A Project Report on

Student Attendance Management System

Submitted in partial fulfilment for the completion of course

Programming Java(SWE1007)

in

M.Tech (SE)
By

VAISHALI.S
18MIS0082

Submitted to
Dr. B. PRABADEVI
Assistant Professor (Sr.)
SITE

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

June 2021

Abstract

The attendance management system’s scheduling features would help teachers and staff in marking the attendance of students, in consideration with various aspects such as daily, session-based, or course-based requirements.

1. Introduction
    1.1. Motivation

        The Motivation of the project is to generate and scan barcode through the mobile application and store it in database in the application of java.

    1.2. Aim of the proposed Work

        The aim of the proposed work is to identify the number of classes students have attended also helpful for the teacher to view the number absentees in a particular course. It also makes students easier to scan and upload the barcode in the application instead of entering the value which brings even more security.

    1.3. Objective(s) of the proposed work

        The objective of the project is to make students to submit their attendance easier checking for the criteria, keeping a track of their attendance. Lao helpful for teachers to view student’s attendance lists and number students attending it. The project also helps in identifying and assigning individual child with the help of unique barcode.

    1.4. Report Organization

2. Analysis & Design of Proposed Work

2.1. Problem Statement

It is very hard for teachers to enter or update attendance of students manually also missing of data is hard to retrieve unless it is saved or backed up .The following ideas have been proposed in order to generate and use barcode to register .

Attendance Management System is developed for daily student attendance in college. It facilitates to access the attendance information of a particular student in a particular class that can be either viewed by the faculty or the student. This system will also help in evaluating attendance eligibility criteria of a student. The system will be able to produce the students' attendance report thus reducing the need for manual labour which leads to human errors and time consuming. The student can only view the attendance record on a daily basis. The faculty can view the attendance record.

The attendance is taken through the barcode image where the barcode is shown and scanned and then entered into the database. Every student is provided with an id card containing a unique barcode. Each barcode represents a unique id of students. Students just have to scan their cards using barcode scanner and the system notes down their attendance as per dates. It then stores all the student’s attendance records and generates list of present students in class.

2.2. Stakeholder identification

Faculty

Admin

Students

2.3. Classes Identification using CRC and Noun Phrase Approach

- Netbeans
    - Login.class

      Login constructor
    - NewUser.class

      NewUser constructor
    - Main.class

      Main constructor
    - History.class

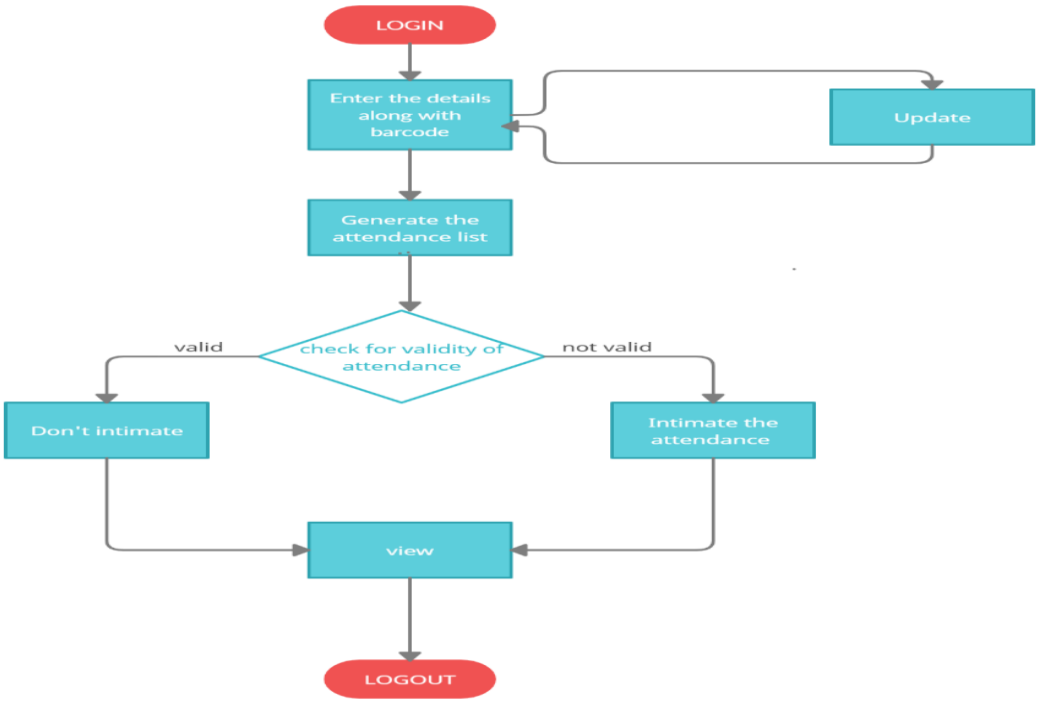
      History constructor
    - Teacher.class

      Teacher constructor
- Android studio
    - MainActivty.class
    - Background.class

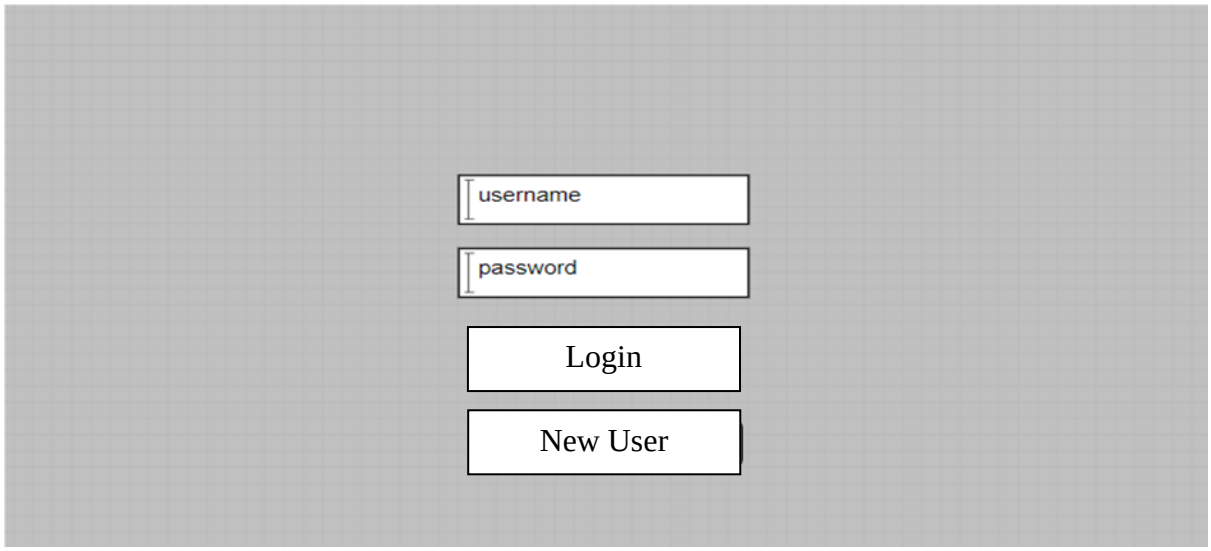
2.4. Gaps identified (How Proposed work differ from Existing)

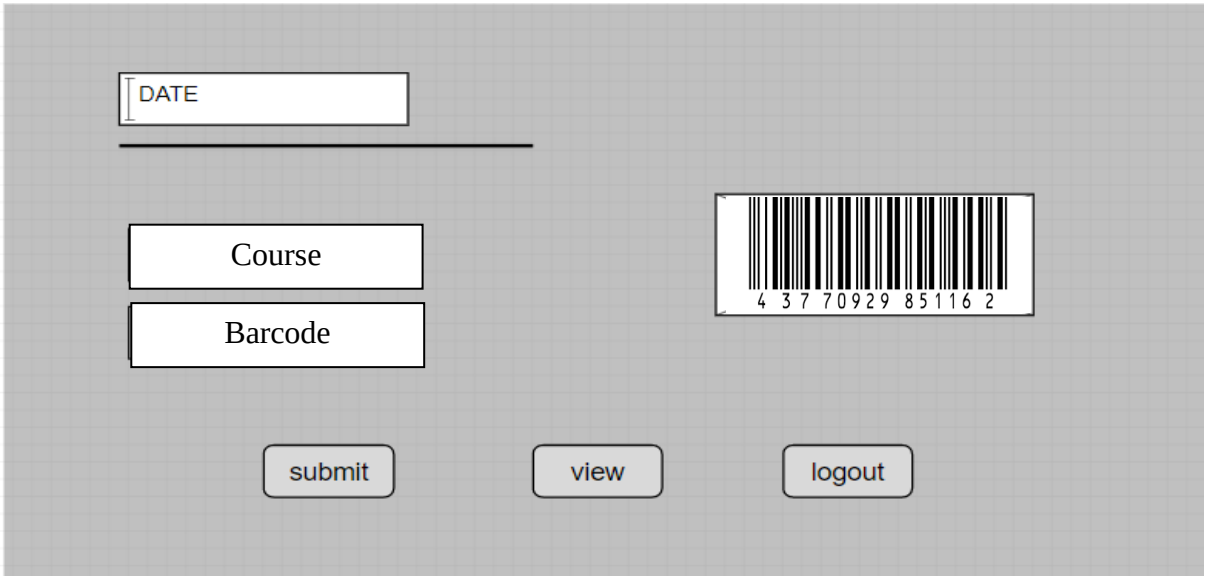
This project

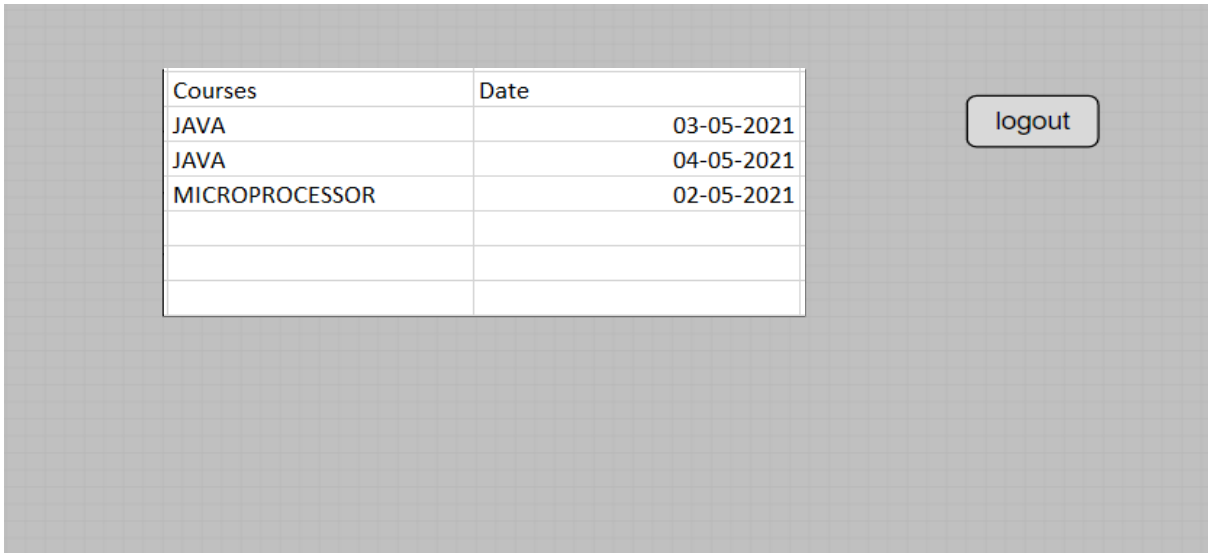
2.5. System Architecture or Flow diagram



2.6 Interfaces







2.7. Module Description

Login

Login module is validating the username and password along with the database of MySQL entered data. This is supported by the jar file for the jdbc connections

NewUser

NewUser is used to get data from user and store it in database also it generates a SMS to the registered number with the barcode with username.

