E BOARD

SOFTWARE REQUIREMENTS SPECIFICATION (SRS)

Version – 3.0

Date – 2016/05/22

Team–ELECTRO BLITZ

PROJECT MEMBERS

[#1] ROBIN PHILIP JOSEPH, [#2] NAGA SHRUTI ADIDAMU, [#3] SHANMUKHA SAI BHEEMISETTY, [#4] SHIVA SAI SUNKARI, [#5] RAJASHEKAR REDDY RAMIDI, [#6] VIGNESH KUNA, [#7] DIVYA KONDAVEETI, [#8] DHARANI NIMMAGADDA , [#9] B V S PREETHIKA POTLURI, [#10] SINDHU VASIREDDY, [#11] ROHITH REDDY JONNALAGADDA

Students of Telecommunication Systems, Blekinge Institute of Technology SE-371 79 Karlskrona, Sweden.

1. Preface

The main aim of the project is to develop a basic Internet Whiteboard, which facilitates communication between employees of the consultancy (ConTech) and its customers. It helps them to work together, despite being far away. This is revised version of the software requirements specification document (version-3.0)

In the remainder of the document, Section 3 presents a high level view of the architecture of the product and describes each module in the architecture. Section 4 describes the user requirements and the system requirements.

1.1 LOG MODIFICATIONS:

Release version 3.0 on 2016/05/22

- Added REQ_USR38(Reconfigurable IP)) to user requirements (section 4.1)
- Added REQ_SYS12 (SHA1), REQ_SYS13 (Open SSL) to system requirements (section 4.2)
- Made changes to section 3.1,3.2 of system architecture

Release version 2.0 on 2016/05/15

- Made changes in section 4.1(REQ_USRQ2,REQ_USRQ6,REQ_USRQ4)
- Identification strings are being identified by REQ_USR instead of REQ_USRQ
- Made changes to section 4.2, REQ_SYS8
- Added separate requirement for reload.

Release version 1.0 on 2016/04/24

- Initial Release

2. Glossary and Abbreviations

API Application Programming Interface

ConTech Consulting firm (Customer)

DB	Data Base
GUI	Graphical User interface
MySQL	Open-source relational database management system
ProjectLibre	Project management software system
Python	Programming language
PDF	Portable Document Format (File format)
RESTful	Representational State Transfer FULL (Software architectural style)
SSL	Secure Sockets Layer
RAM	Random Access Memory
JSON	Java Script Object Notation
IP	Internet Protocol

3. System Architecture

The whole system is broadly divided into three parts, the front end, back end and an interface between the front-end and back-end. This is shown below in figure 1.0

