INTERNET WHITEBOARD

PROJECT SPECIFICATION DOCUMENT

Technocrats

Team Members:

1. Shaik, Adil
2. Tamanampudi, Monica
3. Tammana, Naga Venkata Satya Sai Manoj
4. Tammana, Sai Surya Akhilesh
5. Tanyi, Elvis
6. Ummadi Setty, Yogitha Manasa
7. Valirad, Sina
8. Viswanadhuni, Giri Sai
9. Vuyyuru, Kaushik Reddy
10. Yalavarthi, Sreelekha

1. Preface

- Version 1.3 release

Following changes are made from version 1.2
In section 1 updated the version history of the document.
In section 3 brief description about the Restful API is added.
In section 5 the user's limitations is added.
In section 7 the people allotted to server maintenance is specified to each person.
In section 9 issue management system in GitLab to track process is added.

- Version 1.2 release.

Following changes are made from initial version,
In section 4 we differentiated the role of administrator and the admin server and inserted the overview of the system with simple diagram. A brief description about the role of moderator. Moreover, the encryption and RestfulAPI are added to the proposed solution.
In section 5 the limitations of the product are clearly specified.
In section 7 we elaborated the types of software process model we plan to use is described.
In section 8 we established the relation between the system building and release management.
In the section 9 description about the issue management is added.
In section 11 all the numbers in the table 3.0 are aligned to right side.

- Initial release

The primary aim of the project is to build the internet white board which basically works as a correspondence path between employees of ConTech and clients, it helps them to communicate and work together. The specification document below is organized into sections which are described below.

Section 2 describes glossary and abbreviations that are used in the document. Section 3 portrays client business and current environment. In Section 4 the role of an administrator and admin server and encryption, RestfulAPI is discussed. In section 5 limitation of the project are discussed. Section 6 describes time plan proposed and table names for tables are included. In Section 7 the software process model is discussed in detailed, difference between WBS segment and sheet management is shown. In section 8 the process which is implemented is explained and describes configuration management that includes version management, system building and release management. Section 9 describes progress tracking and issue management system in Gitlab to track progress. Section 10 describes quality control and testing is done. Section 11 contains risk management and it explains about the relevance of number of programming languages and exceeded and business level under execution risk. Section 12 describes System release plan which include testing plan, packaging and documentation plan and time plan.

2. Glossary and Abbreviations

1. **IP Address**: It is known as Internet Protocol Address. It is a unique number assigned to each system which are connected in a Network.

2. **SQL:** Standard Queuing Language. It is a special purpose language which is usedto manage related data.

3. **GUI:** Graphical User Interface. It enables the user to interact with the system through visual indicators.

4. **RESTful API:** Representational State Transfer (REST) is an architectural style that specifies constraints, such as the uniform interface, that if applied to a web service induce desirable properties, such as performance, scalability, and modifiability, that enable services to work best on the Web. API is Application Programmable Interface.

5. **PyMySQL:** PyMySQL is a database connector for Python programing language libraries and its used to enable Python programs to talk to a MySQL server [2]

6. **MySQLdB:** Its having same functionality as PyMySQL [2]

7. **Timestamp:** It is considered as a series of characters or encoded information that identify the occurrence of an event. Mostly expressed based on a calendar year.

8. **Python Tkinter:** Tkinter is one of the standard Pythons Graphical User Interface(GUI) package. [3]

9. **PHP:** Hyper Text Preprocessor is a server scripting programing language that is used for making dynamic and user interactive web based pages.

10. **FLASK**: It is an implementation of the web browsable **APIs** like Django REST framework. It gives proper content negotiated responses. It also provides smart request parsing. we can start building kick-ass web browsable APIs using FLASK .

3. Background

Our client is ConTech, a company that is expanding internationally. Managers of ConTech have decided to make measures to reduce costs, by suggesting that their employees provide services to their customers through remote locations. To make this possible, complete access to an Internet Whiteboard application is required by us from ConTech, so that they can enhance their communication with their customers.

A moderator (who will be the creator of the session) must be designated once communication is established with the customer by ConTech employee. ConTech requests that there be both user to user interaction along with user to server interaction both based on a RESTful API with JSON data encoding. The company also suggests that certificates be used for each user as well as for the central server hosting the database.

4. Proposed Solution

According to the company's requirements, we propose a solution of an internet White board online application which keeps track of all users and their events with security.

- A front end in the following application includes a Web page with all the options for respective type of users. There will be three types of users and those are Admin, Employee and users.
- Admin will be able to create an employee and a user, whereas employee will be able to create a user and start a white board session

- A white board session consists of three sheets and the running application and certain number of users, various tools to draw are provided in each sheet.

- All the events appearing in the sheets i.e.; the session will be saved in the database along with the user details.

- The main server will be handled by administrator and the other users or employees are connected by the administrator.

- All the modifications are saved in the admin server in the list provided and a playback is provided by having a delay between each modification.

