Contents

Lab Authentication System

Security…. is a myth.

Rohit Lingala
1602-18-735-031
Vasavi College of Engineering

Vishnu Kaushik
1602-18-735-059
Vasavi College of Engineering

Vishnu Vardhan
1602-18-735-060
Vasavi College of Engineering

ABSTRACT:

Lab authentication system is a JavaFX application that authenticates the users to access the system in the lab. This application is based on a client-server model. All the systems are initially present in Kiosk mode, and only the authenticated users will be able to use any system present in the lab. The server identifies the user and recognizes the person using a facial recognition system. Any new users are required to train their faces in order to recognize their faces next time. The application also provides a way to login into the systems using username and password when the face recognition system is inaccessible to the user or when the system crashes. The application assigns the systems to authenticated users. The server stores information about the login history of all the systems including the users which enables the administration to analyze the information.

Keywords—OpenJDK 16, PhpMyAdmin, MySQL, Kiosk mode, Server Client model, JavaFX, multi threading, subnetting.

I. MOTIVATION

In this growing digital world, maintaining a record for a process sounds like an obsolete approach. Keeping in point of view the efficiency, labor and accessibility. On the same lines of thought, one of our lecturers Mr. Gupta Sir (Lab Incharge) has suggested an idea to implement a system which automates the whole process of the lab management system. From Authenticating the user to allocating the system.

In light of the above suggestion we wanted to design a comprehensive system, which satisfies the above need, simultaneously making it user-friendly. The vision of this project stands still to establish a sustainable system, which can be managed by students, ensuring optimal security and maintaining a clear record of users accessing the computers, mainly for research or projects.

This paper is the end result of our intense trials and scrutinization which started as a system to authenticate users using a barcode system. But ended up being a face recognizing system. Initially we had many thoughts but keeping in mind the budget and feasibility, we have arrived at this solution.

II. INTRODUCTION

This project comprises 2 main applications named server.jar and client.jar. A JAR (Java Archive) is a package file format typically used to aggregate many Java class files and associated metadata and resources (text, images, etc.) into one file to distribute application software or libraries on the Java platform. In simple words, a JAR file is a file that contains compressed versions of .class files, audio files, image files or directories [1]. So the only requirement that a system should have to run our application is a pre-installed MySQL server with proper privileges, still we are trying to automate that process. We have developed our entire application in java, so our application is platform independent, installation of dependencies is not at all required, everything comes in the jar file.
server.jar has to be loaded into any of the PC, which would be acting as the master/central system and all the remaining systems in the lab should be loaded with client.jar. In detail explanation is covered in the deployment phase in section V.

III. PROPOSED BLOCK DIAGRAM

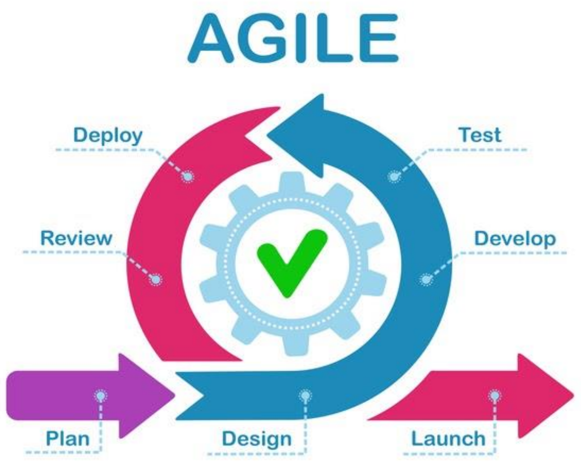


FIG. 1. DIAGRAM DEPICTING AGILE MODEL OF SOFTWARE DEVELOPMENT

[2] Though Most of the developers use SDLC (Software development life cycle), we have this time followed AGILE Development model which is the fastest way to finish the project with the least bugs as it provides the chance to alter the code and database structures whereas in SDLC Software development is a cumbersome activity requiring proper identification of requirements, their implementation, and software deployment.

