INTERNET WHITEBOARD

Design document

Group name: 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

The primary aim of the project is to build the internet white board, which works as a correspondence path between employees of ConTech and clients, it helps them to communicate and work together.

Release v1.3 on 10-11-2017

Design Document organized into five sections, each section describes respective requirements. In Section 3, information provided is about the modules and the labelling of figures modified as well as the information regarding the explanation on working of each module given. Screen shots were removed anddetails about the tests whether it is a User or System requirement mentioned. Under the figure that presented the detailed design, starting a session modified which is a functionality of moderator. In section 4, mentioned the functioning of Restful API, communication between user-server and user - server interaction using RestfulAPI

Release v1.2 on 28-10-2017

Design Document organized into five sections, each section describes respective requirements.. In section 1, type of document is mentioned. In section 2, glossary is updated and explained each term. In section 3, explanation for structure of module have been updated, step-step instruction were updated for each unit test cases. In section 4, Functioning of RestfulAPI in mentioned.

Release v1.1 on 28-05-2017

-Initial release

The Design Document is organized into five sections, each section describes respective requirements. In Section 2 information is provided for glossary and abbreviations that are used in the document, section 3 which describes Detailed Design figure has been modified and under Unit Test Plan details of the settings that are performed before the test is done is mentioned, section4 describes the module2 and tests carried out for module2 section 5 describes references.

