**MINI PROJECT – II**

**(2018-19)**

**Social Networking Site (ChitChat)**

**Group No:**

**Software Requirement Specification(SRS)**



**Institute of Engineering & Technology**

**Team Members**

Nishant Jindal (161500362)

Nikhil Singhal (161500358)

Saurabh Singh (161500492)

Yogesh Bhargav (161500643)

**Supervised By**

Dr. Anand prakash gupta

**Asst. Professor**

**Department of Computer Engineering & Applications**

**Table of Contents**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1. | **Introduction** | | | 3 |
|  | 1.1 | Purpose | | 3 |
|  | 1.2 | Scope | | 3 |
|  | 1.3 | Definitions & acronyms | | 3 |
|  | 1.4 | Intended Audience | | 3 |
|  | 1.5 | Overview | | 3 |
| 2. | **Overall Description** | | | 4 |
|  | 2.1 | Product Perspective | | 4 |
|  | 2.2 | Product Functions | | 4 |
|  | 2.3 | Operating Environment | | 4 |
|  | 2.4 | User characteristics | | 4 |
|  | 2.5 | Constraints | | 4 |
|  | 2.6 | Assumptions and Dependencies | | 4 |
|  | 2.7 | Apportioning of requirements | | 4 |
| 3. | **Specific Requirements** | | | 5 |
|  | 3.1 | External interface requirements | | 5 |
|  |  | 3.1.1 | User interfaces | 6 |
|  |  | 3.1.2 | Hardware Interfaces |  |
|  |  | 3.1.3 | Software interfaces |  |
|  |  | 3.1.4 | Communication interfaces |  |
|  | 3.2 | Functional requirements | | 6 |
|  |  | 3.2.1 | Classes for classification of specific requirements |  |
|  |  | 3.2.2 | User Registration profile and settings – FRU |  |
|  |  | 3.2.3 | Friends and Suggestions – FRF |  |
|  |  | 3.2.4 | Posts and Shares (Posts) – FRP |  |
|  |  | 3.2.5 | Chat - FRM |  |
|  | 3.3 | Use Cases | | 10 |
|  |  | 3.3.1 | Use Case UC1 |  |
|  |  | 3.3.2 | Use Case UC2 |  |
|  |  | 3.3.3 | Use Case UC3 |  |
|  |  | 3.3.4 | Use Case UC4 |  |
|  |  | 3.3.5 | Use Case UC5 |  |
|  |  | 3.3.6 | Use Case UC6 |  |
|  |  | 3.3.7 | Use Case UC7 |  |
|  |  | 3.3.8 | Use Case UC8 |  |
|  |  | 3.3.9 | Use Case UC9 |  |
|  |  | 3.3.10 | Use Case UC10 |  |
|  |  | 3.3.11 | Use Case UC11 |  |
|  |  | 3.3.12 | Use Case UC12 |  |
|  | 3.4 | Non-functional Requirements | | 13 |
|  |  | 3.4.1 | Performance requirements |  |
|  |  | 3.4.2 | Security requirements |  |
|  |  | 3.4.3 | Software Quality Attributes Requirements |  |

**Introduction**

**1.1 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 (Chitchat) for the web.

**1.2 Scope**

It allows users to connect with their friends and family through a common platform. Furthermore, users can share their posts and message with friends. These chats can be saved or deleted as per the user’s wishes. Users can also maintain, update or delete their account.

**1.3 Definitions & acronyms**

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

|  |  |
| --- | --- |
| **Term/Acronym** | **Description/Definition** |
| Chitchat | A social networking website. |
| FR | Functional Requirement |
| UC | Use Case |
| Team | The Group who are using this SRS for the project |
| Posts | Post is a post shared by the user on his slate. |
| Profile Posts | Posts by him or Profile users post |
| Messages | Chatting activity between two users |

**1.4 Intended Audience**

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

**1.5 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.

**2 Overall Description**

**2.1 Product Perspective**

Chitchat 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 posts and engage in Message conversations. This is an independent product and will be the first release of the product.

