

# Aloha Social Network

## Software Requirement Specification (SRS)

|  |  |
| --- | --- |
| Document/Version Number: | Version 1.1 |
| Creation Date: | February 5, 2015 |
| Account/Client: | SOFTA Inc. |
| Author: | Milind Gokhale; Renuka Deshmukh; Mrunal Pagnis; Vivek Supe |
| Editor: | Milind Gokhale; Renuka Deshmukh; Mrunal Pagnis; Vivek Supe |
| Last Edit Date: | March 9, 2015 |
| File Name: | SRS\_Aloha\_My\_Group.docx |

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Description** | **Author** | **Comments** |
| 02/25/2015 | Initial Draft | Vivek Supe, Mrunal Pagnis | Initial draft of the document |
| 02/26/2015 | Version 0.1 | Vivek Supe, Milind Gokhale, Renuka Deshmukh, Mrunal Pagnis | Inserted the basic requirements. |
| 02/27/2015 | Version 1.0 | Renuka Deshmukh, Milind Gokhale | First Revision |
| 02/28/2015 | Version 1.1 | Vivek Supe, Milind Gokhale, Renuka Deshmukh, Mrunal Pagnis | Added use cases and Formatting changes |

# Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

|  |  |  |  |
| --- | --- | --- | --- |
| **Signature** | **Printed Name** | **Title** | **Date** |
|  | John Doe | Senior Architect | February 28, 2015 |
|  | Balaprasath Rajan | Professor | February 28, 2015 |
|  |  |  |  |

# Table of Contents

[Aloha Social Network 0](#_Toc412929778)

[Software Requirement Specification (SRS) 0](#_Toc412929779)

[Revision History 0](#_Toc412929780)

[Document Approval 0](#_Toc412929781)

[Table of Contents 0](#_Toc412929782)

[1 Introduction 2](#_Toc412929783)

[1.1 Purpose 2](#_Toc412929784)

[1.2 Scope 2](#_Toc412929785)

[1.3 Definitions & acronyms 2](#_Toc412929786)

[1.4 Intended Audience 2](#_Toc412929787)

[1.5 Overview 2](#_Toc412929788)

[2 Overall Description 3](#_Toc412929789)

[2.1 Product Perspective 3](#_Toc412929790)

[2.2 Product Functions 3](#_Toc412929791)

[2.3 Operating Environment 3](#_Toc412929792)

[2.4 User characteristics 3](#_Toc412929793)

[2.5 Constraints 3](#_Toc412929794)

[2.6 Assumptions and Dependencies 3](#_Toc412929795)

[2.6.1 Assumptions 3](#_Toc412929796)

[2.7 Apportioning of requirements 4](#_Toc412929797)

[3 Specific Requirements 5](#_Toc412929798)

[3.1 External interface requirements 5](#_Toc412929799)

[3.1.1 User interfaces 5](#_Toc412929800)

[3.1.2 Hardware Interfaces 5](#_Toc412929801)

[3.1.3 Software interfaces 5](#_Toc412929802)

[3.1.4 Communication interfaces 5](#_Toc412929803)

[3.2 Functional requirements 6](#_Toc412929804)

[3.2.1 Classes for classification of specific requirements 6](#_Toc412929805)

[3.2.2 User Registration profile and settings - FRU 6](#_Toc412929806)

[3.2.3 Friends and Suggestions - FRF 9](#_Toc412929807)

[3.2.4 Posts and Shares (Scribbles) - FRP 10](#_Toc412929808)

[3.2.5 Chat - FRC 12](#_Toc412929809)

[3.3 Use Cases 14](#_Toc412929810)

[3.3.1 Use Case UC1 14](#_Toc412929811)

[3.3.2 Use Case UC2 15](#_Toc412929812)

[3.3.3 Use Case UC3 15](#_Toc412929813)

[3.3.4 Use Case UC4 15](#_Toc412929814)

[3.3.5 Use Case UC5 15](#_Toc412929815)

[3.3.6 Use Case UC6 16](#_Toc412929816)

