***Enterprise Chat System***

Project Specification V1.2

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:** Project Specification

**Version Number :**1.2

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

**Summary of contents:**

Section 1: Preface

Section 2: Glossary and abbreviations

Section 3: Background

Section 4: Proposed Solution

Section 5: Limitations

Section 6: Time Plan

Section 7: Project Organization

Section 8: Configuration Management

Section 9: Progress Tracking

Section 10: Quality Control

Section 11: Risk Management

Section 12: System Release Plan

Section 13: References

**1.PREFACE**

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 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.  The administrator can also view statistics of records graphically as tables/graphs (GUI) available on main database server (SQL). The messages and the metadata must remain accessible so a user can browse the history of their conversations with various peers. Users can select their chat status (idle, busy, offline, available) based upon their interest. When the user is offline, messages should be sent as email to the user to their mail address. Authentication of the user depends on their username and password. And those credentials are stored in the database. This process is necessary to avoid any security breaches by the unknown third party and to maintain the confidentiality of communication between them. When an account is created an e-mail is sent to the owner of the account with login details and validity period for the account. Chat system must have friendly GUI providing all necessities to edit and read messages, configure account, add / remove users.

**Release version 1.2 on 4/06/2017**

* Administrative role is described and the remainder of document is organised in section 1
* In section 3we added a chat system architecture.

**Release version 1.1 on 07-05-2017**

* Administrative role is described in section 1
* Milestone are more specific and what is delivered in each milestone is described in section 6
* Changes made in configuration management in section 8.
* Updated information of progress tracking is done in section 9.
* A detailed description of quality control of the product is updates in section 10
* Time schedule is revised in section 12

**Release version 1.0 on 23-04-2017**

* Initial version

The remainder of the document is organized as follows:

* Section 2 :- Describes the glossary and abbreviations which include technical words mentioned in the document.
* Section 3: - Provide an overview of the customer's problems or needs that will be addressed by the product. Feel free to use figures if you think it enhaces the clarity of the information (this goes for the entire document).
* Section 4: - Gives the accurate description of how the customer wants the problems/needs of his system to be addressed.
* Section 5: - Describes the limitations of the project
* Section 6: - Shows the time plan of the project and what will be delivered at each milestone/tollgate.
* Section 7: - Describes the Project organization between the team members.
* Section 8: - Shows the configuration management, differences between the previous and existing versions.
* Section 9: - Describes the tracking of the project progress to reach the final milestone/tollgate.
* Section10: - Describes how to ensure the quality of the project before delivering it to the customer
* Section11: - Describes the types of risks that will affect the project management and strategies to overcome those risks.
* Section 12: - This shows the System Release Plan which ensures that the quality product is being released properly. They are divided into three categories: Testing Plan, Packaging Plan, Documentation Plan. Sections 13 adds References.