Main

Main is used to register the courses with the help of barcode scanned from app and stored in database.

History

This class provides the list of courses registered for attendance with data and also the pie chart of the count of courses registered for attendance so far.

Teacher

Teacher class is used to find out the number of students attended the classes on a particular date. This class also generates an excel sheet for the teacher to vie and update and download for any other necessary purposes.

3. Implementation

3.1. Softwares used with version

Apache Netbeans IDE 12.2

Android Studio 4.2.1

Xampp 3.2.4

3.2. Java concepts used

Inheritance

Interfaces

Encapsulation

Class

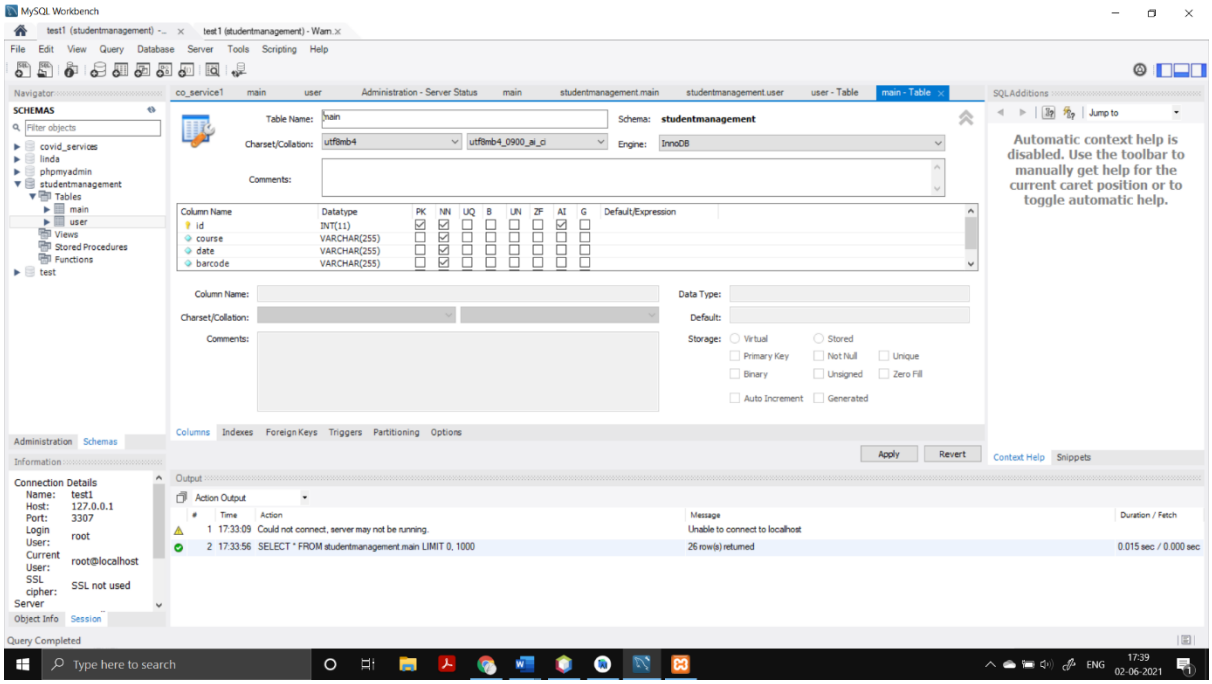
Object

Connectivity with jar files and API Key

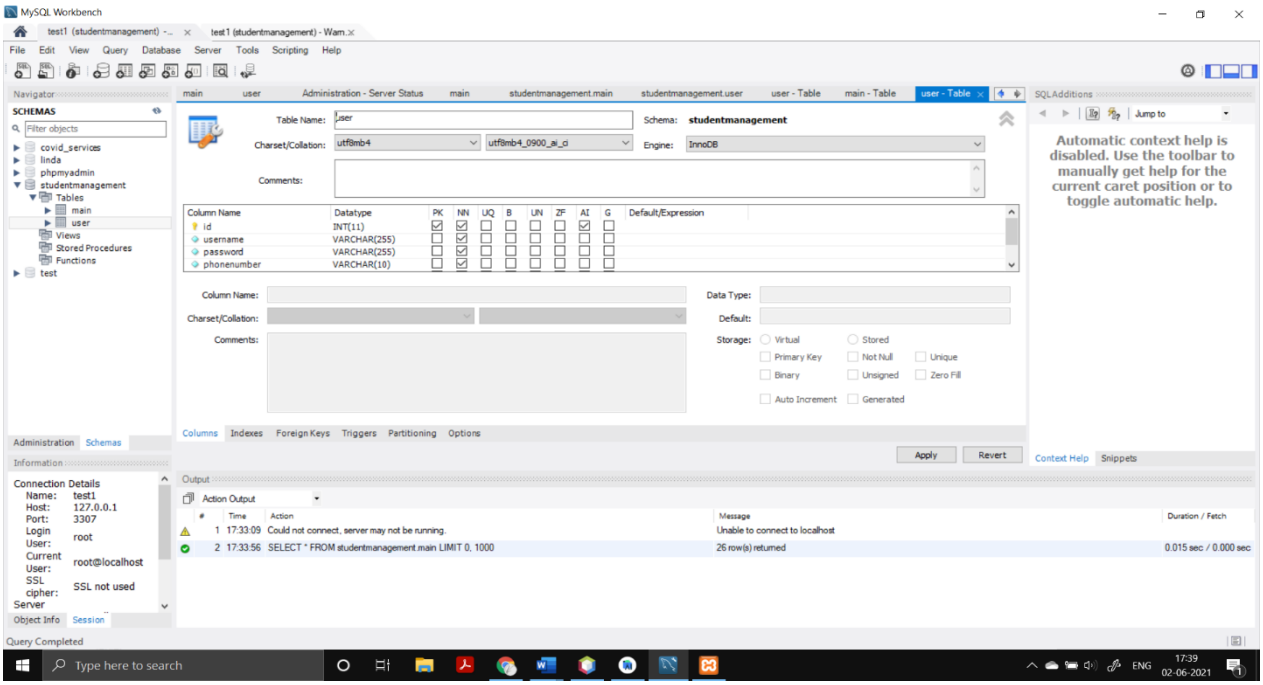
3.3. Database Design

3.3.1. Schema

Main table schema

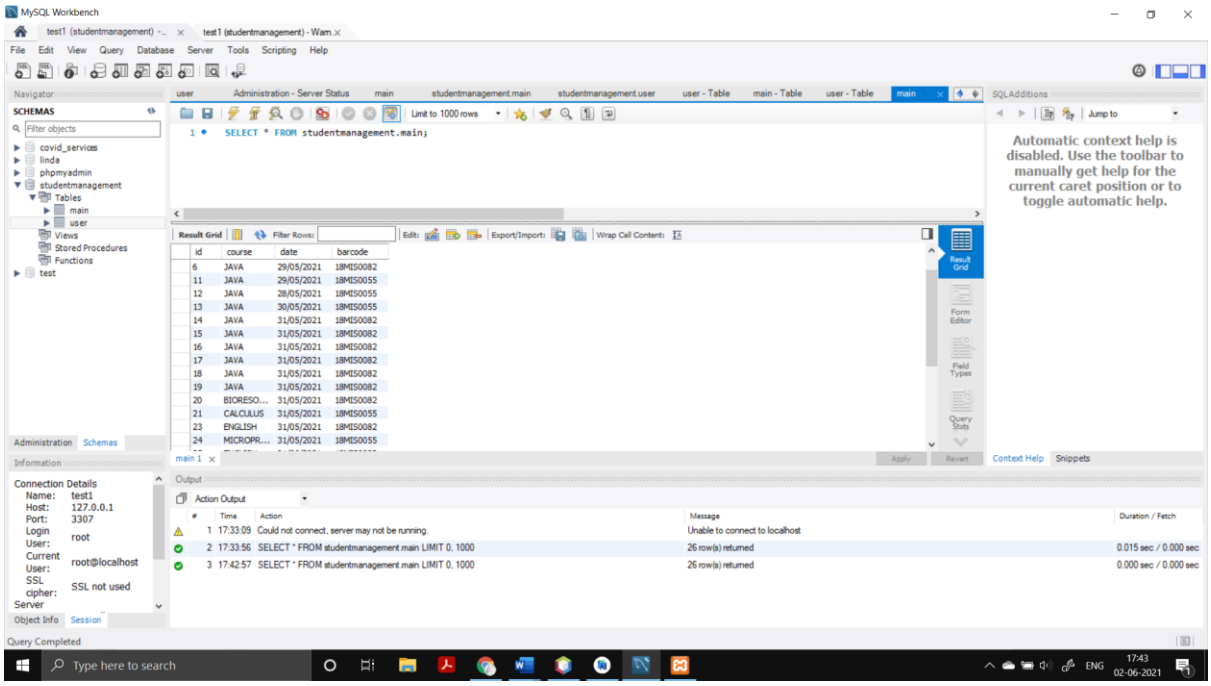


User table schema

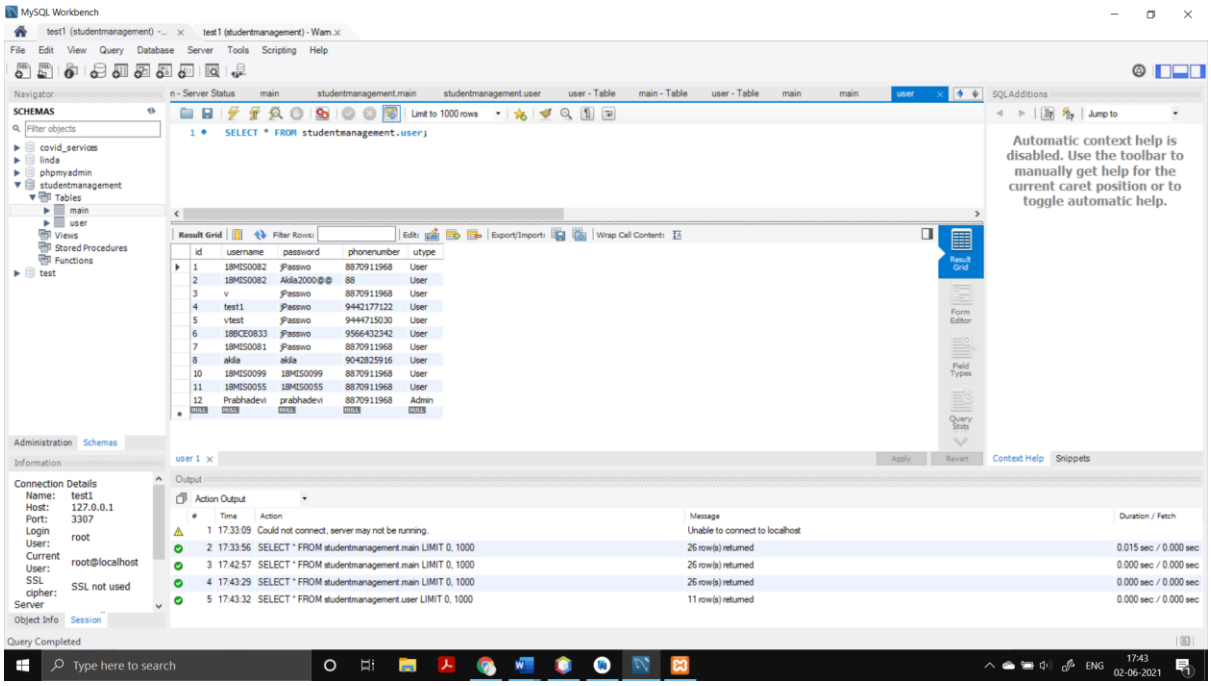


3.3.2. Tables with values

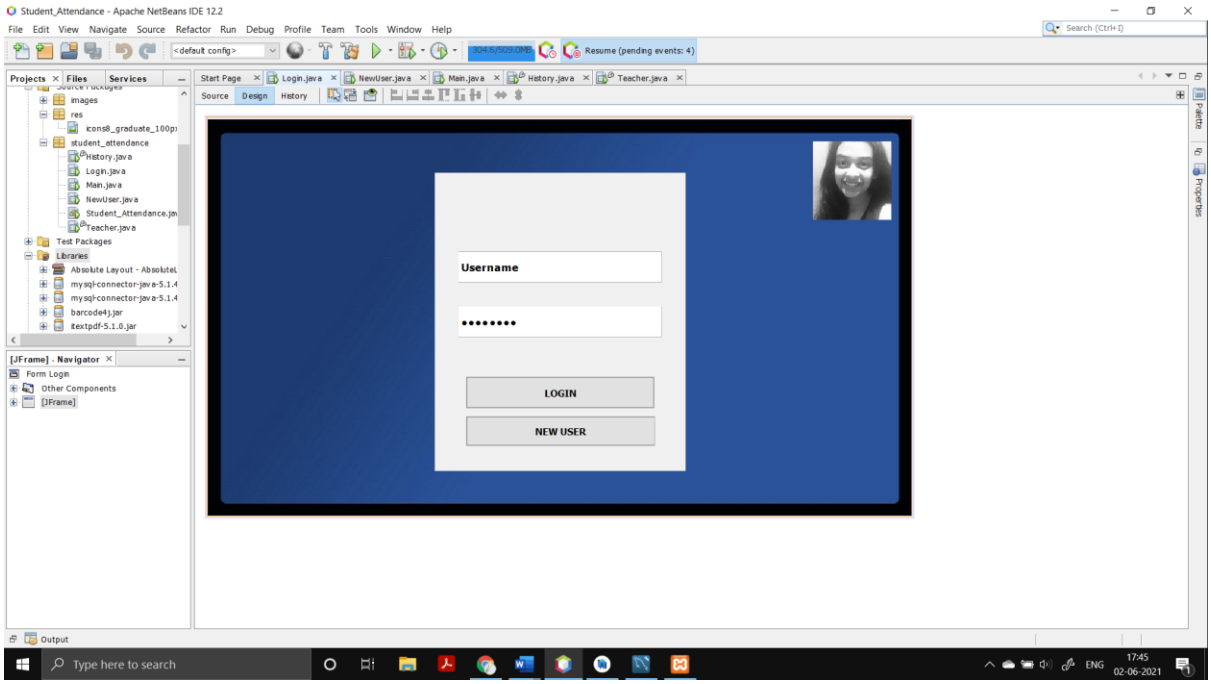
Main table

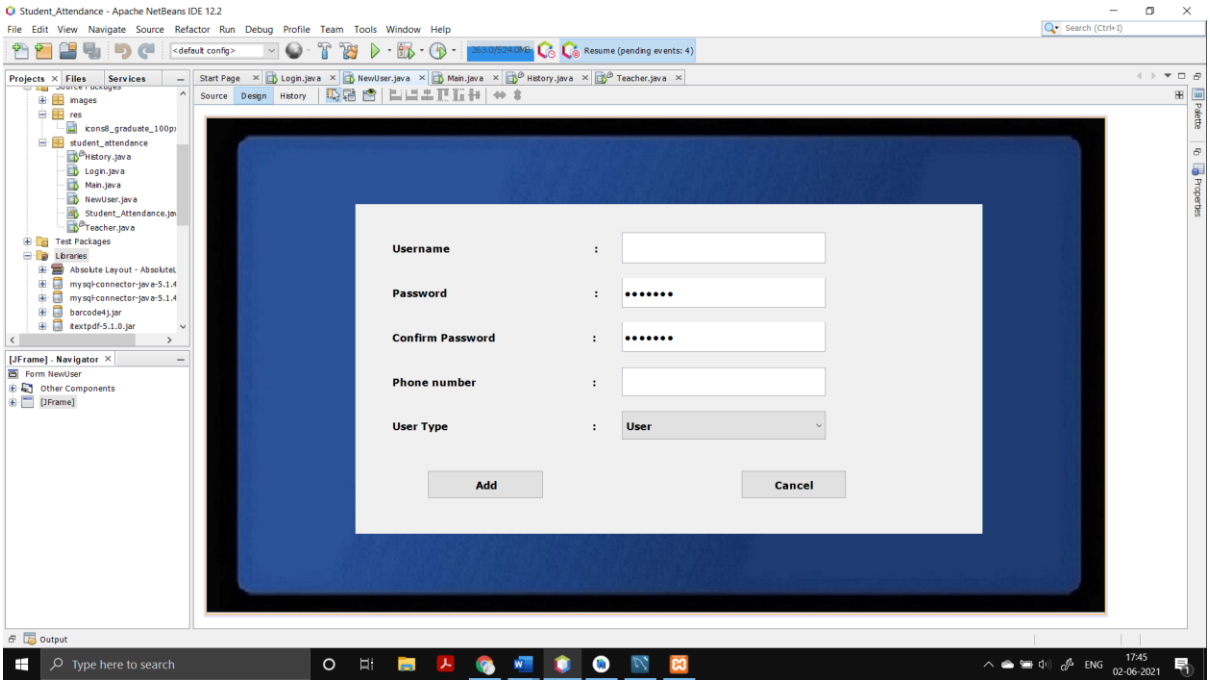


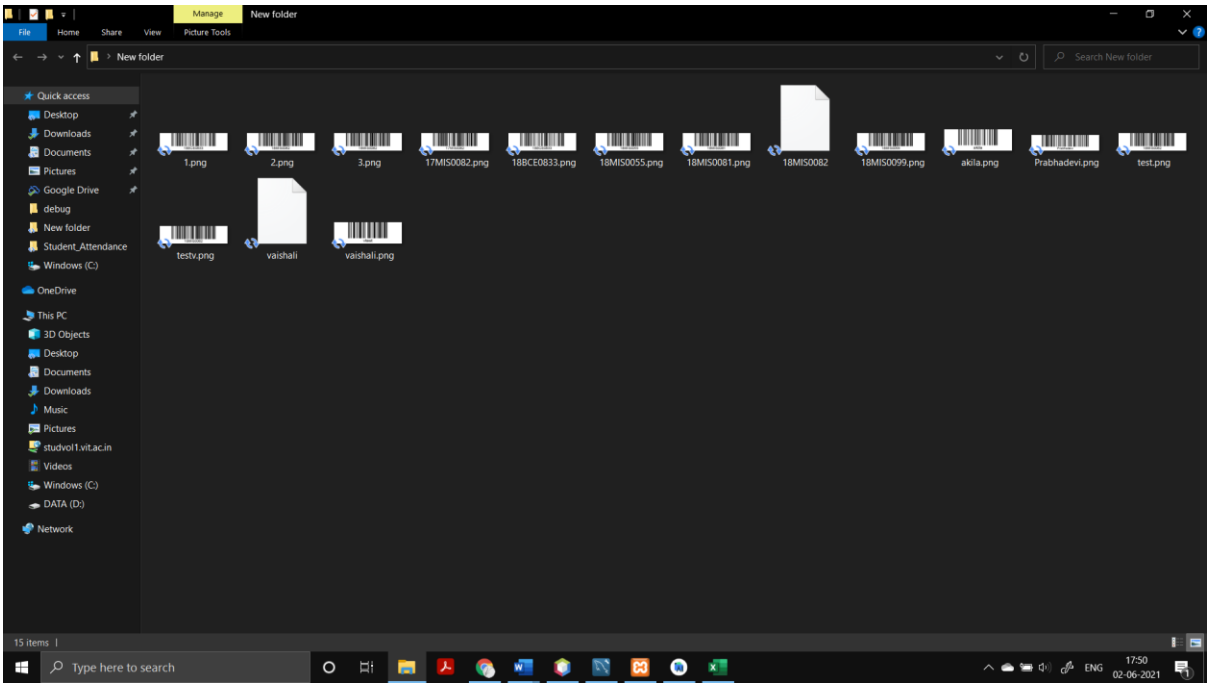
User table

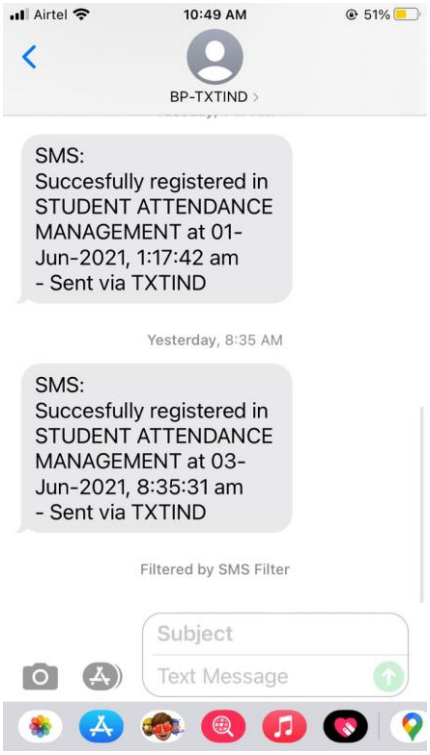


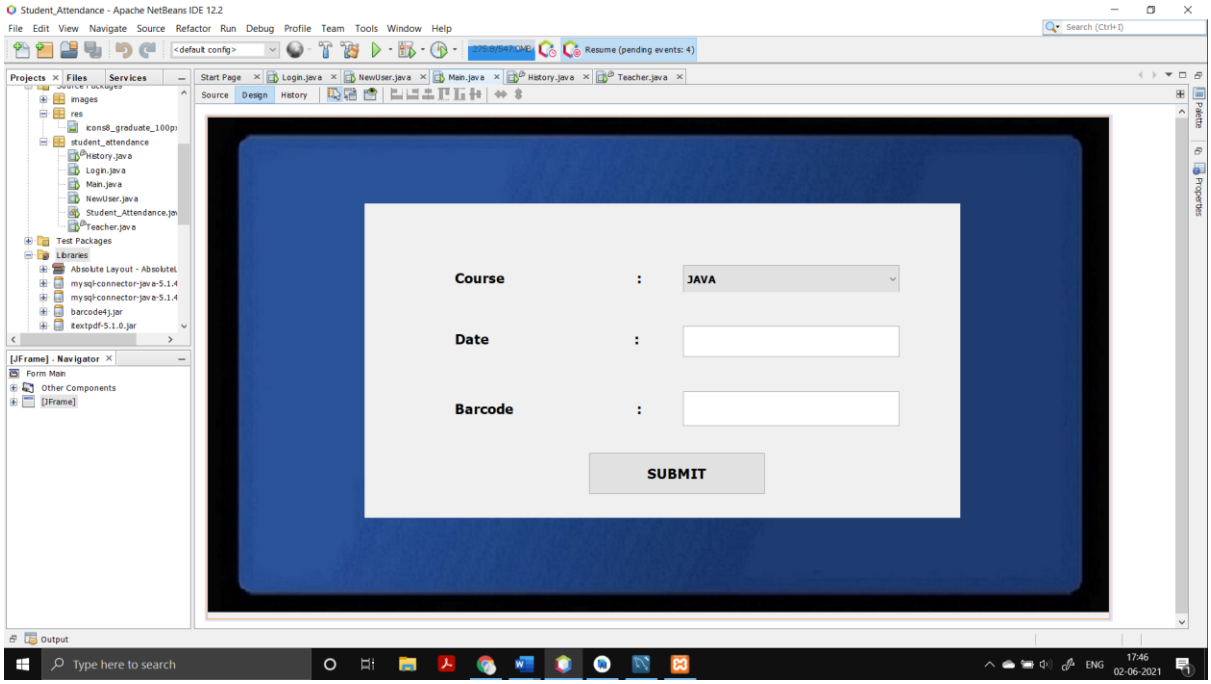
3.4. Screenshots of the system

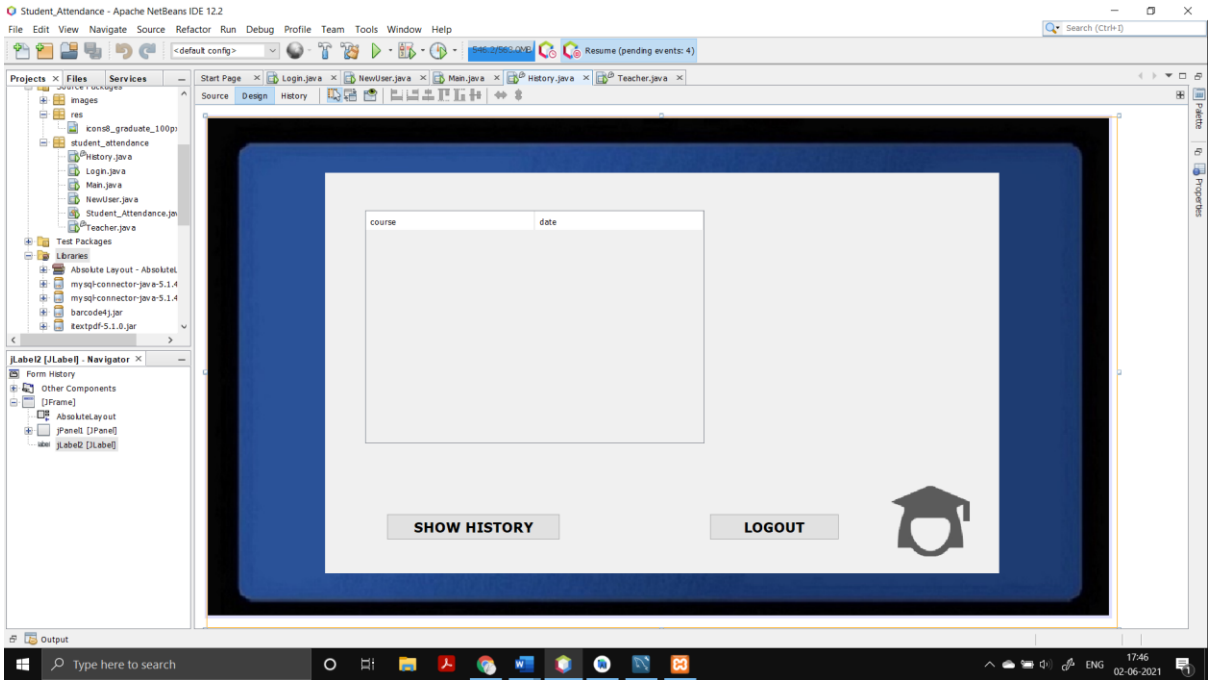


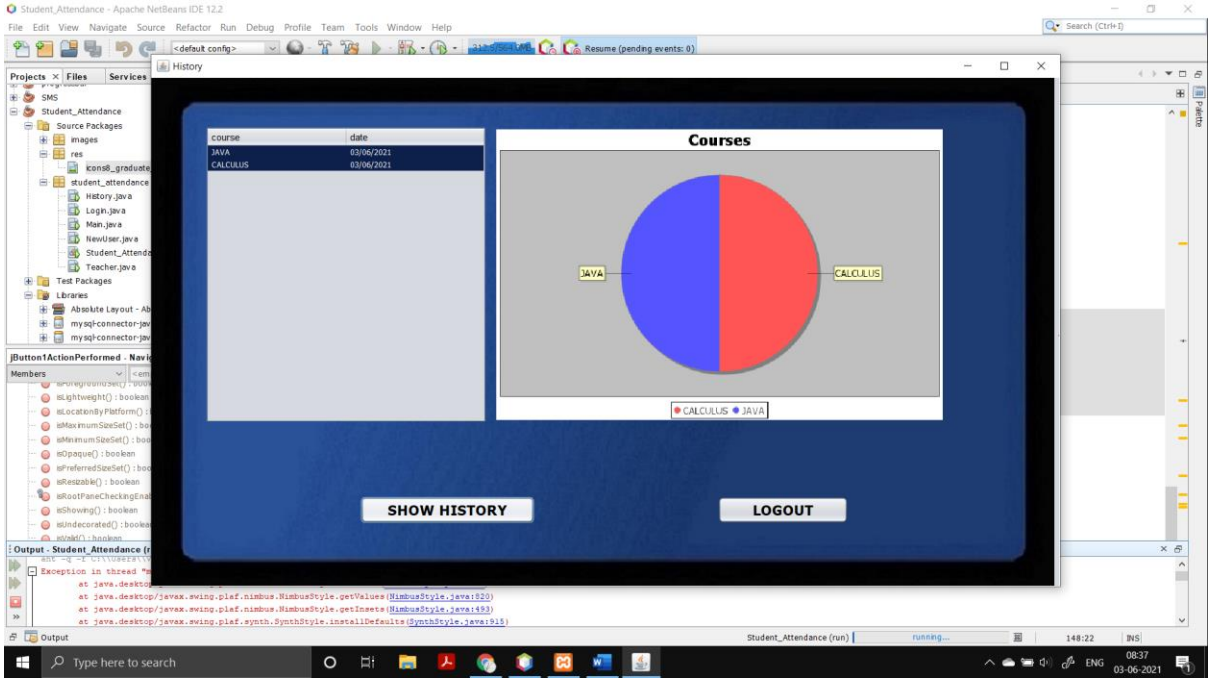