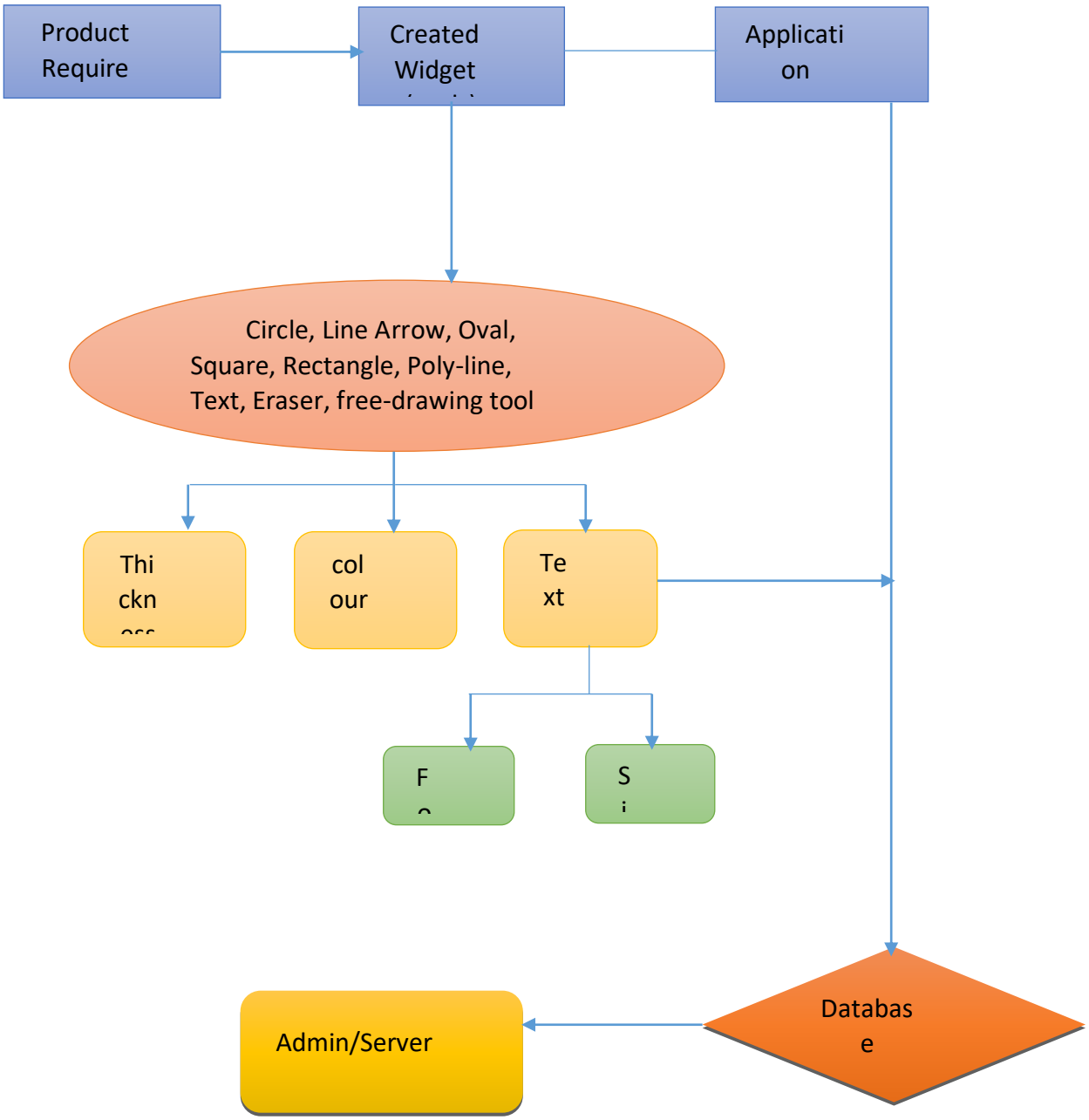
2.Glossary and Abbreviations

1. **IP Address**: It is known as Internet Protocol Address. It is a unique number assigned to each system, which connected in a Network.
2. **SQL:** Standard Queuing Language. It is a special purpose language, whichused to 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, which enable services to work best on the Web. API is Application Programmable Interface.

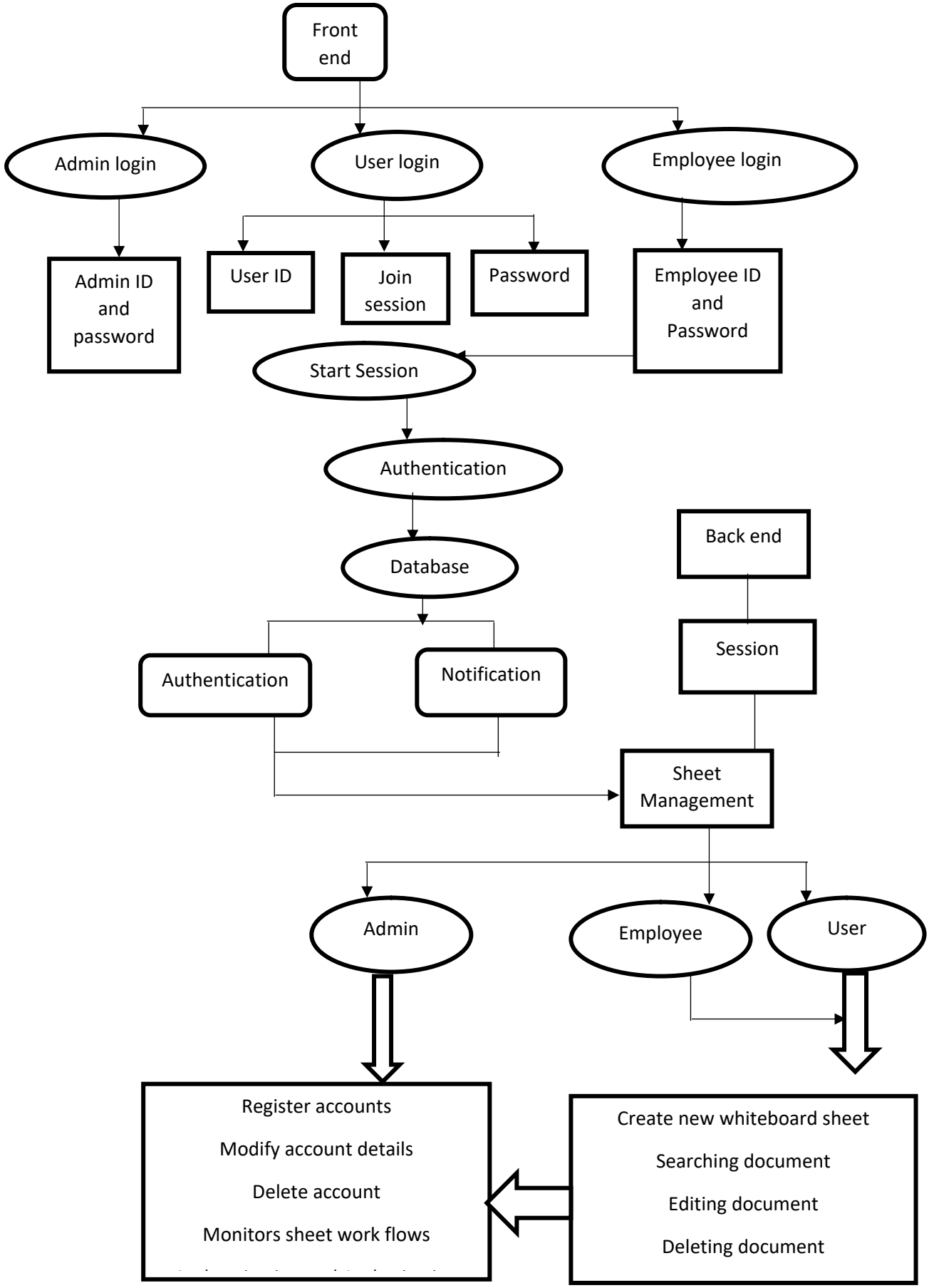
5. **PyMySQL:** PyMySQL is adatabaseconnectors for Python programing language libraries and its used to enable Python programs to talk to a MySQL server [2]
6. **MySQL dB:** Its having same functionality as PyMySQL [2]
7. **Timestamp:** It considers as a series of characters or encoded information that 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)
   a. package. [3]
9. **PHP:** Hyper Text Preprocessor is a server scripting programing language that 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.Module 1 Front End

