***Enterprise Chat System***

Software Requirements SpecificationsV1.2

Date: 11-06-2017

**Team Name:**

**Wireshark**

**Team Members: -**

1. **Ginka Anusha**
2. **Gurram Karthik**
3. **Hossen Saddam**
4. **Injeti Ravi Varma**
5. **Ivvala Venkata Sai Krishna Chaitanya**
6. **Kadamati Venkata Sai Sidhartha**
7. **Karumanchi Mahesh Chowdary**
8. **Kondepati Divya Naga Krishna**
9. **Kovvuru Nasirali**
10. **Sathi Santhosh Reddy**
11. **Janagam Anirudh**

**Type of Document:** Software Requirements Specification

**Version Number :**1.2

**Publication Date:** June 11th, 2017

**Summary of contents:**

Section 1: Preface

Section 2: Glossary and abbreviations

Section 3: System Architecture

Section 4: Requirements

4.1 User Requirements

4.2 System Requirements

Section 5: References

**1.PREFACE**

The aim of this project is to develop an Enterprise Chat System such that the employees of Xtreme Security can communicate safely. A central database is developed to store user information associated with each sent (or) received the message. Users have the capacity to check whether the messages sent have been received by the recipient (or), not alongside the timestamp. Users can also have an administrative role such as delete a user, block and unblock a user, emergency broadcast messages. Users can select their chat status (idle, busy, offline, available) based upon their interest. All the data is stored in the database in which admin can view the statistics of the user.

**Release version 1.2:**

* Changes are made in section 3: System architecture is explained in detail, Frontend module , Database module and backend module are described in detail using UML diagrams.
* Changes made in section 4.1 such that requirements of the user are described in detail with creation date, dependencies etc.
* Updated section 4.2 such that system requirements of the system are described in detail.

**Release version 1.1:**

* Changes made in section 1
* Requirements in section 4 are more clearly described.

**Release version 1.0:**

* Initial version

The remainder of the document is organised as follows

Section 2: Shows the technical terms and their meaning used in this document

Section 3: Shows the system architecture how the project is designed to work.

Section4: Shows the user and system requirements.

**2.Glossary and Abbreviations:**

**GUI:** Graphical User Interface.

**API:** Application programming interface

**SQL:** Standard Queuing Language.

**IP ADDRESS:** Internal Protocol Address.

**PHP:** Hypertext Pre-processor

**HTML:** Hyper Text Markup Language

**JSON:** JavaScript Object Notation

**REST:** Representational State Transfer

**API**: Application program interface

**MYSQL:** An open source relational database management system

**3. System Architecture:**

**3.1 System Architecture:**

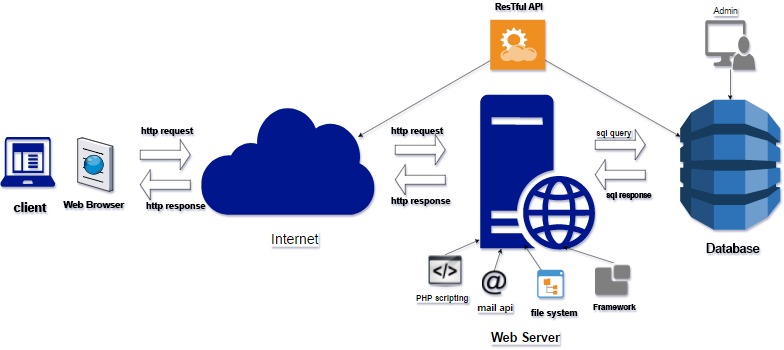


Fig 3.1 System Architecture

**System Architecture Description:**

**Browser: -**A browser is a software application used for retrieving, presenting and transforming information across World Wide Web. They also used to access information provided by the web server.

**JavaScript Engine: -**It is a program used to execute JavaScript Code.

**Web Server: -** A web server is a computer system that processes request over HTTP.

**Restful API** (Representational State Transfer): - It is used to access the database using different software.

**PHP** (Hyper Text Pre-processor): - It is a server-side scripting language. It is embedded into HTML.

**Framework**: -It is an application specific software providing the generic functionalities can be selectively changed by the user.