**2.Glossary and Abbreviations:**

**GUI:** Graphical User Interface.

**API:** Application programming interface

**PDF:** Portable document format

**Project Libre:** Project management software system

**SQL:** Standard Queuing Language.

**IP ADDRESS:** Internal Protocol Address.

**OFFLINE:** It implies that the client is not connected with the system.

**AVAILABLE:** It shows the state where the client is associated and prepared tostart a discussion

**BUSY:** It shows the state where the client is connected but unable to start adiscussion

**IDLE:** It shows the state that the client is connected to the system yet there is noresponse from the client

**PHP:** Hypertext Pre-processor

**HTML:** Hyper Text Mark-up Language

**JSON**: JavaScript Object Notation

**REST:** Representational State Transfer

**API**: Application program interface

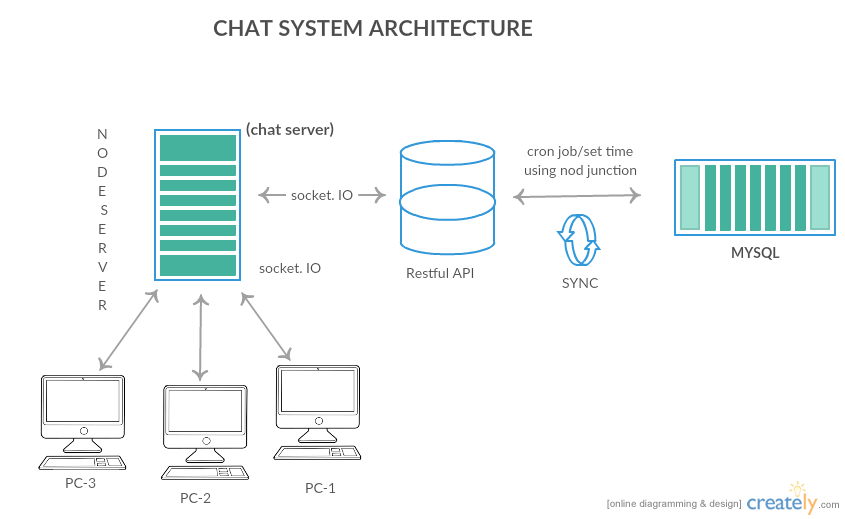
**MYSQL:** An open source relational database management system

**3.Background:**

The company Xtreme Security has grown tremendously in the last five years expanding its points of presence throughout the world. It offers security to governmental and military organizations. Hence Xtreme Security offers security services they hold confidential data which make them very paranoid about security breaches where third party might access confidential data that is being exchange among users. Company want an Enterprise chat System that supports text messages of length (up to 4 kb) and file sharing possibility.

**4.Proposed Solution:**

Customers are provided with a Chat Framework so that there is a safe and secure communication between the organization representatives. Chat system is provided with a GUI to edit and read messages to configure the user account, add / remove / browse users in the local address book. Certificates must be used for each user as well as for the central server hosting database. The application is enabled with encryption features both an authentication and chat messages will use this feature. All interactions are user to user and user to server be based on RESTful API interface and with JSON data encoding.



**5.Limitations:**

* In our project, we have excluded profile pictures and some extra things which are beyond requirements, moreover they may require additional labour to create databases and integration tasks.
* Multimedia files like Video & Audio may not be supported in this framework.

**6.Time Plan:**

**Milestones:**

M1: Project Specification deadline : 23-04-2017

* A detailed description of the project will be presented so that they can develop the project according to the specifications given.

M2: SRS document deadline : 30-04-2017

* In software requirements specifications a detailed description of all the softwares used for both frontend; backend and database development

M3: Design document deadline : 07-05-2017

* Gives the brief description of design of the project

M4: Acceptance plan : 14-05-2017

* Gives the description of the testing phase for the project

M5: RC1 deadline : 21-05-2017

* At this stage the project is almost complete to deliver to the customer after the testing done in the acceptance test plan.

M6: RC2 deadline : 28-05-2017

* At this stage if there are any requirements leftover can be rectified and improved product can be delivered to the customer after the product demo.

Product Demo : 07-11-2017

* At this stage the completed project should be held for a demo to the CEO before delivering it to the customer.

Product release : 11-06-2017

* This is the stage where the final product will be delivered to the customer .

Time plan which is designed is shown below using project libre.