Block diagram 1



3.1 Detailed design



In this section, we present a detailed picture of the module 2 structure presented above. This done by use of UML diagrammatical structure as seen below. Our systems framework involves the product requirements, created widgets (tools), application modifications, database and admin server respectively. The product requirements are considered as the series of tasks that should be completed in one session within the software.

3.2 Unit test plan

Each unit or component is mentioned below and according to the requirement, testing is done for each component.

Test: Line

Purpose: Considering test for line, 'Line' functions that tool according to requirement.

Requirements: The required ID that connected to this case is UR3.

Identifier: 3.1.6

Environment: The environment used for this case is Python Tkinter, which is a standard Graphical User Interface.

Operation: First, open on the Ubuntu. After opening the Ubuntu, run the application. After running the application, select the sheet on which the operation is to be done. Now, select the option 'choose a drawing tool', then select the tool 'Line' in the tool tab and then draw the line by using cursor.

Ubuntu → Run Application Select Sheet Choose Drawing tool Select Line →

Draw.→

Expected result: After the operation is carried out a straight line has to be appear in the sheet.

Comment: Run the application and select the tool that is to be tested.

Test: Pencil

Purpose: Considering the test for Pencil, 'Pencil' functions as per the requirement that tool.

Requirements: The required ID is connected to this case is UR18.

Identifier: 3.1.7

Environment: The environment used for this case is Python Tkinter, which is a standard Graphical User Interface.

Operation: First,open on the Ubuntu. After opening the Ubuntu, run the application. After running the application, select the sheet on which the operation is to be done. Now, select on the option 'choose a drawing tool', then select the tool 'Pencil' in the tool tab and then draw the line by using cursor.

Ubuntu → Run Application Select Sheet Choose Drawing tool Select →

Pencil Draw.

Expected result: After operations carried out a picture should be obtained on the sheet.

Test: Arrow

Purpose: Considering the test for Arrow, 'Arrow' functions as per the requirement that tool.

Requirements: The required ID that is connected to this case is UR4.

Identifier: 3.1.8

Environment: The environment used for this case is Python Tkinter, which is a standard Graphical User Interface.

Operation: First, open on the Ubuntu. After opening the Ubuntu, run the application. After running the application, select the sheet on which the operation is to be done. Now, select the option 'choose a drawing tool', then select the tool 'Arrow' in the tool tab and then draw by using the cursor.

Ubuntu Run Application Select Sheet Choose Drawing tool Select Arrow

Draw

Expected result: After operations carried out picture of arrow should be obtained on the sheet.

Test: Circle

Purpose: Considering the test for Circle, 'Circle' functions as per the requirement that tool.

Requirements: The required ID that is connected to this case is UR6.

Identifier: 3.1.10

Environment: The environment used for this case is Python Tkinter, which is a standard Graphical User Interface.

