Enterprise Chat System

Team Name: **Proj. spec. v1.3**

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

As a part of this project, we have to develop an Enterprise chat system. It is expected to help the employees of Xtreme security to communicate safely. A central database is developed which stores the metadata of the sent and received messages for the users to browse chat history easily. The users can also track if the message which they have sent has been seen by the receiver or not including the time. The user can set up his status (Available or busy) based on their interest. Furthermore, for security purposes users have to create a username and password to authenticate themselves. The data given by the users while creating their usernames is stored in central database. Finally, this product is developed to be a secured chat system and secures confidential data against 3rd party security breaches.

Release v1.2 2016-o5-22

- Changes made in section 8
- Changes made in section 9 detailed division of sub tasks is made and is allocated to each member of the team
- Changes made in section 10 about quality control measures implanted during the system build.
- Changes made in section 12 dates are included in the test plan.

Release v1.1 on 2016-04-15

- Changes made to specify type of texts, i.e. text messages and not text files.
- Milestones made more specific and updated in ProjectLibre (screenshots placed) in Section 6, Time plan.
- Section 7 states the type of software process model used is specified
- Provided solutions for probable and eminent risks that can be faced in Section 11.
- Realistic dates provided for documentation and testing in Section 12.1 and 12.3.

Release v1.0 on 2016-04-17

- Initial release

The remainder of the document is organized as follows: Section 2 describes the Glossary and Abbreviations which includes technical words used in later part of document. Section 3 includes Background the need to develop the project and company's necessity. Section 4 is about how this system is developed and meet the company requirements. Section 5 is Limitations of the application which will be developed. Section 6 refers to the Time plan of the group, including the toll gates in order to obtain the result in the stipulated time. Section 7 is Project Organization. It says about how the project is being organized between the group members, and gives responsibilities each member in the team. Section 8 is Configuration Management. Differences between the newly developed version and already existing versions is described in this section. Section 9 is about tracking the progress of the group. It checks how close we are to achieve the final product. Section 10 is about how we ensure the users or customers that we have developed a quality product. Section 11 is Risk Management. Section 12 is System Release Plan. It shows how we intend to release the product. There are 3 categories in System Release Plan which ensures that the quality product is being released properly. They are: Testing Plan, Packaging Plan, Documentation Plan. Sections 13 adds References.

2. Glossary and Abbreviations

- **IP Address**: It is known as Internet Protocol Address. It is a unique number assigned to each system which are connected in a Network.
- **SQL**: Standard Queuing Language. It is a special purpose language which is used to manage related data.
- **GUI:** Graphical User Interface. It enables the user to interact with the system through visual indicators.
- **Available**: It indicates the state where the user is connected and ready to begin a conversation
- **Busy**: It shows that the user is connected but is not presently in a situation to chat
- **Offline**: It means that the user is not connected to the network.
- **Idle**: It indicates that the user is connected to the network but there is no work going on.

3. Background

The company Xtreme Security has large number of offices in many countries. The company offers security services to a large number of military and governmental organizations. They often hold the confidential information from their customers. So they do not trust the usage of conventional chat tools for their employees and sales men as the third party may access the confidential data that is being exchanged among their employees. The company wants an enterprise chat system that supports the exchange of text messages of a maximum of 4 KB and binary files of arbitrary length.

4. Proposed Solution

The customer is provided with a chat application in order to maintain safe and secure communication between the company employees. The chat system is provided with friendly graphical user interface (GUI) to edit and read messages, to configure the user account, add/remove/browse users in the local address book. The application is enabled with encryption features both on authentication and chat message. Communication between user-to-user and user-to-server is based on RESTful API interface and with JSON data encoding.

