

Telescope ERP System

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Science Degree in Computer Science

By

SURYA PRAKASH.P (Reg. No. 38290094)

PAVITHRAN.V (Reg. No. 38290065)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

**SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
JEPPIAAR NAGAR, RAJIV GANDHI
SALAI, CHENNAI – 600119, TAMILNADU**

APRIL 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC
(Established under Section 3 of UGC Act, 1956)

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI- 600119

www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of
SURYAPRAKASH.P (Reg. No. 38290094) and PAVITHRAN.V (Reg. No. 38290065) who carried out the project entitled "**TELESCOPE ERP SYSTEM**"
under our supervision from November 2020 to April 2021.

Internal Guide

Dr. D.DEEPA M.E., Ph.D.,

Head of the Department

Dr. S. VIGNESHWARI , M.E., Ph.D.,

Submitted for Viva voce Examination held on _____

DECLARATION

I, SURYA PRAKASH.P (Reg. No. 38290094) hereby declare that the Project Report entitled "TELESCOPE ERP SYSTEM" done by me under the guidance of Dr.D.DEEPAA, M.E.,Ph.D., is submitted in partial fulfillment of the requirements for the award of Bachelor of Science degree in Computer Science

DATE:

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. SASIKALA, M.E., Ph.D.**, Dean, School of Computing
and

Dr. S. VIGNESHWARI, M.E., Ph.D., and Dr. L. LAKSHMANAN, M.E., Ph.D., Head of the Department, Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.D.DEEPA.,M.E , Ph.D** for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of Computer Science and Engineering who were helpful in many ways
For the Completion of the project.

INDEX		
Sl.No.	Particulars	Page No.
1	Synopsis	1-2
2	Requirement Specification	3
3	Introduction	4-5
4	Objective	6-7
5	Tool Description	8-9
6	System Analysis	10-12
7	ER Diagram	13-14
8	DFD	15-20
9	SCHEMA	21-22
10	Source Code	23-99
11	System Testing	100-103
12	Snapshots	104-110
13	Future Enhancements	111-112
14	Conclusion	113-114
15	Bibliography	115-116

Synopsis

TELESCOPE – ERP

SYNOPSIS

Title: TELESCOPE - Enterprise Resource Planning

OBJECTIVE:

Tele-Scope ERP is an application software that focuses on improved collaboration and workflows among the students, faculties and the administration of the institution. It can be used to increase productivity by generating information that can be accessed by the members hassle free. Using this application, members can get precise reports for planning. This application can be used for getting updated information on time. Using this application, expenditure being incurred by an institution presently will be reduced.

Functionalities of Modules:

Admin can perform the following operations:

- Generating Attendance report.
- Addition of course, student exam results, sports schedules, library books ·
- Updating the facilities.
- Processing admission request
- Generating ID card
- Preparation of summary
- Preparation of result analysis.

Faculty / Student have the following privileges:

- Obtaining membership
- Viewing Attendance
- Viewing Results
- Feedback on available facilities
- Viewing Time Table
- Changing password **PAGE**

TELESCOPE – ERP

REQUIREMENTS SPECIFICATION

Hardware Requirements:

- Processor :64 BIT, Intel core i3 2nd gen
- RAM :4GB RAM or higher
- Hard disk :20 GB (Minimum)
- Monitor :1024x768(Resolution)
- Keyboard
- Mouse
- Printer

Software requirements:

- Operating system : Windows 10
- Front end : Eclipse(Java Swing)
- Back end :ORACLE 11g

TELESCOPE – ERP

Introduction

TELESCOPE – ERP

INTRODUCTION

Introduction:

Computerization is basically concerned with the automation of task using computer-based information system. This process with the people involved in doing a certain task, which being replaced by particular machineries. Computerization would also mean less effort exerted, since manual could be done in shorter span of time. In any organization, computer is a necessity. Computerization to an organizational establishment is important in order to do task easily and efficiently.