- Extra functionalities will be provided to the moderator and those are Undo operation, a feature to lock and unlock the sessions. There will also be an option to change the moderator,

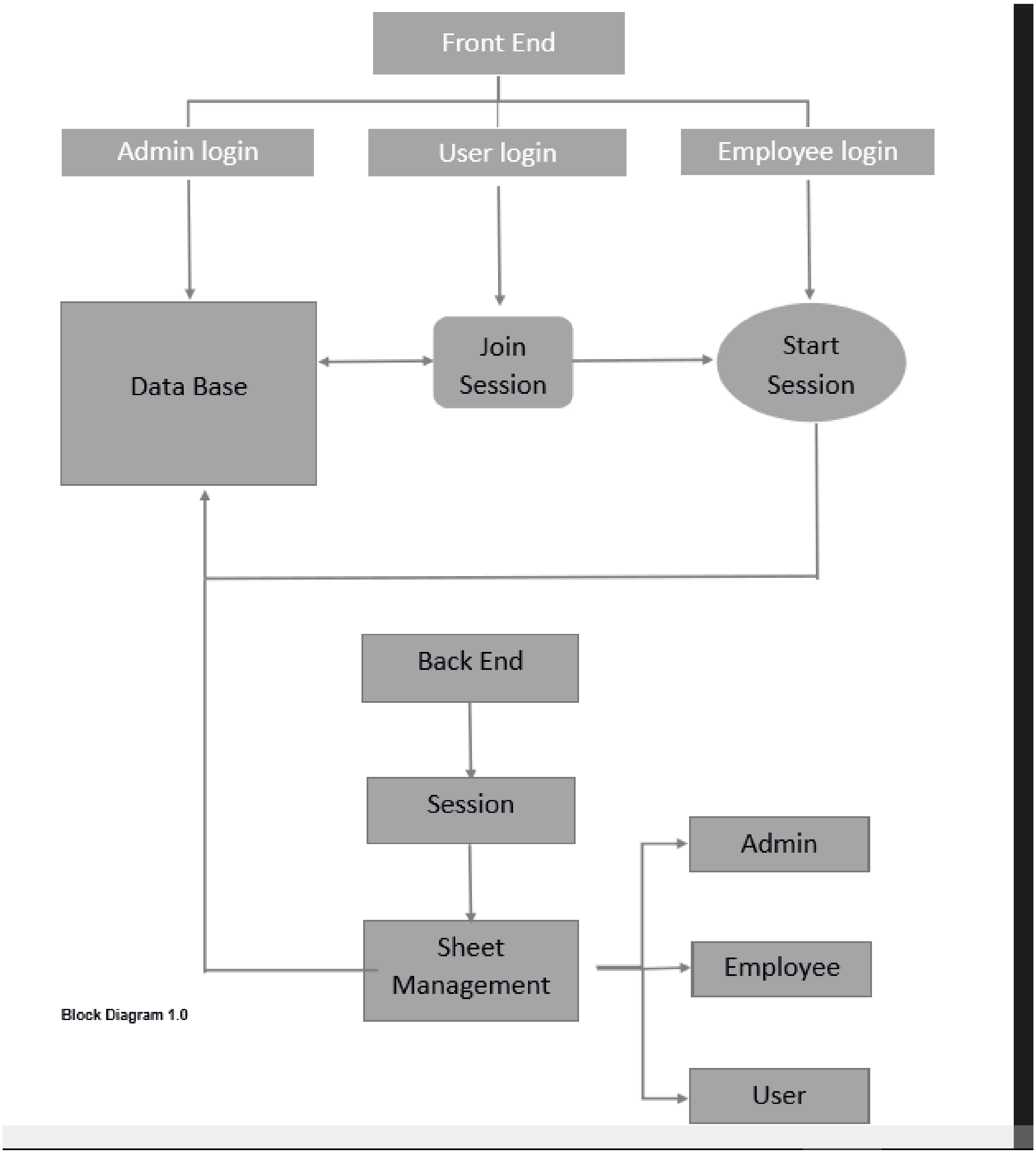
- A moderator by default is the creator of a session. The moderator can lock access to a sheet, sequentially undo modifications and change to a different sheet.

- A reload option is provided to the user, where a clear whiteboard reloads the modifications saved before.

- Finally, a whole application is made user friendly by saving all the modifications in the database of the admin server.

- All forms of communication are encrypted. Certificates are used for each user as well as for the central server hosting the database. Certificates are self-signed in the release to the customer.

- All interaction user-to-user and user-to-server are based on a RESTful API with JSON data encoding.



Block diagram 1.0

5. Limitations

Due to limited resource one computer is utilized as a server and other computers are connected to the server via Lan Switch or Wi-fi. Hence failure in the server may result in failure of whiteboard session. Moreover, a maximum of three users can access simultaneously.

Moreover, the application is limited to Ubuntu.

6. Time Plan

Milestones:

1. 03-04-2017: Project Proposal-A project request for an internet whiteboard application is proposed by Contech(customer).

2. 23-04-2017: Project Specifications- A detailed document relating to the specifications of the project is prepared and shared with the CEO and the development team.