[3.3.7 Use Case UC7 16](#_Toc412929817)

[3.3.8 Use Case UC8 16](#_Toc412929818)

[3.3.9 Use Case UC9 16](#_Toc412929819)

[3.3.10 Use Case UC10 17](#_Toc412929820)

[3.3.11 Use Case UC11 17](#_Toc412929821)

[3.3.12 Use Case UC12 17](#_Toc412929822)

[3.4 Non-functional Requirements 18](#_Toc412929823)

[3.4.1 Performance requirements 18](#_Toc412929824)

[3.4.2 Security requirements 18](#_Toc412929825)

[3.4.3 Software Quality Attributes Requirements 18](#_Toc412929826)

[3.5 Requirement Traceability Matrix 19](#_Toc412929827)

[4 Change Control Mechanism 20](#_Toc412929828)

[4.1 Documenting the change request: 20](#_Toc412929829)

[4.2 Formal assessment: 20](#_Toc412929830)

[4.3 Designing and testing: 20](#_Toc412929831)

[4.4 Final assessment: 20](#_Toc412929832)

[5 Appendices 21](#_Toc412929833)

[5.1 Appendix 1: Glossary 21](#_Toc412929834)

# Introduction

## Purpose

The purpose of this Software Requirement Specification document is to give an overview of the functional and non-functional requirements of a social networking website for the web.

## Scope

Aloha is a web portal which allows users to connect with their friends and family through a common platform. Furthermore, users’ can share scribbles and *ChitChat* with their friends. These chats can be saved or deleted as per the users’ wishes. Users can also maintain, update or delete their account.

## Definitions & acronyms

The following table explains terms and acronyms specific to this SRS.

|  |  |
| --- | --- |
| *Term/Acronym* | *Description/Definition* |
| *Aloha* | *Aloha is a social networking website.* |
| *FR* | *Functional Requirement* |
| *UC* | *Use Case* |
| *Scribble* | *Scribble is a post shared by the user on his slate.* |
| *ChitChat* | *Chatting activity between two users registered on Aloha.* |
| *Slate* | *Wall where users scribble.* |

## Intended Audience

This document is intended for a varied set of audiences. This document serves as a contract agreement with SOFTA. This document is intended for the SOFTA authorities, the design team, developers, project manager, team leads, supervisors, security analysts, testing team and the QA team to better understand the system requirements.

## Overview

The remaining document has three sections. The second section gives a generic overview of the systems including requirements and functions in a broader view. It gives an insight into the user characteristics, constraints and the operating environment of the system.

The third section gives a detailed look of the functional and non-functional requirements of the system. It also gives a complete description explaining these requirements. This section also gives information about the quality attributes the system needs to satisfy. Section four explains the methodology followed in case of any change in requirements. Section five is a list appendix that helps to better understand the SRS.

# Overall Description

## Product Perspective

Aloha is a free online social networking website which allows users to connect with their friends and family. It gives a global platform for users around the world to connect with each other. Users can also share *scribbles* and engage in chitchat conversations. This is an independent product and will be the first release of the product.

## Product Functions

User can register to the social network and login whenever he wishes to do so. User can build his profile, add or edit information, set visibility to various profile sections, see his own profile and upload profile picture to his profile.

With this application users can connect with their friends and family through a common platform. A user can search for his friends and will be shown appropriate results if matching profiles are found. Any user can post scribbles to his profile which will be visible to his connections.

## Operating Environment

* Minimum: 2 CPU Cores (Recommended: 4 CPU Cores or more)
* Recommended memory: 2 GB for Windows platforms, 1 GB for Linux
* Minimum disk space: 500 MB
* Recommended disk space: 1024 MB
* Java Platform Enterprise Edition (Java EE)
* Apache 8.x
* JDK 6+ for Spring Framework
* MySQL 5.x.x

## User characteristics

Standard users may belong to any demographic group including any gender, nationality that can use computer’s browser. Aloha requires its users to be above 18 years of age. Aloha does not require its users to have any specific computer knowledge. User interface is in English language so users should have a Basic English reading knowledge.