**2.2 Product Functions**

User can register to the Chitchat and login whenever he wishes to do so. User can build his profile, add or edit information, see his own profile and upload profile picture to his profile, can also change password .With this site 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 to his profile which will be visible to his connections and separately view profile posts by him/for him.

**2.3 Operating Environment**

Minimum: 2 CPU Cores (Recommended: 4 Cores or more)

Recommended memory: 2 GB for Windows platforms, 1 GB for Linux

Recommended disk space: 1024 MB

Required Applications: WAMP (Windows, Apache, MySQL, PHP)

**2.4 User characteristics**

Standard users may belong to any religion including any gender, nationality that can use computer’s browser. Chitchat 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.

**2.5 Constraints**

Chitchat will support the following browsers – IE 8+, Chrome 27.0+, and Firefox 30+.

Chitchat supports English language only. The first version of the website is intended to be viewed best on a desktop browser.

**2.6 Assumptions and Dependencies**

There will be availability of Internet via 3G, 4G or Wi-Fi.

The user of the site will be acquainted with Basic English language.

The user should have a valid email address in order to register into Chitchat.

Central server of the system must be able to handle all the incoming requests simultaneously.

The user has basic knowledge of computers and internet.

**2.7 Apportioning of requirements**

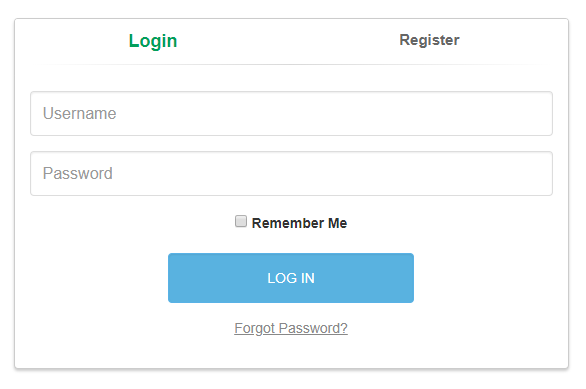
Requirements like login verification through email and many more will be planned in the future releases of the product. Also video sharing in Posts and Message may be included in future. Certain machine learning algorithms will be included for friend suggestions in future releases.

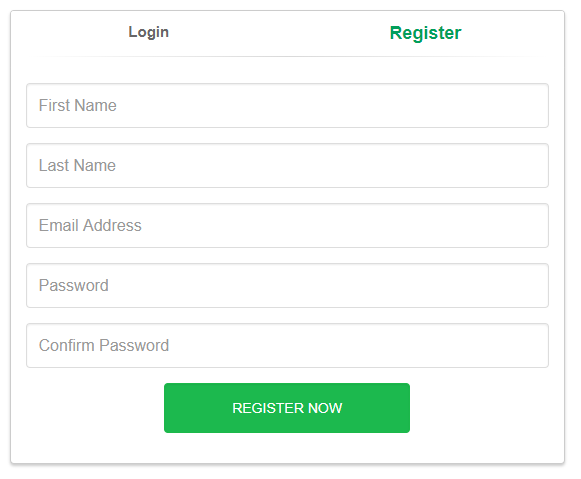
**3 Specific Requirements**

**3.1 External interface requirements**

**3.1.1 User interfaces**

Standard users will use the web browser to use Chitchat. Thus, it shall have a login page for users to login to Chitchat. 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.

**3.1.2 Hardware Interfaces**

N/A

**3.1.3 Software interfaces**

A free-tier AWS Account to deploy our website (Chitchat) on EC2 server.