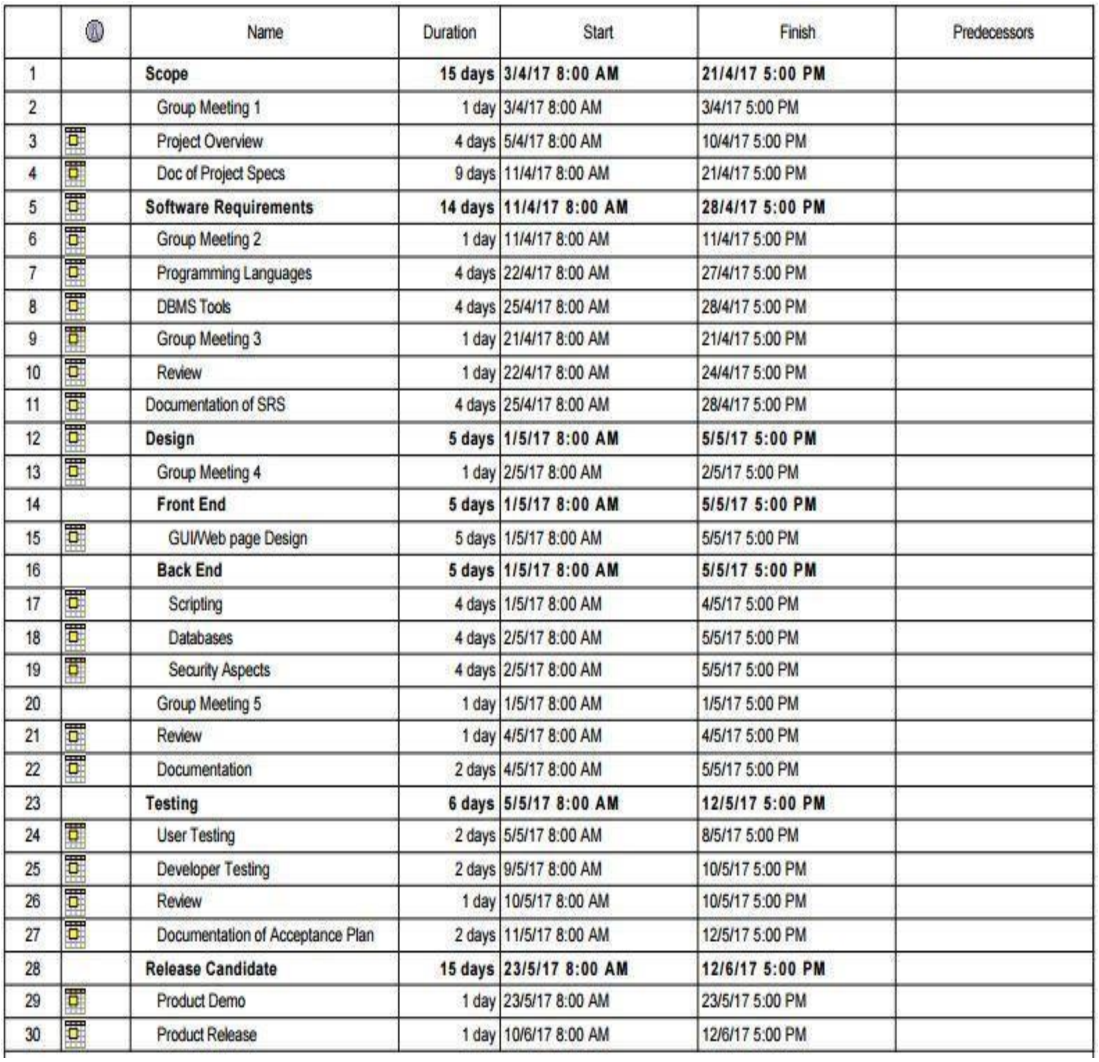
3. 30-04-2017: Software Requirement Specifications- Various softwares to be used to achieve at completion of the project are specified to the CEO and development team.

4. 07-05-2017: Design plan- A smooth and efficient design plan is laid down to the team of developers.

5. 14-05-2017: Acceptance Test Plan- An acceptance test plan is conducted by the CEO.

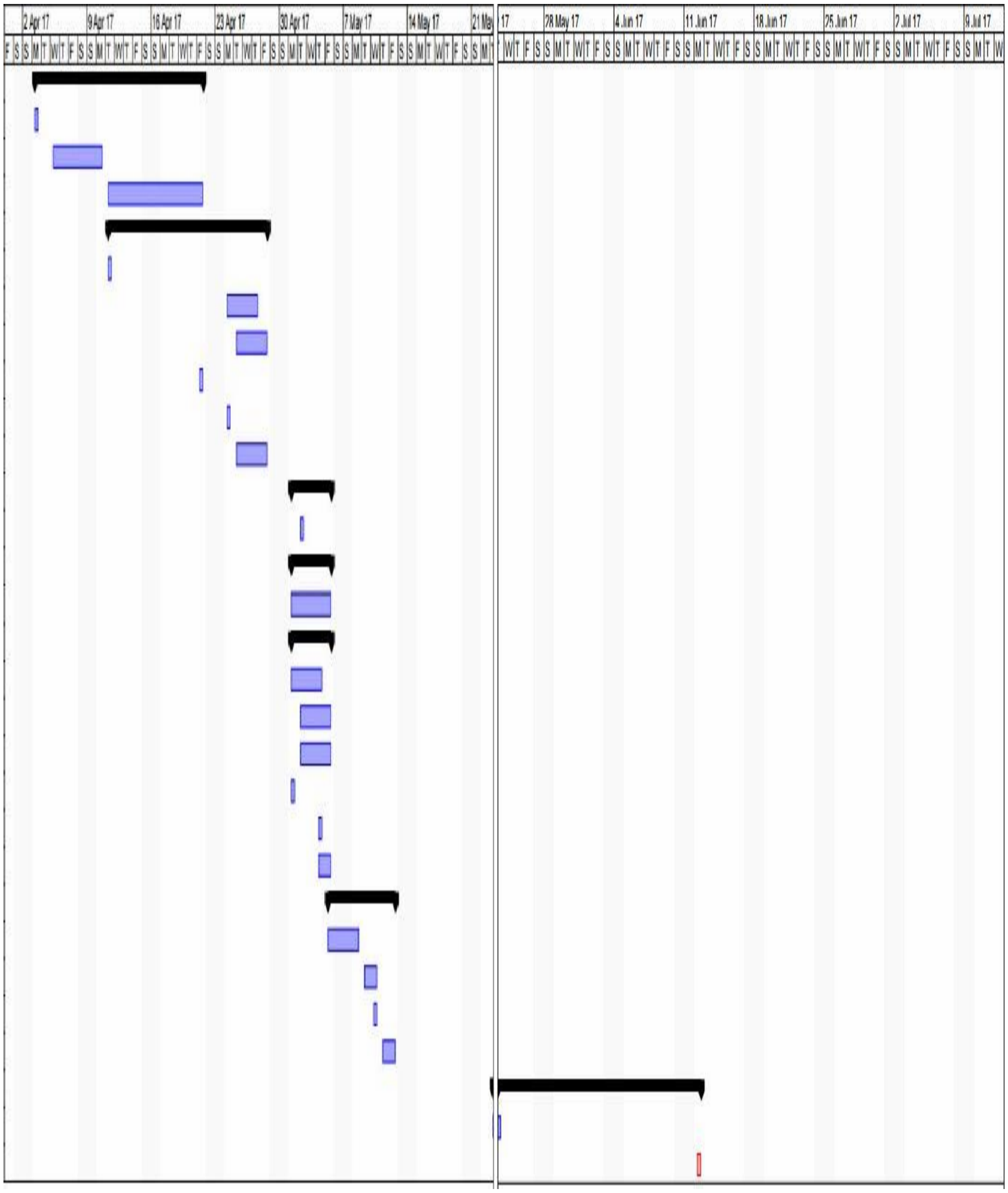
6. 22-05-2017: Project Demo- A sample project demo in its rudimentary form is conducted to the customer(contech) for approval and further improvement.