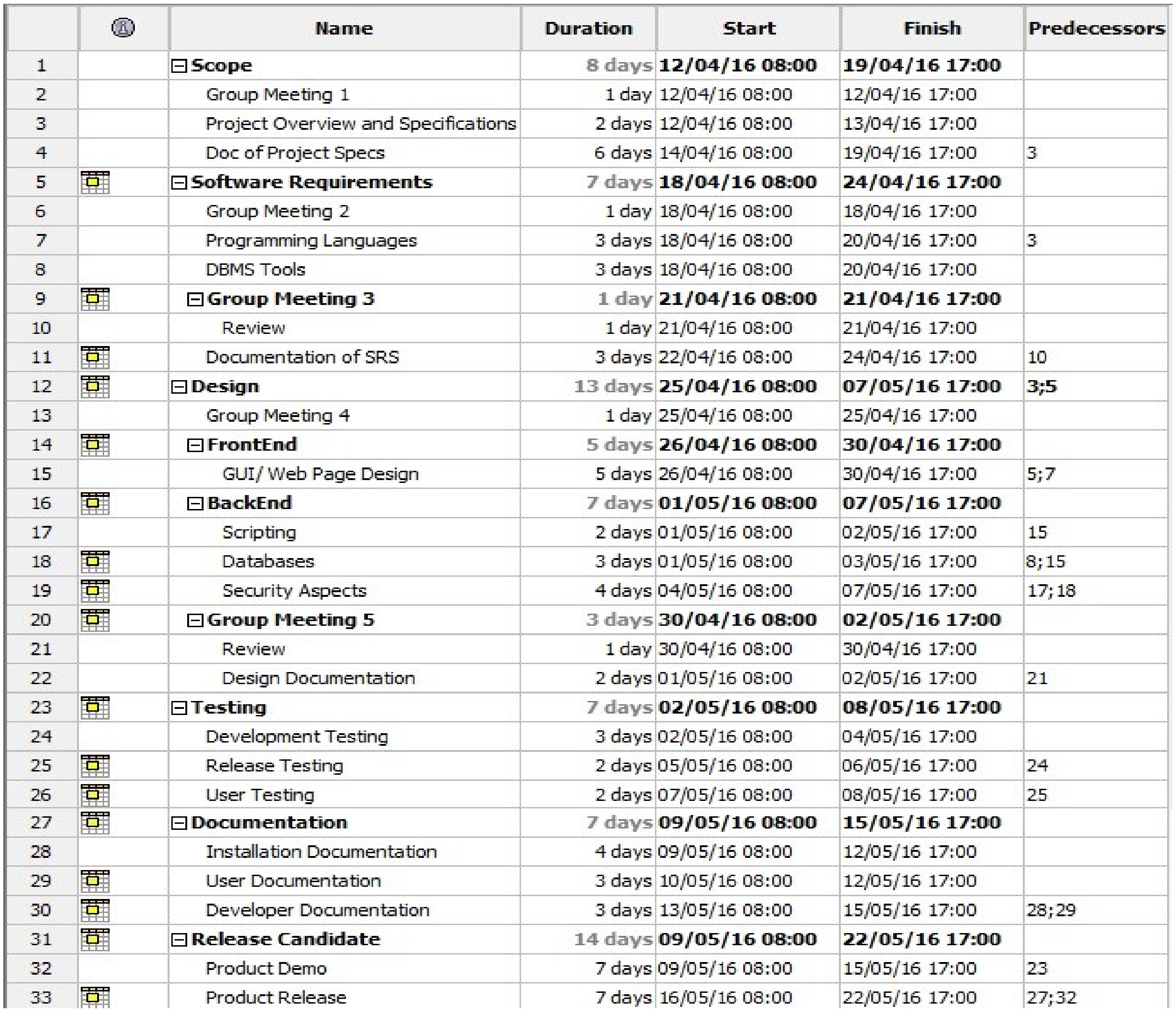
5. Limitations

- Text messages of size 4KB can be transmitted at a time.
- Multimedia files like audio and video are not supported.

- Compatibility with different browsers.
- Compatibility with system OS(Windows, Linux, Mac OS X)
- Project demonstration is on 3 peer users.

6. Time Plan

i) 11-04-2016: Project proposal
ii) 17-04-2016: Project specification
iii) 24-04-2016: Software Requirement Specifications
iv) 01-05-2016: Design document
v) 08-05-2016: Acceptance test plan
vi) 15-05-2016: Project Demo
vii) 22-05-2016: Project Submission and Documentation