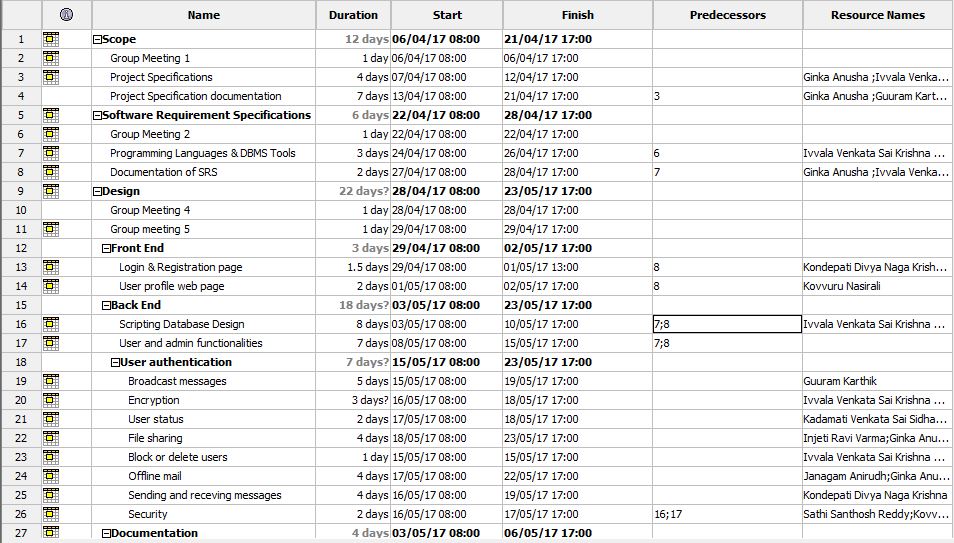
 

Figure 1. Time Plan

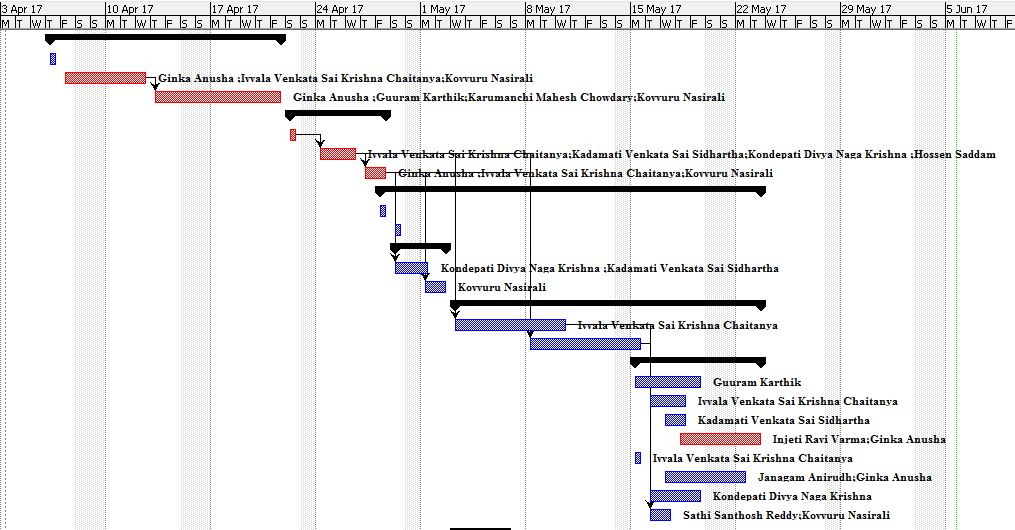


Figure 2.1.Gantt model in Project Libre

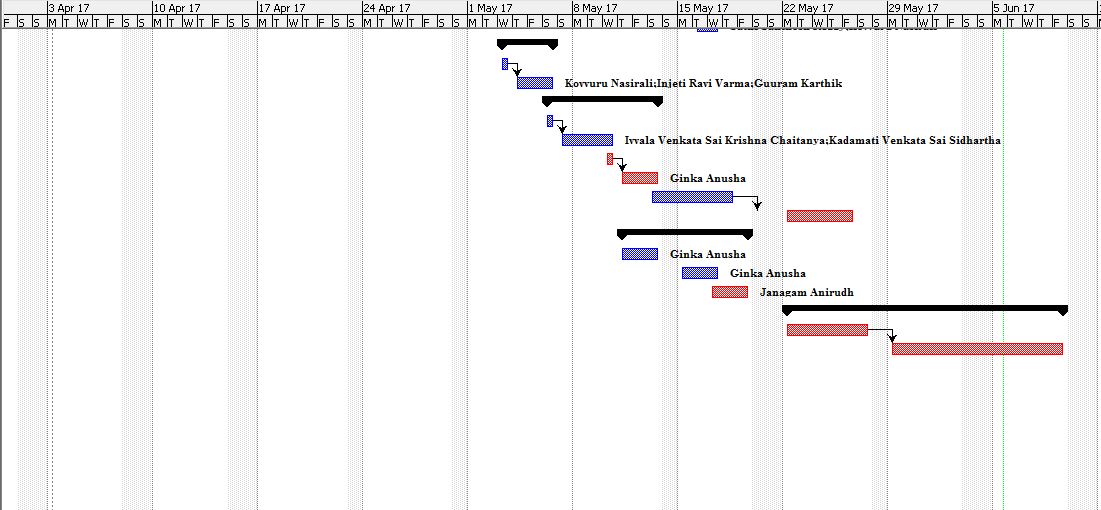


Figure 2.2. Gantt model in Project Libre

**7.Project Organization**

Agile Waterfall Model is chosen to our project. The reason why we chose this model is:

* Continuous feedback of our work.
* More flexible to changes in the works.
* Team motivation can be done.
* Very transparent when tracing progress
* Delays can be monitored and can take necessary steps to tackle those challenges.

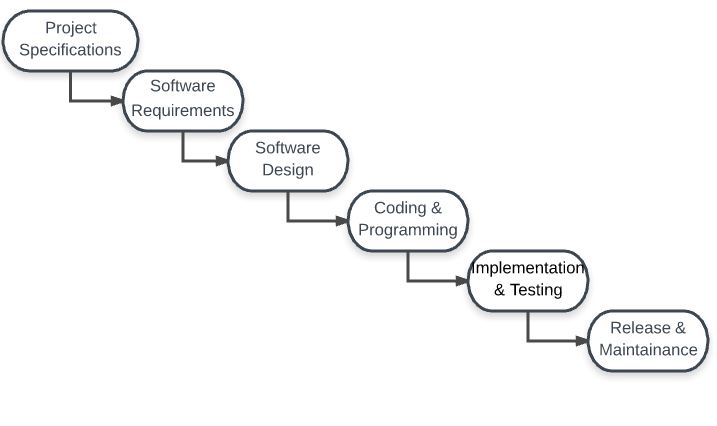


Figure 4. Waterfall model

|  |  |  |
| --- | --- | --- |
| **WORK** | **TEAM MEMBER** |  |
|  |  |  |
| **1. Programming:** |  |  |
|  |  |  |
| **A. Frontend:** | **Kondepati, Divya Naga Krishna** |  |
| **Kovvuru, Nasirali** |  |
|  |  |
|  | **Kadamati, Venkata Sai Sidhardha**  **Karumanchi Mahesh Chowdary**  **Janagam Anirudh** |  |