7. 11-06-2017: Project Submission and Documentation- The final version of the project is submitted along with the documentation.



	ⓘ	Name	Duration	Start	Finish	Predecessors
1		**Scope**	**15 days**	**3/4/17 8:00 AM**	**21/4/17 5:00 PM**	
2		Group Meeting 1	1 day	3/4/17 8:00 AM	3/4/17 5:00 PM	
3	📅	Project Overview	4 days	5/4/17 8:00 AM	10/4/17 5:00 PM	
4	📅	Doc of Project Specs	9 days	11/4/17 8:00 AM	21/4/17 5:00 PM	
5	📅	**Software Requirements**	**14 days**	**11/4/17 8:00 AM**	**28/4/17 5:00 PM**	
6	📅	Group Meeting 2	1 day	11/4/17 8:00 AM	11/4/17 5:00 PM	
7	📅	Programming Languages	4 days	22/4/17 8:00 AM	27/4/17 5:00 PM	
8	📅	DBMS Tools	4 days	25/4/17 8:00 AM	28/4/17 5:00 PM	
9	📅	Group Meeting 3	1 day	21/4/17 8:00 AM	21/4/17 5:00 PM	
10	📅	Review	1 day	22/4/17 8:00 AM	24/4/17 5:00 PM	
11	📅	Documentation of SRS	4 days	25/4/17 8:00 AM	28/4/17 5:00 PM	
12	📅	**Design**	**5 days**	**1/5/17 8:00 AM**	**5/5/17 5:00 PM**	
13	📅	Group Meeting 4	1 day	2/5/17 8:00 AM	2/5/17 5:00 PM	
14		**Front End**	**5 days**	**1/5/17 8:00 AM**	**5/5/17 5:00 PM**	
15	📅	GUI/Web page Design	5 days	1/5/17 8:00 AM	5/5/17 5:00 PM	
16		**Back End**	**5 days**	**1/5/17 8:00 AM**	**5/5/17 5:00 PM**	
17	📅	Scripting	4 days	1/5/17 8:00 AM	4/5/17 5:00 PM	
18	📅	Databases	4 days	2/5/17 8:00 AM	5/5/17 5:00 PM	
19	📅	Security Aspects	4 days	2/5/17 8:00 AM	5/5/17 5:00 PM	
20		Group Meeting 5	1 day	1/5/17 8:00 AM	1/5/17 5:00 PM	
21	📅	Review	1 day	4/5/17 8:00 AM	4/5/17 5:00 PM	
22	📅	Documentation	2 days	4/5/17 8:00 AM	5/5/17 5:00 PM	
23		**Testing**	**6 days**	**5/5/17 8:00 AM**	**12/5/17 5:00 PM**	
24	📅	User Testing	2 days	5/5/17 8:00 AM	8/5/17 5:00 PM	
25	📅	Developer Testing	2 days	9/5/17 8:00 AM	10/5/17 5:00 PM	
26	📅	Review	1 day	10/5/17 8:00 AM	10/5/17 5:00 PM	
27	📅	Documentation of Acceptance Plan	2 days	11/5/17 8:00 AM	12/5/17 5:00 PM	
28		**Release Candidate**	**15 days**	**23/5/17 8:00 AM**	**12/6/17 5:00 PM**	
29	📅	Product Demo	1 day	23/5/17 8:00 AM	23/5/17 5:00 PM	
30	📅	Product Release	1 day	10/6/17 8:00 AM	12/6/17 5:00 PM	

Time plan1.1



Gantt structure 1.2

7. Project Organization

The project organization for the Internet White Board has 9 WBS segments which are mentioned below, the work is divided based on WBS segments and members in the group are made responsible to one or multiple tasks. The alignment of members in the group to their respective tasks are listed as follows

1. **Generating frontend white board Structure:** This basically means to create a simple white board including registration and login pages for the Admin, User and Employee.

2. **Building the backend white board session:** In this segment, the created white board is developed and functions are added to the whiteboard

3. **Sheet Management control:** This deals with managing and modification of sheets, different access is determined to user, employee and admin respectively. Moderator is provided with lock access to the sheets so that to prevent editing wars.

4. **Server maintenance:** This segment deals with server linkup and ensuring every modification that has been made is regularly updated in the database.

5. **Security**: Security plays a prominent roll and it deals with the encryption of communication thus, one members in the group have been assigned to complete the task.

6. **Debugging:** This segment deals with identifying the errors and fixing them, this also included fixing errors that have been reported by the users

7. **Packaging:** The segment deals with the integration and packaging of different modules into the system.