Our ERP System can achieve this with utmost security and accuracy.

TELESCOPE – ERP

Objective

TELESCOPE – ERP

OBJECTIVE:

The main objective of this project is to computerize the courier management process. Here we record the details of the consignment and generate the bill and process the shipment through the formulated system. The user can also check the receipt of the bill. The software has been created with the aim to ensure a smooth relationship with our clients.

PRESENT AND PROPOSED SYSTEM:

- The existing system is limited with few modules and not easy to access for the novice users.
- The proposed system gives easy access to information using a structured GUI.
- Has better data security.
- Gives a platform to interact.
- INPUT: Registration, Updation and View of various modules by aspiring students, parents, faculty and admin.

Tool description

TELESCOPE – ERP

EclipseVersion: 2020-09 (4.17.0)

Eclipse Version: 2020-09 (4.17.0) is the version of Eclipse used to develop this application. The Eclipse Foundation provides our global community of individuals and organizations with a mature, scalable, and business-friendly environment for open source software collaboration and innovation.

Oracle 11g

Oracle Database (commonly referred to as **Oracle DBMS** or simply as **Oracle**) is a multimodel database management system produced and marketed by Oracle Corporation.

It is a database commonly used for running online transaction processing (OLTP), data warehousing (DW) and mixed (OLTP & DW) database workloads. Oracle Database is available by several service providers on-prem, on-cloud, or as hybrid cloud installation. It may be run on third party servers as well as on Oracle hardware (Exadata on-prem, on Oracle Cloud or at Cloud at Customer). Exclusively for Cloud customers Oracle offers Oracle Autonomous Database providing fully automated operation procedures. The current version of Oracle Database is the result of over 30 years of innovative development.

System analysis

Feasibility Study

Feasibility study is the measure of how beneficial or practical the development of an information system will be to an organization. The Feasibility analysis is a cross life cycle activity and should be continuously performed throughout the system life cycle. Feasibility study lets the developer foresee the future of the project and the usefulness. The study on feasibility is done on the basis of a few factors. They are:

Operational feasibility:

By automating the ERP works both the administrators and users will feel better than when it was manual. Users will get a very quick service by reducing the manual recordings. Also administrators will feel comfortable by reduction of their work. Recording errors will be reduced and it is easy to handle a large database. Losing of records will also be avoided.

Considering all these factors we can conclude that all the users and end users will be satisfied by the system.

Technical feasibility:

The system must be evaluated from the technical point of view. The assessment of this feasibility must be based on an outline design of the system requirement in terms of input, output, programs and procedures.

For the design and development of the system, several software products have been accommodated.

Database design – ORACLE 11g

Interface design – Eclipse 2020-09(Java Swing)

This software has enough efficiency in producing the system. Therefore the project is technically feasible.

Schedule feasibility:

TELESCOPE – ERP

The duration of time required for the project has been planned appropriately and it is the same as the duration of time expected by the user. Therefore the product can be delivered to the customer within the expected time duration, satisfying the students. Hence the project is feasible in scheduling.

Economic feasibility:

According to the resources available and the project scheduling process it is estimated that the expenses allocated for the software to be developed, by the user is sufficient enough. Hence the economical factor has been considered feasible.

Behavioural Feasibility:

This includes the following questions:

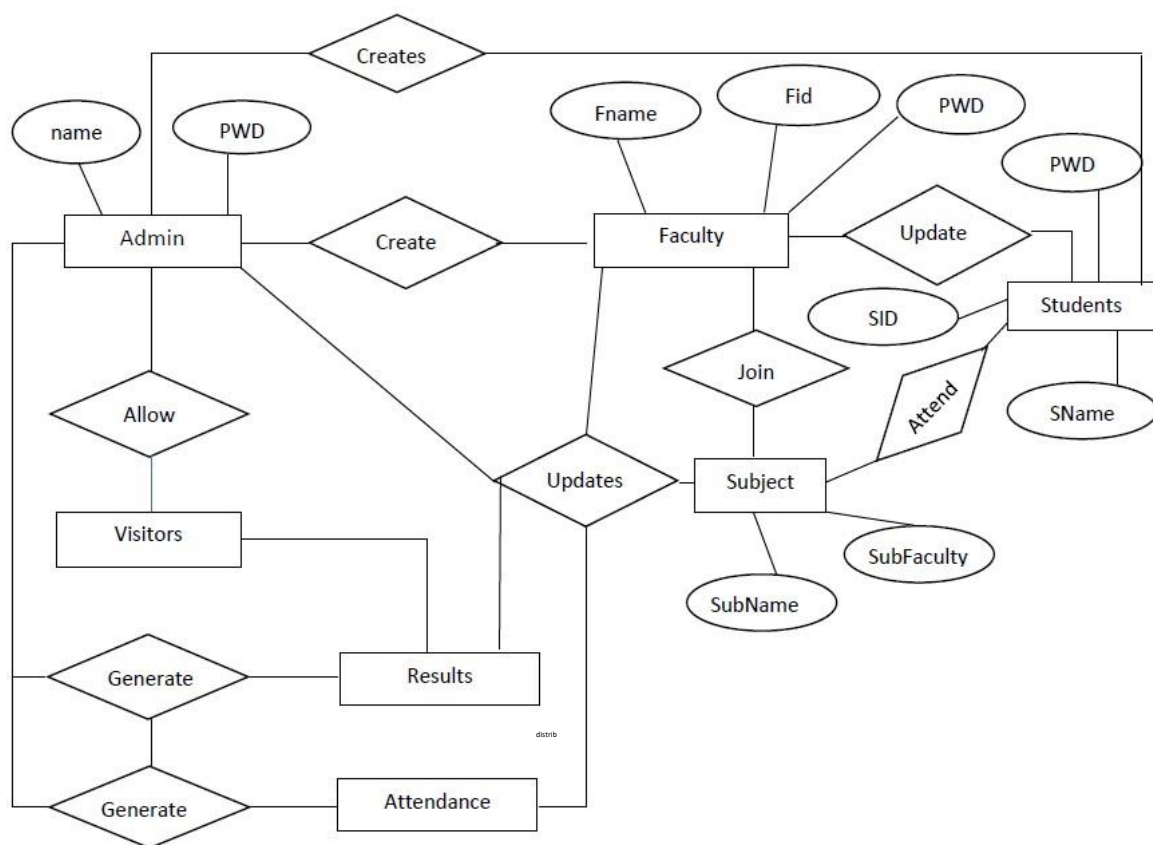
- Is there sufficient support for the users?
- Will the proposed system cause harm?

This project would be beneficial because it satisfies the objectives when developed and installed. All behavioural aspects are considered carefully and conclude that the project is behaviourally feasible.

SYSTEM DESIGN

E-R diagram

E-R Diagram



Data flow diagram

DATA FLOW DIAGRAM

Data Flow diagram:

A data-flow diagram (DFD) is a graphical representation of the "flow" of data through an information system. DFDs can also be used for the visualization of data processing (structured design).

On a DFD, data items flow from an external data source or an internal data store to an internal data store or an external data sink, via an internal process.

A DFD provides no information about the timing or ordering of processes, or about whether processes will operate in sequence or in parallel. It is therefore quite different from a flowchart, which shows the flow of control through an algorithm, allowing a reader to determine what operations will be performed, in what order, and under what circumstances, but not what kinds of data will be input to and output from the system, nor where the data will come from and go to, nor where the data will be stored (all of which are shown on a DFD)

The idea behind the explosion of a process into more process is that understanding at one level of details is exploded into greater detailed at the next level. This is done until

further explosion is necessary and an adequate amount of detail is described for analyst to understand the process.