## Constraints

1. Aloha will support the following browsers – IE 8+, Chrome 27.0+, Firefox 30+.
2. Aloha supports English language only.
3. Aloha requires certificates issued by CA to use HTTPS.
4. The first version of the website is intended to be viewed best on a desktop browser.

## Assumptions and Dependencies

### Assumptions

1. There will be availability of Internet via 3G, 4G or Wi-Fi.
2. The user of the site will be acquainted with Basic English language.
3. The user should have a valid email address in order to register into Aloha.
4. Central server of the system must be able to handle all the incoming requests simultaneously.
5. The user has basic knowledge of computers and internet.
6. In the current functionality the user will be able to invite one friend at a time to Aloha network.

## Apportioning of requirements

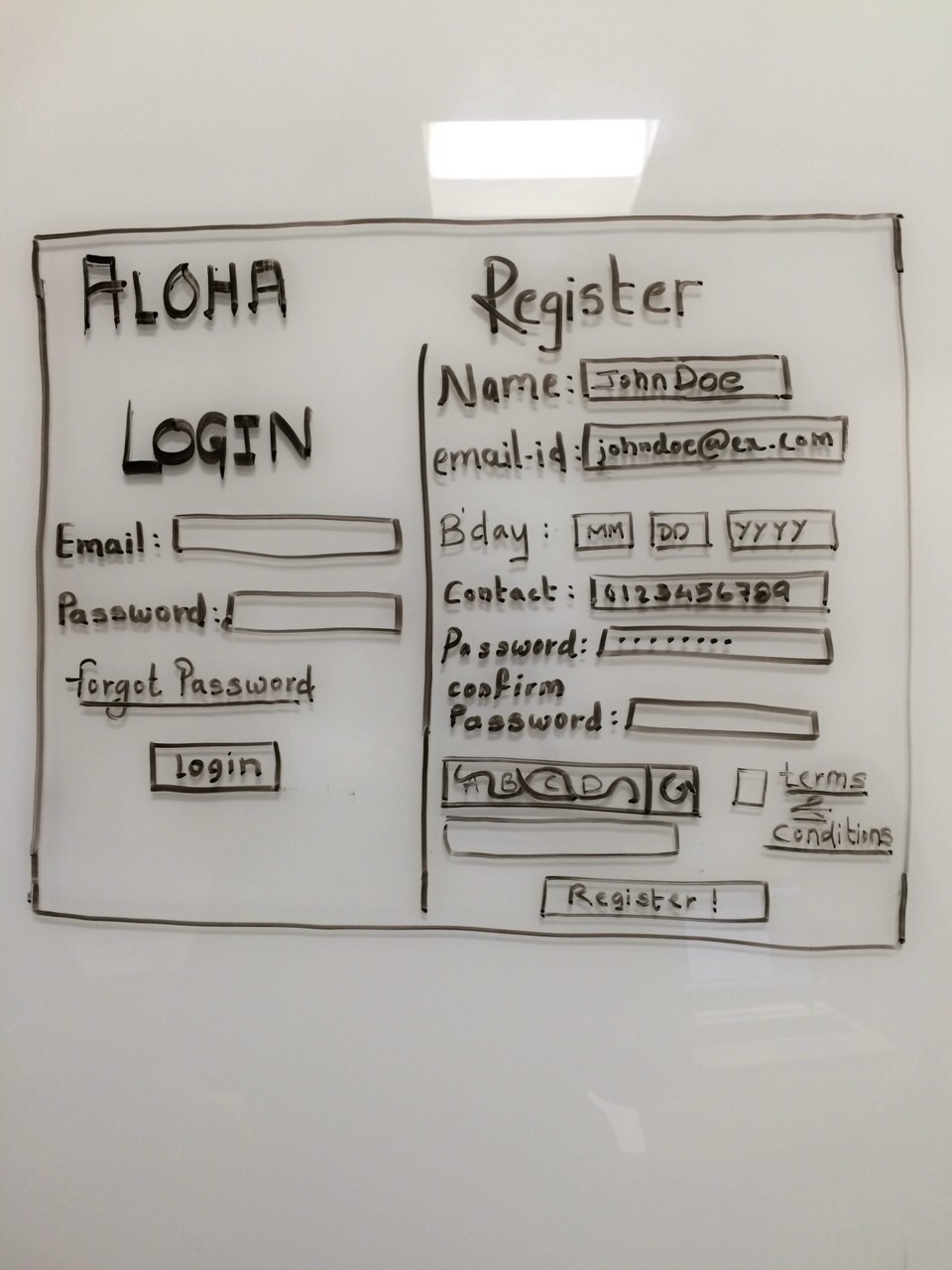
Requirements like multiple invites from user’s Gmail contacts will be planned in the future releases of the product. Also video sharing in *Scribbles* and *ChitChat* may be included in future. Certain machine learning algorithms will be included for friend suggestions in future releases.