**File System: -**It is to organise the different files in the web server which user can access them through Web Browser.

**Data Base**:-It is the collection of organised information in the form of Query, tables, schemes, reports, records, objects which can be create, update, delete easily.

**Encryption** of the web server is designed by using SSL certificates

**3.2 Use Case Module :**

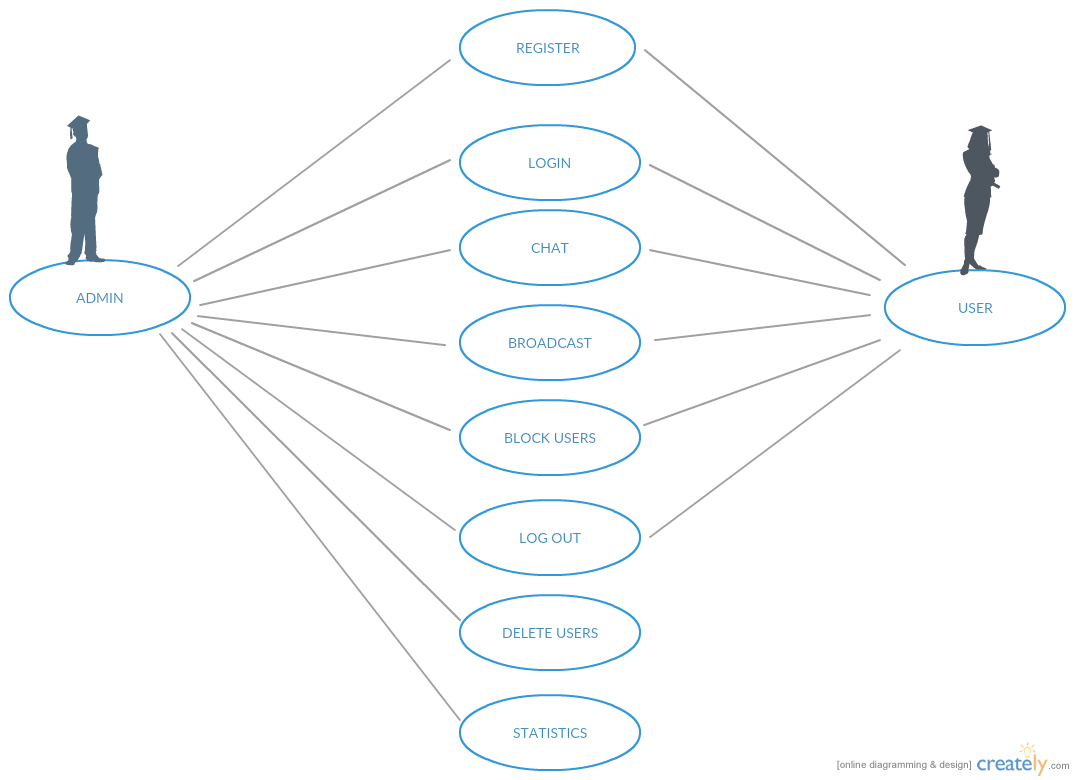


Fig 3.2 Use Case

* **Admin can perform the following activities :**

1. Admin registration
2. Admin login
3. Chat
4. Broadcast chat
5. Block users
6. Logout
7. Delete Users
8. Statistics

* **User can perform the following activities:**

1. User registration
2. User login
3. Chat
4. Broadcast chat
5. Block users
6. Logout

**3.3 Functional Module :**

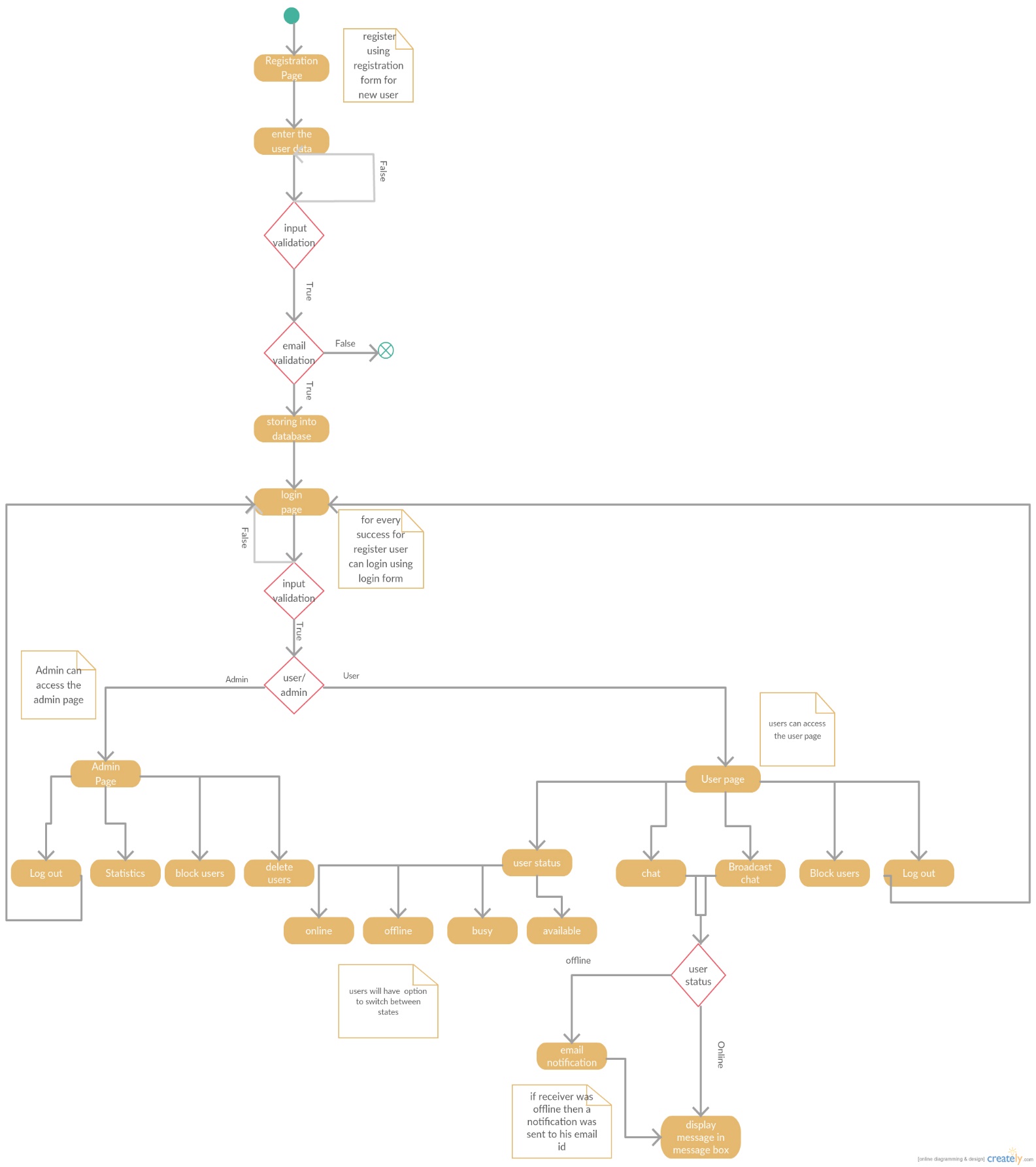


Fig 3.3 Activity diagram of system

* When user enter into the application they must register using the registration form.
* Input validation will be to the user.
* Email will be sent to users email id to activate users account.
* Users data will be store int database.
* After successful registration then user can login using login form.
* After successful login users can view their profile page.
* Form the profile page users can send messages to other users in the address book.
* If the users is offline the an email will be sent to the users email to notify the message.
* Users will have additional functonalities like block the users, emergency broadcast message and switching through different states(online,offline,idel,busy)
* Admin can moniter the users, block and delete the users, view the users statistics and manage the database.