Larry Constantine first developed the DFD as a way of expressing system requirements in a graphical form, this lead to modular design.

A DFD is known as a “bubble chart” has the purpose of clarifying system requirements and identifying major transformation they will become program in system design. So it is the starting point of the design to lowest level of details. A DFD consists of series of bubbles joined by data flows in the system.

DFD Symbols

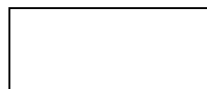
In DFD, there are four Symbols

1. A square-defines a source or destination system data
2. An arrow identified data flow. Its is the pipeline through which the information flow
3. A circle or a bubble represents a process that transforms
4. Incoming data flow into outgoing data flows
5. An open rectangle is a data source, data at rest or a temporary of data

Process that transforms data flow



Source or destination of the data



Data flow



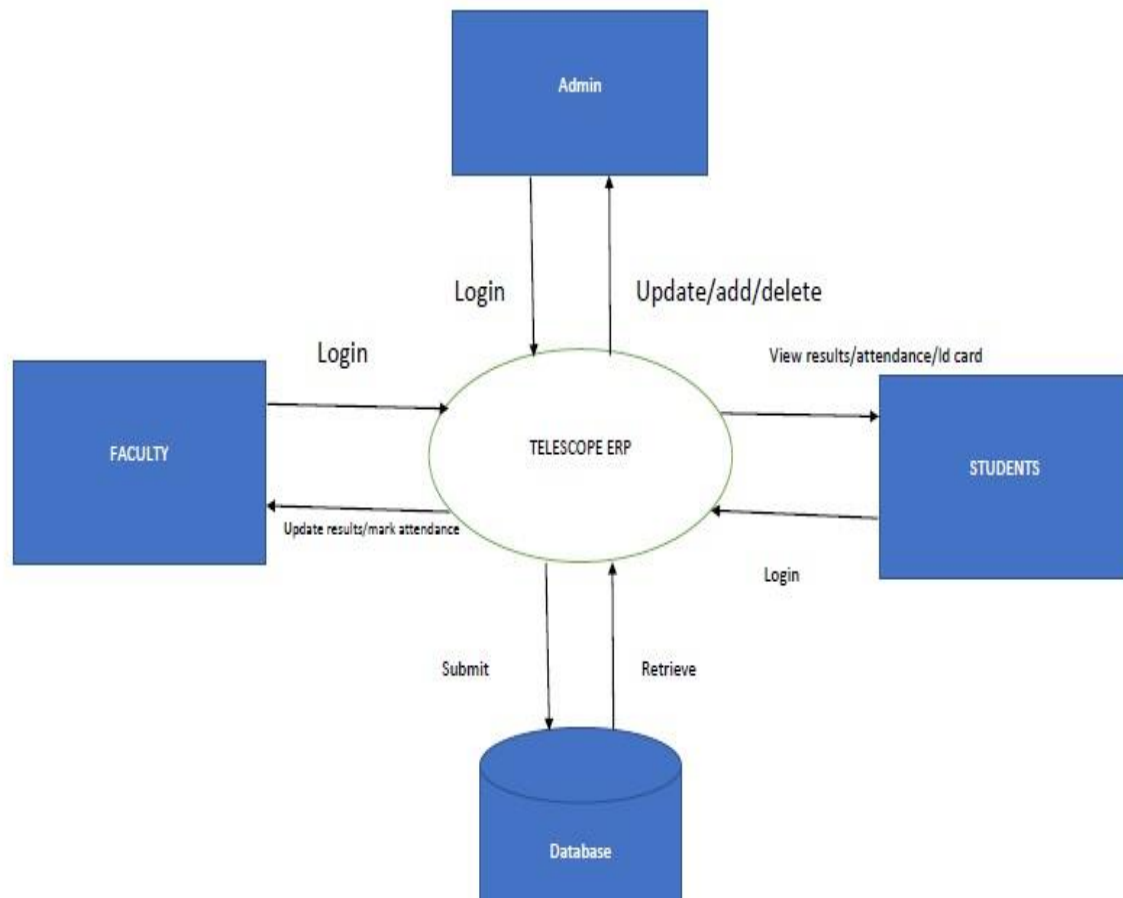
TELESCOPE – ERP

Process for DataFlow

--	--	--

CONTEXT LEVEL :