# Specific Requirements

## External interface requirements

### User interfaces

Standard users will use the web browser to use Aloha. Thus, it shall have a login page for users to login to Aloha. For those who are first time users will fill up registration details. After logging into the system, the user will be shown his profile and the *slate* containing the posts shared by user’s friends. User is given options to logout, search people and change any of his personal settings on his home page.



### Hardware Interfaces

N/A

### Software interfaces

* E-mail Interface

This interface uses the SMTP/POP service provided by third party to send emails to required user. This service will help notify users with required actions through a conventional medium.

* Captcha Service Interface

Captcha service will be used for human identification.

### Communication interfaces

ALOHA is based on HTTP as it is a website. Moreover TCP/IP is used for chatting functionality.

## Functional requirements

### Classes for classification of specific requirements

The functional requirements are grouped as below

1. User Registration profile and settings – [FRU]
2. Friends and suggestions – [FRF]
3. Posts and Shares (*Scribbles*) – [FRP]
4. *ChitChat* – [FRC]

### User Registration profile and settings - FRU

#### Functional Requirement FRU1

ID – FRU1

Name – User Registration

Description – User will create an account if there is no existing account associated with user’s email-id. User should enter account details such as name, email-id, contact number, birth date, captcha and password. All these are required fields and account creation cannot proceed unless these are filled. User is registered after this information is validated and user agrees with the terms and conditions. User must be 18 years old or more. Contact number should be a 10 digit numeric field. First and last name should not exceed 20 characters each. Password should be at least 8 characters long, at most 15 characters including at least one capital and small letter, a special character and a number. Any of the above fields must not have ( ‘ , “, : , ; , <, >, {, }, -, %, $, |, ], [).

Dependency – None

#### Functional Requirement FRU2

ID – FRU2

Name – Email Verification

Description – When the user has registered, a verification link is sent to the email-id set at the time of registration so as to verify given email-id as genuine. After user has verified the account it will be marked as verified. Account privileges will not be given to the user until verification is complete. User should be able to re-send the verification link to his email-id.

Dependency – FRU1

#### Functional Requirement FRU3

ID – FRU3

Name – Login

Description – User must be registered in order to Login. Input will be the email-id and password for login. The hash of password should match to the hash of password stored in the database. If three attempts of password are wrong then a captcha should appear. If more than 8 attempts go wrong, then the account will be locked and an e-mail will be sent to the user notifying about the wrong password attempts. To unlock the account, user must click on the unlock link sent to him in the email. If user clicks on forgot password then reset password link should be emailed to the user. Input to each field must be validated.

Dependency – FRU1

#### Functional Requirement FRU4

ID – FRU4

Name – Add/Edit Education

Description – A part of building user profile. User should be able to add the educational details. These fields can remain empty. User must be logged in and his account should be marked as verified to be able to add education. Input to each field must be validated.

Dependency – FRU2, FRU3

#### Functional Requirement FRU5

ID – FRU5

Name – Add/Edit Personal information

Description – A part of building user profile. User should be able to add the personal details like language, gender, about user and his interests. These fields can remain empty. User must be logged in and his account should be marked as verified to be able to add personal information. Input to each field must be validated.