	ⓘ	Name	Duration	Start	Finish	Predecessors
1		⊟ **Scope**	8 days	**12/04/16 08:00**	**19/04/16 17:00**	
2		Group Meeting 1	1 day	12/04/16 08:00	12/04/16 17:00	
3		Project Overview and Specifications	2 days	12/04/16 08:00	13/04/16 17:00	
4		Doc of Project Specs	6 days	14/04/16 08:00	19/04/16 17:00	3
5	▦	⊟ **Software Requirements**	7 days	**18/04/16 08:00**	**24/04/16 17:00**	
6		Group Meeting 2	1 day	18/04/16 08:00	18/04/16 17:00	
7		Programming Languages	3 days	18/04/16 08:00	20/04/16 17:00	3
8		DBMS Tools	3 days	18/04/16 08:00	20/04/16 17:00	
9	▦	⊟ **Group Meeting 3**	1 day	**21/04/16 08:00**	**21/04/16 17:00**	
10		Review	1 day	21/04/16 08:00	21/04/16 17:00	
11	▦	Documentation of SRS	3 days	22/04/16 08:00	24/04/16 17:00	10
12	▦	⊟ **Design**	13 days	**25/04/16 08:00**	**07/05/16 17:00**	**3;5**
13		Group Meeting 4	1 day	25/04/16 08:00	25/04/16 17:00	
14	▦	⊟ **FrontEnd**	5 days	**26/04/16 08:00**	**30/04/16 17:00**	
15		GUI/ Web Page Design	5 days	26/04/16 08:00	30/04/16 17:00	5;7
16	▦	⊟ **BackEnd**	7 days	**01/05/16 08:00**	**07/05/16 17:00**	
17		Scripting	2 days	01/05/16 08:00	02/05/16 17:00	15
18	▦	Databases	3 days	01/05/16 08:00	03/05/16 17:00	8;15
19	▦	Security Aspects	4 days	04/05/16 08:00	07/05/16 17:00	17;18
20	▦	⊟ **Group Meeting 5**	3 days	**30/04/16 08:00**	**02/05/16 17:00**	
21		Review	1 day	30/04/16 08:00	30/04/16 17:00	
22		Design Documentation	2 days	01/05/16 08:00	02/05/16 17:00	21
23	▦	⊟ **Testing**	7 days	**02/05/16 08:00**	**08/05/16 17:00**	
24		Development Testing	3 days	02/05/16 08:00	04/05/16 17:00	
25	▦	Release Testing	2 days	05/05/16 08:00	06/05/16 17:00	24
26	▦	User Testing	2 days	07/05/16 08:00	08/05/16 17:00	25
27	▦	⊟ **Documentation**	7 days	**09/05/16 08:00**	**15/05/16 17:00**	
28		Installation Documentation	4 days	09/05/16 08:00	12/05/16 17:00	
29	▦	User Documentation	3 days	10/05/16 08:00	12/05/16 17:00	
30	▦	Developer Documentation	3 days	13/05/16 08:00	15/05/16 17:00	28;29
31	▦	⊟ **Release Candidate**	14 days	**09/05/16 08:00**	**22/05/16 17:00**	
32		Product Demo	7 days	09/05/16 08:00	15/05/16 17:00	23
33	▦	Product Release	7 days	16/05/16 08:00	22/05/16 17:00	27;32