8. **Testing:** Testing is expected to be wide and crucial, every member of the group are expected to perform the each task individually and the whole package is tested by the group members accordingly.

9. **Documentation:** This segment deals with all the documentations needed that is Installation documentation and user documentation.

WBS segments	Members
1. Generating simple white board (Front End)	**adil, Monica, Manoj,**
2. Building backend white board session (Back End)	**Manasa, Sina, Kaushik**
3. Sheet Management control	**Manasa, Sina, Giri**
4. Server Maintenance	**Adil, Akhilesh, Giri**
5. Security	**Manoj, Akhilesh, Monica**
6. Debugging	**Manasa, Sina, Akhilesh**
7. Packaging	**Manasa, Kaushik, Manoj**
8. Testing	**sreelekha, Elvis, Akhilesh**
9. Documentation	**Sreelekha, Elvis, Adil**

Table 1.3 Work management table

Software Process model: The software process model used in this project is spiral model which suites our project. This model consists of prototypes, we begin with determining the requirements of the project and then generate the simple white board then developing it with required tools and features that are mentioned by the client which comes under development section after implementing we further test every component involved in the project and deliver the product to the client.

8. Configuration Management

8.1 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 the 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 followed by 1.1,1.2,1.3 ……. etc. IF it is a branch, then Integration part will be done at last subversion of the branch and this code is pushed into trunk. GitLab server is used as a part of the development to store the versions.

8.2 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.

8.3 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.

9. Progress Tracking

- The whole project is distributed among the group members and regular internal meetings are made to track the progress and to make sure the product is delivered on time.
- Each time when internal meetings happen issues are discussed and make status update on open issues and concern member of the group is assigned to solve and close the issue with in certain time.
- Using GitLab any member in the group can open an issue, assign it to other members and when the task is done the issue can be closed.
- In addition to it the timeline has been made using Project Libre to ensure the work being done as planned.
- Each member in the group is provided with specific responsibilities and time plan, should act according to timeline assigned.

10. Quality Control

- To ensure smooth run of the application, each developer of the group is assigned with single element that is to be tested. Later interconnected modules of the product are tested by specific members of the group and the significance of this test is to ensure communication between the components is working fine.

- Encryption for user modules and data are secured in the data base, Authentication is required to create and modify the data.

- The test cases as mentioned in the acceptance test plan are tested periodically. Moreover, the final product package is checked as per the user requirements before delivering it to the customer.

- The bugs in the system are identified and fixed for every version release, the user report and feedbacks are accepted, and actions are conducted to fix the issues

11. Risk Management

Risk are future uncertain events with either a probability value of occurrence or having an effect. The issue of risk in software development project has been emphasized in many research papers and this makes it a prime concern and top priority in any software project. This is because lots of software's have been intended and developed but never used[1] [2]

In other for us to complete our project we identified a series of risk typed: that is project that that tends to affect our scheduling and resource to be used, product risks that will affect the quality and or performance of software to be developed and finally business risk that might affect our client. These three groups of risks typed identified was further reclassified into 10 subgroups comprising of; users, technology, team members, planning and control, financial, legal implications, complexity, scope and requirement, project estimation and execution and classified them into subgroups of specified below.

POSSIBLE RISK IDENTIFICATION AREAS IN OUR PROJECT		
RISK TYPE	**AFFECTED**	**POSSIBLE RISKS INVOLED**
Users	Project	Lack of user participation 1
	Project	Lack of trust 2
	Project	Users resistance to change 3
	Project	Users with negative attitude toward software 4
	Project	Large number of users expected and affected 5
	Project	Conflict between users 6
Technology	product	Database used in system incompatible 7
	product	Software and other applications not desired 8
	Business	Fast changing technological trends and user's desires 9
Team members	Project	In experienced team members 10
	Project	Members lack of specialized skills required for the project 11
	Project	Lack of trust 12
	Project	Communication problems 13
	Project	Divergent working style 14
	Project	Team composition 15

	Project	Negative attitude and behaviours 16
Planning and Control	Project and product	Project mile stone not clearly analysed and defined 17
	Project	Ineffective communications problems 18
	Project and product	Poor project planning strategies 19
	Project and product	Project progress not monitored closely enough as required 20
	project	Unrealistic estimation and forecast of schedule and required software component parts 21
	Project and product	Poor user expectation management 22
Financial	Project and product	Fear of credit point reduction as penalty 23
Legal implications	Project and product	Improper referencing, quoting and respect of patent rights 24
Complexity	Project	First time involved in such a project magnitude 25
	Project	Larger project team and many clients 26
	Project	Use of technology for the first time 27
	Product	Wrongly developed components 28
	Product	Technical and performance problems 29
	Product	Difficulty to understand by client 30
Scope and Requirements	Project and product	Frequently changing requirements 31
	Project and product	Incorrect, unclear and inadequately identified system requirement 32
	Project and product	Ill-defined project 33
	Business	Not based on realistic business case 34
Project Estimations	Project and product	Project completion time underestimated 35

	Project and product	Time for testing and quality assurance underestimated 36
	Project and product	Little or no time for backup plans 37
	Project and product	Size of software underestimated 38
Execution	Product and business	Wrongly developed components 39
	Product and business	Noncompliance with project specifications-methodology 40
	Product and business	Technical and performance problems 41
	Business	Technological discontinuity 42
	Business	Change of clients’ interest due to fast technological changes. 43

Table 1.4: possible risk identification areas in our project

RISK ANALYSIS

In risk analysis, we accessed the impact on each identified risk to the part or entire project work. Here we used parameters from low medium to high to express the probability of the risk occurring and we used parameters ranging between low to catastrophic to qualify the impact to the project as seen on the table and figure below