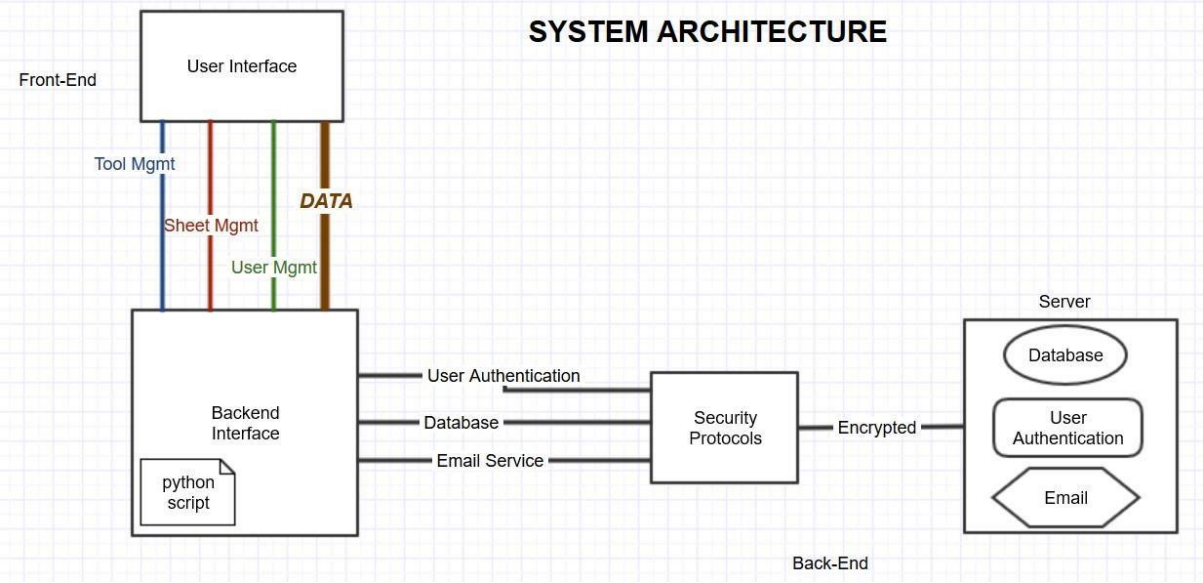

Figure 1.0

The front-end includes everything that the user can see and interact with. The back-end is responsible for the storage and management of sheets and data, including encryption and authentication. The Back-end interface is just the python script which acts as a mediator.

3.1 Front-End:

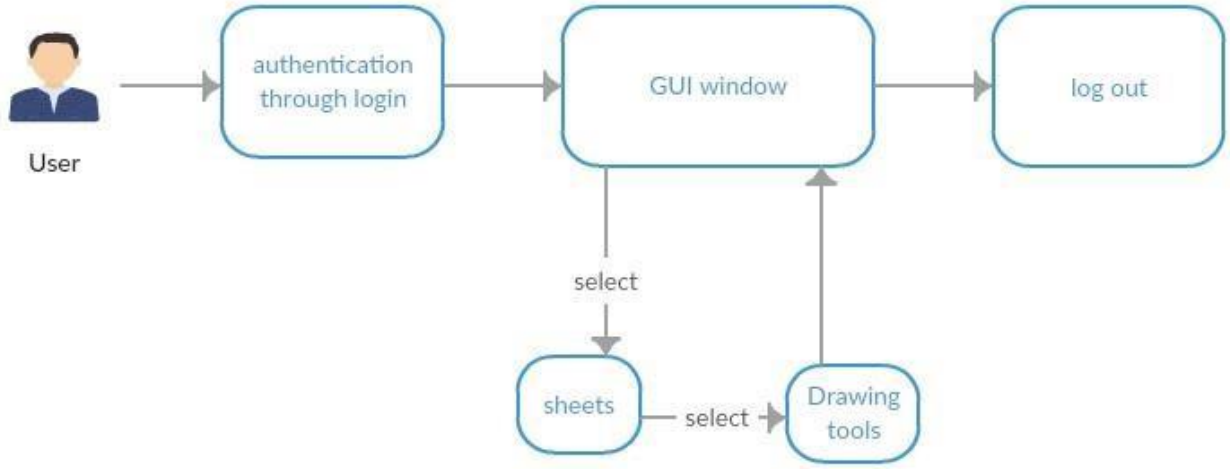

Figure 2

This module represents the front end of the system architecture. The user can join the whiteboard session through the login page. After successful login user gets access to the GUI window. Depending on the type of user (Admin, Employee and Customer), tools are available in the GUI window. A sheet is where the user can draw using the set of standard shape drawing tools, line, arrow, circle, square, rectangle, oval, text, eraser and free drawing tool. For each tool the user will be able to select thickness, the drawing color and the filling color. The employee who starts the session is made the moderator and can lock and undo sheets.