IV. EXPLANATION

We have broadly classified our project into the following categories.

A. *Database schema:*

1) auth : to store admin and user credentials
2) logs: to record check in and check out of a user
3) clients: to store the clients information

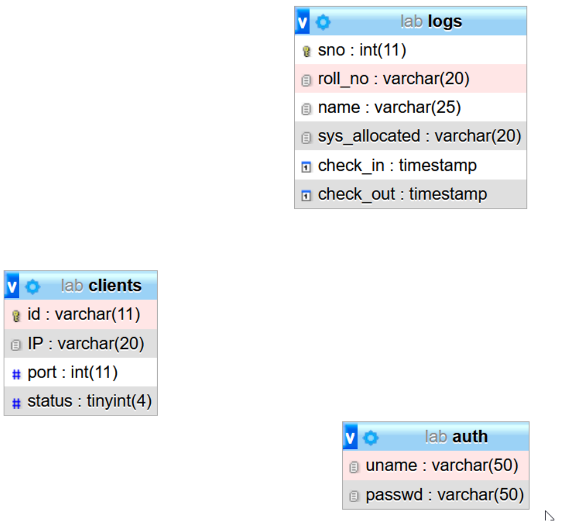


FIG. 2. RELATIONAL SCHEMA OF THE DATABASE

B. *Back-end:*

1) We have clearly explained how to set up the required network in the implementation section. Our back-end is a simple client server model [6], using which are successful in locking and unlocking all the systems in the lab using a single master PC.

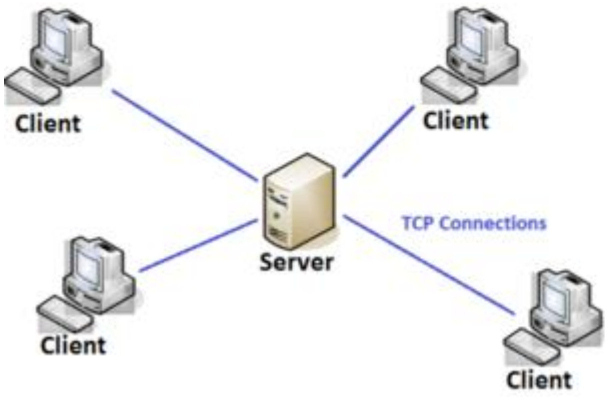


FIG. 3. NETWORK DIAGRAM OF THE PROJECT

2) We have created 2 means by which a user logs in; one is through the server and in order to login from the client side (if in case our system crashes) we have created a simple login system. So the client has to do 2 actions simultaneously here, we have implemented this feature using a multi threading concept.

3) We have identified one of the major vulnerability in our system which is as follows:

a) If a hacker is in the same network then he can easily disrupt the client server communication because standard client server communication occurs on a fixed port. We have patched that vulnerability by making the port dynamic. Only the client and server knows the port of operation and that changes time to time.

4) Kiosk mode: We lock each and every client using kiosk mode in which all the keys except alphanumeric keys will be blocked, so that a user can’t bypass our application by any means. As soon as the system boots our applications start running since we will be adding our application to startup. So in order to use the systems the user should identify his identity at the server side.

C. *Front-end:*

1) We have used JavaFX and scene builder to create different screens for our application. Which includes server side interface, Client side interface.

2) We have used CSS to style the front end screens.