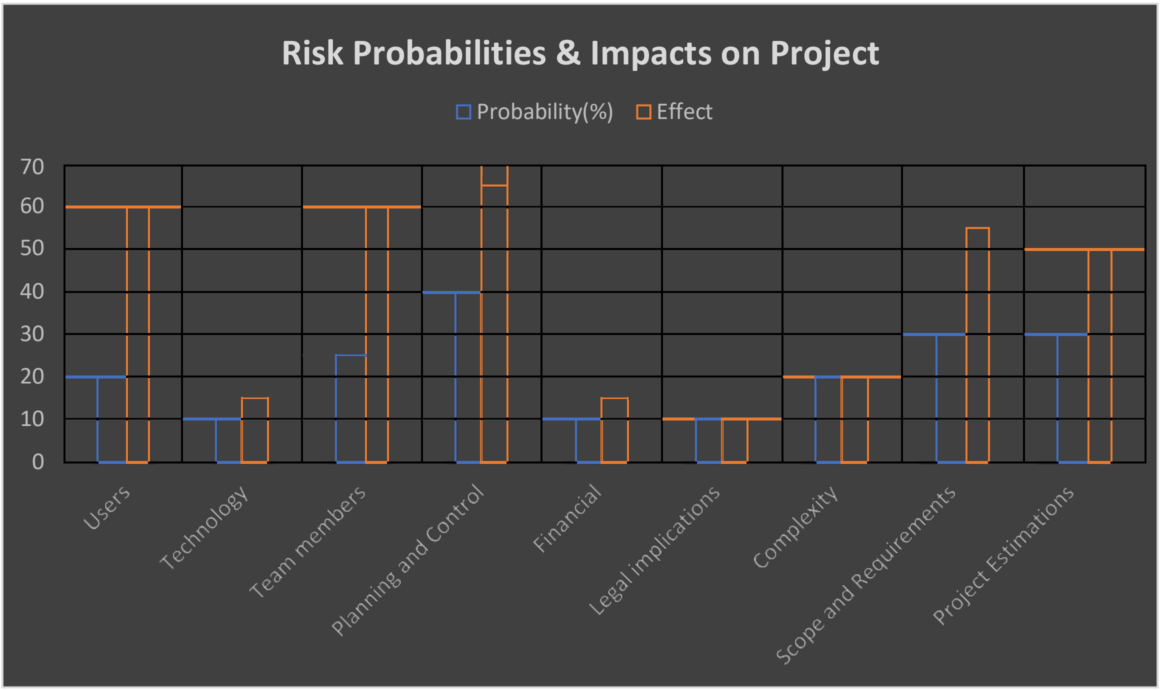


Fig 1.5: Compares of the various risk probabilities and likely degree of impact on the project

RISK ANALYSIS, PROBABILITY AND IMPACT ON PROJECT				
RISK TYPE	**AFFECTED**	**POSSIBLE RISKS INVOLED**	**PROBABILITY**	**EFFECTS**
Users	Project	Lack of user participation 1	Low	High
	Project	Lack of trust 2	Low	High
	Project	Users resistance to change 3	Low	Low
	Project	Users with negative attitude toward software 4	Low	moderate
	Project	Large number of users expected and affected 5	low	Low
	Project	Conflict between users 6	low	catastrophic
Technology	Product	Database used in system incompatible 7	High	Catastrophic
	Product	Software and other applications not desired 8	High	Serious

	Business	Fast changing technological trends and user's desires 9	moderate	High
Team members	Project	In experienced team members 10	Moderate	Serious
	Project	Members lack of specialized skills required for the project 11	High	Serious
	Project	Lack of trust 12	Low	Serious
	Project	Communication problems 13	Low	Serious
	Project	Divergent working style 14	low	Insignificant
	Project	Team composition 15	High	Insignificant
	Project	Negative attitude and behaviours 16	Low	Serious
Planning and Control	Project and product	Project mile stone not clearly analysed and defined 17	High	Catastrophic
	Project	Ineffective communications problems 18	Low	Serious
	Project and product	Poor project planning strategies 19	High	Very serious
	Project and product	Project progress not monitored closely enough as required 20	High	Catastrophic
	project	Unrealistic estimation and forecast of schedule and required software component parts 21	High	Tolerable
	Project and product	Poor user expectation management 22	Low	Serious

Financial	Project and product	Fear of credit point reduction as penalty 23	High	Catastrophic
Legal implications	Project and product	Improper referencing, quoting and respect of patent rights 24	High	Very serious
Complexity	Project	First time involved in such a project magnitude 25	High	Serious
	Project	Larger project team and many clients 26	High	serious
	Project	Use of technology for the first time 27	Low	Tolerable
	Product	Wrongly developed components 28	High	Catastrophic
	Product	Technical and performance problems 29	High	Serious
	Product	Difficulty to understand by client 30	Low	Tolerable
Scope and Requirements	Project and product	Frequently changing requirements 31	Moderate	Serious
	Project and product	Incorrect, unclear and inadequately identified system requirement 32	High	Catastrophic
	Project and product	Ill-defined project 33	High	Serious
	Business	Not based on realistic business case 34	High	Catastrophic
Project Estimations	Project and product	Project completion time underestimated 35	Very high	Catastrophic
	Project and product	Time for testing and quality assurance underestimated 36	High	Tolerable
	Project and product	Little or no time for backup plans 37	Moderate	Serious

	Project and product	Size of software underestimated 38	High	Tolerable
Execution	Product and business	Wrongly developed components 39	High	Tolerable
	Product and business	Noncompliance with project specifications-methodology 40	low	Catastrophic
	Product and business	Technical and performance problems 41	Moderate	Tolerable
	Business	Technological discontinuity 42	High	Catastrophic
	Business	Change of clients' interest due to fast technological changes. 43	Low	Serious

Table 1.6: Risk analysis, probability and impact on project

RISK RANKING

Ranking the risks helps us to map a risk impact assessment from their respective probabilities and impacts. This enables us to prioritise and focus resources and time on higher risks first [1]. This is seen in table below.