**3.4 Front end module :**

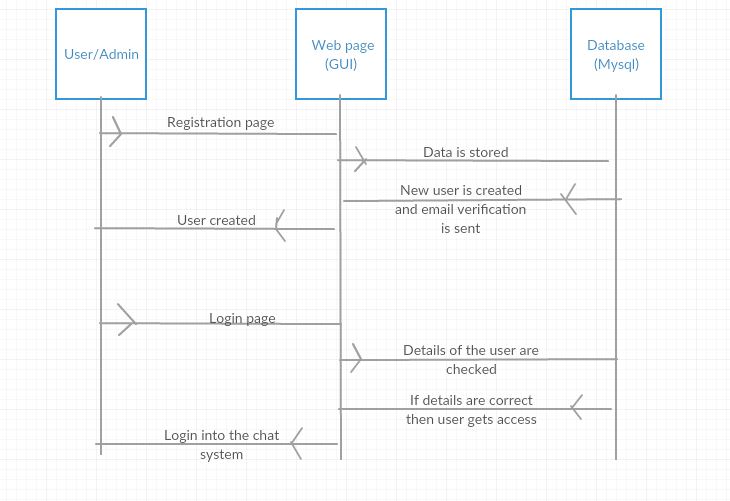
****

Fig 3.5 Front-end module

**Front end module functions as the following steps :**

* User/Admin needs to resgister to get the access of the chat system by filling the details given in the registration page and an email is sent to the mail id of the user to activate their account and then user account gets activated.
* User/Admin can log-in into the system, database verifies the details entered by the user matches to the records of data stored and if details are correct then user gets access into the chat system and can perform activities

**3.4 Database Module and Backend module :**

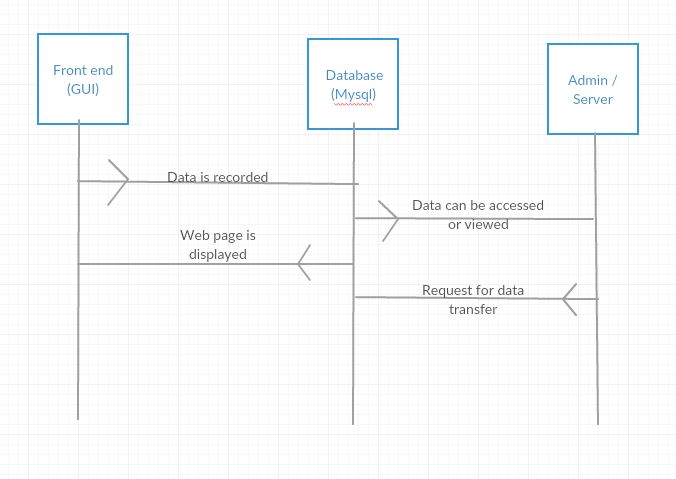
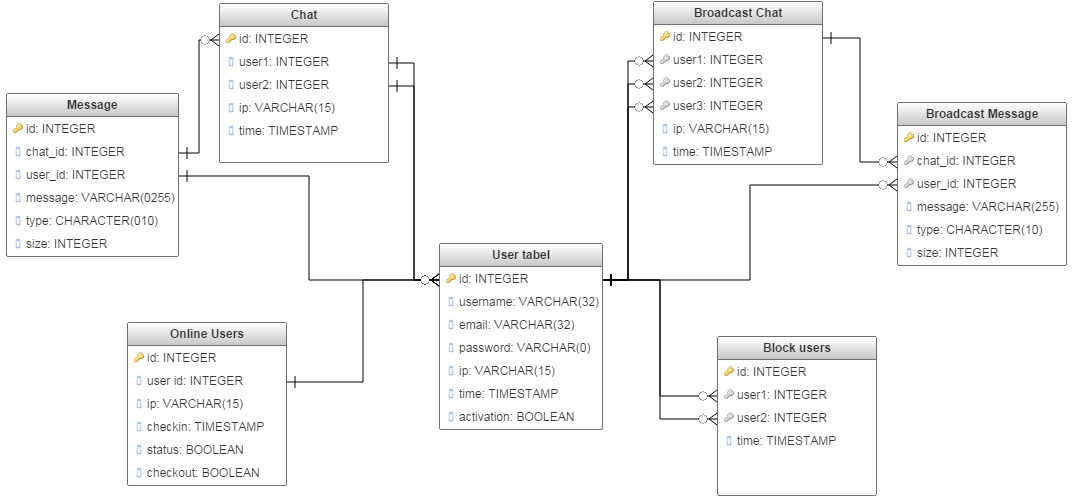


