



INDEX

* INTRODUCTION
* FEATURES OF THE PROJECT
* DBMS (SQL)
* Spring frame work and JDBC
* Comparison between JDBC and Spring Frame work
* Database table
* APPLICATION CONTROLLER (CLASS)
* APPLICATION DAO (PACKAGE)
* ONE FRAME FOR ALL
* DESIGN AND CODE
* THOUGHTS ON FURTHER IMPROVEMENT
* ACKNOWLEDGEMENTS

**INTRODUCTION**

**What is Phoenix Attendance?**

Phoenix Attendance is a Software tool, which offers a wide range of features to the user.Its core function is to allow teachers, to have a dynamic attendance taking software, which is easy to use and also extremely user friendly.

**Why is Phoenix Attendance special?**

Conventional attendance taking includes outdated mechanisms like roll-call, which are cumbersome and also the data storage is inefficient.  
After storing this data, accessing it is also difficult, as it is mostly maintained in hard copies like, attendance registers.

Phoenix Attendance is an extremely empowering tool, as it includes error free mechanisms to take attendance figures.

It has the dynamic attendance feature.

Dynamic attendance here means that the teacher can now take attendance of the students before the start of every class.

Thus a database can be created which has the attendance statistics of every student for INDIVIDUAL classes.

**What are the different features of this software?**

**Teacher Login :** All the teachers have a unique username and password, which they will use to login in to the software. Teachers can also change their usernames and passwords once they log into their accounts.

**Speedy attendance taking :**  This is achieved because, as soon as the teacher logs in during school hours, the software automatically calculates the class that the teacher is in, according to the preset teachers timetables, and displays the names of the students of that class.  
After that, taking attendance is just a few clicks away.

**Teachers and Students time tables :** When the teacher completes taking the attendance or if he/she logs in to their account in a free period, they can access the timetables of other teachers, and the timetables of all the senior secondary classes.

**The Change class feature** **:**  When a Teacher wants to view the timetable of a particular class for the week , she could use this combo Box and Do the same. This Would Enable The Teacher To Efficiently Manage His /Her Work. Such as If a Teacher Wishes to exchange a period with another teacher then she wants to view the timetables. This would be a tiring task to do with the Hardcopies.

**The Change Teacher feature :**  When a Teacher wants to view the timetable of a particular Teacher for exchanging a period with another teacher, she could use this combo Box and Do the same. This Would Enable The Teacher To Efficiently Manage His /Her Work.

**Accessing the attendance Database:**  Every time the teacher takes attendance and submits it, the data is added to the grand attendance database. Every Teacher can access this database. He/she just has to set the filter options such as class, period, and the date. He/she will get the attendance figures of the entered class, on a given day of the particular entered period.

**Admin:** Admin is a person who has access to more sensitive operations, like adding new teacher’s profiles, deleting old profiles.  
He also has access to the login log. This log has records which shows, which teacher logged in, the time of logging in, the time spent logged in, and the time of logout.

**Report Problem:** No program in this world is perfect. Every program will encounter bugs or malfunctions. Our program is no different. If any bugs or malfunctions occur, then the user can report it in the Report problem section, and we will immediately act upon it and try to fix the bug as soon as possible.

Users can also suggest any improvements, which can be made in the program through the report problem feature.

**DATABASE MANAGEMENT**

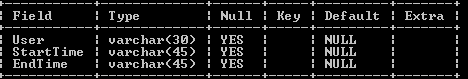
The database system for this program is dynamic and error free. We have made sure that the database systems occupy as little space as possible, with our special “3 pronged” Isolation method. This method uses 3 inputs from the user to pinpoint the required data and extracts it.

For example, in the teacher’s time table, in addition to the period column and the 6 days of the week, we also included a column, for the teacher’s name. So instead of creating 8 tables for 8 teachers, we created only one table.