RANK	SOFTWARE RISK	DIMENSION
1	Project mile stone not clearly analysed and defined 17	Planning and control
2	Ineffective communications problems 18	Planning and control
3	Lack of user participation 1	Users
4	Project progress not monitored closely enough as required 20	Planning and control
5	Incorrect, unclear and inadequately identified system requirement 31	Scope and requirement
6	Frequently changing requirements 30	Scope and requirement
7	In experienced team members 10	Team members

8	Wrongly developed components 28	Complexity
9	Project completion time underestimated 34	Project Estimation
10	Database used in system incompatible 7	Technology

Table 1.7: Ranking of the identified risk types based on priority

RISK PLANNING

Here we aim at handling the identified risk based on their analysis. Top priority risks are selected and strategies designed to either avoid or minimise their occurrences. Also, risk planning helps come up with a contingency/action(backup) plan based on cost benefit approach[3].

SOFTWARE RISK	DIMENSION
Project mile stone not clearly analysed and defined	- Brainstorming deliverables first and then deriving the work package per deliverable. -High priority task and decisions critical to the completion of the project highly analysed and monitored -checkpoints should be understandable, controllable, focused on decisions, logical, summarised and graphically presented.
Ineffective communications problems	-Frequent team meetings before deliverable deadlines -Use standardized project communication practices proactively and effectively (software's in line with project mile stones) -Scheduled feedbacks from team members and client.
Lack of user participation	-Create a user infrastructure that outlines clearly the roles and responsibilities of all users. the communication links and services to support users -Ensure use of pilot trials, simulation, workshops, exercises/prototypes to promote users understanding and participation
Project progress not monitored closely enough as required	-Place supporting plan for team, communication, user that reminds them of deliverable dates -Use milestones dates to track and access the projects progress and accomplishments
Incorrect, unclear and inadequately identified system requirement	-Continues involvement of team and users to see that requirements are feasible and verifiable via thorough brainstorming of the products to be developed.

	-Ensure proactive requirement tracing and its validation included in the projects schedule
Frequently changing requirements	Use Mission Critical Requirements Stability Risk Assessment Metrics (MCRSRM
In experienced team members	-Negotiate additional support from team members ---Provide more, adjust plan and WBS to reflect actual support being given to the project -Inform client in due time for a possibility of delay.
Wrongly developed components	-Reanalysis of project mission, milestones, and quality assurance factor-testing
Project completion time underestimated	-Project Libre update and monitor -Work Breakdown Structure monitoring
Database used in system incompatible	-Investigate the possibility of switching to higher performing database and backups -Simulating-prototyping for better options

Table 1.8: Strategies designed to either avoid or minimise, avoid the identified risk types

RISK MONITORING

Basically, Indicators are tools for decision-support, knowledge and control of measured risk entities and their respective attributes. The application of indicators for risk assessment enables the continuous risk identification, analysis and risk controlling, since it allows analysing the effectiveness of actions through the project lifecycle. Also, risk monitoring accesses how project team reacts to identified risks. Finally, risk monitoring accesses each identified risk regularly to decide whether it is becoming less or more probable so that other contingency plans if needed implemented[2].

Risk TYPE	**INDICATORS**
Users	Client experience and participation level
Technology	Number of programing languages, product stability
Team members	Motivation, turnover, effort rate and level of expertise on the development process, maturity level
Planning and Control	Failure for feedbacks, conflicts between team and client, testing difficulties

Financial	Number of hours put in per credit earned, blame on team members to carry penalty
Legal Implications	Patent breach, referencing problems,
Complexity	Requirement complexity level and dependence
Scope and requirement	Requirement stability/changes, process changes, requirement clarity and completeness
Project estimation	Delivery date not met, project size
Execution	Software malfunctioning, methodology not respected, business level

Table 1.9: Risk monitoring indicators

The relevance of number programming languages to the risk of technology is

considered as having non-functional (environmental risk) requirement to the software development.

Also, the aspect of business level as a risk indicator is considered as external dependence level. It tries to access if there is similar product already in market or how competitive this product will be after being launched. So this will help to shape the software products value competitively[4].

12. TESTING PLAN

As earlier mentioned, testing is expected to be wide and crucial. Here every project team member(s) are expected to perform each of the required testing task individually and the whole package is tested by the group members accordingly as mentioned below.

Type	Test objective	Testing method	Testing period	Test milestone	Tested by	Test area
User	Ensures that the user freely, securely and comfortably -Navigate within the application/system	-Create and modify test for each window -Test performance per window and	8/5/17-27/5/17	System/Application well understood by user and each window meets the benchmark	Tamanampudi, Monica Kaushik Reddy	user-user and user-server