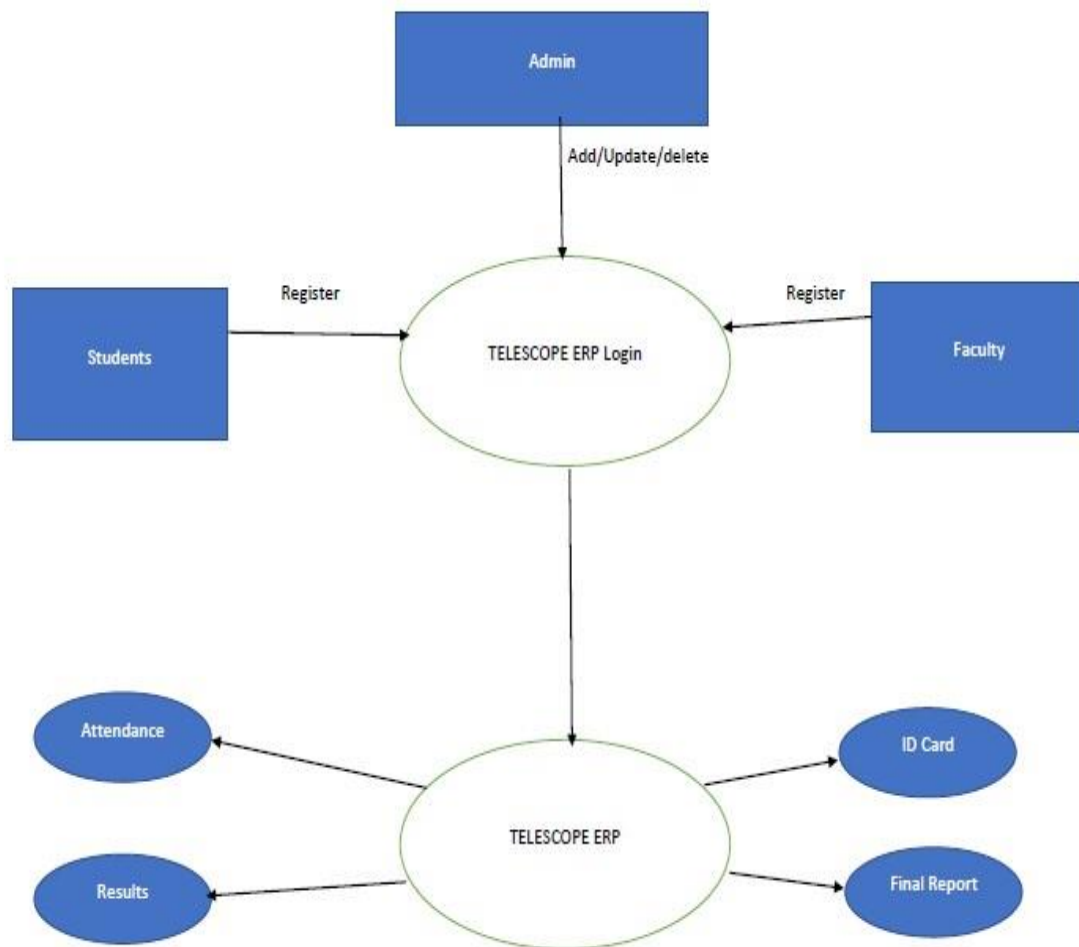
TELESCOPE ERP DATA FLOW DIAGRAM(DFD) - 0 level DFD