Student_Attendance - Apache NetBeans IDE 12.2
Username :
Password :
Confirm Password :
Phone number :
User Type :
User
Add
Cancel

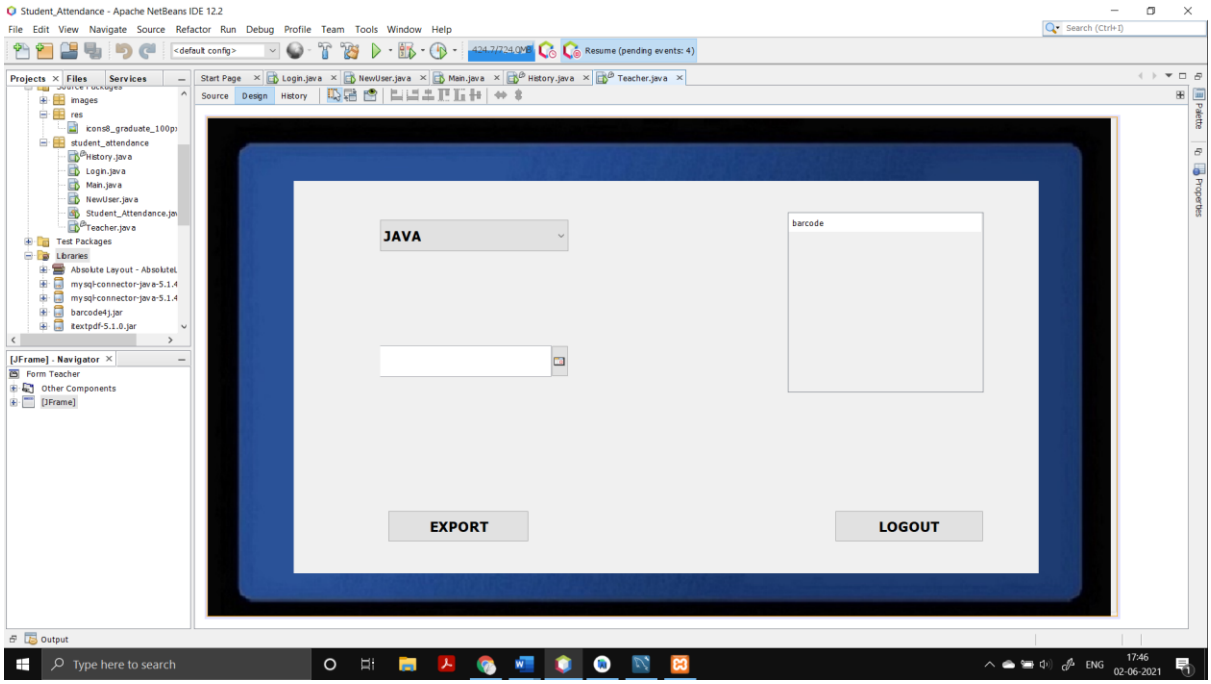

New folder
1.png
2.png
3.png
17MIS0082.png
18BCE0833.png
18MIS0055.png
18MIS0081.png
18MIS0082
18MIS0099.png
akila.png
Prabhadevi.png
test.png
testv.png
vaishali
vaishali.png

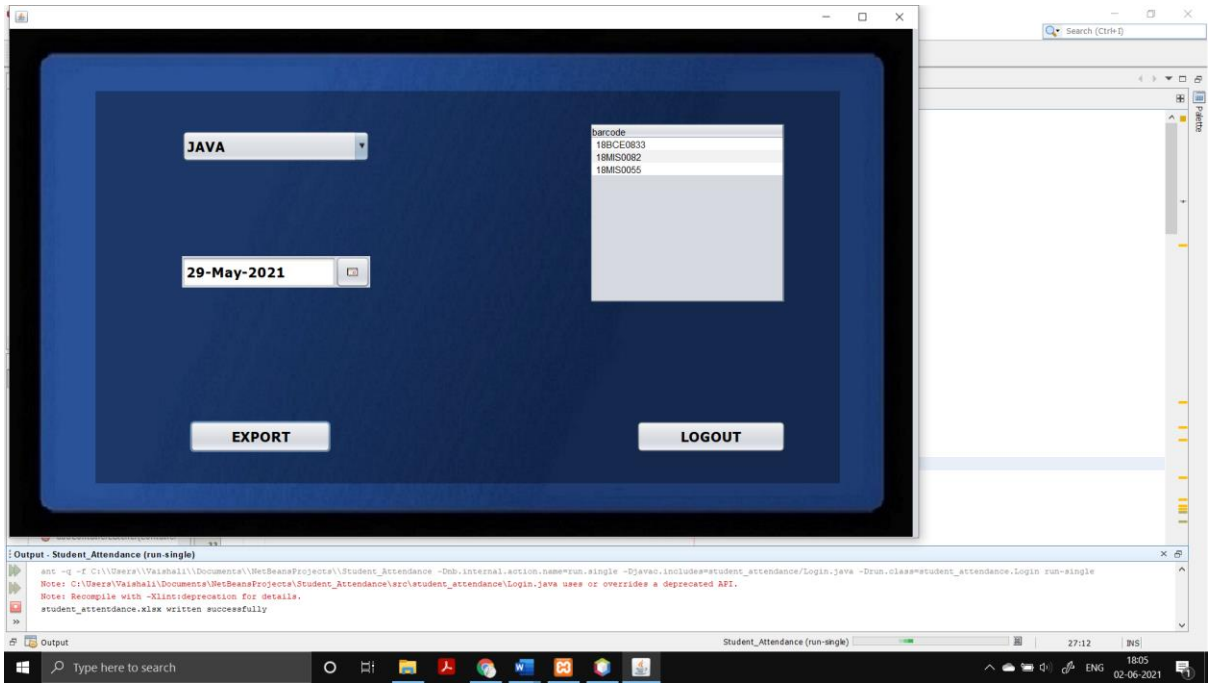

Airtel
10:49 AM
51%
BP-TXTIND
SMS:
Succesfully registered in STUDENT ATTENDANCE MANAGEMENT at 01-Jun-2021, 1:17:42 am - Sent via TXTIND
Yesterday, 8:35 AM
SMS:
Succesfully registered in STUDENT ATTENDANCE MANAGEMENT at 03-Jun-2021, 8:35:31 am - Sent via TXTIND
Filtered by SMS Filter
Subject
Text Message