Figure 1. Time Plan

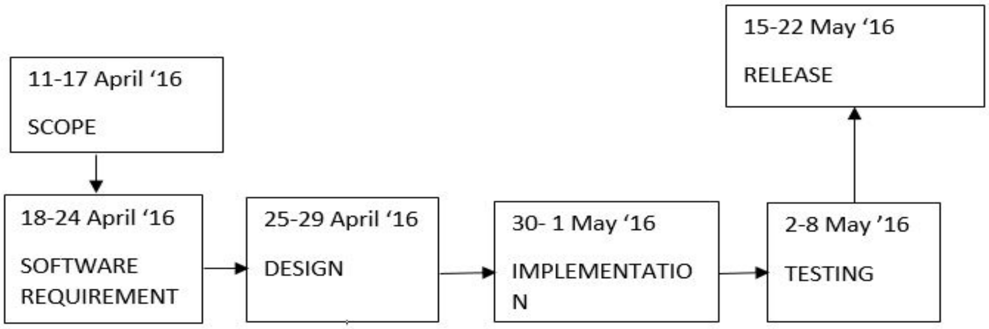


Figure 2. System Architecture

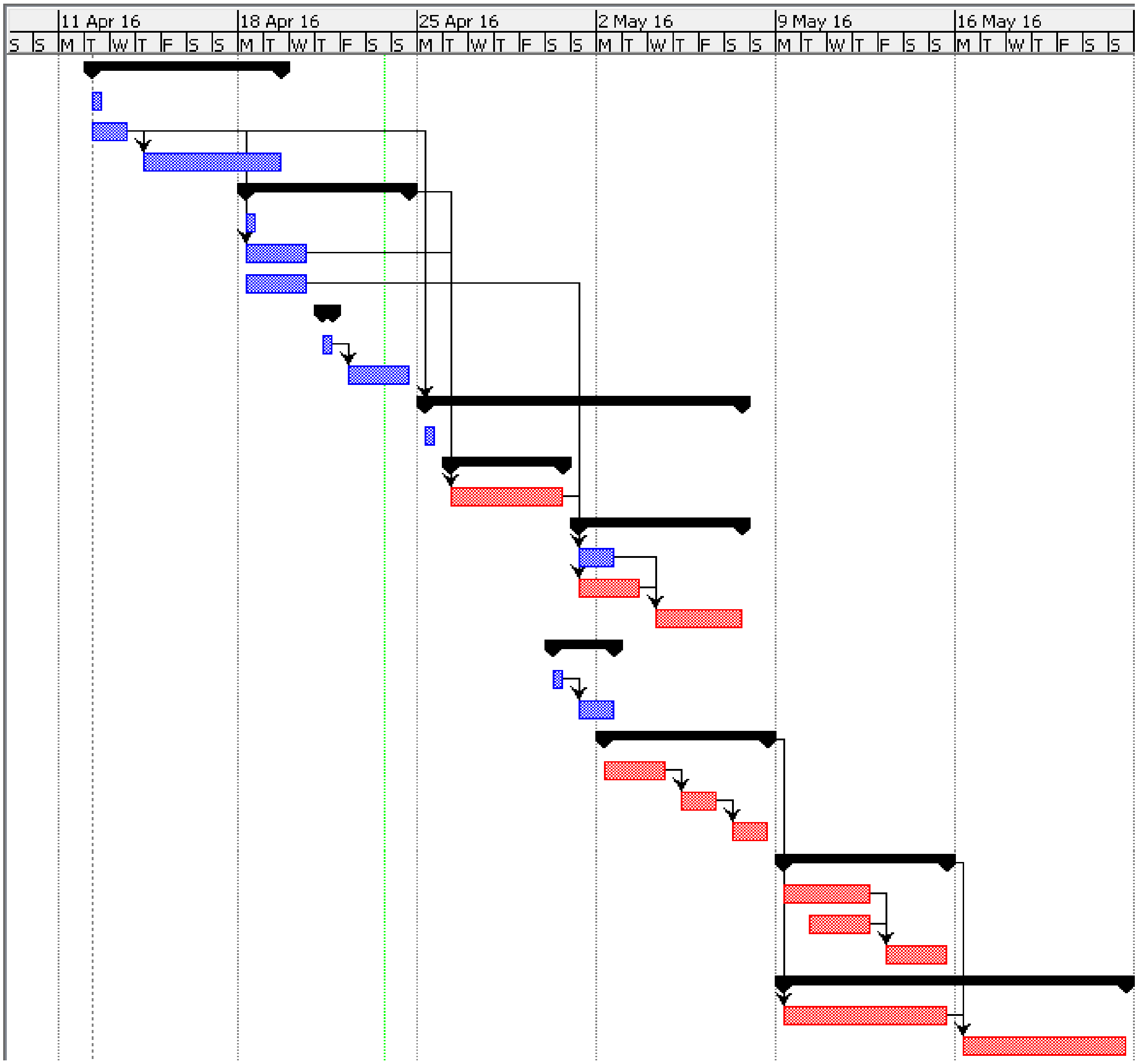


Figure 3. Gantt Structure in Project Libre

7. Project Organization

To organize and plan for a project, we need to make sure what type of process we will be using to apply and implement software present at hand to meet the requirements of the customer. To do that, the activities are described so that a specific data model can be referred to while performing these tasks. In that manner, the type of software process model we have used is a waterfall model. This model's separate and distinct phases of specifications and development provide an ambitiously plan-driven model which was most suitable to complete this project.

Various steps are defined so that the project can be completed in a specific manner. Although this model is highly successful, the main drawback is that each step in it is dependent on its predecessor. This may not be a problem unless and until change has to be accommodated when the project is underway.