Fig 3.5 Database and Backend achitecture

 Fig 3.6 Database Design

The database will contain the tables and their relation as mentioned in Fig 3.5

**4.Requirements:**

Requirements for a software development project is divided into two parts. They are:

**4.1 User Requirements:**

**The user requirement(s) specification** is a document usually that specifies what the user expects the software to be able to do.

Following are the requirements provided for the user.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Identification String** | **Requirement** | **Creation date** | **Change date** | **Module** | **Type** | **Dependencies** | **Test** | **Assignee** | **Description** |
| **UR1** | Web based GUI | **04-05-2017** | **28-05-2017** | Front-end | Functional | **SR1, SR3** |  | Krishna  Siddartha | It is used ti describe the GUI for the users. |
| **UR2** | User registration | **05-05-2017** | **28-05-2017** | Front End | Functional | UR1,SR1,SR2 |  | Krishna;  Siddartha | It is the HTML page for the user to display registration page |
| **UR3** | User login | **05-05-2017** | **28-05-2017** | Front end | Functional | UR1,UR2,SR1,SR2 |  | Krishna;  Siddartha | It is the HTML page for the user to display login page |
| **UR4** | Users list to select a user and start a chat | **15-05-2017** | **28-05-2017** | Front End | Function nil | UR3,SR1,SR2 |  | Nasirali  Anirudh | It is the HTML page to display the user functionalities |
| **UR5** | Chat service – send messages to other users | **25-05-2017** | **28-05-2017** | Front End | Functional | UR4,SR1,SR2 |  | Saddam;  Chaitanya; | Provide an option to send messages to other users |
| **UR6** | Chat service – receive messages from other users | **25-05-2017** | **28-05-2017** | Front End | Functio nal | UR4,SR1,SR2 |  | Chaitanya;  Santosh | User must be able to receive messages from other users |
| **UR7** | Validation of Registration and login pages of user and admin | **04-05-2017** | **28-05-17** | Front-end | Functional | UR2,UR3 |  | Krishna  Santosh | User and admin need to provode correct credentials to register or login into the chat system |
| **UR8** | Offline messages to be sent as email | **27-05-2017** | **28-05-2017** | Back End | Functio nal | UR5,UR22 |  | Anusha;  Anirudh;  Ravi Varma | If the users is offline the an email will be sent to the users email to notify the message. |
| **UR9** | Send and receive files | **01-06-2017** | **05-06-2017** | Front End | Function nil | UR4, SR1, SR2 |  | Karthik;  Mahesh | Users can be able to send and receive files from other users |
| **UR10** | Admin-  view users | **29-06-2017** | **04-06-2017** | Back End | Function nil | UR13, SR1, SR2 |  | Chaitanya;  karthik | Admin must be able to view all users from database. |
| **UR11** | Admin-Delete users | **26-05-2017** | **04-06-2017** | Back end | Functional | UR13, SR1, SR2 |  | Chaitanya;  Karthik | Admin must be able to delete users from the database. |
| **UR12** | Admin –Block/  deactivate users | **28-05-2017** | **04-06-2017** | Backend | Functional | UR13, SR1, SR2 |  | Chaitanya;  Karthik | Admin must be able to block/deactivate users from not accessing the chat service |
| **UR13** | Admin Authentication | **28-05-2017** | **04-06-2017** | Back end | Functional | UR25, SR2, SR2 |  | Chaitanya  Saddam | Only authorized users must be able to use the service |
| **UR14** | Admin-Unblock/  activate users | **03-06-2017** | **04-06-2017** | Back End | Functional | UR13, SR1, SR2 |  | Chaitanya;  Karthik | Admin must be able to unblock/activate users. |
| **UR15** | Message Status(read/unread) | **08-05-2017** | **22-05-2017** | Front-end | Functional | UR4, UR5 |  | Santosh;  Krishna | User must able to see the message which are sent or received |
| **UR16** | Message history | **08-05-2017** | **23-05-17** | Front-end | Functional | UR4, UR5, UR15 |  | Chaitanya,  Karthik | Previous messages can be viewed |
| **UR17** | Timestamp of sent and received messages | **08-05-2017** | **23-05-2017** | Front-end | Functional | UR15 |  | Santosh;  Mahesh | User must be able to see when the message is sent and received through the webpage |
| **UR18** | Encryption-web server | **11-05-2017** | **24-05-2017** | Back-end | Functional | SR1,  SR2 |  | Ravi Varma;  Saddam | Web server must be encrypted by using suitable algorithms/certificates |
| **UR19** | Blocked user cannot login | **15-05-2017** | **04-06-2017** | Front-end | Functional | UR14, UR3 |  | Chaitanya;  Anusha | User when blocked by admin cannot login into the chat system. |
| **UR20** | View sent and received files | **10-05-2017** | **20-05-2017** | Front-end | Functional | UR9, SR1, SR2 |  | Chaitanya;  Anirudh;  Sidartha | User must be able to view send and receive the binary files from other users |
| **UR21** | Restful API | **17-05-2017** | **02-06-2017** | Back-end | Functional | SR1,SR2 |  | Saddam;  Nasir | It is used for the user-server interaction |
| **UR22** | Manage Status of the user(available,busy,ofline) | **09-05-2017** | **24-05-2017** | Front-end | Functional | UR3, SR1, SR2 |  | Siddhartha;  Krishna | User must be able to manage their status based upon their interest |
| **UR23** | Activation of account through email | **14-05-2017** | **27-05-2017** | Back-end | Functional | UR3,  SR1, SR2 |  | Anusha,  Ravi varma  Chaitanya | After user registration necessary email must be sent to user for account activation |
| **UR24** | Admin-  View user statistics | **28-05-2017** | **04-06-2017** | Back-end | Functional | UR13, SR1, SR2. |  | Karthik  Chaitanya | Admin must be able to see the user statistics from the database in tabular/graphical manner |
| **UR25** | Admin registration | **28-05-2017** | **02-06-2017** | Back-end and Front-end | Functional | UR1, SR1, SR2 |  | Karthik  Mahesh | Admin need to register and authenticate to get access to admin functionalities |

