

## **Enterprise Chat System**

### **Software Requirement Specification**

**Version: 1.2**

**Date: 24/04/2016**

**Team Name:**

**Invictus**

**Team Members:**

- 1. Balusu, Geethanjali**
- 2. Budda, Shiva Tarun**
- 3. Byreddy, Sreenibha Reddy**
- 4. Kalaparti, Adithya**
- 5. Mamidi, Mounica**
- 6. Mara, Nikhil**
- 7. Mohammed, Zaahid Mukarram**
- 8. Neelam, Sankeerthana**
- 9. Sanagari, Vivek**
- 10. Srinivas, Sri Krishna**
- 11. Yerrapragada, Sita Rama Suryateja**

## 1. Preface

The product, which is to be developed, must perform secure chat communication through internet and must secure confidential data against 3<sup>rd</sup> party breaches. A central database is developed which stores the meta data of sent and received messages for the user to browse chat history easily. User is also provided to view the status (Available, idle, busy, offline) of the other user. The user can set up the status- available or busy. Call record, timestamped information (date and time) about the sender and receiver of the message s and corresponding IP addresses, along with metadata such as message size, type is stored in MySQL database and only authorized users i.e., administrator at the central server can access it.

### **Release v1.1 on 15-05-2016**

-Changes made in section 3 replaced fig.1.

-Changes made in section 4 updated unit testing plan of all components and also updated user requirements.

### **Release v1.0 on 24-04-2016**

-initial release

The remainder of the document is organized as follows. In Section 2, Glossary and Abbreviations of terms used in the document is provided. Section 3 adds System Architecture and describes the modules-frontend, database and backend. Section 4 includes Requirements- User and System Requirements. In Section 5, References are added.

## 2. Glossary and Abbreviations

- ☐ **PHP:** Hypertext Preprocessor  
PHP is open source general-purpose scripting language and an interpreter used for web development and which can be embedded into HTML.
- ☐ **JSON:** JavaScript Object Notation  
JSON is a minimal format for structuring data. Easy to Read and Write. It is used to exchange data between server and web application.
- ☐ **AJAX:** Asynchronous JavaScript and XML  
AJAX is the method of data exchange with a server and updating a part of web page, without reloading the entire page. It can transmit and receive information in a various format - JSON, XML, HTML, and even text files.
- ☐ **GUI:** Graphical User Interface  
GUI allows the user to interact with graphical icons and visual indicators with electronic devices.
- ☐ **API:** Application Programming Interface

It is a set of programming instructions, routines and protocols to build web-based software or applications.

□ **REST:** Representational State Transfer

REST is an architectural style and is an approach to communicate development of Web services. Web services using REST architecture are called RESTful APIs.

