trigger** | The click on the Personal Profile Page link |
| **Exception** | Cannot load the correct Page |
| **Assumptions** | User has a web browser and Internet connection |

**2.2.1.4 Functional Requirements 4**

|  |  |
| --- | --- |
| **ID** | FR-1 |
| **Title** | Use Search Box to search friend |
| **Description** | User types the search keywords into the text field for searching |
| **Pre-condition** | The user is on Search Friend page. |
| **Post-condition** | Data retrieved from database (friend(s) satisfying the search conditions) will be displayed underneath the search box |
| **Trigger** | User uses keyboard types in the keywords and Click on Search Button |
| **Exception** | **:** Cannot type any text in the search box; the length of text exceeds the limitation; Cannot retrieve correct result. |
| **Assumptions** | User has a web browser and Internet connection |

**2.2.1.12 Functional Requirements 5**

|  |  |
| --- | --- |
| **ID** | FR-1 |
| **Title** | See friend’s brief information |
| **Description** | User could see brief information of friend in both searched friend List and recommended friend list |
| **Pre-condition** | The user is on Search Friend page. |
| **Post-condition** | Friend’s brief profile information will display near the mouse |
| **Trigger** | Hold mouse on the friend record for two seconds |
| **Exception** | Extra information doesn’t appear; the information is incorrect |
| **Assumptions** | User has a web browser and Internet connection |

**2.2.1.13 Functional Requirements 6**

|  |  |
| --- | --- |
| **ID** | FR-1 |
| **Title** | See friend’s profile page |
| **Description** | User could see personal profile of friend |
| **Pre-condition** | The user is on Search Friend page. |
| **Post-condition** | The page is redirected to the friend’s profile page |
| **Trigger** | Make right click on friend’s record and choose the “profile” button |
| **Exception** | The friend’s profile page doesn’t load; load the wrong page |
| **Assumptions** | User has a web browser and Internet connection |

**2.2.1.18 Functional Requirements 7**

|  |  |
| --- | --- |
| **ID** | FR-1 |
| **Title** | Show recommended friend(s) |
| **Description** | User see their potential/possible friend(s), the server will retrieve the user’s friend list and try to create a potential friend list for the user based on recommendation algorithm. |
| **Pre-condition** | The user is on Search Friend page. |
| **Post-condition** | The potential friend list is displayed in recommendation area |
| **Trigger** | Make right click on friend’s record and choose the “profile” button |
| **Exception** | Cannot load the recommendation area or show completely irrelevant people. |
| **Assumptions** | User has a web browser and Internet connection |

**2.2.1.18 Functional Requirements 8**

|  |  |
| --- | --- |
| **ID** | FR-1 |
| **Title** | Add friend |
| **Description** | User could add friend. The server will send request to the friend. If the request is accepted, insert data into database. If rejected, return rejection information. Otherwise, show “request is sent, please wait for acceptance” information |
| **Pre-condition** | The user is on Search Friend page. |
| **Post-condition** | The new friend is added to the friend list/rejection information/waiting information. |
| **Trigger** | Click on ‘Add’ button on the right side of searched/potential friend list. |
| **Exception** | Cannot add friend/cannot send request to friend |
| **Assumptions** | User has a web browser and Internet connection |

**2.2.1.19 Functional Requirements 9**

|  |  |
| --- | --- |
| **ID** | FR-1 |
| **Title** | logout |
| **Description** | User could logout the application. The server will change user’s login status, and redirect to Login Page |
| **Pre-condition** | The user is on Search Friend page. |
| **Post-condition** | The Login Page is load into browser, and user cannot return to the previous page. |
| **Trigger** | Click on ‘logout’ link |
| **Exception** | Cannot load the Login Page; user could return to previous page |
| **Assumptions** | User has a web browser and Internet connection |

**3.3 Non-functional Requirements**

Non-functional Requirements are describing the high level definition of the social networking application. This section will introduce the assumed expectation from user side.

**3.3.1 Performance**

**3.3.1.1 Login and Register**

The login feature should be fast and should not take more than 2 seconds.

The Registration page should not have many entries, just relevant information required for registering a user.

Registration should be fast and not take more than 2 seconds.

**3.3.1.2 Password Reset**

Resetting the Password would only require updating the password field in the database, hence it should be quick.