| **B. Database:** | **Ivvala Venkata Sai Krishna Chaitanya,** |  |
|  | **Hossen, Saddam**  **Sathi Santhosh Reddy** |  |
|  | **Ivvala Venkata Sai Krishna Chaitanya,** |  |
|  | **Hossen, Saddam** |  |
| **C. Backend:** | **Ginka Anusha** |  |
| **Kadamati Venkata Sai Sidhartha** |  |
|  | **Injeti, Ravi Varma** |  |
|  | **Kondepati, Divya Naga Krishna** |  |
|  | **Gurram Karthik** |  |
|  |  |  |
| **2.Management:** |  |  |
|  | **Ivvala Venkata Sai Krishna Chaitanya,** |  |
|  | **Sathi Santosh Reddy** |  |
|  |  |  |
| **3.Testing:** |  |  |
|  | **Ivvala Venkata Sai Krishna Chaitanya,** |  |
|  | **Injeti Ravi Varma** |  |
|  | **Gurram Karthik** |  |
|  |  |  |
|  |  |  |
| **4.Documentation:** | **Ginka, Anusha** |  |
|  | **Gurram, Karthik** |  |
|  | **Kovvuru, Nasirali** |  |
|  | **Karumanchi,Mahesh Chowdary** |  |
|  | **Anirudh ,Janagam** |  |
|  |  |  |