3.2 Back-End:

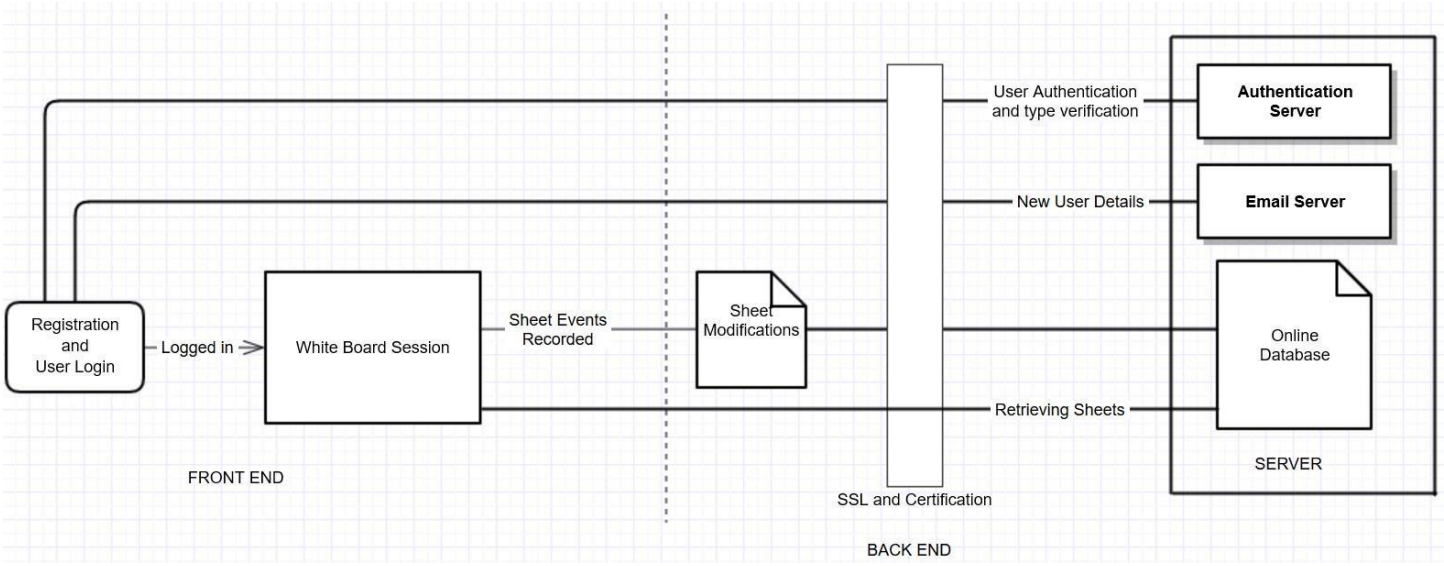

Figure 3

The users should have an account for using the white board. Once an account is registered, the email server sends the user their login credentials and the user is authorized through the authentication server. Once the users of the application login to their whiteboard session, they can draw or write on the sheet and use the tools available. Each modification event is recorded and the data is sent to the MySQL database. A user can reload the events in a sequence using this data. The application will constantly retrieve data from the database and thus every user will see the modifications replicated in their session.

4. Requirements

This section is a description of the product to be developed. It lays out functional and non-functional requirements. It also includes a list of actions that describe user interactions that the product must provide. It comprises of user requirements and system requirements.

4.1 User Requirements

This section describes the services provided for the user by the product.

S.no	REQUIREMENT	IDENTIFICATION-STRING	CREATION DATE	MODULE	TYPE	DEPENDICIES	TEST	DESCRIPTION
1	Unlimited number of Users	REQ_USR1	2016/04/18	Front End	Functional	REQ_SYS5	TEST_FE21	• Product should allow any number of users to be created.

2	Admin privileges	REQ_USR2	2016/04/18	Back End	Functional	REQ_SYS5	TEST_BE3	• Product should allow Admin to create customer and employee account
3	Creation of an Employee account	REQ_USR3	2016/04/18	Back End	Functional	REQ_SYS5	TEST_BE4	• Product should allow employee to create customer accounts and start whiteboard sessions.
4	Creation of an Customer account	REQ_USR4	2016/04/18	Back End	Functional	REQ_SYS5	TEST_BE5	• Product should allow customer to join an existing whiteboard session.
5	Validity period of an account	REQ_USR5	2016/04/18	Back End	Functional	REQ_SYS5	TEST_BE7	• Every employee and customer account shall have a validity period.
6	Authentication	REQ_USR6	2016/04/18	Back End	Functional	REQ_SYS5 REQ_SYS8	TEST_BE12	• Unauthorized users shall be prevented from entering the whiteboard session.
7	Acknowledgement via E-mail	REQ_USR7	2016/04/18	Back End	Functional	REQ_SYS2	TEST_BE8	• Upon registration login details should be sent to the user via E-mail.