Dependency – FRU2, FRU3

#### Functional Requirement FRU6

ID – FRU6

Name – Add account information

Description – A part of building user profile. User should be able to add the account details like name, birth date, email-id, phone number and address. Name, birth date and email-id cannot be empty while the other two can remain empty. User must be logged in. Input to each field must be validated.

Dependency –FRU3

#### Functional Requirement FRU7

ID – FRU7

Name – Edit account information

Description – A part of building user profile. User should be able to edit the account details like name, birth date, email-id, phone number and address. Name, birth date and email-id cannot be empty while the other two can remain empty. If email is changed then email verification module should be called. User must be logged in. Input to each field must be validated.

Dependency –FRU2, FRU3

#### Functional Requirement FRU8

ID – FRU8

Name – Upload Profile Picture

Description – User should be able to upload the profile picture. The size of profile picture should be at least 160px \* 160px. User may leave this field empty. Profile image should be visible to all.

Dependency –None

#### Functional Requirement FRU9

ID – FRU9

Name – Set account visibility settings

Description – User should be able to make account details private or public according to his wish. By default the account details will be visible to public.

Dependency –None

#### Functional Requirement FRU10

ID – FRU10

Name – Set personal information visibility

Description – User should be able to make personal information private or public according to his wish. By default the personal information will be visible to public.

Dependency –None

#### Functional Requirement FRU11

ID – FRU11

Name – Show Profile

Description – Any logged-in user should be able to see any user’s profile on Aloha.

Dependency –FRU1, FRU2, FRU3

#### Functional Requirement FRU12

ID – FRU12

Name – Delete Account

Description – User may delete his account permanently from ALOHA depending on his wish. User will be removed from his friends list once deleted. User information will be cleared.

Dependency – FRU1, FRU2, FRU3

### Friends and Suggestions - FRF

#### Functional Requirement FRF1

ID – FRF1

Name – Add Friend

Description – A registered user of Aloha should be able to send add friend requests to other users of Aloha. User can send friend request to already registered users of Aloha. User can search his friend on Aloha among the registered users and then send a friend request to him/her.

Dependency – FRU1

#### Functional Requirement FRF2

ID – FRF2

Name – Accept/Ignore Friend Request

Description – If a user gets a friend request from another user, then the user receiving the request should have an option to accept or ignore the friend request. If the user selects to accept the friend request, the requestor is added to the friend list of the acceptor. If the user selected to ignore the request, the sender continues to see the status of the request as “Request Sent” and the request is removed from the queue of the request receiver.

Dependency – FRF1

#### Functional Requirement FRF3

ID – FRF3

Name – Delete Friend

Description – A registered user of Aloha should be able to un-friend any of the existing friends in his/her friend’s list.

Dependency – FRU1, FRF1

#### Functional Requirement FRF4

ID – FRF4

Name – List Friends

Description – Whenever the user is on any person’s profile page, he should have an option to view currently open user profile’s friends on a page. User can have many friends and the maximum number of friends shown at a time will be 20, and paging will be implemented to show more friends if the user has more than 20 friends.

Dependency – FRU3, FRU1, FRF1

#### Functional Requirement FRF5

ID – FRF5

Name – Search User

Description – User should be able to search for his/her friends who may be registered on Aloha. User will put the name of the friend and click search after which the search results will display all the people registered on Aloha by the searched name. When the search results are displayed the maximum number of results shown at a time will be 20, and paging will be implemented to show more results.

Dependency – FRU3

#### Functional Requirement FRF6

ID – FRF6

Name – Friend Suggestions

Description – User will be given suggestions for adding more friends. The friend suggestions will be based on the users having maximum number of mutual friends with the user. The maximum number of friend suggestions displayed will be 20 at a time, and paging will be implemented to show more suggestions.

Dependency – FRU3

#### Functional Requirement FRF7

ID – FRF7

Name – Invite Friend

Description – User should be able to invite more friends to join the Aloha network. User will be given an option to enter the email address of the person to invite. User will be able to invite one friend at a time.