The Forget password would send an email to the user upon successful answering the questions. The email sending should be instant.

**3.3.1.3 Page Loading**

Loading page should cost less than 1 second.

**3.3.1.4 Chat History Loading**

After click on history button, the time for messages shown on history window should be less than 2 second.

**3.3.1.5 Chat Request Notification**

The time for searching friends and adding friends to a chatting room should be less than 1 second

**3.3.1.6 Instant Chatting**

After click on send button, the time for messages shown on display window should be less than 2 second.

**3.3.1.7 Data Synchronization**

When user has changed his profile information, the database should keep the data synchronous.

**3.3.1.8 Friend Searching**

The time for searching friends based on input conditions from database should be less than 3 seconds. The expected performance is near real-time processing.

**3.3.1.9 Friend Recommendation**

Potential/possible friend list should be displayed within 2 seconds.

**3.3.1.10 Friend Request Sending/Accepting**

Request of friend should be sent and received within 5 seconds.

**3.3.1.11 Friend List Synchronization**

The friend list should be synchronized with database.

**3.3.1.12 Submitting post**

The time for uploading the post to Post View Window and database should be less than 1 second. The expected performance is near real-time processing.

**3.3.1.13 Post View Window Synchronization**

Post View Window should be loaded dynamically.

**3.3.2 Reliability**

**3.3.2.1 Communication Reliability**

The request should be processed correctly. The server should not update the data into wrong item.

The message should include enough information, and nothing will be missing during messaging step.

**3.3.2.2 Friend List Sync Reliability**

When user add/delete/block/release a friend, the database should keep data relationship synchronously.

**3.3.2.3 Real-Time Notification Reliability**

The notification will be kept in Friend List until user process these notification.

**3.3.2.4 Real-Time Chatting Reliability**

When a friend starts a chat with user, user could see the chat notification on any page except Login Page.

**3.3.3 Availability**

The application should be tested more than 5 hours work:

**1.** Jump to correct Pages

**2.** Post messages correctly (include pictures, text and stickers)

**3.** View posts in the Post View Window (could click ‘like’, add comments)

**4.** View Friend List correctly (click to chat, add friend, delete friend, block friend, and view friend’s brief and entire profile)

**5.** Add tutorials for the beginners to know all functionalities.

**6.** Query database efficiently.

**7.** Update chat history dynamically into database.

**8.** Send messages to target directly.

**9.** Load/update Profile Page efficiently.

**3.3.4 Security**

**3.3.4.1 Password Security**

The password should be secure in the database and there should not be any memory leakage leading to information circulation.

**3.3.4.2 Message Security**

Normal users must log in to access their chatting history. The system will automatically shut down the dialog box if the message involve financial information or sensitive topic.

**3.3.4.3 Database Security**

The data should be saved in the database, and invalid request cannot access the database.

**3.3.5 Maintainability**

**3.3.5.1 Application Extendibility**

The application should be easy for new company to maintain and develop. And there should are some potential ports for implementing new functions.

**3.3.5.2 Application Testability**

There should be a reliability environment to test new added functions easily. And the application should be debugged easily with providing clear error reports.

**3.3.6 Portability**

The application could be developed to install in other hardware and systems, like iSO, android, desktop and so on.

**3.4 Design Constraints**

The application size should be small.

**3.5 Other Requirements**

The user interface should be brief and easy to understand. After user registers and logs in the account, there should be tutorial to guide user how to use all functions.

**A. Appendices**

This section introduces the User Interface of the social networking web application.

**A.1 Appendix 1 – Login and Register Page**

REGISTER

SECURITY QUESTION 2

SECURITY QUESTION 1

GENDER (MALE/FEMALE)

DOB (MM/DD/YYYY)

EMAIL

PASSWORD

USERNAME

NEW USER

NAME OF APPLICATION

RESET

PASSWORD

FORGET

PASSWORD

LOGIN

PASSWORD

USERNAME/EMAIL

EXISTING USER

**A.2 Appendix 2 – Home Page**

****

**A.3 Appendix 3 – Friend Search Page**

****

**A.4 Appendix 4 – Personal Profile Page**

****

**A.5 Appendix 5 – Chat Page**

****

**Priority Table**