8	Selection of sheets	REQ_USR8	2016/04/18	Front End	Functional	REQ_SYS2	TEST_FE13	• Product should enable user to select and change sheets.
9	Replication of events	REQ_USR9	2016/04/18	Front End	Functional	REQ_SYS2	TEST_FE22	• Contents should be replicated in real-time on all other devices in the current session.
10	Line	REQ_USR10	2016/04/18	Front End	Functional	REQ_SYS2	TEST_FE3	• Product shall enable user to draw a line.
11	Arrow	REQ_USR11	2016/04/18	Front End	Functional	REQ_SYS2	TEST_FE20	• Product shall enable user to draw an arrow.
12	Circle/Oval	REQ_USR12	2016/04/18	Front End	Functional	REQ_SYS2	TEST_FE5	• Product shall enable user to draw circle/oval.
13	Square/ Rectangle	REQ_USR14	2016/04/18	Front End	Functional	REQ_SYS2	TEST_FE6	• Product shall enable user to draw square/rectangle.
14	Poly-line	REQ_USR16	2016/04/18	Front End	Functional	REQ_SYS2	TEST_FE4	• Product shall enable user to draw poly-line.

15	Text	REQ_USR17	2016/04/18	Front End	Functional	REQ_SYS2	TEST_FE8	• Product shall enable user to enter text.
16	Eraser	REQ_USR18	2016/04/18	Front End	Functional	REQ_SYS2	TEST_FE2	• Product shall enable user to erase contents on the whiteboard canvas.
17	Free drawing tool (Pencil)	REQ_USR19	2016/04/18	Front End	Functional	REQ_SYS2	TEST_FE1	• Product shall enable user to freely draw on the whiteboard.
18	Thickness	REQ_USR20	2016/04/18	Front End	Functional	REQ_SYS2	TEST_FE10	• Thickness also can be selected for all drawing tools.
19	Color	REQ_USR21	2016/04/18	Front End	Functional	REQ_SYS2	TEST_FE11	• Outline color for tools can also be selected from the menu.
20	Color Fill	REQ_USR22	2016/04/18	Front End	Functional	REQ_SYS2	TEST_FE12	• Color fill shall be available for all shapes.
21	Font size	REQ_USR23	2016/04/18	Front End	Functional	REQ_SYS2	TEST_FE14	• Font size for text can be selected.

22	Font style	REQ_USR24	2016/04/18	Front End	Functional	REQ_SYS2	TEST_FE15	• Font styles for test also shall be available
23	Bold and Italic	REQ_USR25	2016/04/18	Front End	Functional	REQ_SYS2	TEST_FE16	• Bold and Italic for text shall also be available.
24	Moderator	REQ_USR26	2016/04/18	Front End	Functional	REQ_SYS2	TEST_FE24	• Product enables creator of session to be moderator.
25	Change of Moderator Role	REQ_USR27	2016/04/18	Front End	Functional	REQ_SYS2	TEST_FE19	• Moderator role can be reassigned to another participant.
26	Lock	REQ_USR28	2016/04/18	Front End	Functional	REQ_SYS2 REQ_SYS3	TEST_FE17	• Moderator must be able to lock the access to sheets.
27	Undo	REQ_USR29	2016/04/18	Front End	Functional	REQ_SYS2 REQ_SYS3	TEST_FE9	• Moderator must be able to undo changes.

28	List of Modifications	REQ_USR30	2016/04/18	Back End	Functional	REQ_SYS3	TEST_BE13	• Index, Time stamp, name of user, type of modification and sheet on which the changes occurred shall be stored in the database.