**8.CONFIGURATION MANAGEMENT:**

**Version Management:**

Version management is the process of keeping track of different version of software components or configuration items and the system in which these components are used. It also involves ensuring that changes made by different developers to these versions do not interfere with each other. Version management systems are used to identify, store and change history according to the development process. The initial version is taken as version 1.0 (For Trunk or branch) followed by 1.1,1.2,1.3……..etc. If it is a branch, then Integration part will be done at last subversion of that branch and this code is pushes into trunk.Gitlab server is used as a part of development to store the versions. With the help of Gitlab server users can merge their codes to create a new version of component. Git will monitor the code history and which has composed with portion of code.

**System Building:**

System building is the process of creating a complete, executable system by compiling and linking the system components, external libraries, configuration files. Creating a executable system involves many processes under it, such as development system which includes development tools such as compliers, source code editors. Developers checkout the code from the version management system into a private workspace before making changes to the system. Developers check-in code to the version management system before it is built. The system build may rely on external libraries that are not included in the version management system

**Release Management:**

System release is a version of software system that is distributed to the customers. It consists of minor release which repairs bugs and fix customer problems that have been reported and major releases which delivers a significant new functionality. System release is nothing but the codes which is used to execute the project, data files, documentation of design specifications and software requirements such that the project can be recreated in the future System release depend on the factors like technical quality, changes in the platform, etc. **Tagging** is done with **git tag**, and the **tags** that are created using **git tag** are the base for the commit identifiers **git** describe creates. In another words, in **Git** we don't **tag** branches. We are **tagging** commits.

**9.Progress Tracking:**

To guarantee on time delivery of the item, we set up a course of events that need to be planned well through the tool by name project libre to ensure that our item is prepared inside the stipulated time. Everyone in the team is required to lookup the time plan and speak with the colleagues in charge of the undertaking tasks to be done. Project tracking can be done in various manners.

Projects manager should monitor each member of the team whether they completed the work which has been assigned to them. Each member of the project is assigned with issues which have to to be completed within a given particular time. When the member of the project completes the specific issues assigned to them in Gitlab, they can close the issue which is assigned to them. Alternate way is to have group meetings according to the time plan to discuss the work done by each individuals and problems faced if any. After each group meeting issues will be assigned to each team member so that they will work on the particular topic for the upcoming milestone/tollgate. When anyone needs help for the task given other team member who is working on the similar topic will help them. Status update is done by the management team so that everything is completed on time without any delay.

Every member will upload the work completed by them on to the Gitlab server. Program can be tracked by anyone who want to know the status of the project with the help of issue management in Gitlab from the following link <https://herkules.comproj.bth.se:6017/G2/P1>

**10.Quality Control:**

Quality control is inspection – driven. Quality Control requires project manager and project team to inspect the work that's been done to determine if the work results are in alignment with the stated and implemented Quality Control. To meet the expectations of the client/customer we will maintain all the standards required using testing – phase. We will also test few codes in a step by step manner and also test the final code after integrating all of them. We will also release different versions of our product. This helps us to identify any bugs/errors in our product software. Our aim will be to deliver an error free software to the client/customer. We will also use encryption for our product. Finally, we will test our product in different versions of operating systems to deliver an error free software. To maintain the required level of quality product for the customer, the testing engineers of quality control team performs the testing and they will report the bugs to the developers. Unit testing is done at developer level after integration of the code, testing group will do overall testing and they correct the code and make sure the verification and validation is done. The next step is converting the tested source code into an executable file and testing that file for bugs, after fixing that bugs our product is ready to deliver. Then developers provide document about their tests and their results in acceptance test plan document. This process ensures that our code will not have any bugs and got validated.