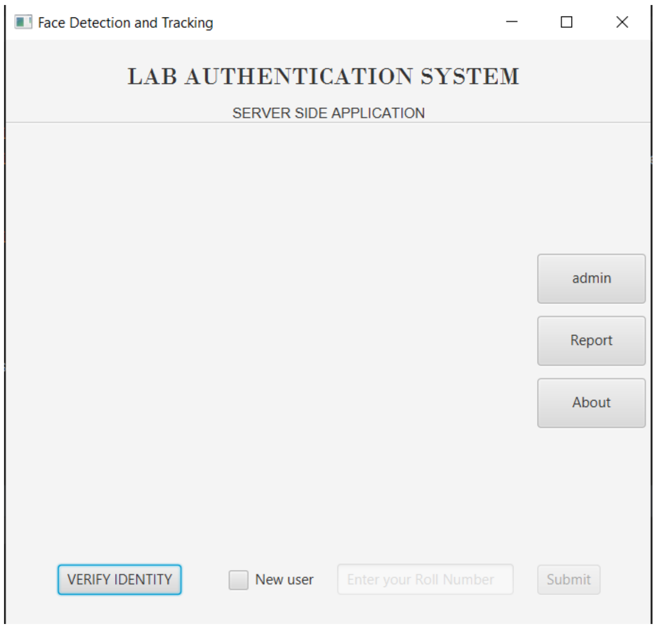


FIG. 4. SERVER SIDE APPLICATION INTERFACE.

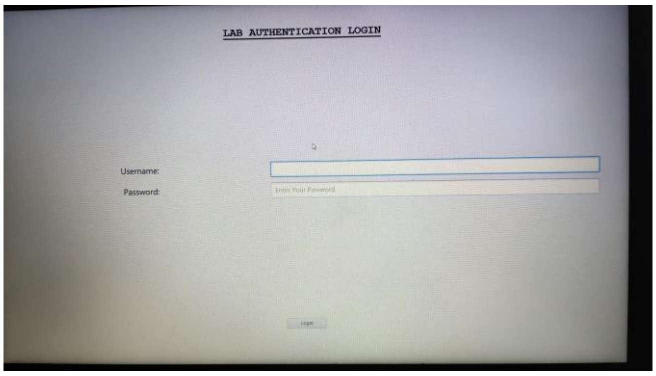
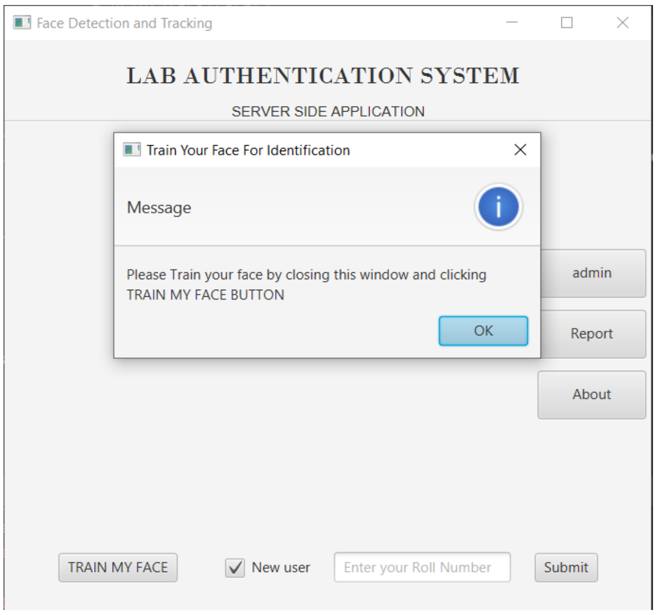


FIG. 5. CLIENT SIDE INTERFACE WITH KIOSK MODE.

D. *Face recognition:*

The basic implementation of face recognition comes from OpenCV. Where we have used the LBPH classifier to detect the face. Once the face is being detected the system starts to take up to 30 pictures, which are stored into a file with a label of the roll number the user entered.
Once the pictures are stored in a file, the data gets trained and when we run the model to recognize the face. It recognizes using the roll number. The complete process of LBPH algorithm is discussed in detail in this page [8].