Operation: First, open on the Ubuntu. After opening the Ubuntu, run the application. After running the application, select the sheet on which the operation is to be done. Now, select the option 'choose a drawing tool', then select the tool 'Circle' in the tool tab then draw by using the cursor.

Ubuntu Run Application Select Sheet Choose Drawing tool Select Circle
Draw.

Expected result: After operations carried out circle should be obtained on the sheet.

Test: Eraser

Purpose: Considering the test for Eraser, 'Eraser' functions as per the requirement.

Requirements: The required ID that is connected to this case is UR9.

Identifier: 3.1.12

Environment: The environment used for this case is Python Tkinter, which is a standard Graphical User Interface.

Operation: First,open on the Ubuntu. After opening the Ubuntu, run the application. After running the application, select the sheet on which the operation is to be done. Now, select the option 'choose a drawing tool', then select the tool 'Eraser' in the tool tab then erase the picture using cursor.

Ubuntu → Run Application → Select Sheet → Choose Drawing tool → Select eraser → Erase.

Expected result: After operations carried out, erased picture should be obtained on the sheet.

Test: Font selection

Purpose: Considering the test for font selection, 'Font selection' functions as per the requirement.

Requirements: The required ID that is connected to this case is UR19.

Environment: The environment used for this case is Python Tkinter, which is a standard Graphical User Interface.

Operation: First, open on the Ubuntu. After opening the Ubuntu, run the application. After running the application, select the sheet on which the operation is to be done. Now, select the option 'choose a drawing tool', then select the tool 'Select Font' in the tool tab.

Ubuntu → Run Application → Select Sheet → Choose Drawing tool → Select Font.

Expected result: After operations carried out selected font should be obtained on the sheet

Test: Polyline

Purpose: Considering the test for Polyline, 'Polyline' functions as per the requirement that tool.

Requirements: The required ID that is connected to this case is UR5.

Environment: The environment used for this case is Python Tkinter, which is a standard Graphical User Interface.

Operation: First, open on the Ubuntu. After opening the Ubuntu, run the application. After running the application, select the sheet on which the operation is to be done. Now, select the option 'choose a drawing tool', then select the tool 'Polyline' in the tool tab then draw by using the cursor.

Ubuntu → Run Application → Select Sheet → Choose Drawing tool → Select → Polyline Draw.

Expected result: After operations carried out desired polyline should be obtained on the sheet.

Test: Rectangle

Purpose: Considering the test for Rectangle, 'Rectangle' functions as per the requirement.

Requirements: The required ID that is connected to this case is UR10.

Environment: The environment used for this case is Python Tkinter, which is a standard Graphical User Interface.

Operation: First, open on the Ubuntu. After opening the Ubuntu, run the application. After running the application, select the sheet on which the operation is to be done. Now, select the option 'choose a drawing tool', then select the tool 'Rectangle' in the tool tab then draw by using cursor.

Ubuntu → Run Application → Select Sheet → Choose Drawing tool → Rectangle → Draw.

Expected result: After operations carried out rectangle should be obtained on the sheet.

Test: Select Filling color

Purpose: Considering the test for selecting color, 'select filling color' functions as per the requirement.

Requirements: The required ID that is connected to this case is UR13.

Environment: The environment used for this case is Python Tkinter, which is a standard Graphical User Interface.

Operation: First, open on the Ubuntu. After opening the Ubuntu, run the application. After running the application, select the sheet on which the operation is to be done. Now, select the option 'choose a drawing tool', then select the tool 'select filling color' in the tool tab then select the desired color.

Ubuntu → Run Application → Select Sheet → Choose Drawing tool → Select Filling Color Select Color.

Expected result: After operations carried out desired color should be obtained on the sheet.

Test: Text

Purpose: Considering the test for text, 'Text' functions as per the requirement.

Requirements: The required ID that is connected to this case is UR8.

Environment: The environment used for this case is Python Tkinter, which is a standard Graphical User Interface.

Operation: First, open on the Ubuntu. After opening the Ubuntu, run the application. After running the application, select the sheet on which the operation is to be done. Now, select the option 'choose a drawing tool', then select the tool 'Text' in the tool tab then write text.

Ubuntu → Run Application → Select Sheet → Choose Drawing tool → Select Text → Write.