### 3. System Architecture

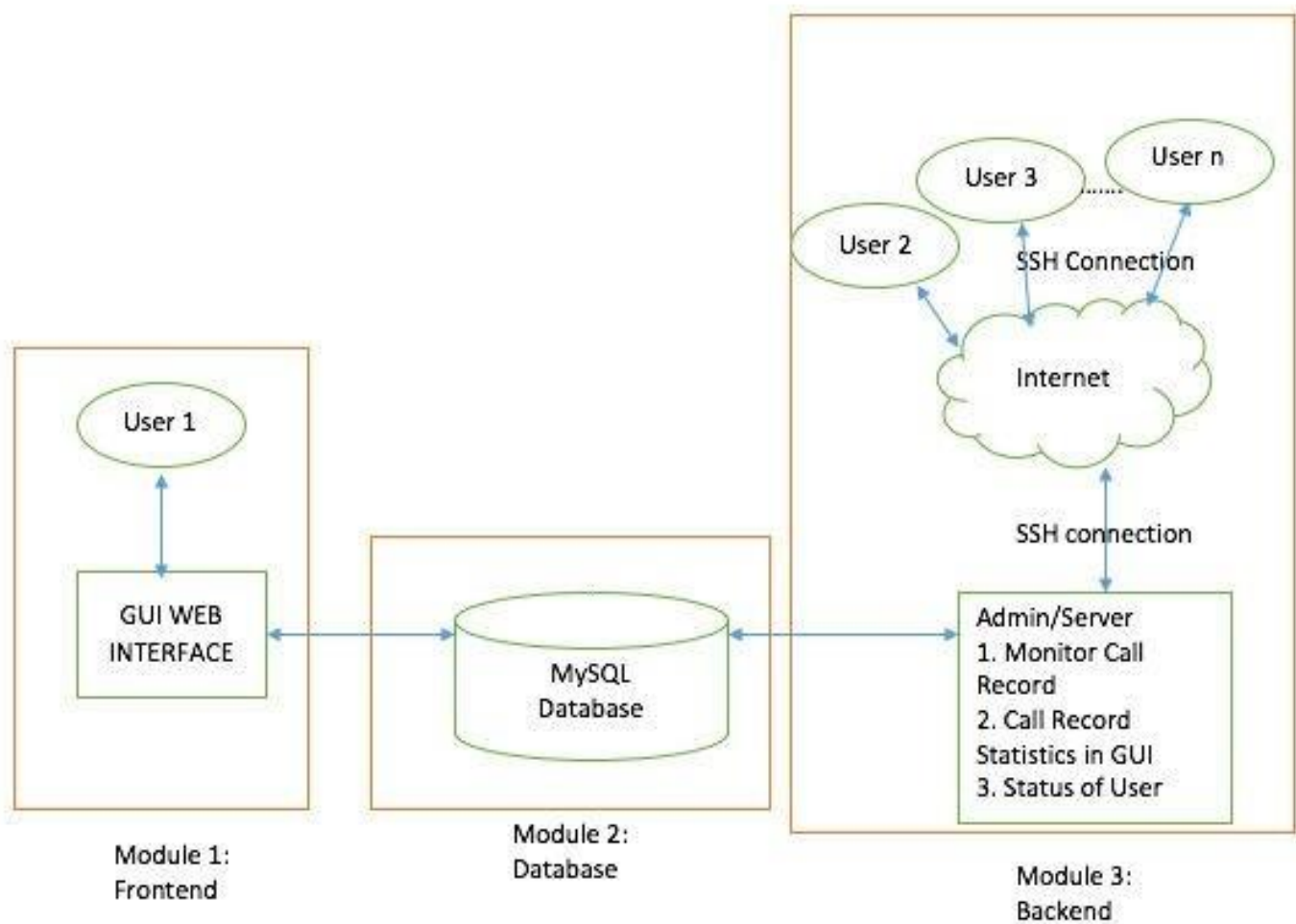


Fig.1 System Architecture

The chat application, must provide a secure chat conversation from user-to-user. The chat system is thus being divided to 3 modules. They are:

Module 1- Frontend: This module describes how the user connects to the other users using the interactive web page.

Module 2- Database: This module stores the required details of the user.

Module 3- Backend: To provide administration and monitoring facility for the administrator