While querying this table, we provide period number, day of the week and also the teacher’s name. So we will get the class in which the entered teacher should be in. This reduces a lot of space and helps in the efficiency of the program.

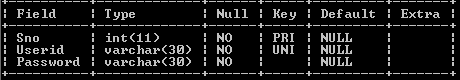
Database Structures

**1. Table Log**



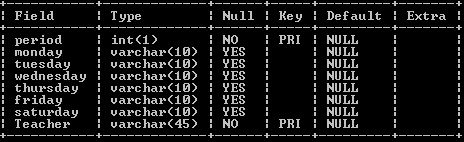
This table will store the name of the user, the time he/she logged in, and the time he/she logged out.

**2. Table Login**



This table contains all the usernames and passwords of the users. During the login process, the values entered are cross checked with these values.

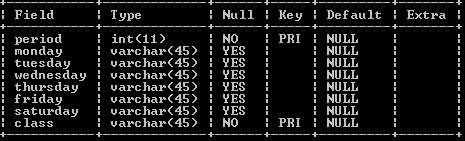
**3. Table staff\_timetable**



This tables stores the teachers time tables.

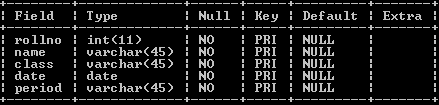
This table has 3 kinds of columns. Period, day of the week, and teacher name.  
While querying this table, the “3 pronged isolation method” is used as mentioned above.

**4. Table student\_timetable**



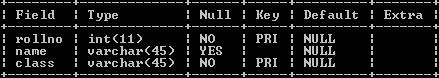
This table Stores all the timetables of the different sections. This table also has three types of columns. Instead of teachers name, here there will be “class”.

**5. Table student\_absent**



This is the grand attendance database. It contains information regarding the students who were marked absent and the periods in which they were absent.

**6. Table student\_list**



This table contains the roll number, name, and class of all the students of the senior secondary wing.

**ApplicationController**

Application Controller is a class which controls the different properties and processes of the whole program. It has different methods , which facilitate the removal and addition of panels into a single frame. Also it is the base of all the transactions of screens taking place and caters for all the requests from different screens.

**ApplicationDao**

Application Dao is a class which is responsible for all the sql transactions and services related to the whole program. This class has methods which perform tasks such as sending a query to the database or updating information in the database. All the database related tasks are performed using this class. This is the base class for the database services.

Advantages of Spring Jdbc template over Jdbc

**1.**They allow you to access JDBC functionality from within Spring container, and they provide some additional simplifications compared to plain JDBC, like connection management and exception handling. Spring is harder to setup but easier to develop .

**2.**Spring utilizes Template pattern to hide all low level details while giving you extension hooks to extend and work with JDBC.

**3**.In Jdbc A lot of code is needed to be return and is not very efficient and wastes programmers valuable time.

**4.**In Spring Jdbc connections need not be opened and closed all the time and also connections donot leak.

**5.**The Spring JDBC template allows to clean-up the resources automatically.

One Frame For All

Our Project Uses Panels unlike Traditional Frames for all the screens. Using Panels Allows use to reuse components .If at all we would have used Frames, it would made our work

longer time to complete and also be more tiring. As Our Program Consists Of Different Screens Which All have things in common such as the Header of the screen which is the same in all the screens i.e only our Center of the screen keeps changing in our program. Using a Header Panel Exempted us from writing alot of code for each screen. Also the center of the screen is host for tables which are used to display information. We have made a single table as a panel and used different methods to change what it displays depending on the user requests. It

took us lot of understanding of the concepts of panels and components to come off with such an idea and be successful in it. Although it was very difficult to think and start of with it but we felt it easier as we progressed into the developing of our projetct.