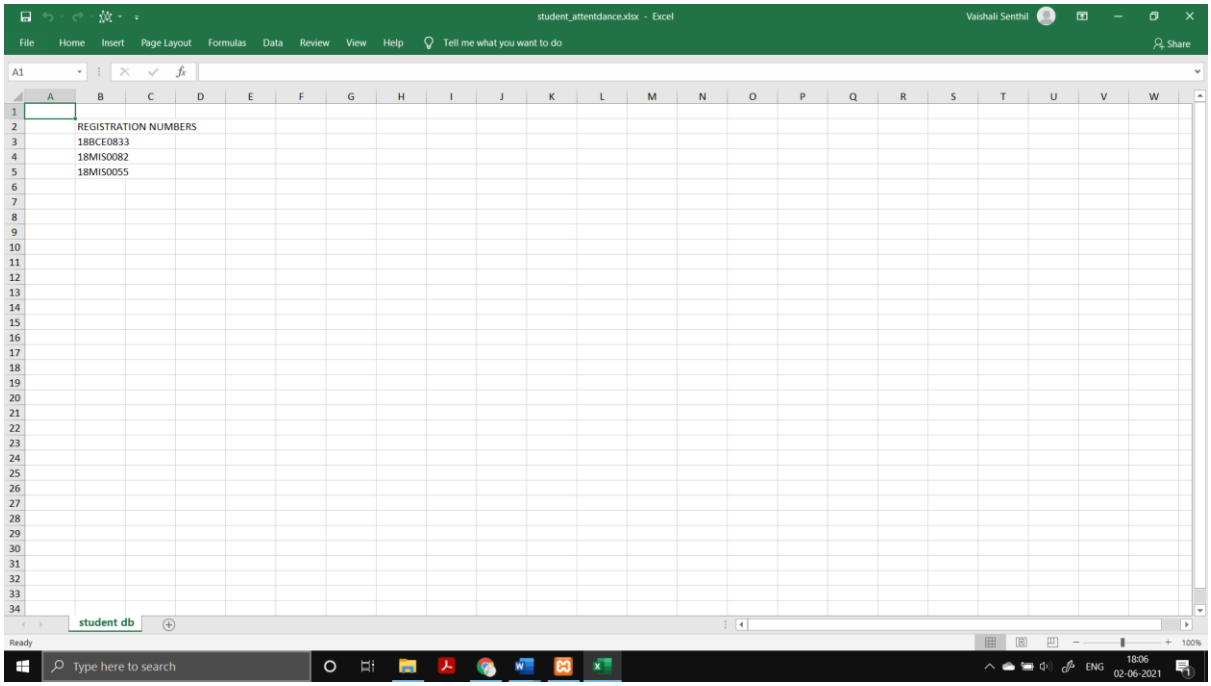

Student_Attendance - Apache NetBeans IDE 12.2
Course
:
JAVA
Date
:
Barcode
:
SUBMIT
Type here to search
ENG
17:46
02-06-2021

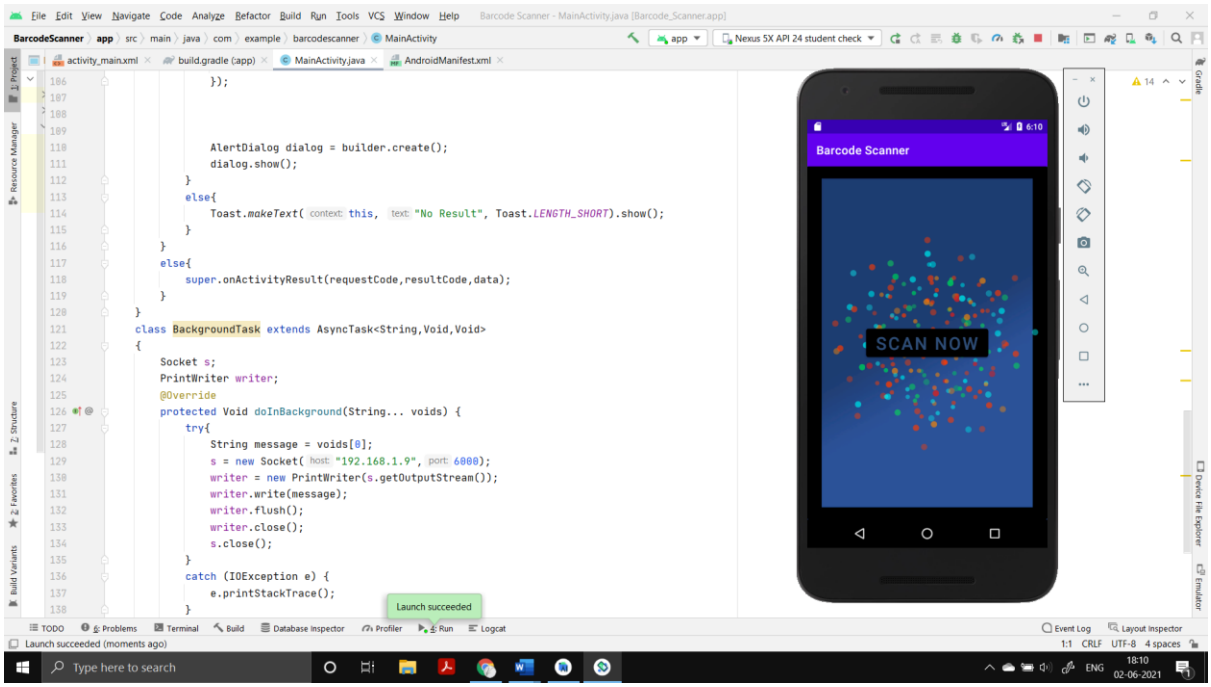

Student_Attendance - Apache NetBeans IDE 12.2
course
date
SHOW HISTORY
LOGOUT

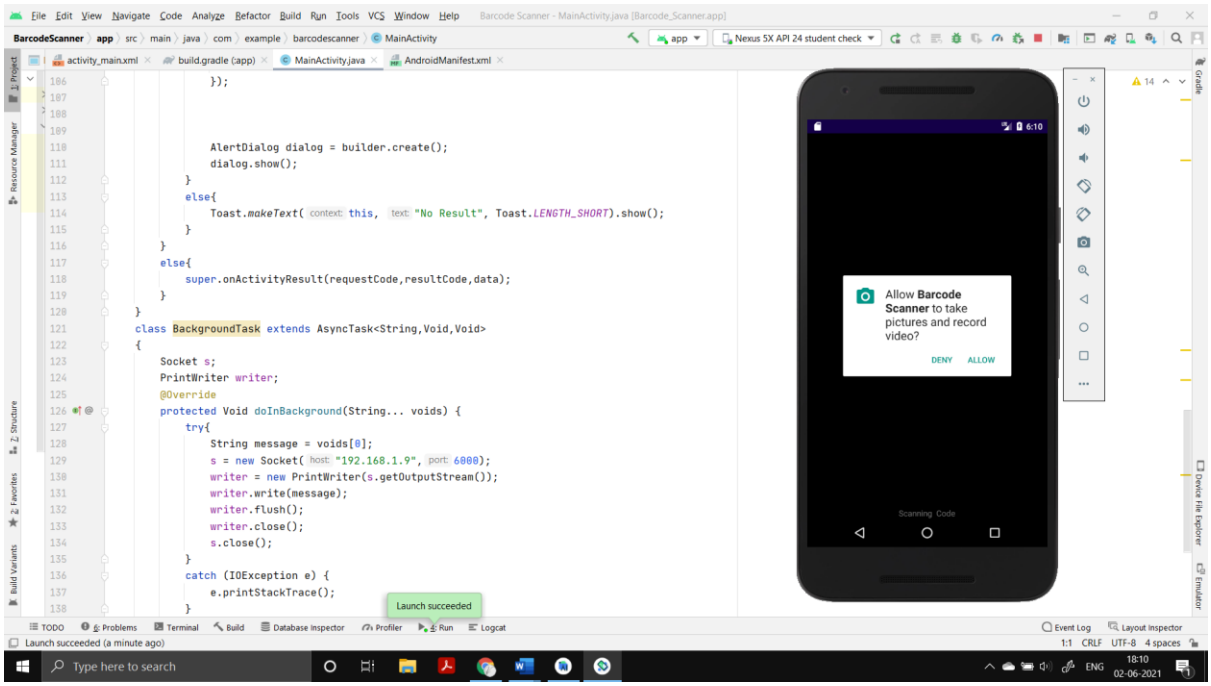

Student_Attendance - Apache NetBeans IDE 12.2
History
course
date
JAVA
03/06/2021
CALCULUS
03/06/2021
Courses
JAVA
CALCULUS
SHOW HISTORY
LOGOUT

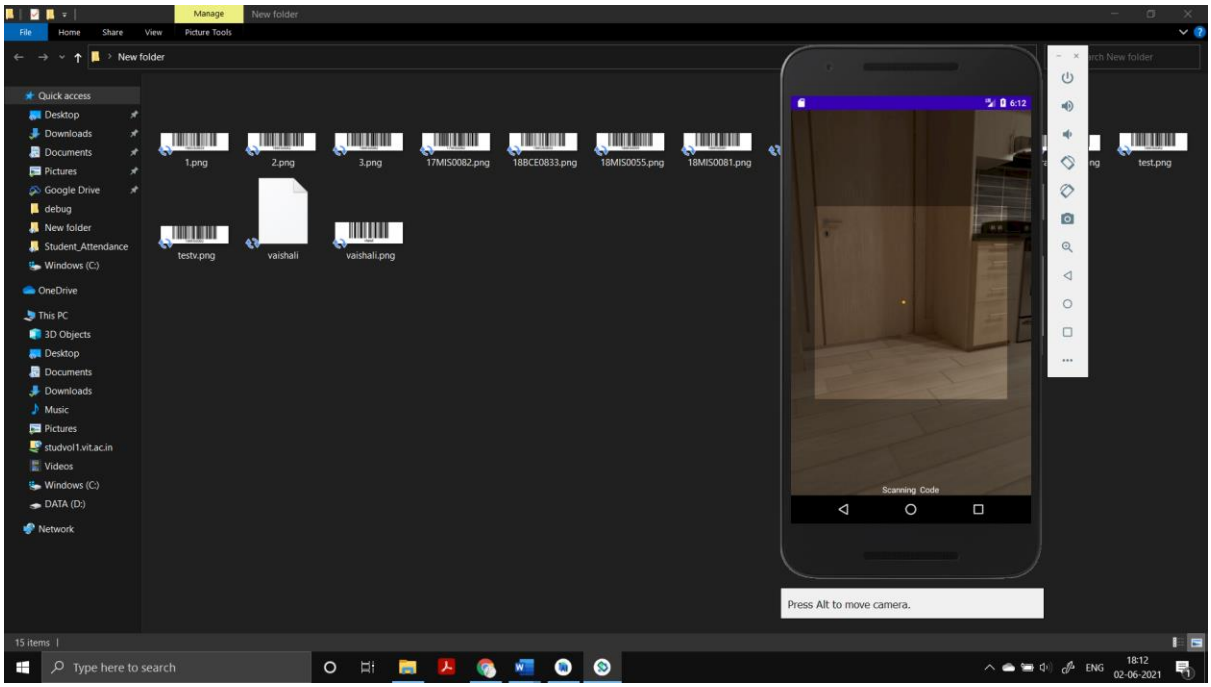

Student_Attendance - Apache NetBeans IDE 12.2
JAVA
barcode
EXPORT
LOGOUT


JAVA
barcode
18BCE0833
18MIS0082
18MIS0055
29-May-2021
EXPORT
LOGOUT
Output - Student_Attendance (run-single)
student_attentdance.xlsx written successfully