Dependency – FRU3, FRU1

#### Functional Requirement FRF8

ID – FRF8

Name – Show Online Friends

Description – User should be able to see currently online friends among all his friends on Aloha. The user can chat with online friends by clicking on the currently available friends in the online friends list.

Dependency – FRU3, FRU1

### Posts and Shares (Scribbles) - FRP

#### Functional Requirement FRP1

ID – FRP1

Name – Scribble

Description – A registered user of Aloha can share *scribbles* with his friends. The scribble can only be plain English text with maximum 500 characters. User can enter the scribble in the designated text area and on clicking Scribble button; it should be visible to the user and his friends.

Dependency – FRU3

#### Functional Requirement FRP2

ID – FRP2

Name – Erase

Description – A user can *erase* a *scribble* that has been posted. After deletion, the *scribble* should not be visible on the users’ or his friends’ *slates*. Deletion of a *scribble* should also remove all the comments associated with that *scribble*.

Dependency – FRU3, FRP1

#### Functional Requirement FRP3

ID – FRP3

Name – List *Scribbles*

Description – After a user logs into Aloha, he can see a listing of *scribbles* posted by him and his friends. The maximum number of *scribbles* displayed at a time will be 20, and paging will be implemented for more *scribbles*.

Dependency – FRU3

#### Functional Requirement FRP4

ID – FRP4

Name – Comment

Description *–* For each *scribble* visible to the user he has the ability to comment on it to express his opinion. The comment can contain only text and cannot exceed 500 characters in length. To comment, user can see a text area under the *scribble*, where he can enter the text.

Dependency – FRU3, FRP1

#### Functional Requirement FRP5

ID – FRP5

Name – Delete Comment

Description – A user can delete a comment written by him or a comment on his *scribbles*. For deleting a *scribble* that he is authorized to delete, a user can see a delete link near the comment.

Dependency – FRU3, FRP4

#### Functional Requirement FRP6

ID – FRP6

Name – Like a *Scribble*

Description – Users can like a *scribble* by clicking on the “Thumb’s Up” button below the *scribble*. For any *scribble*, all users can see a count of the “Liked” *scribble*.

Dependency – FRU3, FRP1

#### Functional Requirement FRP7

ID – FRP7

Name – Unlike a *Scribble*

Description – Users can dislike a *scribble* by clicking on the “Thumb’s Down” button below the *scribble*. For any *scribble*, all users can see a count of the “Disliked” *scribble*.

Dependency – FRU3, FRP1

### Chat - FRC

#### Functional Requirement FRC1

ID – FRC1

Name – *ChitChat*

Description – User should be able to initiate *ChitChat* with any user in the friend list who is online. The other user must seamlessly understand that the *ChitChat* has been initiated and the *ChitChat* message must be forwarded to the intended user. User should be able to receive *ChitChat* from any user in the friend list who is online and within the friend list of that user. Thus the receiving entity can send message back to the originator, thus, completing a two-way conversation. The message cannot be greater than 250 characters in length.

Dependency – FRU3, FRF8

#### Functional Requirement FRC2

ID – FRC2

Name – Send File

Description – Using this functionality user can send text files to online users. This helps user to share text documents with users on this platform. The maximum file size would be limited to 25 MB.

Dependency – FRU3, FRF8

#### Functional Requirement FRC3

ID – FRC3

Name – Store *ChitChat*

Description – User should be able to store *ChitChat* on exit. So that if the user wants to keep history of the previous session he can keep it to be brought back.

Dependency – FRU3, FRF8

#### Functional Requirement FRC4

ID – FRC4

Name – Retrieve *ChitChat*

Description – User should be able to retrieve *ChitChat* when the *ChitChat* when the user wants to see *ChitChat* history.