	windows via access methods	when multiple windows		-Security and access measures restricted to those concerned	Ummadi Setty, Yogitha Manasa Valirad, Sina	
Database	Verify data and update of users (employees, administration)	-Manually Inspect the database to ensure the data was populate correctly and retrieve the data to ensure that the correct data was retrieved	9/5/17-27/5/17	-All database logging, access methods and processes function as designed, -Update and without any data corruption.	Shaik Adil Tamanampudi, Monica Tammana, Naga Venkata Satya Sai Manoj Tammana, Sai Surya Akhilesh Viswanadhuni, Giri Sai	Admin server
Performance	It validates the systems response time for each activity perfumed	Script or test carried out on users' computer and on multiple users, dedicated time and on actual and modified database	9/5/17-27/5/17	There should be a successful completion of test script without any failure and within acceptable time limits for both single user and multiple users.	Tamanampudi, Monica Kaushik Reddy Ummadi Setty, Yogitha Manasa Valirad, Sina	Debugging based on White board interaction between -user-user and -user-server
Load	Ensure system works perfectly beyond standard capacity	-Modify data files -Increase users per given time	9/5/17-27/5/17	-Multiple users should be able to work on different whiteboard sheets and windows -Also server to operate at full capacity, and teste results compared	Tamanampudi, Monica Kaushik Reddy Ummadi Setty, Yogitha Manasa Valirad, Sina	- Admin Server -White board session
Security and access control	Securing third party from access to application(data) and system security(login)	-database security test -user type test and data permitted to	9/5/17-27/5/17	-Users login-registration, data accessed uniquely reviewed and restricted to system administration, except otherwise	Tammana, Naga Venkata Satya Sai Manoj	- Admin Server

		-System access test			Tammana, Sai Surya Akhilesh	-White board session
Configuration	Ensures software works properly and compatible with clients working environment	-Uses integration and system test scripts -Analyse users' workstation and test software on different user PCs	10/5/17-27/5/17	The software prototype should be compatible with users' workstation without failure and or delays	Shaik Adil Viswanadhuni, Giri Sai Tammana, Naga Venkata Satya Sai Manoj Tammana, Sai Surya Akhilesh	Test of complete software (system)
Installation	-Ensures that users personally install and set up the new software package without developer's help -Also ensure user can restore workstation to previous state in the software installation fails	-Run and install the completed package software on series of operating systems and similar user workstation applications	11/5/17-27/5/17	There should be little or no disruption in synchronisation of data in both the test environment and users' workspace	Shaik Adil Tanyi, Elvis Yalavarthi, Sreelekha	Test of complete software (system)
Functional	It ensures proper navigation, data entry, processing and retrieval both at login and level of white board	Test each unit parts (login, registration, database and white board functionalities	10/5/17-27/5/17	All test and identified bugs resolved	Shaik Adil Tanyi, Elvis Yalavarthi, Sreelekha	-White board sheet -Admin server
System Testing	Ensures that the software is fully integrated and functional	-Test each input in the application (login, register, which board and database) ensure expected outcome -Test security measures functional	12/6/17-27/5/17	the software satisfied by user -all modules operating effectively and secured	Project team	Test final software package based on clients working environment.

Table 2.0: Testing types, test area, test schedule and responsible

PACKAGING-DOCUMENTATION PLANS

PACKAGING PLANS

Based on our project specification, our software will be packaged and documentation together as seen on the tables below. Also, the due dates are based on the specified time plans as seen on the WBS above.

PACKAGING PLANS					
Plan type	**Format**	**Version**	**Description**	**Date**	**Responsible**
Packaging	Zip/tar file	1.0	Package based on users work station	09/6/17 to 10/6/17	Shaik Adil Tanyi, Elvis Yalavarthi

Table 2.1: Packaging-documentation plans

DOCUMENTATION PLANS

The documentation plan is separated into three subtypes; the installation, user document and the developers document. In this area, we clearly define the various steps to install, configure and running the software, and uninstall in case of failure. This document will be in the form of a Zip/tar file and available in Web interface.

The user document will be on a PDF format and aims at defining the procedures on how to access the system and using functionality of the application

Finally, the developers sub-document shows how System modification and upgrading can be proceeded with the specified programming languages.

DOCUMENTATION PLANS

Plan-type	Sub-type	Format	Purpose	Due date	Responsible
Documentation	Installation	-Zip/tar file -Web interface.	Clear and define steps to install, configure and running the software -Clearly defined steps to uninstall the software in case of failure to install.	09/6/17 to 10/6/17	Documentation team (Shaik Adil Tanyi, Elvis Yalavarthi, Sreelekha)
	User Doc.	PDF	-Reference manual clearly defined step by step to understand the functionality of the application system (login and whiteboard buttons, etc.) -Clearly defined steps on how to use the whiteboard to perform a task, obtain save, retrieve and modify result -Complementary hardware- software specification to aid the software effectiveness -Troubleshooting guidelines		
	Developer Doc.	-PDF	System modification and upgrading		-Frontend (Tamanampudi, Monica Tammana, Naga Venkata Satya Sai Manoj Tammana, Sai Surya Akhilesh Viswanadhuni, Giri Sai) -Back end (Ummadi Setty, Yogitha Manasa Valirad, Sina Kaushik Reddy)

Table 2.2: Documentation plans

13. REFERENCES

[1] B. W. Boehm, “Software risk management: principles and practices,” *IEEE Softw.*, vol. 8, no. 1, pp. 32–41, Jan. 1991.

[2] W. Hwang, B. Hsiao, H.-G. Chen, and C.-C. Chern, “Multiphase Assessment of Project Risk Interdependencies: Evidence from a University ISD Project in Taiwan: Multi-Phase Assessment of Project Risk Interdependencies,” *Proj. Manag. J.*, vol. 47, no. 1, pp. 59–75, Feb. 2016.

[3] Roy Schmidt, Kalle Lyytinen, Mark Keil, Paul Cule, “Identifying Software Project Risks: An International Delphi Study,” *J. Manag. Inf. Syst.*, vol. 17, no. 4, pp. 5–36, Mar. 2001.

[4] “Defining Indicators for Risk Assessment in Software Development Projects - PDF.” [Online]. Available: http://docplayer.net/6516868-Defining-indicators-for-risk-assessment-in-software-development-projects.html. [Accessed: 02-May-2017]