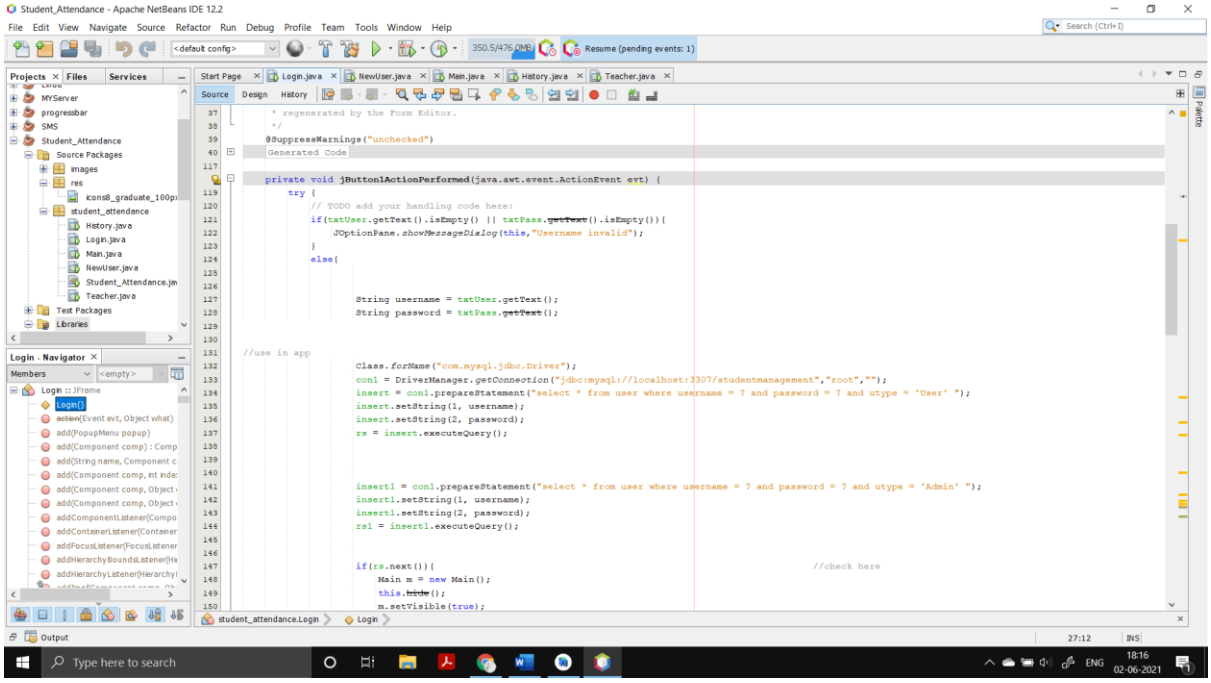

student_attentdance.xlsx - Excel
Vaishali Senthil
File Home Insert Page Layout Formulas Data Review View Help Tell me what you want to do
Share
A1
REGISTRATION NUMBERS
18BCE0833
18MIS0082
18MIS0055
student db
Ready
Type here to search
ENG
18:06
02-06-2021

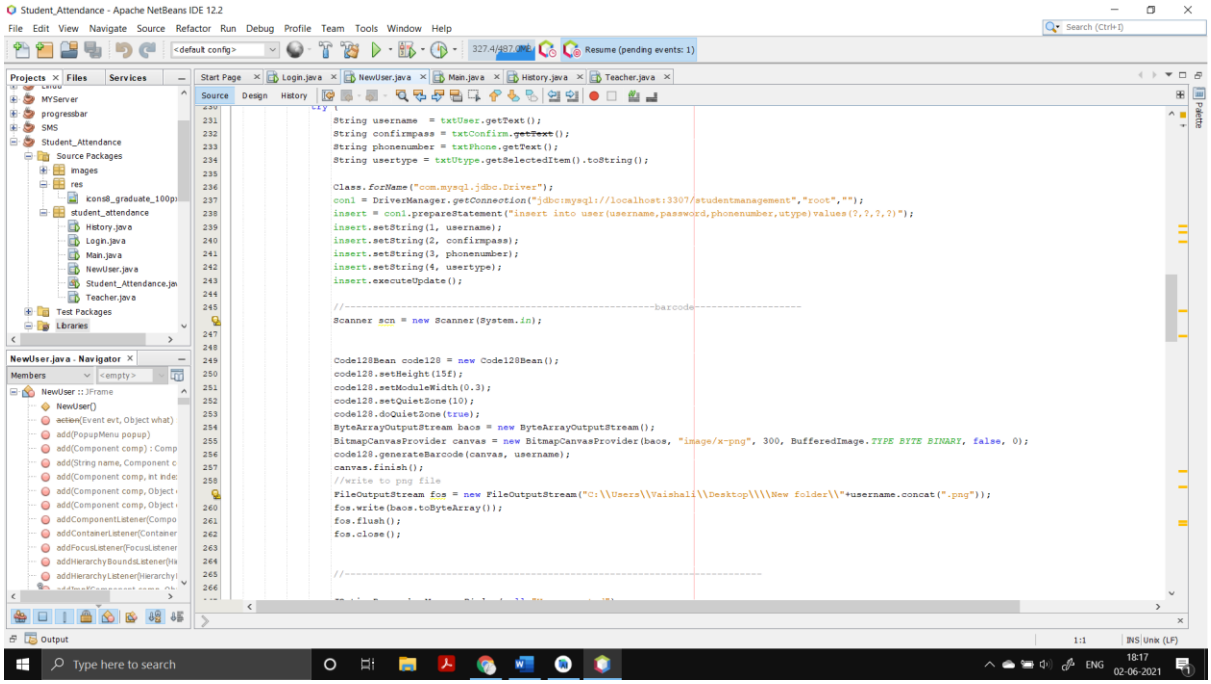

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Barcode Scanner - MainActivity.java [Barcode_Scanner.app]
BarcodeScanner app src main java com example barcodescanner MainActivity
activity_main.xml build.gradle (:app) MainActivity.java AndroidManifest.xml
app
Nexus 5X API 24 student check
AlertDialog dialog = builder.create();
dialog.show();
Toast.makeText(context: this, text: "No Result", Toast.LENGTH_SHORT).show();
super.onActivityResult(requestCode,resultCode,data);
class BackgroundTask extends AsyncTask<String,Void,Void>
Socket s;
PrintWriter writer;
@Override
protected Void doInBackground(String... voids) {
String message = voids[0];
s = new Socket(host: "192.168.1.9", port: 6000);
writer = new PrintWriter(s.getOutputStream());
writer.write(message);
writer.flush();
writer.close();
s.close();
catch (IOException e) {
e.printStackTrace();
Launch succeeded
Barcode Scanner
SCAN NOW
TODO Problems Terminal Build Database Inspector Profiler Run Logcat
Event Log Layout Inspector
Launch succeeded (moments ago)
1:1 CRLF UTF-8 4 spaces
Type here to search
ENG
18:10
02-06-2021

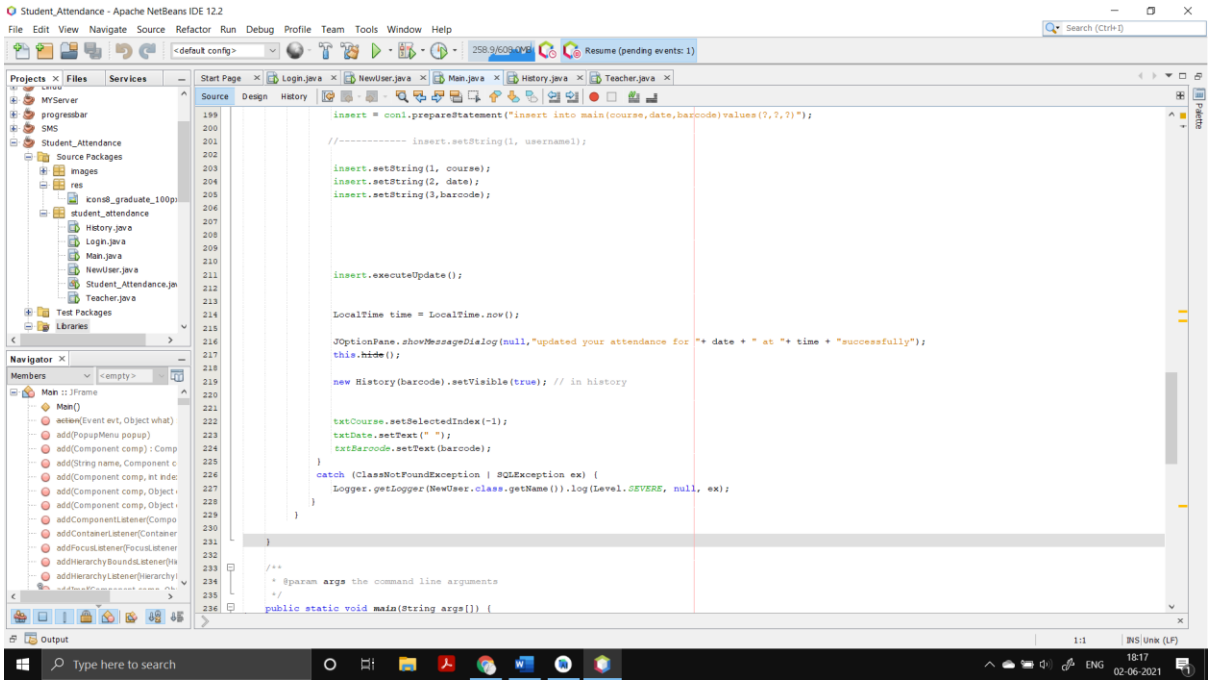

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Barcode Scanner - MainActivity.java [Barcode_Scanner.app]
BarcodeScanner app src main java com example barcodescanner MainActivity
app
Nexus 5X API 24 student check
activity_main.xml
build.gradle (:app)
MainActivity.java
AndroidManifest.xml
});
AlertDialog dialog = builder.create();
dialog.show();
}
else{
Toast.makeText(context: this, text: "No Result", Toast.LENGTH_SHORT).show();
}
}
else{
super.onActivityResult(requestCode,resultCode,data);
}
}
class BackgroundTask extends AsyncTask<String,Void,Void>
{
Socket s;
PrintWriter writer;
@Override
protected Void doInBackground(String... voids) {
try{
String message = voids[0];
s = new Socket(host: "192.168.1.9", port: 6000);
writer = new PrintWriter(s.getOutputStream());
writer.write(message);
writer.flush();
writer.close();
s.close();
}
catch (IOException e) {
e.printStackTrace();
}
Launch succeeded
Allow Barcode Scanner to take pictures and record video?
DENY
ALLOW
Scanning Code
TODO
Problems
Terminal
Build
Database Inspector
Profiler
Run
Logcat
Event Log
Layout Inspector
Launch succeeded (a minute ago)
Type here to search
ENG
18:10
02-06-2021