Dependency – FRU3, FRF8

## Use Cases

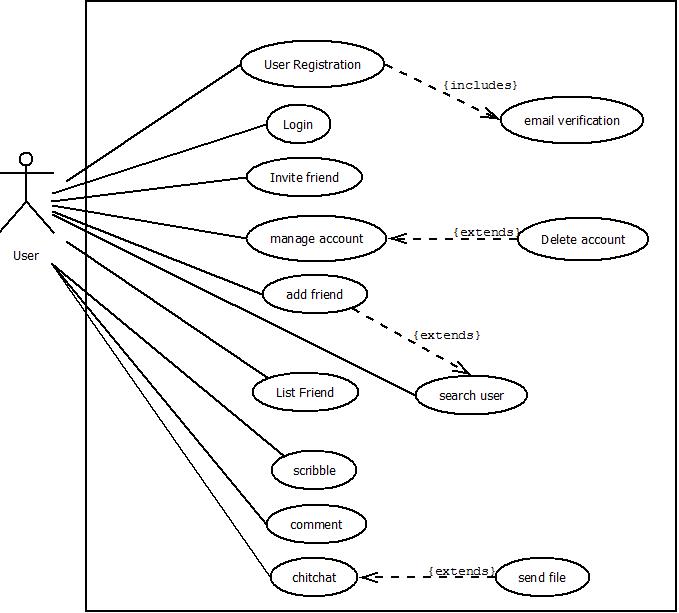


Figure 1: Use Case Diagram

### Use Case UC1

#### Objective

Registration

#### Priority

High

#### Actors

End-User

#### Pre-conditions

The user has internet and opens the website on his browser.

#### Post-conditions

The user is now a member of Aloha website.

### Use Case UC2

#### Objective

Email Verification

#### Priority

High

#### Actors

End-User

#### Pre-conditions

The user is registering to aloha website as a new user.

#### Post-conditions

The user is now a registered user on Aloha and can update account, post scribbles, add friends and chat with other registered friends.

### Use Case UC3

#### Objective

Login

#### Priority

High

#### Actors

End-User

#### Pre-conditions

User is a registered member of Aloha.

#### Post-conditions

The user logs in successfully and can now view his profile, friends and scribbles.

### Use Case UC4

#### Objective

Manage Account

#### Priority

Medium

#### Actors

End-User

#### Pre-conditions

A user is logged into Aloha.

#### Post-conditions

The user account has been successfully updated as per his requirements, like, update account, visibility setting, etc.

### Use Case UC5

#### Objective

Delete Account

#### Priority

Medium

#### Actors

End-User

#### Pre-conditions

User is a registered member of Aloha.

#### Post-conditions

Account is successfully deleted.

### Use Case UC6

#### Objective

Add Friend

#### Priority

High

#### Actors

End-User

#### Pre-conditions

User is a registered member of Aloha.

#### Post-conditions

Friend is added successfully.

### Use Case UC7

#### Objective

List Friends

#### Priority

High

#### Actors

End-User

#### Pre-conditions

User is a registered member of Aloha.

#### Post-conditions

User sees a list of his friends.

### Use Case UC8

#### Objective

Invite Friend

#### Priority

Low

#### Actors

End-User

#### Pre-conditions

User is a registered member of Aloha.

#### Post-conditions

Friend invite is sent successfully on his email allowing him to register as a new user.

### Use Case UC9

#### Objective

Scribble

#### Priority

High

#### Actors

End-User

#### Pre-conditions

User is a registered member of Aloha.

#### Post-conditions

Posted scribbles are visible to the user and his friends.

### Use Case UC10

#### Objective

Comment

#### Priority

High

#### Actors

End-User

#### Pre-conditions

User is a registered member of Aloha.

#### Post-conditions

Posted comments are visible to the user and his friends.

### Use Case UC11

#### Objective

*ChitChat*

#### Priority

High

#### Actors

End-User

#### Pre-conditions

User is a registered member of Aloha.

#### Post-conditions

The user successfully performs direct chitchat with his friend.

### Use Case UC12

#### Objective

Send File

#### Priority

High

#### Actors

End-User

#### Pre-conditions

User is a registered member of Aloha.

#### Post-conditions

The second user successfully receives the send file.

## Non-functional Requirements

### Performance requirements

#### Scalability

System should be able to handle a large number of users. For e.g. Handling around thousand users at the same time.