\* Above Mentioned are for only Non-Admin Services

DESIGN AND CODE

LOGIN PAGE:



**code for clear:**

jTextField1.setText(“”);

jTextField2.setText(“”);

code for exit:

System.exit(0);

Code for Admin Login:

Admin newadm = new Admin();

newadm.setVisible(true);

**Code for login button:**

Calendar cal1 = Calendar.getInstance();

cal1.getTime();

SimpleDateFormat sdg = new SimpleDateFormat("yyyy.MM.dd 'at' hh:mm:ss");

String b=(sdg.format(cal1.getTime()));

int c=0;

if(jTextField1.getText().equals("")||jPasswordField1.getPassword().equals("")){

JOptionPane.showMessageDialog(null, "UserID or Password not entered.Please Try Again"); }

else{

try{

String username=jTextField1.getText();

String passW=new String(jPasswordField1.getPassword());

Class.forName("java.sql.DriverManager" )Connectioncon=DriverManager.getConnection("jdbc:mysql://localhost:3306/Student,"teja)

Statement stmt=con.createStatement();

String query="select \* from login;";//to get table logins data

ResultSet rs=stmt.executeQuery(query);

while (rs.next()){

String userid=rs.getString("Userid");

String password=rs.getString("Password");

if(userid.equals(userid)&&password.equals(passW)){

new Admin().setVisible(true);

dispose();

} else{

JOptionPane.showMessageDialog(null, "Please Check Your Details");

}

break;

} con.close();

stmt.close();

rs.close();

}

catch (Exception e){

JOptionPane.showMessageDialog(null, e.getMessage());

**\*If the teacher logs in non-school hours, then this page comes up.**



This consists of two components. The header panel and the lower panel. The header panel remains the same but the lower panel keeps changing.

This is the default page which shows the teachers time-table.

The change class drop box, has all the classes of the senior secondary wing, when we select any class, the table below is filled with the timetable of that class.

Similarly the change teacher drop box contains the all the teachers of the senior secondary wing, and when we select any teacher, that teachers timetable, is filled in the table.

**Code for change class/change teacher(from application controller):**

void showLoggedInTeacherTimetable() {

ensureTimetablePanelAtCenter();

getTimetablePanel().showTeacherTimetable(loggedInUsername);

}

private void ensureTimetablePanelAtCenter() {

BorderLayout layout = (BorderLayout)applicationFrame.getContentPane().getLayout();

Component atCenter = layout.getLayoutComponent(BorderLayout.CENTER);

if (atCenter != getTimetablePanel()) {

if (atCenter != null)

applicationFrame.getContentPane().remove(atCenter);

applicationFrame.getContentPane().add(getTimetablePanel(), BorderLayout.CENTER);

applicationFrame.revalidate();

applicationFrame.repaint();

}

}

void showLoggedInTeacherClass() {

applicationFrame.getContentPane().add(getHomePagePanel(), BorderLayout.CENTER);

getHomePagePanel().showStudentData();

}

void showSelectedTeacherTimetable(String value) {

ensureTimetablePanelAtCenter();

getTimetablePanel().showTeacherTimetable(value);