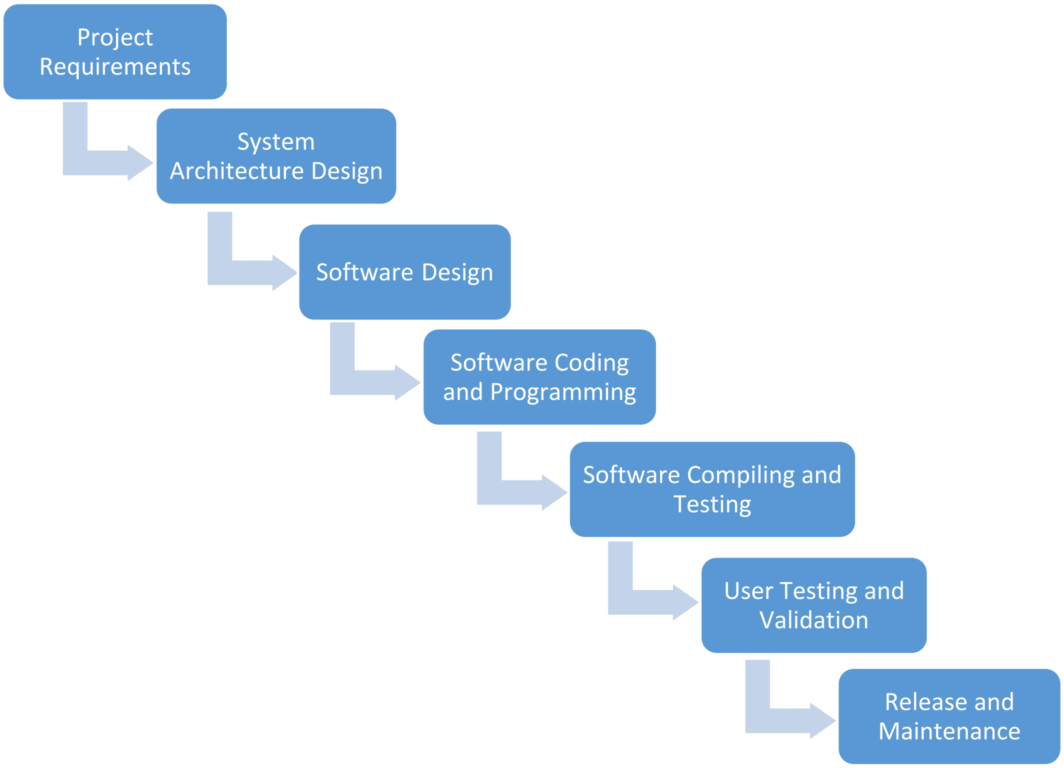

Project Requirements
System Architecture Design
Software Design
Software Coding and Programming
Software Compiling and Testing
User Testing and Validation
Release and Maintenance

Figure 4. Waterfall Model

WORK	TEAM MEMBERS

1. Programming:	
a. Front-end:	Mohammed Zaahid, Mounica Mamidi, Neelam Sankeerthana, Byreddy Sreenibha. Yerrapragada Sita Rama Suryateja
b. Database:	Kalaparti Adithya, Sanagari Vivek, Balusu Geethanjali.
c. Back-end:	Mara Nikhil, Srinivasa Srikrishna, Shiva Tarun
2. Testing:	Sanagari Vivek, Budda Shiva Tarun.
3. Management:	Neelam Sankeerthana, Byreddy Sreenibha.
4. Documentation:	Mohammed Zaahid, Kalaparti Adithya.

8. Configuration management

To successfully manage and release a product on time requires three major steps to be carried out. They are:

Version management

The concepts of version management are used to identify, store and access the different versions of the software components. Changes made to those versions by different developers doesn't interfere to each other. These modifications are saved as two separate versions called blobs The Git Lab server created is used to maintain the versions in stable state. It may be necessary to merge code line branches to create a new version of the component, where different developers work independently on different versions of the source code.

System building

The product enterprise system is mostly build using html and PHP scripting. The front end part of the system build using html along with java script .The styling of web base at GUI is done through CSS the back end of the system is build using PHP scripting. The interaction between one web page to another is done through restful API'S with JSON encoding .the communication between the users will be encrypted format frontend and backend are connected to Mysql data base which consists of a single data base with 3 tables . 3 tables include credentials which stores user login details, info base which stores chat messages between users and the associated meta data. Contacts of each user are stored in a common table named contacts. User can dynamically search for contacts in Mysql data base and add to his contacts, this is done by including Ajax library. The entire activity of users is governed by administrator. Administrator part of the system is also done by PHP scripting. The modules are merged in a proper manner to build the complete system with all the specified functionalities.