Expected result: After operations carried out written text should be obtained on the sheet.

Test: Registration form (Front end)

Purpose: Considering the test for Registration page, when employee registers the credentials of the user, after the registration the credentials is saved in a database, which is used later.

Requirements: The required ID that is connected to this case is UR2.

Environment: The environment used for this case is HTML and CSS for styling.

Operation: First, open the Ubuntu. Then click on the HTML file and open in browser then provide username, email address, set a desired password and repeat the password in the placeholder and then click on the signup button.

Ubuntu → Select HTML → Open Browser → Provide Email address → Set a Password → Repeat the password → Log in.

Expected result: After operations carried out registration form credentials must be saved in the database should be obtained.

Test: Login form (Front end)

Purpose: Considering the test for Login page, that is checking response when the login details are provided.

Requirements: The required ID that is connected to this case is UR1.

Environment: The environment used for this case is HTML and CSS for styling.

Operation: First, open the Ubuntu. Then click on the HTML file and open in browser then provide user email address, type the password that has been set during registration and then click on the login button.

Ubuntu → Select HTML → Open Browser → Provide Email address → provide Password Login.

Expected result: After logging into the session user must be able to see the user login page.

Test: Playback (Backend)

Purpose: Considering the test for Playback, All the modifications that is done on a sheet by the user is provided. Modifications done are provided with a time delay between each modification.

Requirements: The required ID that is connected to this case is UR29.

Environment: The environment used for this case is Python Tkinter, which is a standard Graphical User Interface.

Operation: First, open on the Ubuntu. After opening the Ubuntu, run the application. After running the application, select the sheet on which the operation is to be done. Now, select the option 'choose a drawing tool', then select any tool in the tool tab and draw. Repeat selecting the tool tab and continue to draw using the cursor .Next click on 'Playback 'then all the previous modifications that is done is provided with a time delay between each modification.

Ubuntu → Run ApplicationSelect Sheet → Choose Drawingtool → Selectany toolDraw on a sheetClick on Playback.

Expected result: After a user selects a sheet and chooses a drawing tool to draw on a sheet using different tools in the tool bar each time to draw then when we click on playback all the modifications that is done previously is provided each modification is provided with a short time delay.

4.MODULE 2Backend

This area discusses the architecture of module 2 and various the working of features requested by customer. It further shows the interrelationship between the database, web interface and the admin server.

Starting with the database sequence structure, we see that it forms two interfaces with the web GUI and the admin server.it stores user login information and displays the saved data upon request. Furthermore, authentication processes are carried out with the admin server as well as storage and retrieval or user data as well.