**3.1.4 Communication interfaces**

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

**3.2 Functional requirements**

**3.2.1 Classes for classification of specific requirements**

The functional requirements are grouped as below

User Registration, profile and settings – [FRU]

Friends and suggestions – [FRF]

Posts and Shares (Posts) – [FRP]

Message – [FRM]

**3.2.2 User Registration, profile and settings - FRU**

3.2.2.1 Functional Requirement FRU1

ID – FRU1

Name – User Registration Dependency – None

Description –

User will create an account if there is no existing account associated with user’s email-id. User should enter all details such as first and last name, email-id, and password. All these are required fields and account creation cannot proceed unless these are filled. User is registered after this information is validated. First and last name should not exceed 25 characters each. Password should be at least 8 characters long, at most 15 characters.

3.2.2.2 Functional Requirement FRU2

ID – FRU2

Name – Email Verification Dependency – FRU1

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.

3.2.2.3 Functional Requirement FRU3

ID – FRU3

Name – Login Dependency – FRU1

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 user clicks on forgot password then reset password link should be emailed to the user. Input to each field must be validated.

3.2.2.4 Functional Requirement FRU4

ID – FRU4

Name – Add/Edit Personal information Dependency - None

Description –

A part of building user profile. User should be able to add the personal details like 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.

3.2.2.5 Functional Requirement FRU5

ID – FRU5

Name – Add/Edit account information Dependency –FRU1

Description –

A part of building user’s profile. User should be able to add/edit the account details like name, birth date, email-id, phone number and address. Name 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.

3.2.2.6 Functional Requirement FRU6

ID – FRU6

Name – Upload Profile and Cover Picture Dependency –None

Description –

User should be able to upload the profile and cover picture. The size of profile picture should be 200px \* 200px and for cover picture it should be 300px \* 150px. User may leave these field empty. Images should be visible to all. By default images should be assigned.

3.2.2.7 Functional Requirement FRU7

ID – FRU7

Name – Set account visibility settings Dependency –None

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.

3.2.2.8 Functional Requirement FRU8

ID – FRU8

Name – Show Profile Dependency –FRU1, FRU2, FRU3

Description –

Any logged-in user should be able to see any user’s public profile on Chitchat.

3.2.2.9 Functional Requirement FRU9

ID – FRU9

Name – Delete Account Dependency – FRU1, FRU2, FRU3

Description –

User may delete his account permanently from Chitchat depending on his wish.

User will be removed from his friends list once deleted. User information will be cleared.

**3.2.3 Friends and Suggestions - FRF**

3.2.3.1 Functional Requirement FRF1

ID – FRF1

Name – Add Friend Dependency – FRU1

Description –

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

3.2.3.2 Functional Requirement FRF2

ID – FRF2

Name – Accept/Ignore Friend Request Dependency – FRF1

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 “Add Friend” and the request is removed from the queue of the request receiver.

3.2.3.3 Functional Requirement FRF3

ID – FRF3

Name – Remove Friend Dependency – FRU1, FRF1

Description –

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

3.2.3.4 Functional Requirement FRF4

ID – FRF4

Name – List Friends Dependency – FRU3, FRU1, FRF1

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 10, and paging will be implemented to show more friends if the user has more than 10 friends.

3.2.3.5 Functional Requirement FRF5

ID – FRF5

Name – Search User Dependency – FRU3

Description –

User should be able to search for user who may be registered on Chitchat. User will put the name of the friend and click search after which the search results will display all the people registered on Chitchat by the searched name. When the search results are displayed the maximum number of results shown at a time will be 10, and paging will be implemented to show more results.

3.2.3.6 Functional Requirement FRF6

ID – FRF6

Name – Friend Suggestions Dependency – FRU3

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 5 at a time, and paging will be implemented to show more suggestions.

3.2.3.7 Functional Requirement FRF7

ID – FRF7

Name – Invite Friend Dependency – FRU3, FRU1

Description –

User should be able to invite more friends to join the Chitchat 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.

3.2.3.8 Functional Requirement FRF8

ID – FRF8

Name – Show Friend Online Dependency – FRU3, FRU1

Description –

User should be able to see currently whether his friend is online or not. The user can chat with online friend.