Face Detection and Tracking

LAB AUTHENTICATION SYSTEM

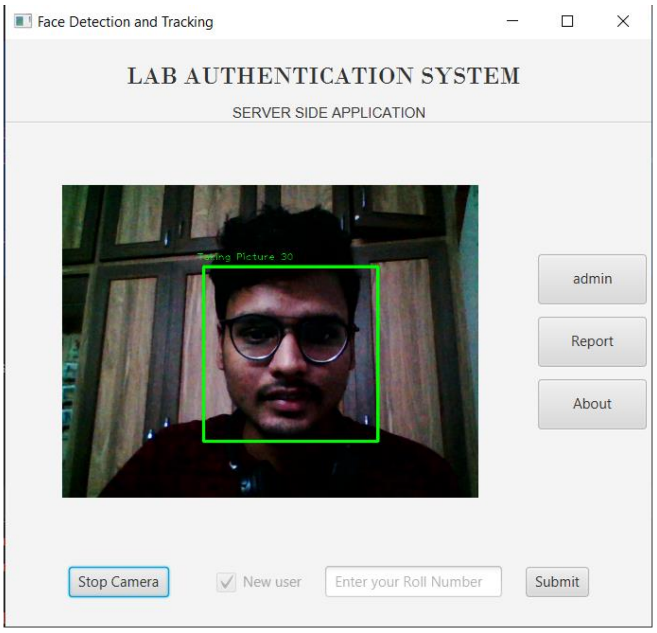
SERVER SIDE APPLICATION

Train Your Face For Identification

Message

Please Train your face by closing this window and clicking TRAIN MY FACE BUTTON

FIG. 6. ROLL NUMBER VERIFICATION


Face Detection and Tracking

LAB AUTHENTICATION SYSTEM

SERVER SIDE APPLICATION

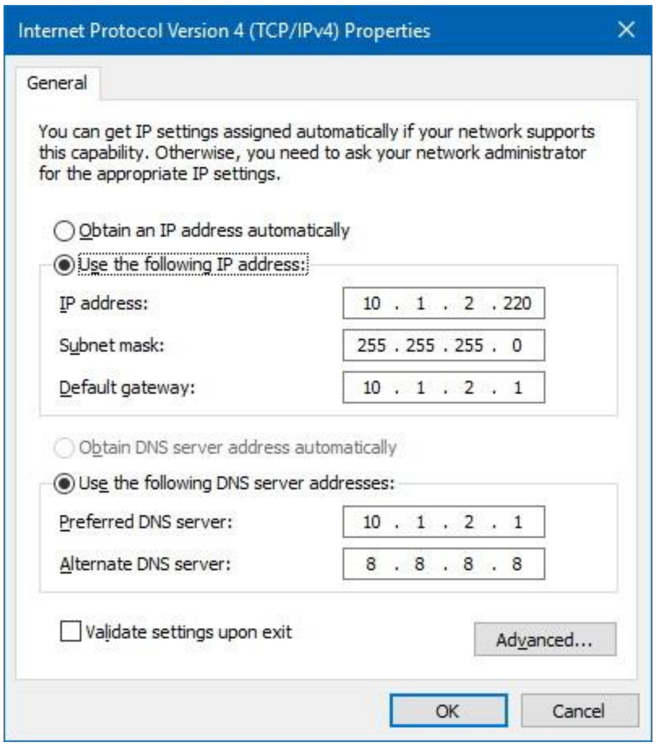
FIG. 7. TRAINING THE USERS FACE

V. IMPLEMENTATION FLOW

Follow the below steps to install our application in the labs.

A. *Basic Setup:*

1) First we need to create a network to facilitate the process of client server communication, if you are new to subnetting simply follow this tutorial [3]. So you can take any network address of your choice, we have taken 172.16.6.0/21. Please check out [4] to know how to assign a static IP to a host in a network. The Master system communicates with all the systems using IP addresses, so we have to ensure that each host has a unique static IP.