29	Playback of Modifications (Auto-play)	REQ_USR31	2016/04/18	Front End	Functional	REQ_SYS4 REQ_SYS5	TEST_FE18	• All modifications shall be played back upon the click of play-back button.
30	Playback of Modifications (Upon click)	REQ_USR32	2016/04/18	Front End	Functional	REQ_SYS4 REQ_SYS5	TEST_FE18	• Modifications shall be played back upon each click of a button.
31	Encryption of web server	REQ_USR33	2016/04/18	Back End	Functional	REQ_SYS7	TEST_BE6	• Communication to and from web server shall be encrypted.
32	Encryption of MySQL	REQ_USR34	2016/04/18	Back End	Functional	REQ_SYS7	TEST_BE9	• Communication to and from MySQL database shall be encrypted.
33	RESTful API	REQ_USR35	2016/04/18	Back End	Functional	REQ_SYS6	TEST_BE10	• Server-client interaction must follow REST principles
34	Documentation	REQ_USR36	2016/04/18	-	Non-Functional	-	-	• Documentation includes user manuals that provide instructions regarding software installations as well as tool operating instructions.
35	Debugging	REQ_USR37	2016/04/18	Back End	Functional	REQ_SYS7	TEST_BE11	• It should be possible to disable encryption in server-client communication

36	Reconfigurable IP	REQ_USR38	2016/04/18	Front End	Functional	REQ_SYS2	TEST_ FE23	• Admin has a privilege to configure IP address.

Table 1

4.2 System Requirements

This section provides details about the system requirements. These are technical requirements that complement the user requirements and provide information for design and implementation of product.

S.No	REQUIREMENT	IDENTIFIC-ATION STRING	CREATION DATE	MODULE	TYPE	DEPENDENCY	TEST	DESCRIPTION
1.	Platform	REQ_SYS1	2016/04/19	-	Non-Functional	-		Operating system required are • Microsoft Windows 7, 8 or 10. • RAM required is 2GB.
2.	Python-Executable	REQ_SYS2	2016/04/19	-	Functional	-		• System should be able to run and execute Python and its corresponding modules.
3.	Web Server	REQ_SYS3	2016/04/19	Back-End	Functional	-		• Application should be able to access the database on the corresponding web server.

4.	Database	REQ_SYS4	2016/04/19	Back-End	Functional	REQ_SYS3		• SQL database is employed where Information related to Whiteboard sessions are stored.
5.	Account Management	REQ_SYS5	2016/04/19	Back-End	Functional	REQ_SYS4		• System should be able to save and retrieve users' data from the database.
6.	Interfacing	REQ_SYS6	2016/04/19	-	Functional	REQ_SYS2 REQ_SYS3 REQ_SYS4		• RESTful API with JSON data encoding.
7.	Encryption	REQ_SYS7	2016/04/19	Back-End	Functional	-		• All data communication by the system are encrypted using openSSL or standard python libraries
8.	Authentication	REQ_SYS8	2016/04/19	Back-End	Functional	REQ_SYS4		• System shall be able to authenticate users.
9.	Testing	REQ_SYS9	2016/04/19	-	Functional	-		• Running the tests to ensure proper functioning of front-end and back-end modules.
10.	Packaging	REQ_SYS10	2016/04/19	-	Non-Functional	-		• All the files are packaged into an executable file.

11.	Debugging Feature	REQ_SYS11	2016/04/19	-	Functional	-		• Encryption can be disabled for debugging purpose
12.	SHA1	REQ_SYS12	2016/04/19	-	Functional	-		• SHA1 algorithm is used for PHP scripts.
13.	Open SSL	REQ_SYS13	2016/04/19	-	Functional	-		• OpenSSL is used to establish secured communication through certificates.

Table 2

5. References

[1] Bourque. P and Fairley, R.E. (2014). "Guide to the Software Engineering Body of Knowledge (SWEBOK)". IEEE Computer Society. https://en.wikipedia.org/wiki/Software_requirements_specification

[2] ConTech product request, P5: Internet Whiteboard, 2016. [Online]. Available for itslearning users: https://bth.itslearning.com/ContentArea/ContentArea.aspx?LocationID=6504&LocationType=1