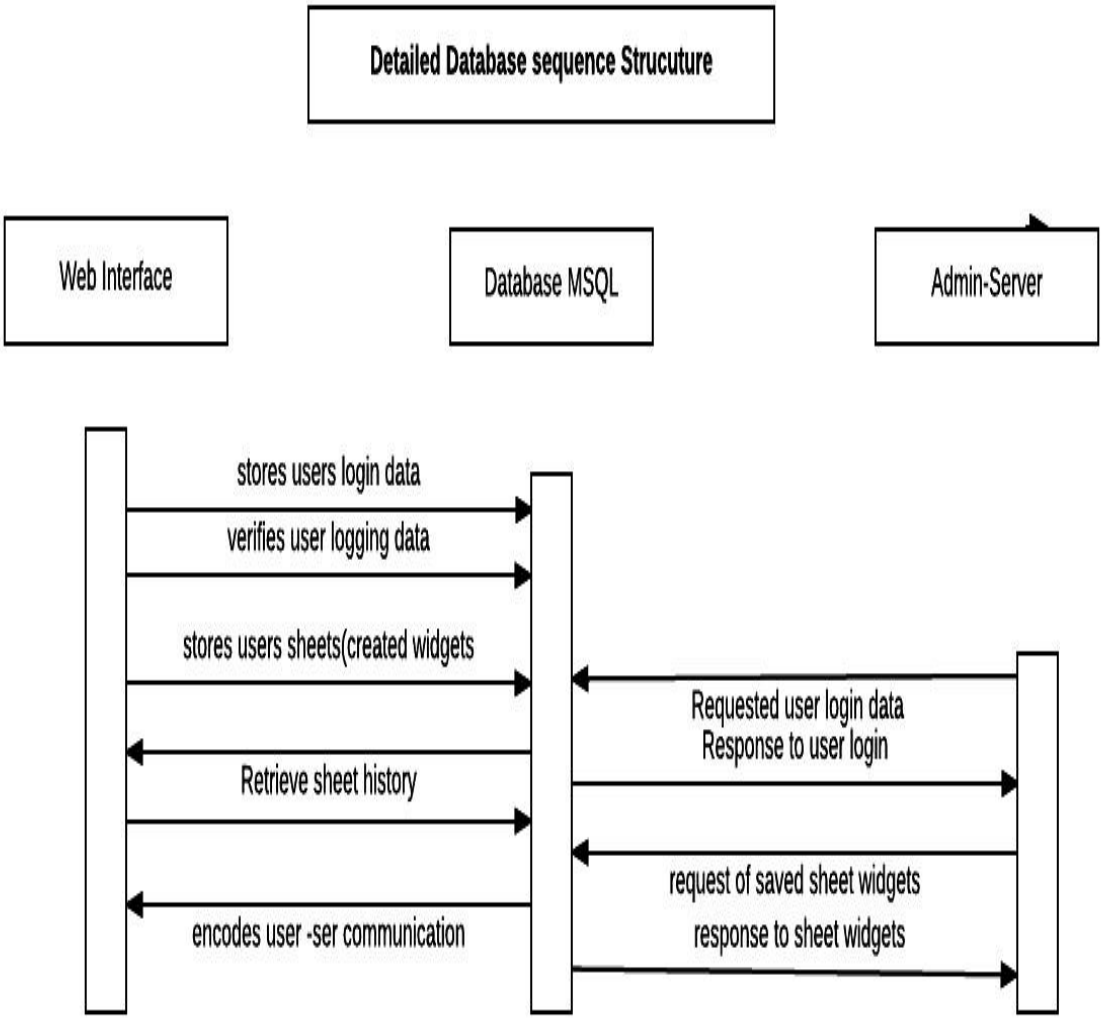


Figure3: database sequence structure

With respect to the backend sequence structure, it includes the interfaces between the frontend and database, and the database and the admin server. Firstly, the database and admins servercarriesout a RESTful API authentication process concerning the Message- Data exchange between users-user and user -server. Admin server verifies and validates new users upon login and relate with database to confirm user authentication and access.

Flask-server: Flask server is implemented in Admin where it is configured in such a way that whenever a request is received from user, it responses with the required information where Restful - API comes into existence.

User-User and Admin-User Interaction: The interaction is done by directly connecting the application to Admin-database. Only for two of the features mentioned below, the interaction is done using Restful API.

Restful API is implemented here for two cases:

1. While retrieving the lock status which is stored in database of admin using an API. For this respective feature, we run a flask server in admin. This flask server provides the information of the lock status by retrieving it from the database of admin.

2. While retrieving the information of moderator which is also done by using a flask server in admin that provides a response with the required information in it when a request is received to the flask server running in admin.

Restful-API is tested by checking whether the response is received or not. The response in the two cases above is lock status and information regarding moderator. By testing lock function and feature that includes moderator change, the restful API is tested and considered working.

According to the customer requirements, we have used pythonTkinter module, Flask server to implement RESTful API and database Mysql which is considered as backend . This process is executed by use of MSQL database and Python scripting language.

Python Tkinter module: Tkinter module is a GUI in python where tools required in white board is implemented which is inter-connected to Database.

Mysql database: Database is used to store the information of drawing tools of employee and user from time to time. The user and employee details are also saved in database.

In software development environment, high-level architectural design simply explains the systematic array of diagrams that clearly explains the roles of the respective components of the software product to be produced. This architectural diagram provides the overview of the complete system, main, sub components, and respective interfaces that would be developed and combined for the software product to operate effectively. It should be noted that the aim of the architecture is to also show the overview of the various solutions, products and services identified, respective platforms, systems and tools used during the development process.[1] [2] best on the Web. API is Application Programmable Interface (P5:M2-SRC DOCS, its learning).

REFERENCES:

[1] "CJ_05_Design.pdf."

[2] "High-level design," *Wikipedia*. 29-Jan-2017.

[3] P5: internet whiteboard, M2-SRC DOCS, its learning

.