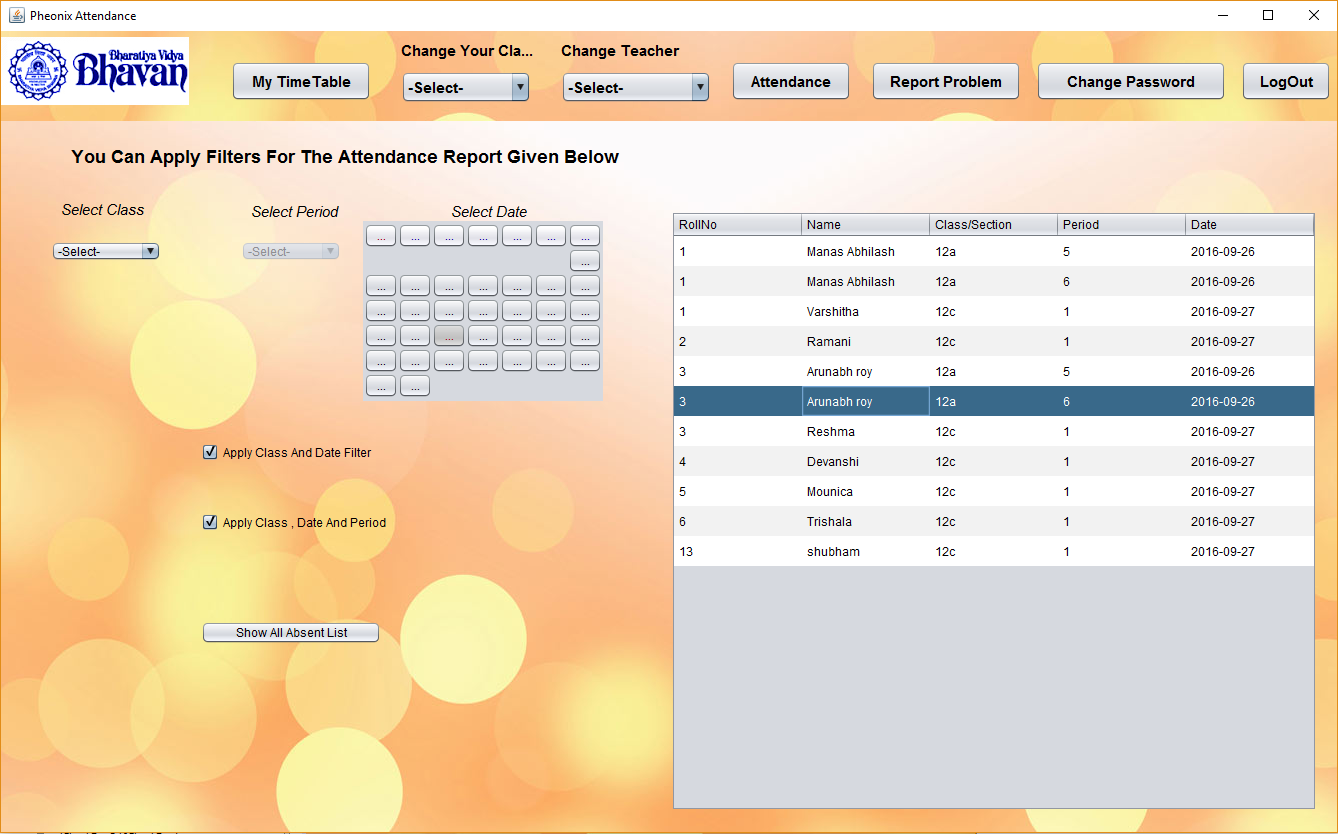
}

void showSelectedClassTimetable(String classname) {

ensureTimetablePanelAtCenter();

getTimetablePanel().showStudentTimetable(classname);}

ATTENDANCE



When the attendance tab is clicked on the header panel, the above picture is presented.

Here we can select the class, period , date and the students who were absent on that specified time, their names will be displayed in the table to the right.

**CODE FOR SHOW ABSENT LIST:**

jComboBox1.setSelectedItem("-Select-");

jComboBox2.setSelectedItem("-Select-");

jComboBox2.setEnabled(false);

getAbsentList(); // REFER APPLICATION CONTROLLER

**LOGOUT:**

From application controller:

public void userLoggedOut() {

applicationDao.logUserLogOff(loggedInUsername, sessionStartTime, new Date());

loggedInUsername = null;

applicationFrame.getContentPane().removeAll();

applicationFrame.getContentPane().add(getLoginPanel(), BorderLayout.CENTER);