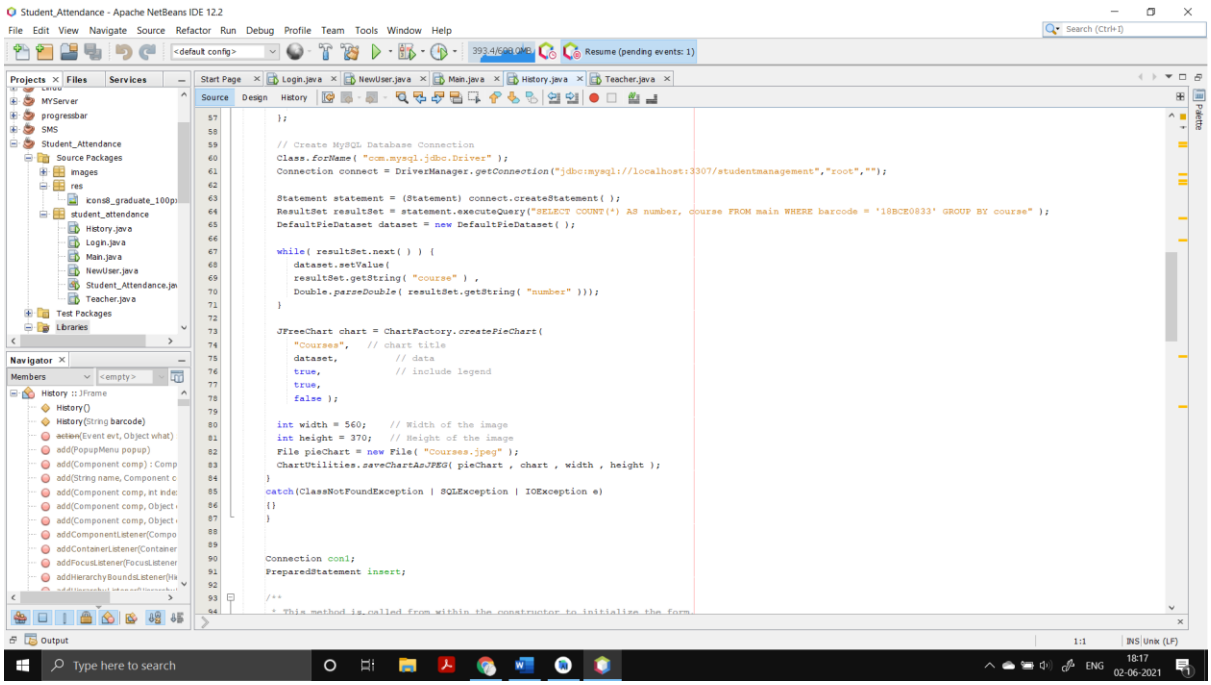

Manage
New folder
File
Home
Share
View
Picture Tools
New folder
Quick access
Desktop
Downloads
Documents
Pictures
Google Drive
debug
New folder
Student_Attendance
Windows (C:)
OneDrive
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
studvol1.vit.ac.in
Videos
Windows (C:)
DATA (D:)
Network
1.png
2.png
3.png
17MIS0082.png
18BCE0833.png
18MIS0055.png
18MIS0081.png
test.png
testv.png
vaishali
vaishali.png
Scanning Code
Press Alt to move camera.
15 items
Type here to search
ENG
18:12
02-06-2021

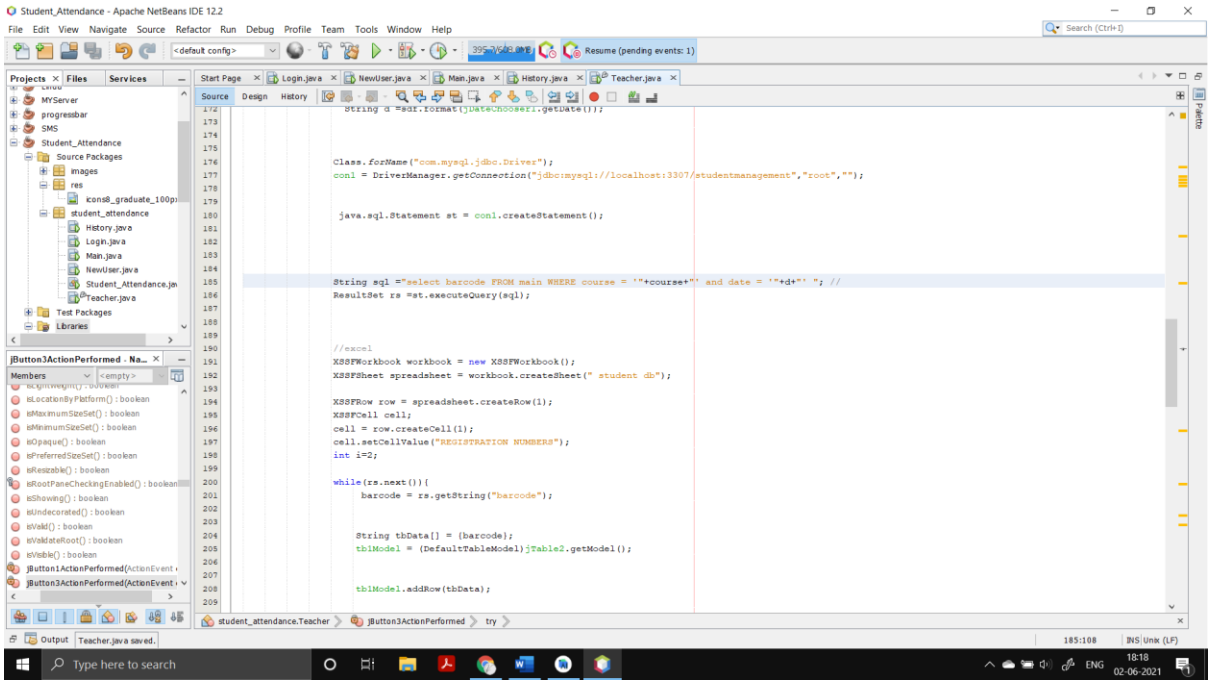
3.5. Sample source code

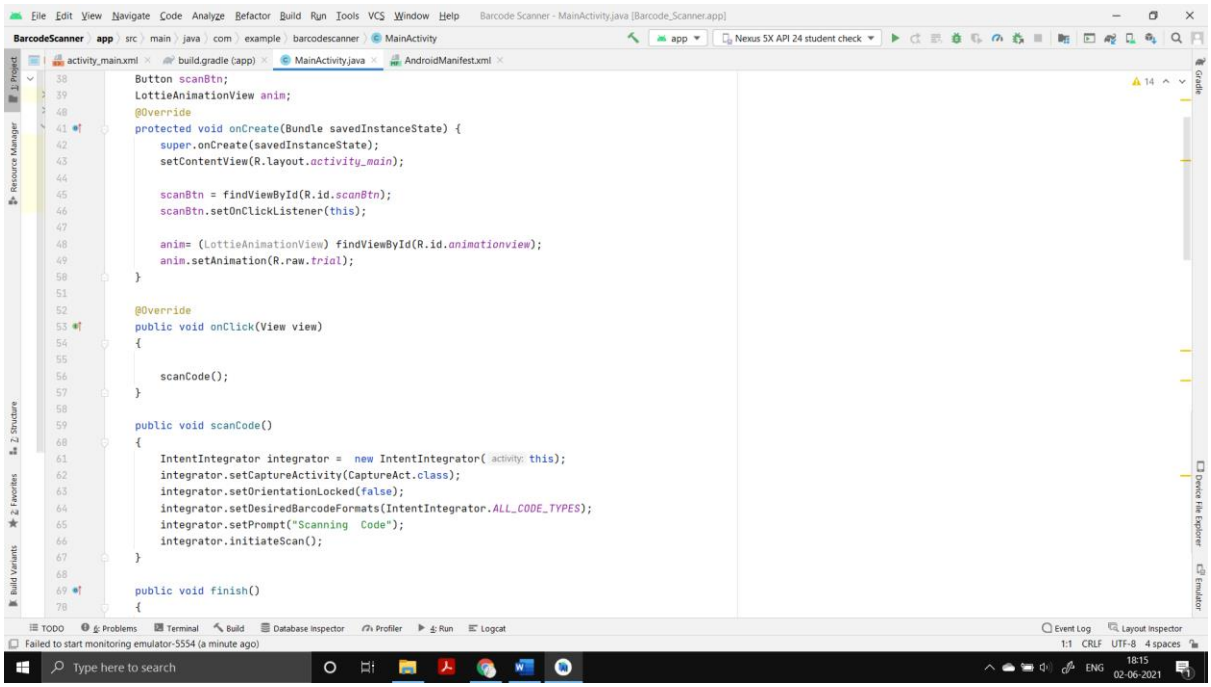


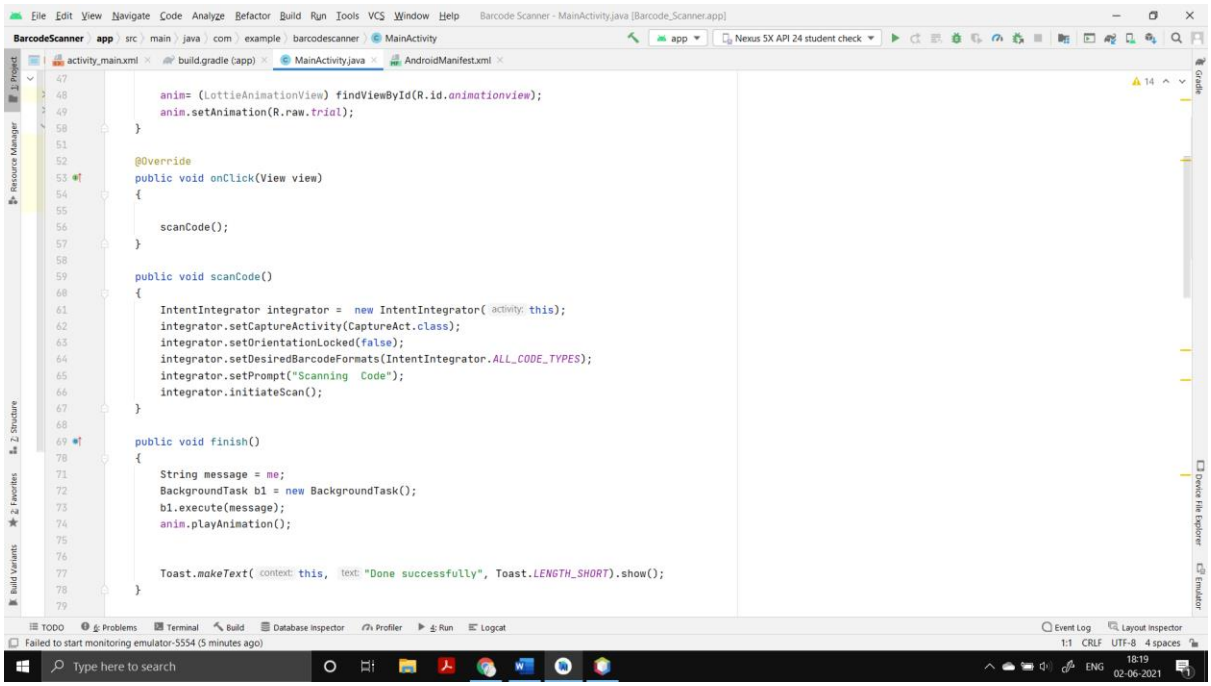



Student_Attendance - Apache NetBeans IDE 12.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Resume (pending events: 1)
Start Page
Login.java
NewUser.java
Main.java
History.java
Teacher.java
Source
Design
History
insert = con1.prepareStatement("insert into main(course,date,barcode)values(?,?,?)");
//------------ insert.setString(1, username1);
insert.setString(1, course);
insert.setString(2, date);
insert.setString(3,barcode);
insert.executeUpdate();
LocalTime time = LocalTime.now();
JOptionPane.showMessageDialog(null,"updated your attendance for "+ date + " at "+ time + "successfully");
this.hide();
new History(barcode).setVisible(true); // in history
txtCourse.setSelectedIndex(-1);
txtDate.setText(" ");
txtBarcode.setText(barcode);
catch (ClassNotFoundException | SQLException ex) {
Logger.getLogger(NewUser.class.getName()).log(Level.SEVERE, null, ex);
/**
* @param args the command line arguments
*/
public static void main(String args[]) {
Type here to search
ENG
18:17
02-06-2021