Internet Protocol Version 4 (TCP/IPv4) Properties

General

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

- Obtain an IP address automatically
- [✓ Use the following IP address:]

IP address: 10 . 1 . 2 . 220
Subnet mask: 255 . 255 . 255 . 0
Default gateway: 10 . 1 . 2 . 1

- Obtain DNS server address automatically
- [✓ Use the following DNS server addresses:]

Preferred DNS server: 10 . 1 . 2 . 1
Alternate DNS server: 8 . 8 . 8 . 8

Validate settings upon exit

FIG 8. STATIC IP CONFIGURATION

2) Check whether you are able to ping from all the clients to server and from server to all the clients, if the pings were unsuccessful try once again by turning off the firewalls.

B. *Server Side Configuration :*

1) To set up the server, first install xampp server on master systems and import the database lab.sql using PhpMyAdmin [5]. Now we need to give access to all the clients to access the database.

a) In the Xampp Control panel → config→ httpd-xamp.conf make the following changes.

```
<Directory "C:/xampp/phpMyAdmin">
    AllowOverride AuthConfig
    **Require local**   Replace with   **Require all granted**
    ErrorDocument 403 /error/XAMPP_FORBIDDEN.html.var
</Directory>```
```

FIG 9. CHANGES TO BE MADE IN HTTPD_XAMP.CONF

b) Now grant access to all the clients using the following command. Modify the highlighted sections accordingly.

GRANT ALL ON . to ***username***@'**IP**' IDENTIFIED BY '***put-your-password***';

c) Now copy the server.jar into the master system. Now the server system is ready.

C. *Client Side requirements:*

1) Just copy the client.jar

Any queries contact us here

VI. CIRCUIT DIAGRAM

●

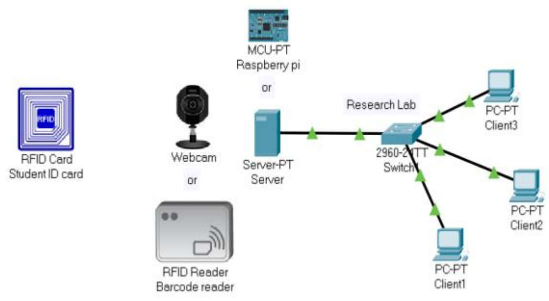
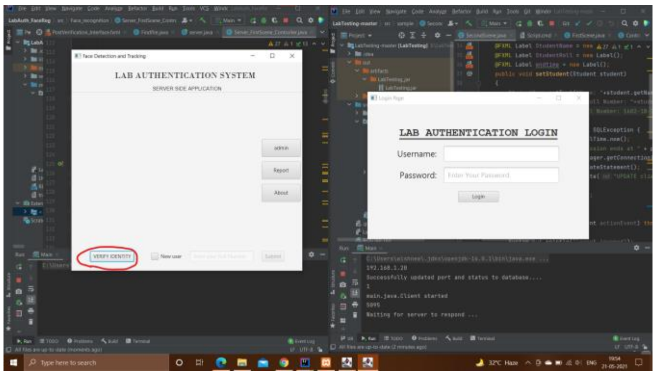


FIG. 10. CIRCUIT DIAGRAM OF THE PROJECT.

VII. FUTURE SCOPE

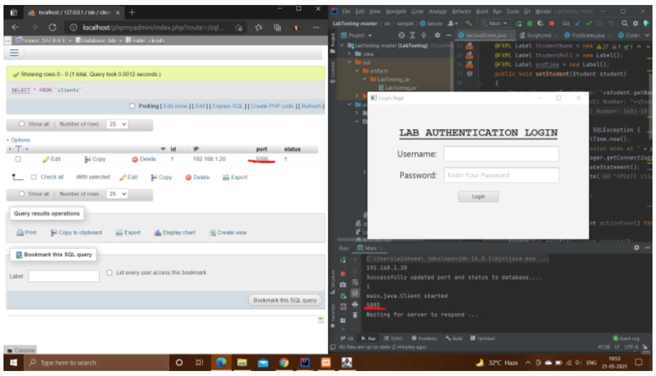
- Allocating the system to users according to their requirement.
- To develop this as complete software.
- To create a dashboard for the admin.
- Increasing the security in the client side application
- Implementing a more robust face recognition algorithm.
- Making the update process of the software dynamic.