Release management

Version of a software system distributed to customers are possibly minor release, which are distributed free of charge, and major release, which are very important economically to the software vendor as customers have to pay for these. System releases include executable code, data files, configuration files, and documentation to ensure that it can be re-created in the future. System release depend on the factors like technical quality, changes in the platform, marketing requirements, etc.

9. Progress Tracking

All the tasks required to be completed in the project are divided in subtasks and these tasks are assigned to each person in the group. These subtasks are merged to develop a particular functionality of the system. Tracking is done in different manners, on a personal level by the project manager to make sure each member finishes the work given. Another way of tracking the process is by having group meetings at a minimum of twice a week to discuss problems faced and progress completed. Finally, each member will be uploading a summary of the work completed by oneself, at the GitLab server by issuing tracking and continuous monitoring by the project manager. The progress can be tracked by anybody who wants to through this html: https://herkules.comproj.bth.se/groups/INVICTUS

10. Quality Control

Quality testing involves breaking entire system into small units, each small unit undergoes several test during each test bugs (if any) are fixed . These small units are merged to form a particular component of the system again several test are performed on each individual component to guaranty the proper functionality. These components are finally merged to build the complete system with all the functionalities that satisfies the customer requirements. The system is again tested to ensure proper functioning of all the components without any errors.

11. Risk Management

- The following are some common type of risks with their likelihood, effects and strategies.

Type of Risk	Probability	Level of Effect	Strategy
Unrealistic schedule/ Underestimating development time.	High	Serious	Detailed schedule planning based on number of members and their capability to deliver in a software development team.
Key team members unavailable.	Moderate	Serious	Cross –training/work sharing by proper communication/relationship among team members.
Crashing of software or hardware components.	Moderate	Serious	Test software and hardware components required and repair defects /reinstall before use if any problems.
Dependencies of various processes on predecessors.	Moderate	Serious	Progress tracking done at high scrutiny level to ensure members deliver in given time.

- Risk management deals with the prediction of risks that affect
    1. The project schedule
    2. Quality of software product developed.

- It also deals with the strategies to manage risks. Some strategies can be used to avoid, some are minimizing and some are for worst scenarios called contingency plans.

12. System Release Plan

12.1 Test Planning

The Test planning is concerned with scheduling and resourcing all of the activities in testing process. The test plan may also include details of the tests to be run on the software.

A commercial software system has to go through three stages

1. Development testing: This testing is done by System designers and Programmers. It involves in discover bugs and related defects in the software. During development, testing is carried out in three stages.

- Unit testing: It involves testing the functionality of objects or methods. This is done on 2nd may 2016.

- Component testing: It is concerned with testing of component interfaces. This is done on 3rd may 2016.
- System testing: It focuses on testing component interactions. This is done on 4th may 2016.

2. Release testing: A separate testing team tests a complete version of system before it is released to users. The testing team intends to discover bugs in the system. It makes sure that the system meets its requirements and is good enough for external use. This done on 5tmay2016.

3. User testing: Users test the system in their own environment. This is done on 7th may and 8th may 2016.s

12.2 Packaging Plan

A compressed file of .tar.gz archive is given to the customer containing the computer code, alpha code, supervisory and data base management files, library files and related documents. The time schedule details of packaging are as follows:

Release candidate:	2016/05/15
Product Release:	2016/05/22

12.3 Documentation Plan

12.3.1 Installation Documentation

The installation documentation is provided in PDF file format. It will contain step-by-step information regarding applications, set-up and installation and configuration of the software and its components. Installation documentation will be done from 9th May to 12th May 2016.

12.3.2 User Documentation

The user documentation is provided in PDF file format. It will contain information regarding functionality of the system, different modules and their linking, how to use the tool, limitations and troubleshooting. User documentation will be done from 10th May to 12th May 2016.

12.3.3 Developer Documentation

This document provides scope for development of our chat system. This document will contain technical aspects i.e., source code and few related algorithms, description of entry functions, Database entries, API's and encryptions used in the project. Developer documentation will be done from 13th May to 15th May 2016.

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] Ian Sommerville. Software Engineering. 9th Ed.

[2] Software Documentation, https://en.wikipedia.org/wiki/Software_documentation