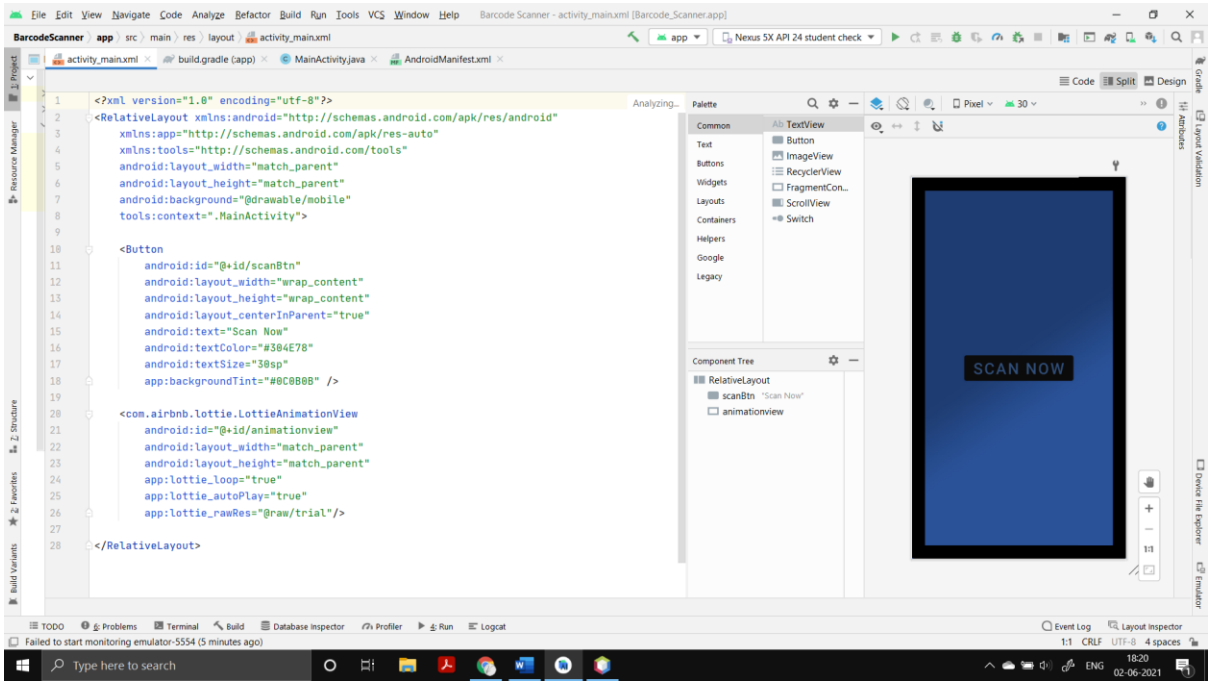

Student_Attendance - Apache NetBeans IDE 12.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Resume (pending events: 1)
Start Page
Login.java
NewUser.java
Main.java
History.java
Teacher.java
Source
Design
History
// Create MySQL Database Connection
Class.forName("com.mysql.jdbc.Driver");
Connection connect = DriverManager.getConnection("jdbc:mysql://localhost:3307/studentmanagement","root","");
Statement statement = (Statement) connect.createStatement( );
ResultSet resultSet = statement.executeQuery("SELECT COUNT(*) AS number, course FROM main WHERE barcode = '18BCE0833' GROUP BY course");
DefaultPieDataset dataset = new DefaultPieDataset( );
while(resultSet.next( )) {
dataset.setValue(
resultSet.getString("course") ,
Double.parseDouble(resultSet.getString("number")));
}
JFreeChart chart = ChartFactory.createPieChart(
"Courses", // chart title
dataset, // data
true, // include legend
true,
false);
int width = 560; // Width of the image
int height = 370; // Height of the image
File pieChart = new File("Courses.jpeg");
ChartUtilities.saveChartAsJPEG(pieChart , chart , width , height);
}
catch(ClassNotFoundException | SQLException | IOException e)
{}
}
Connection con1;
PreparedStatement insert;
/**
Type here to search
ENG
18:17
02-06-2021


Student_Attendance - Apache NetBeans IDE 12.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Projects Files Services
Start Page Login.java NewUser.java Main.java History.java Teacher.java
Source Design History
Class.forName("com.mysql.jdbc.Driver");
con1 = DriverManager.getConnection("jdbc:mysql://localhost:3307/studentmanagement","root","");
java.sql.Statement st = con1.createStatement();
String sql ="select barcode FROM main WHERE course = '"+course+"' and date = '"+d+"' "; //
ResultSet rs =st.executeQuery(sql);
//excel
XSSFWorkbook workbook = new XSSFWorkbook();
XSSFSheet spreadsheet = workbook.createSheet(" student db");
XSSFRow row = spreadsheet.createRow(1);
XSSFCell cell;
cell = row.createCell(1);
cell.setCellValue("REGISTRATION NUMBERS");
int i=2;
while(rs.next()){
barcode = rs.getString("barcode");
String tbData[] = {barcode};
tblModel = (DefaultTableModel)jTable2.getModel();
tblModel.addRow(tbData);
Type here to search
ENG
18:18
02-06-2021


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Barcode Scanner - MainActivity.java [Barcode_Scanner.app]
BarcodeScanner app src main java com example barcodescanner MainActivity
activity_main.xml build.gradle (:app) MainActivity.java AndroidManifest.xml
Button scanBtn;
LottieAnimationView anim;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
scanBtn = findViewById(R.id.scanBtn);
scanBtn.setOnClickListener(this);
anim= (LottieAnimationView) findViewById(R.id.animationview);
anim.setAnimation(R.raw.trial);
}
@Override
public void onClick(View view)
{
scanCode();
}
public void scanCode()
{
IntentIntegrator integrator = new IntentIntegrator(activity: this);
integrator.setCaptureActivity(CaptureAct.class);
integrator.setOrientationLocked(false);
integrator.setDesiredBarcodeFormats(IntentIntegrator.ALL_CODE_TYPES);
integrator.setPrompt("Scanning Code");
integrator.initiateScan();
}
public void finish()
{
Failed to start monitoring emulator-5554 (a minute ago)
Type here to search
ENG
18:15
02-06-2021



```
        anim= (LottieAnimationView) findViewById(R.id.animationview);
        anim.setAnimation(R.raw.trial);
    }

    @Override
    public void onClick(View view)
    {

        scanCode();
    }

    public void scanCode()
    {
        IntentIntegrator integrator =  new IntentIntegrator( activity: this);
        integrator.setCaptureActivity(CaptureAct.class);
        integrator.setOrientationLocked(false);
        integrator.setDesiredBarcodeFormats(IntentIntegrator.ALL_CODE_TYPES);
        integrator.setPrompt("Scanning  Code");
        integrator.initiateScan();
    }

    public void finish()
    {
        String message = me;
        BackgroundTask b1 = new BackgroundTask();
        b1.execute(message);
        anim.playAnimation();


        Toast.makeText( context: this,  text: "Done successfully", Toast.LENGTH_SHORT).show();
    }
```



```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/mobile"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/scanBtn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true"
        android:text="Scan Now"
        android:textColor="#304E78"
        android:textSize="30sp"
        app:backgroundTint="#0C0B0B" />

    <com.airbnb.lottie.LottieAnimationView
        android:id="@+id/animationview"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:lottie_loop="true"
        app:lottie_autoPlay="true"
        app:lottie_rawRes="@raw/trial"/>

</RelativeLayout>
```

4. Testing

4.1. Login

Testcases	Check Item	Testcase Objective	Steps to execute	Input	Expected Result
TC1	Login page	Leave all fields blank	Click login	Fill all the fields	Filling the fields is mandatory
TC2	username	Enter invalid username	Click login	Invalid username and password	Enter correct credentials
TC3	password	Enter invalid password	Click login	Invalid username and password	Enter correct credentials
TC4	Username and password	Correct username and password	Click login	XYZ ****	XYZ **** Which is correct response

4.2. Username

Testcases	Check Item	Testcase Objective	Steps to execute	Input	Expected Result
TC1	All fields	Empty fields	Click Add	Fill all the fields	Filling the fields is mandatory
TC2	Password and confirm password match	No matching password	Click Add	Password does not match	Enter correct credentials
TC3	Valid phone number	Enter invalid password	Click Add	Invalid username and password	Enter 10 correct digits
TC4	All fields	Valid passwords and phone number	Click Add	Correct credentials and add to db	Password: *** Confirm pass:*** Phone number: xxxxxxxxxx

4.3. Main

Testcases	Check Item	Testcase Objective	Steps to execute	Input	Expected Result
TC1	Barcode	Empty field	Click Submit	Fill the field	Filling the fields is mandatory with the same username
TC2	Date	Empty field	Click Submit	Fill the field	Enter correct credentials
TC3	Barcode	Socket	Click submit	Invalid port number	Enter correct socket number as in client
TC3	All fields	Enter all fields	Click Submit	valid Credentials	Barcode=username Date=dd/MM/yyyy Socket: xxxx

4.4. History

Testcases	Check Item	Testcase Objective	Steps to execute	Input	Expected Result
TC1	Submit When empty	To check for history attendance	Click View history	Empty data	Cannot view the table history
TC2	Submit When data is available	To check for history attendance	Click view history	Data present	Can view the history

4.5. Teacher

Testcases	Check Item	Testcase Objective	Steps to execute	Input	Expected Result
TC1	Date	Empty field	Click Export	Fill the field	Filling the fields is mandatory with the same username
TC2	Barcode table	Empty fields and empty data	Click Export	Fill the field	Enter correct credentials
TC3	Barcode table	Data view	Click Export	valid Credentials	Date :dd/MM/yyyy Jdropfield And data available

4.6 Barcode app

Testcases	Check Item	Testcase Objective	Steps to execute	Input	Expected Result
TC1	Barcode	Empty field with no valid barcode	Click Scan now	Fill the field	Empty scanning with no result
TC2	Barcode	Empty fields with no scan	Click Finish	Scanned	Scan valid barcode
TC3	Barcode	Data view	Click Scan again	valid Credentials	To scan again with valid barcode
TC4	Barcode	Update in application	Click scan now and then finish	Valid barcode scanning	Updated to the application

5. Conclusion

The student attendance management system is assisted with barcodes and application to scan them along with to view the attendance is build in order to eliminate the manual human errors with the help of Java language and software.

6. References

https://www.tutorialspoint.com/apache_poi/apache_poi_database.htm
www.youtube.com
https://developer.android.com/reference/android/os/AsyncTask