VIII. OUTPUTS



```
192.168.1.20
Successfully updated port and status to database....
1
main.java.Client started
5095
Waiting for server to respond ...
```

FIG. 11. VERIFYING IDENTIFY ON SERVER SIDE.



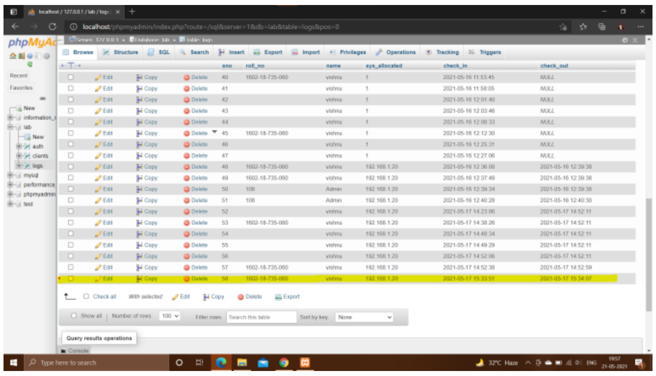
Showing rows 0 - 0 (1 total, Query took 0.0012 seconds.)

```
SELECT * FROM `clients`
```

id	IP	port	status
1	192.168.1.20	5095	1

```
192.168.1.20
Successfully updated port and status to database....
1
main.java.Client started
5095
Waiting for server to respond ...
```

FIG. 12. CLIENT WAITING ON A RANDOM PORT



localhost/phpmyadmin/index.php?route=/sql&server=1&db=lab&table=logs&pos=0

sno	roll_no	name	sys_allocated	check_in	check_out
40	1602-18-735-060	vishnu	1	2021-05-16 11:53:45	NULL
41		vishnu	1	2021-05-16 11:58:05	NULL
42		vishnu	1	2021-05-16 12:01:40	NULL
43		vishnu	1	2021-05-16 12:03:46	NULL
44		vishnu	1	2021-05-16 12:08:33	NULL
45	1602-18-735-060	vishnu	1	2021-05-16 12:12:30	NULL
46		vishnu	1	2021-05-16 12:25:31	NULL
47		vishnu	1	2021-05-16 12:27:06	NULL
48	1602-18-735-060	vishnu	192.168.1.20	2021-05-16 12:36:08	2021-05-16 12:39:38
49	1602-18-735-060	vishnu	192.168.1.20	2021-05-16 12:37:49	2021-05-16 12:39:38
50	108	Admin	192.168.1.20	2021-05-16 12:39:34	2021-05-16 12:39:38
51	108	Admin	192.168.1.20	2021-05-16 12:40:28	2021-05-16 12:40:30
52		vishnu	192.168.1.20	2021-05-17 14:23:06	2021-05-17 14:52:11
53	1602-18-735-060	vishnu	192.168.1.20	2021-05-17 14:38:26	2021-05-17 14:52:11
54		vishnu	192.168.1.20	2021-05-17 14:48:34	2021-05-17 14:52:11
55		vishnu	192.168.1.20	2021-05-17 14:49:29	2021-05-17 14:52:11
56		vishnu	192.168.1.20	2021-05-17 14:52:06	2021-05-17 14:52:11
57	1602-18-735-060	vishnu	192.168.1.20	2021-05-17 14:52:38	2021-05-17 14:52:59
58	1602-18-735-060	vishnu	192.168.1.20	2021-05-17 15:33:51	2021-05-17 15:34:07

FIG. 13. LOG SAVED INTO DATABASE.