**11. Risk Management:**

* In the following table, we have divided Risks into 4 major categories and we have approximated their probability of occurrence, their level of Effect and strategy to tackle those challenges.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.NO** | **Type of Risk** | **Probability** | **Level of Effect** | **Strategy** |
| 1. | Underestimating  Development Time. | High | Serious | Serious Planning regarding Communication between development team members and predicting Do’s and Don’ts before starting coding part |
| 2. | Team member’s  unavailability | Moderate | Serious | By using divide and conquer approach, each work is divided into two or more members. Such if one person is unavailable another person can tackle the problem. |
| 3. | Crashing of Software or Hardware components | Moderate | Serious | Keep track of plugin’s we have used and perfect using of cloud infrastructure for sharing the code. |
| 4.  5.  6.  7. | Dependencies  May run out of time for deliverable documents  Testing  No Encryption | Moderate    Low  High    Medium | Serious    Low  Serious    Serious | There may be multiple preceding tasks and succeeding tasks.The main relationship of the dependencies approach will be a Finish to start approach If 1 and 2 are two tasks, task 2 cannot be started before the completion of task 1.    Work breakdown is used to deliver the documents on time.  The high number of tests may not be done manually so the better approach will be to utilise through automated phase.    Suitable libraries needed to be utilised for setting up encryption and also for a better secure communication. |

**12.System Release Plan:**

## 12.1 Test Planning

* Mainly testing is used to know the presence of errors in the code. There are different tools in the market to Accomplish this task.
* Test planning is nothing but scheduling and resourcing all the activities in our testing process.
* Mainly testing is done in 3 stages in a project; they are
  1. **Testing in Development phase:**

Elementary testing is done by the development team. In this phase, they concern about the most predominant things in testing like component interface testing and Integration of components.**This testing is done from 12th may to 20th may 2017.**

* 1. **Release testing:**

A separate testing team tests the complete source code of the system before it is released to users. In this phase, this individual makes sure that the system meets its requirements and is good enough for external use. **This testing is done from 24th may to 30th may 2017.**

* 1. **User testing**

This is the exact place where they can judge their system. Here users test the system in their own environment.

Most of the MNC’s knows their actual result from this phase only. The company like Skype, TransferWise etc. manages their call centres very efficiently (Meanwhile they Appoint their testing crew members and knows the customer feedback about performance and validate their system). **This testing is done on 30th may to 6th may 2017.**

## 12.2 Packaging plan

A compressed zip file is given to the customer containing the code, Management files, library files, executable files,readme.txt.How to set up and install the product will be explained in the readme.txt file.Changes in software version and any changes in the software will be explained in the version.txt file.We are planning to start the packaging process from 19-05-2017 to 01-06-2017

**12.3** **Documentation Plan:**

### 12.3.1 Installation Documentation

This document is provided in PDF file format. It consists of a step-by-step procedure to install and configure the application and its components.

Installation document will be done from 11-05-2017 to 13-05-2017[3days].

### 12.3.2 User Documentation

This document is provided in PDF file format. It contains Do’s and Don’ts of the customer, Troubleshooting, and how to use the system in a more efficient way.

User documentation will be done from 13-05-2017 to 15-05-2017

### 12.3.3 Developer Documentation

It contains all technical aspects regarding system (From designing to Implementation). What type of algorithms we have used, Database entries, apps and encryptions used in the project.

Developer documentation will be done from 15-05-2017 to 19-05-2017

The documentation takes us a total of 7 days, as the last part of the documentation has dependencies over the first 2 parts.

**13.References**

1. Software Documentation, <https://en.wikipedia.org/wiki/Softwaredocumentation>
2. Ian Sommerville. Software Engineering. 9th Ed.