#### Speed

The application should be fast. It should not slow down with increase in the number of users. Search functionality should be fast to enable better end-user experience. The system should be quick enough to be able to respond to the user actions within a short period of time. E.g. the search user functionality of Aloha should perform a quick search among the users on the database.

### Security requirements

* During user registration, the given email address is validated.
* The password should be at least 8 characters, containing at least a small character and one capital, a number and a special character.
* Password is stored as a hash value in database.
* We are transferring all data via HTTPS i.e. via SSL so that the data is encrypted during the transit. Thus safeguarding the user information.
* Re-capcha service is used during registration for human identification.

### Software Quality Attributes Requirements

#### Usability

* Aloha User interface should be simple and clear to be able to understand by any user.

#### Availability

* The system should be available at all times. It should be ensured that there should be minimum or no downtime to ensure better user experience.

#### Reliability

* The system should be reliable. It should yield correct results if a user performs searches for a person. Also, if the user sends chitchat message or media, the system should ensure that the correct message is delivered to the correct destination without any loss or change in content.

#### Testability

* The application should be testable. A separate test environment should be set up where testers and the Quality Assurance engineers can test the application for bugs and/or incomplete or missed requirements.

#### Maintainability

* The system should be developed in such a way that it is extensible. It should be easy to incorporate new features requirements or accommodate a change in the existing requirements.

## Requirement Traceability Matrix

The requirement traceability matrix is available at – <https://documentcenter.aloha.com/srs/requirements/RTM1.xlsx>

The Link for requirement traceability matrix will be updated from time to time and will be made available in this section of the SRS.

# Change Control Mechanism

## Documenting the change request:

Before the change is considered the client must document the change and its requirement so that developer can understand the implication that the changes have.

## Formal assessment:

If a new change requirement is brought in, it must be approved by the change control board to be considered for development.

## Designing and testing:

If the request is accepted it is queued for development. Once the request is picked up by the developer the configuration files are checked out by the developer and changes are made on those. Once the changes are fixed the system will be tested on a sandbox with various baseline test cases.

## Final assessment:

Once that is proven to be a success the product is pushed into the repository. But if the change is too drastic it will be posted to the next version of the software.

# Appendices

## Appendix 1: Glossary

|  |  |
| --- | --- |
| *Term/Acronym* | *Description/Definition* |
| *SRS* | *Software Requirement Specification* |
| *API* | *Application Programming Interface* |
| *HTTP* | *Hyper Text Transfer Protocol* |
| *HTTPS* | *Hyper Text Transfer Protocol over SSL* |
| *SSL* | *Secure Socket Layer* |
| *IE* | *Internet Explorer* |
| *TCP/IP* | *Transfer Control Protocol/Internet Protocol* |
| *CA* | *Certificate Authority* |
| *CAPTCHA* | *Completely Automated Public Turing test to tell Computers and Humans Apart* |

*References:*

1. *Software requirements specification,* [*http://en.wikipedia.org/wiki/Software\_requirements\_specification*](http://en.wikipedia.org/wiki/Software_requirements_specification)
2. *Donn Le Vie, Jr., “Writing Software Requirements Specifications (SRS)”* [*http://techwhirl.com/writingsoftwarerequirementsspecifications/*](http://techwhirl.com/writingsoftwarerequirementsspecifications/) *(accessed Jan 30, 2015)*
3. *Software Requirements Specification Instructions,* [*https://www.dir.texas.gov/SiteCollectionDocuments/IT%20Leadership/Framework/Framework%20Extensions/SDLC/SDLC\_softwareRequirements\_instructions.pdf*](https://www.dir.texas.gov/SiteCollectionDocuments/IT%20Leadership/Framework/Framework%20Extensions/SDLC/SDLC_softwareRequirements_instructions.pdf)
4. *"What Is Change Control? - Definition from* [*WhatIs.com.*](http://whatis.com/)*" Search Disaster Recovery. Accessed March 1, 2015.* <http://searchdisasterrecovery.techtarget.com/definition/change-control>
5. <http://csis.pace.edu/~marchese/cs615sp/L12New/se_l12new_files/image014.jpg>.