**4.2 System Requirements:**

These are the technical requirements that complement the user requirements and provide information for design and implementation of product.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Identification string** | **Requirement** | **Creation Date** | **Change date** | **Module** | **Type** | **Depen**  **dencies** | **Test** | **Assignee** | **Description** |
| SR1 | Server | **30-04-2017** | **28-05-2017** | Backend | Functio  nal | SR2,  SR3 |  | Chaitanya;  Saddam;  Krishna;  Karthik | Apache server is used to deliver the web page |
| SR2 | Database | **30-04-2017** | **28-05-2017** | Backend | Function  nil | SR3 |  | Chaitanya;  SaiSidhartha;  Nasirali  Anusha | MySQL database will be such that statistics of the data can be details of the users are stored |
| SR3 | Design and Programming  Languages: | **30-04-2017** | **28-05-2017** |  | Functional |  |  | **FRONTEND:**  Krishna;  SaiSidartha;  Saddam;  Nasirali  **BACKEND:**  Chaitanya;  Anusha;  Karthik;  Ravi Varma;  Santosh  **DATABASE:**  Chaitanya;  Anirudh;  Mahesh;  Krishna | HTML;  CSS;  PHP are used for frontend.GUI  PHP;  JSON;  MYSQL;  JAVA SCRIPT;  AJAX  Will be used accordingly for backend. |

**5.References:**

[1] Software Documentation, <https://en.wikipedia.org/wiki/Softwaredocumentation>

[2]. Ian Somerville UU4 Software Engineering. 9th ed.