### 3.1 FRONTEND

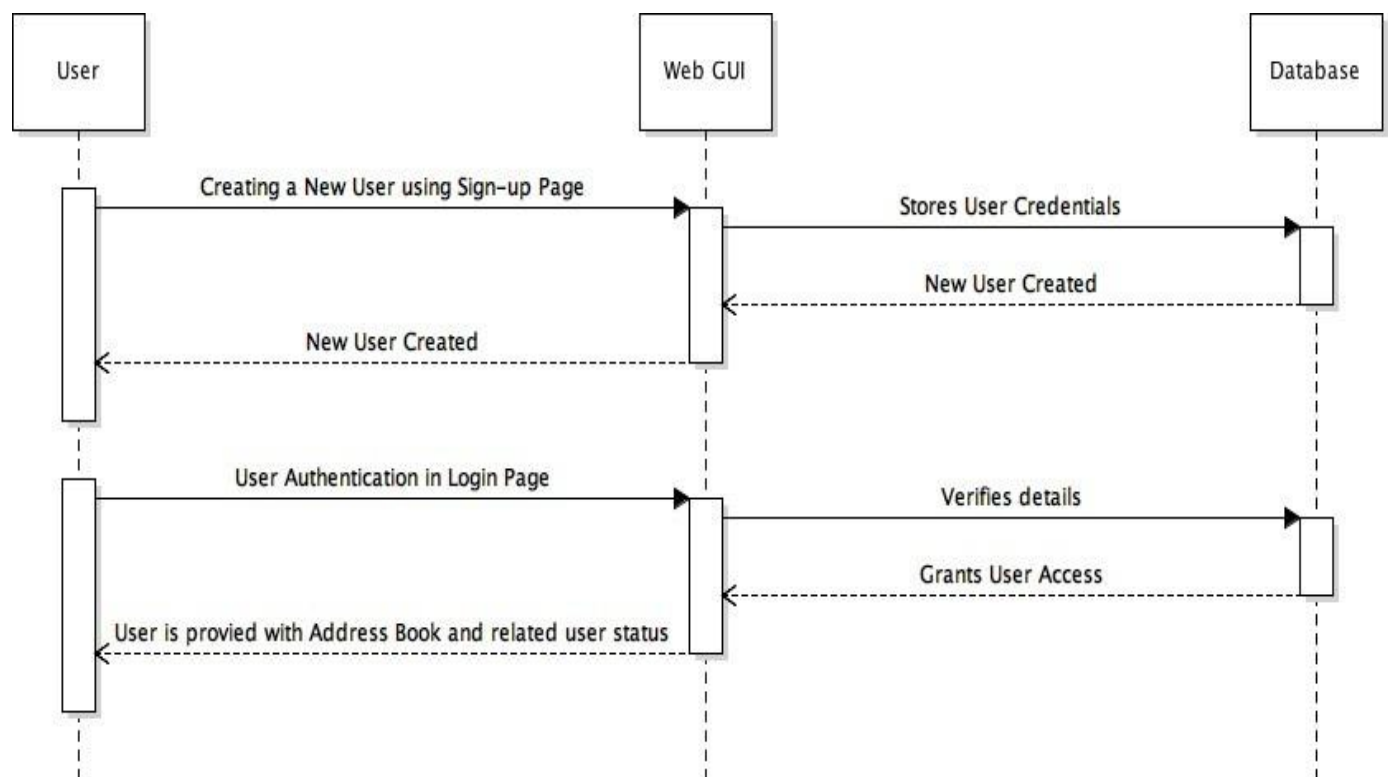


Fig.2 Front End Architecture Module

It must perform the following two tasks:

1. Sign-up Page: To register a New User using User name, mail ID and Password through Web-GUI and store the user credentials at database.
2. Authentication: (Login Page): User provides his Email Id and password in Web GUI. Then, verify User credentials from database and provide User access. User is provided with his Address book and status of respective users- online, offline, busy or idle. The User can browse his conversation history with other users.

The login and signup pages are developed using HTML, CSS, JavaScript.