**3.2.4 Posts and Shares (Posts) – FRP**

3.2.4.1 Functional Requirement FRP1

ID – FRP1

Name – Post Dependency – FRU3

Description –

A registered user of Chitchat can share posts with his friends. The Post can only be plain English text with maximum 60000 characters. User can enter the Post in the designated text area and on clicking Post button; it should be visible to the user and his friends.

3.2.4.2 Functional Requirement FRP2

ID – FRP2

Name – Delete Post Dependency – FRU3, FRP1

Description –

A user can erase a Post that has been posted. After deletion, the Post should not be visible on the user’s Profile Posts. Deletion of a Post should also remove all the comments associated with that Post.

3.2.4.3 Functional Requirement FRP3

ID – FRP3

Name – List Posts Dependency – FRU3

Description –

After a user logs into Chitchat, he can see a listing of posts posted by him and his friends. The maximum number of posts displayed at a time will be 10, and paging will be implemented for more posts.

3.2.4.4 Functional Requirement FRP4

ID – FRP4

Name – Comment Dependency – FRU3, FRP1

Description –

For each Post 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 250 characters in length. To comment, user can see a text area under the Post, where he can enter the text.

3.2.4.5 Functional Requirement FRP5

ID – FRP5

Name – Delete Comment Dependency – FRU3, FRP4

Description –

A user can delete a comment written by him.

3.2.4.6 Functional Requirement FRP6

ID – FRP6

Name – Like a Post Dependency – FRU3, FRP1

Description –

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

3.2.4.7 Functional Requirement FRP7

ID – FRP7

Name – Unlike a Post Dependency – FRU3, FRP1

Description –

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

**3.2.5 Message - FRM**

3.2.5.1 Functional Requirement FRM1

ID – FRM1

Name – Message Dependency – FRU3, FRF8

Description –

Users can be able to initiate messaging with other user by get to their profile. User can see whether user is online or not. User should be able to receive message from any user who is online and within the profile 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 500 characters in length.

3.2.5.2 Functional Requirement FRM2

ID – FRM2

Name – Store Message Dependency – FRU3, FRM1

Description –

User can store the message. So that if user came back to online he/she can see the message from message notification.

3.2.5.3 Functional Requirement FRM3

ID – FRM4

Name – Retrieve Message Dependency – FRU3, FRM2

Description –

User should be able to retrieve Message when the Message when the user wants to see Previous Message history.

**3.3 Use Cases**

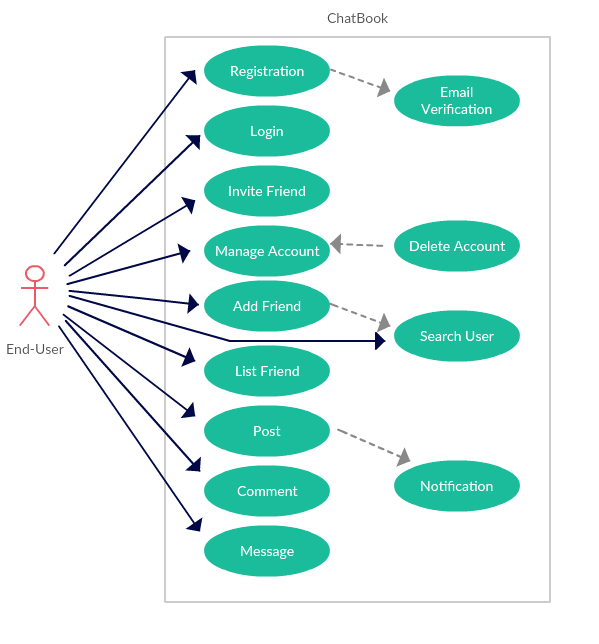


Figure 1: Use Case Diagram

**3.3.1 Use Case UC1**

|  |  |
| --- | --- |
| **Objective** | Registration |
| **Priority** | High |
| **Actor** | End-User |
| **Pre-Conditions** | The user has internet and opens the website on his browser. |
| **Post-Conditions** | The user is now a member of Chitchat website. |