Tools used:

1. Xampp Server - 3.2.4
2. packet tracer - 8.0.0.0212
3. openjfx-15.0.1_windows-x64_bin-sdk
4. javafx-sdk-15.0.1
5. Jna -
6. mysql-connector-8.0.23

IX. CONCLUSIONS

Hence, the system has been successfully tested by implementing the server-client model in a single PC. Our main goal lies in implementing the system in the Project Lab ECE. Lab Authentication system successfully makes it easier to manage the project's lab. So, as to know who is responsible for any discrepancy in the lab.

The Face recognition system makes the process of identification more secure and easy. On a whole this system serves the purpose, it was intended to be.

● X. LIMITATIONS

1. The Face Recognition model used here, needs a controlled environment to work at its maximum efficiency.
2. There are Chances of mocking the face recognition system by using individuals' photographs.
3. Client side application doesn't hold complete privileges for locking the system in Kiosk mode, User can log in to the PC using some specific shortcuts.
4. The updating process of the software is currently static, which means it needs to be reinstalled whenever new updates are patched.

ACKNOWLEDGMENT

We would like to thank Vasavi College of Engineering for providing us the facilities and opportunity for building this project. Our Special thanks to V. Krishna Mohan sir(Assistant Professor ECE) and Gupta Sir(Lab Assistant) for helping us in developing the project.

REFERENCES

[1] https://www.geeksforgeeks.org/jar-files-java/

[2] https://www.techopedia.com/definition/22193/software-development-life-cycle-sdlc

[3] https://www.computernetworkingnotes.com/ccna-study-guide/subnetting-tutorial-subnetting-explained-with-examples.html

[4] https://pureinfotech.com/set-static-ip-address-windows-10/

[5] https://help.dreamhost.com/hc/en-us/articles/214395768-phpMyAdmin-How-to-import-or-restore-a-database-or-table

[6] https://www.geeksforgeeks.org/socket-programming-in-java/

[7] M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.