### 3.2 DATABASE

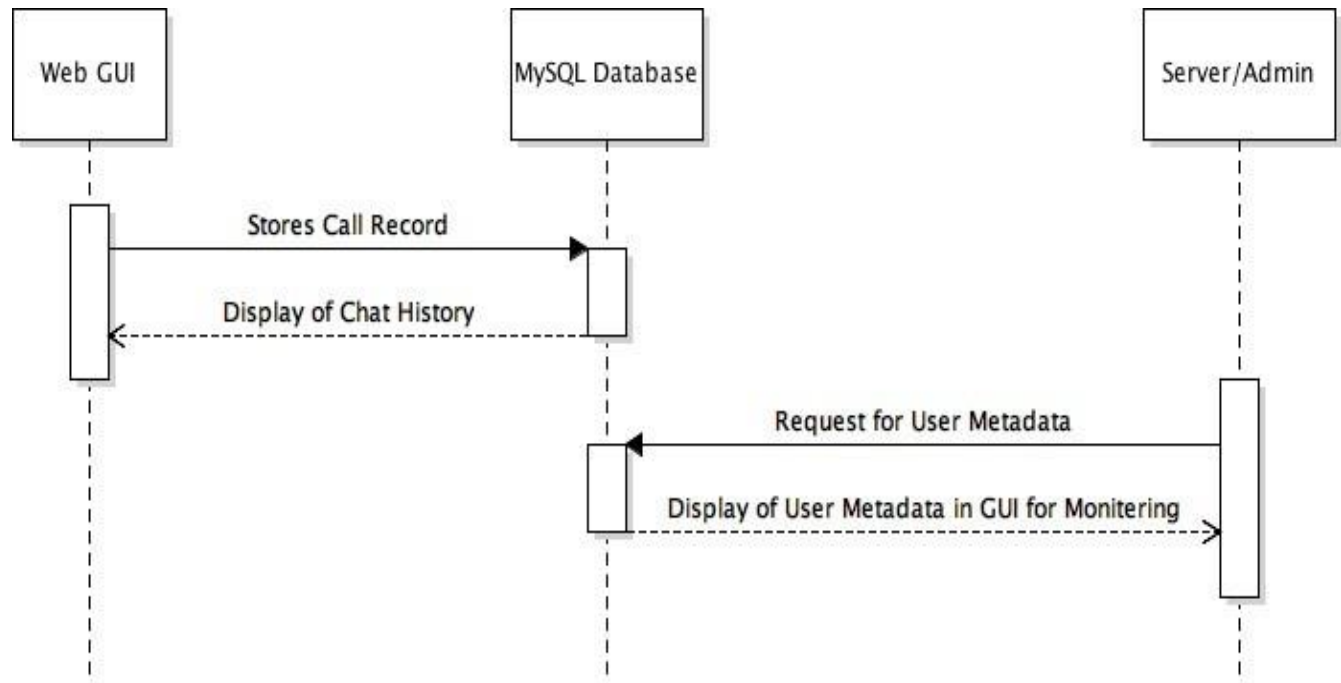


Fig.3 Database Architecture Module

It must perform the following tasks:

1. Stores conversation details of the user
2. To provide chat history to the user
3. To provide metadata information for administration to Admin/Server in the form of graphs, table, etc.

MySQL database is expected to store the following data:

- ☐ Authentication data of Registered Users
- ☐ Send and received messages
- ☐ Date and Time of the messages
- ☐ Name of sender and recipient
- ☐ If the recipient has seen the message