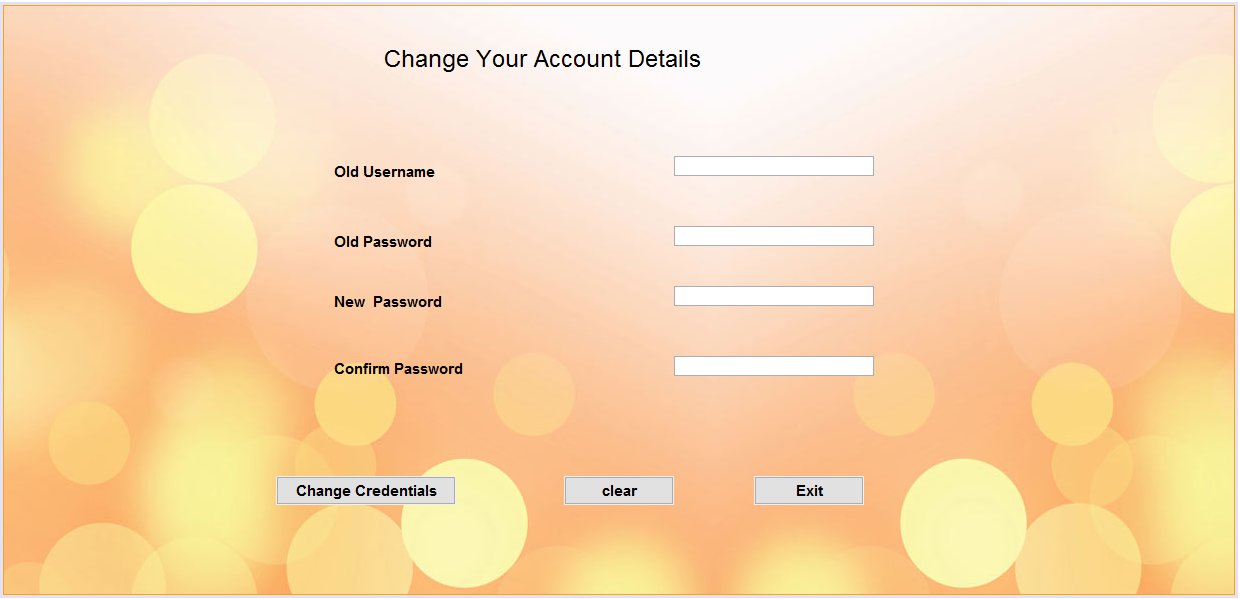
applicationFrame.revalidate();

applicationFrame.repaint();}

**Code for change Password:**

**Code for change Password:**

applicationController.getchangePassword();

**CHANGE PASSWORD:**

**Code for clear:**

jTextField1.setText(“”);

jTextField2.setText(“”);

jTextField3.setText(“”);

jTextField4.setText(“”);

**Code For exit:**

System.exit(0);

Code For change credentials:

String username=oldUsername.getText();

String old\_pass=new String(jPasswordField1.getPassword());

String new\_pass1 = new String (newpass1.getPassword());

String new\_pass2 =new String( newpass2.getPassword());

boolean validUser=applicationController.getApplicationDao().validateUser(username,old\_pass);

if (validUser) {

applicationController.changePassword(username,new\_pass1,new\_pass2); }

else { JOptionPane.showMessageDialog(null, "Wrong User ID or Password. Try again");

oldUsername.setText("");

jPasswordField1.setText("");

newpass1.setText(null);

newpass2.setText(null); }

THOUGHTS ON FURTHER IMPROVEMENT

* This program can be extended to the entire school, instead of just senior secondary wing.
* A better user interface
* Better storage and querying of the attendance figures.

ACKNOWLEDGEMENTS

* Our team sincerely thank our Informatic Practices teacher, Mrs. Deepshika mam for supporting us throughout the development of this project. We would like to thank her for providing us with solutions for the problems we faced during the development of the project.
* We would like to thank the school authorities for providing us with the computer lab, and all the resources that we needed.
* We would like to thank our class mates, especially Siva Charan who inspired us to name the project. And all others who helped and guided us.