**3.3.2 Use Case UC2**

|  |  |
| --- | --- |
| **Objective** | Email Verification |
| **Priority** | Low |
| **Actor** | End-User |
| **Pre-Conditions** | The user is registering to Chitchat website as a new user. |
| **Post-Conditions** | The user is now a registered user on Chitchat and can update account, post posts, add friends and chat with other registered friends. |

**3.3.3 Use Case UC3**

|  |  |
| --- | --- |
| **Objective** | Login |
| **Priority** | High |
| **Actor** | End-User |
| **Pre-Conditions** | User is a registered member of Chitchat. |
| **Post-Conditions** | The user logs in successfully and can now view his profile, friends and posts. |

**3.3.4 Use Case UC4**

|  |  |
| --- | --- |
| **Objective** | Manage Account |
| **Priority** | Medium |
| **Actor** | End-User |
| **Pre-Conditions** | User should logged into Chitchat. |
| **Post-Conditions** | The user account has been successfully updated as per his requirements, like, update account etc. |

**3.3.5 Use Case UC5**

|  |  |
| --- | --- |
| **Objective** | Delete Account |
| **Priority** | Medium |
| **Actor** | End-User |
| **Pre-Conditions** | User is a registered member of Chitchat. |
| **Post-Conditions** | Account is successfully deleted. |

**3.3.6 Use Case UC6**

|  |  |
| --- | --- |
| **Objective** | Add Friend |
| **Priority** | High |
| **Actor** | End-User |
| **Pre-Conditions** | User and friend are registered members of Chitchat. |
| **Post-Conditions** | Friend is added successfully. |

**3.3.7 Use Case UC7**

|  |  |
| --- | --- |
| **Objective** | List Friends |
| **Priority** | High |
| **Actor** | End-User |
| **Pre-Conditions** | User and friends are registered members of Chitchat. |
| **Post-Conditions** | User sees a list of his friends. |

**3.3.8 Use Case UC8**

|  |  |
| --- | --- |
| **Objective** | Invite Friend |
| **Priority** | Low |
| **Actor** | End-User |
| **Pre-Conditions** | User is a registered member of Chitchat. |
| **Post-Conditions** | Invite Friend on his email to register as a new user. |

**3.3.9 Use Case UC9**

|  |  |
| --- | --- |
| **Objective** | Post |
| **Priority** | High |
| **Actor** | End-User |
| **Pre-Conditions** | User is a registered member of Chitchat. |
| **Post-Conditions** | Posted posts are visible to the user and his friends. |

**3.3.10 Use Case UC10**

|  |  |
| --- | --- |
| **Objective** | Comment |
| **Priority** | High |
| **Actor** | End-User |
| **Pre-Conditions** | User is a registered member of Chitchat. |
| **Post-Conditions** | Posted comments are visible to the user and his friends. |

**3.3.11 Use Case UC11**

|  |  |
| --- | --- |
| **Objective** | Message |
| **Priority** | High |
| **Actor** | End-User |
| **Pre-Conditions** | User is a registered member of Chitchat. |
| **Post-Conditions** | The user successfully performs direct Message with other User. |

**3.3.12 Use Case UC12**

|  |  |
| --- | --- |
| **Objective** | Notification |
| **Priority** | Low |
| **Actor** | End-User |
| **Pre-Conditions** | User posted or commented. |
| **Post-Conditions** | Notification sends to the friends. |

**3.4 Non-functional Requirements**

**3.4.1 Performance requirements**

3.4.1.1 Scalability

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

3.4.1.2 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 Chitchat should perform a quick search among the users on the database.

**3.4.2 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 safe guarding the user’s information.

**3.4.3 Software Quality Attributes Requirements**

3.4.3.1 Usability

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

3.4.3.2 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.

3.4.3.3 Availability

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

3.4.3.4 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.

3.4.3.5 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.