- ☐ Read time of recipient
- Chat conversation with various peers

### 3.3 BACKEND

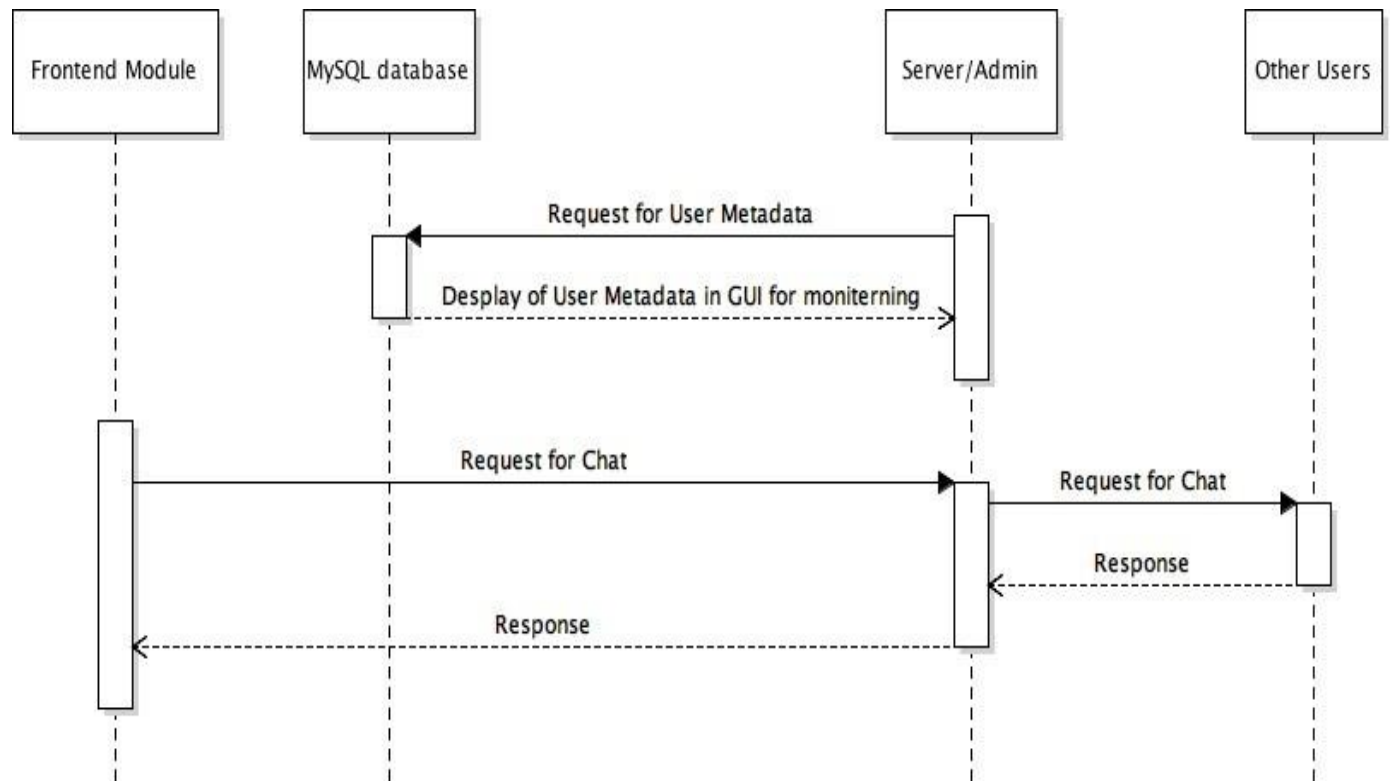


Fig.4 Server Architecture Module

It includes server and Administrator. Administrator is provided with following features:

- ☐ Call Record:
  1. Time stamp information of Sender and Receiver
  2. IP Addresses
  3. Metadata like message size, message type- message or file format
- ☐ Statistics of communication is provided from the database in GUI format- tables, graphs, etc.
- ☐ Messages from user-to-user or user-to-server are encrypted to provide security. Messages are encoded with RESTful API with JSON encoding.

It must also add the following features:

- ☐ On creating a new user, mail concerning account details and validity period of the account.
- ☐ User is allowed to send text of 4 KB or a arbitrary size Binary File.
- ☐ User can:
  1. Delete his account
  2. Block or unblock user
  3. Emergency Broadcast message like crash problem
- ☐ If user has not read a message, then he is sent mail regarding the message information.

Backend scripting is done using PHP and information exchange between database and server is done by JavaScript and AJAX programming language.

#### 4. Requirements

The requirements involve user requirements and system requirements.

##### 4.1 User requirements

The following are the services provide for the user.

Requirements	Creation date	Change date	Module	Type	Dependencies	Test	Assignee	description
Web based GUI (UR1)			Front end	Functional	SR1,SR3		Zaahid, Sreenibha, Mounica	User Requires simple web based GUI for easy usage.
User authentication (UR2)			Front end	Functional	SR1, SR2, SR3		Adithya, Vivek, Surya Teja	The user is required to authenticate himself as admin/user.
Configure user account (UR3)			Front end	Functional	SR1, SR2, SR3		Sankeert hana,Shiv a Tarun	The user must be able to change his account credentials.