[8] https://towardsdatascience.com/face-recognition-how-lbph-works-90ec258c3d6

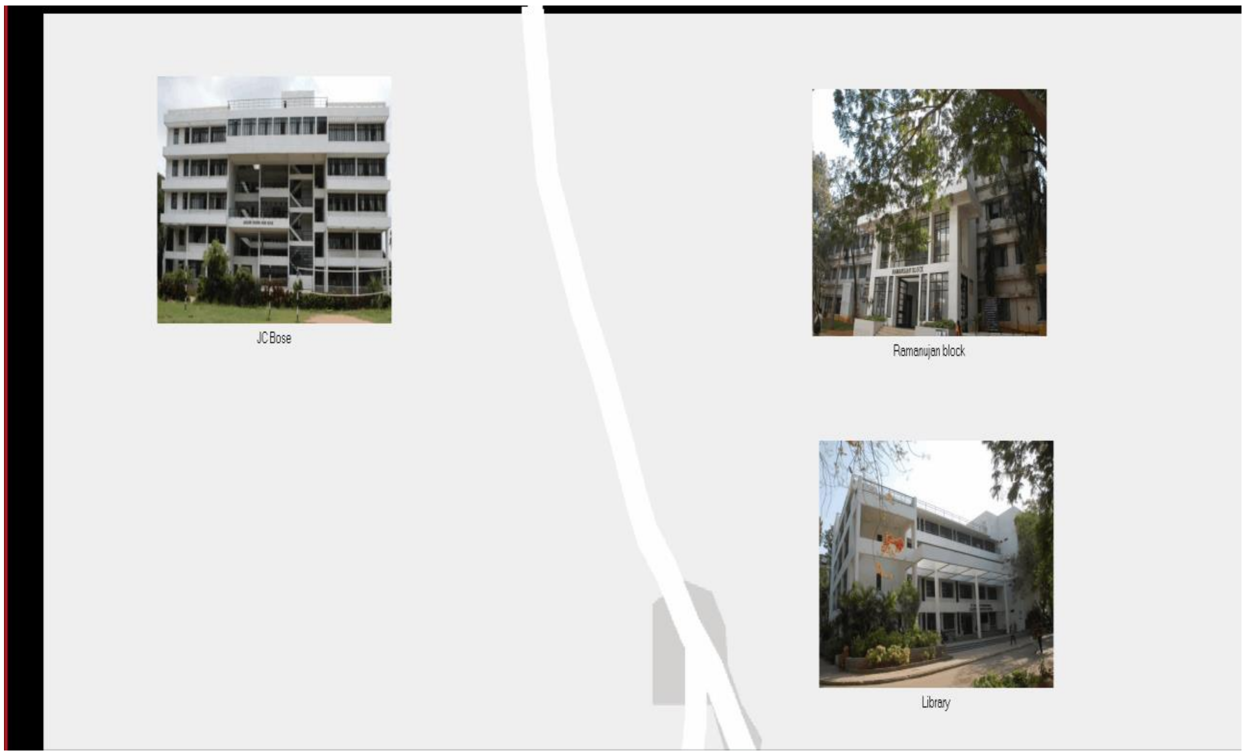


FIG 14. PHYSICAL VIEW

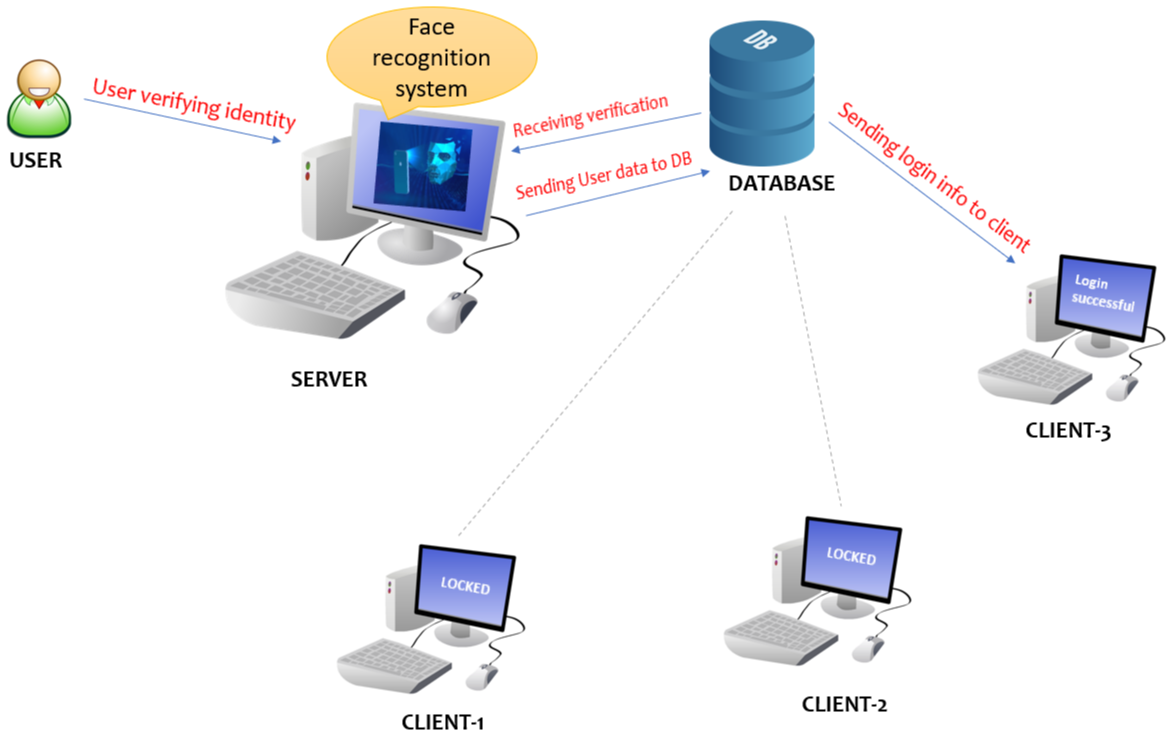


FIG 8. SYSTEM WORKING MECHANISM