Status of other users available in phone book (UR4)			Back end	Functional	SR1, SR2, SR3		Sankeerthana, Sree Nibha	The status of other users who are available in the phone book shall be shown.
Unread message (phone book) (UR5)			Back End	Functional	SR1, SR2, SR3		Geethanjali, Adithya	Number of unread messages with other user must be shown in phone book
Read time (phone book) (UR6)			Back end	Functional	SR1, SR2, SR3		Vivek, Adithya	Message read time must be displayed.
Block (phone book) (UR7)			Back end	Functional	SR1, SR2, SR3		Sri Krishna, shiva tarun	The user must be able to block any user from phone book who does'nt want to communicate
Search Box (UR8)			Front end	Functional	SR1, SR2, SR3		Vivek, Geethanjali, surya teja	The user must be able to search other users who have account in database.



User status (UR9)			Front end	Functional	SR1, SR2, SR3		Nikhil, Shiva Tarun, Sri Krishna	The Status of the user must change between offline/available/busy/idle and must be updated in database.
Administrative functions- Broadcast mail (UR10)			Back end	Functional	SR1, SR2, SR3		Zaahid, A Adithya	Administrator must be given provision of broadcasting an emergency mail.
Administrative functions-View records (UR11)			Back end	Functional	SR1, SR2, SR3		Nikhil, sree nibha	Administrator must be given provision of view database records.
Administrative functions-View Statistics. (UR12)			Back end	Functional	SR1, SR2, SR3		Sankeert hana, Adithya	Administrator must be given provision of monitoring statistics of records available at the database.
Chat messages(send and receive) (UR13)			Back end	Functional	SR1, SR2, SR3		Adithya, Nikhil, Sri Krishna	The user must be able to send and receive messages from other Users upto 4Kb.

Binary files(send and receive) (UR14)			Back end	Functional	SR1, SR2, SR3		Mounika, Zaahid	The user must be able to send and receive Binary files from other Users upto 4Kb.
Manages Status (UR15)			Back end	Functional	SR1, SR2, SR3		Nikhil, Geethanjali	The user must be able to manage status from available and busy.
Email services-offline users (UR16)			Back end	Functional	SR1, SR2, SR3		Shiva Tarun, Sankeert hana	The offline users must be notified about any message from other user in order to reduce delay through mail.
Email services-on creating a new account (UR17)			Back end	Functional	SR1, SR2, SR3		Mounika, Sri Krishna	The users must be notified about his login details and validity period through mail.
User manual (UR18)			Frontend	Non Functional	SR3		Sri Krishna, sankeert hana	The user needs to be guided how to use the service.

## 4.2 System requirements

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

Requirements	Creation date	Change date	Module	Type	Dependencies	Test	Assignee	description
Web Server (SR1)			Back end	Functional			Nikhil, Sri Krishna Shiva Tarun Sankeerthana	This is used to deliver web pages. (Example: APACHE local server)
Database (SR2)			Back end	Functional			Adithya Shiva Tarun Surya Teja, Geethanjali Vivek.	To access the data statistics and details of the users.
Programming language (SR3)				Non Functional			Frontend- Zaahid Sreenibha Mounica Vivek.  Back end- Nikhil Sri Krishna, Shiva Tarun, Sankeerthana .  Database- Adithya Shiva Tarun, Surya Teja, Geethanjali, Vivek.	Front end-HTML, CSS, JAVA Script for web based GUI. Back end-PHP, SQL, AJAX. JSON for encryption.

## 5. References

[1]. Ian Sommerville. Software Engineering. 9th ed. [2]. Xtreme Security Project Request, available at link <https://drive.google.com/file/d/0B-IPzXOrtPFSeWZVY1pzSnRIV0E/view>