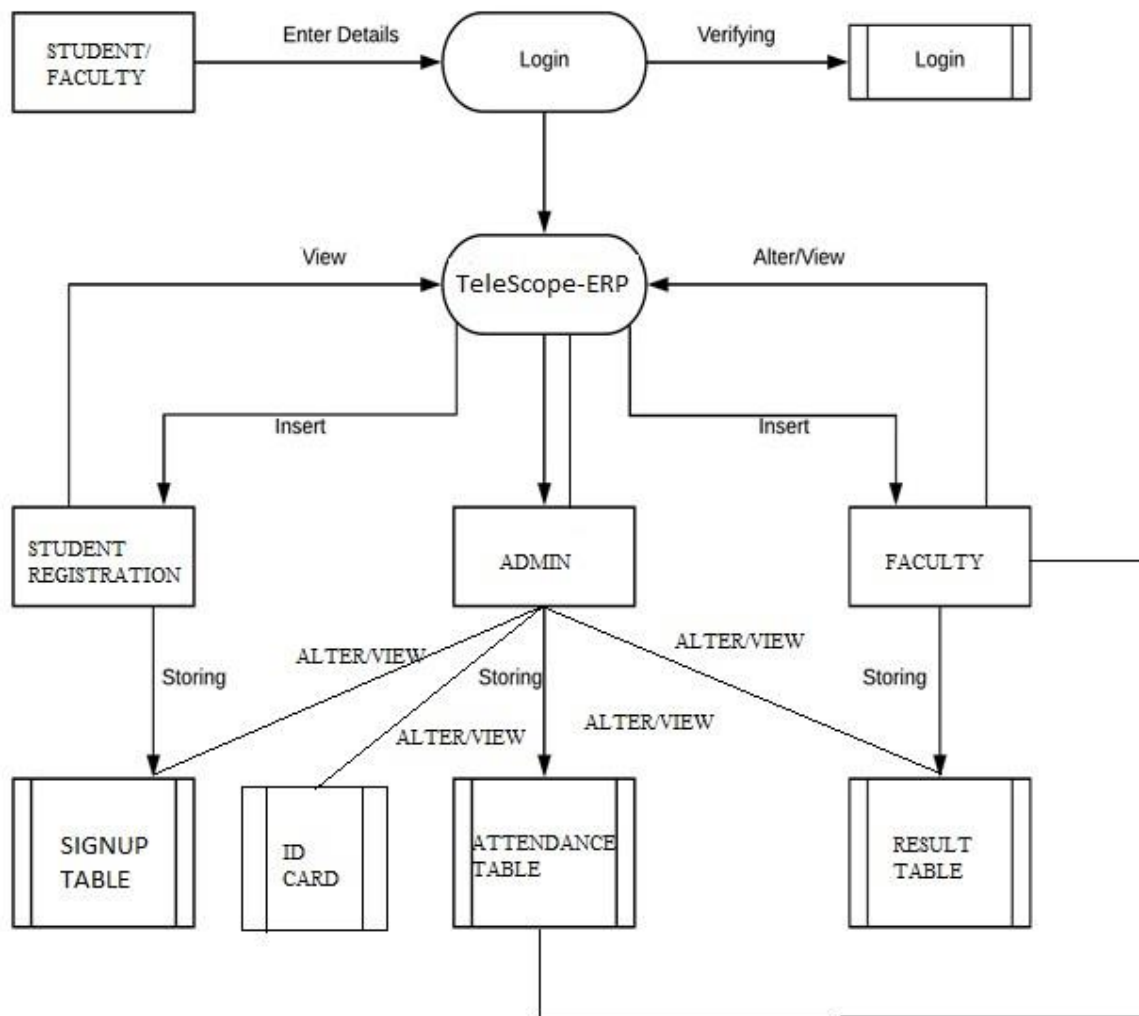
LEVEL 1 :

TELESCOPE – ERP

TELESCOPE ERP DATA FLOW DIAGRAM(DFD) - 1 level DFD

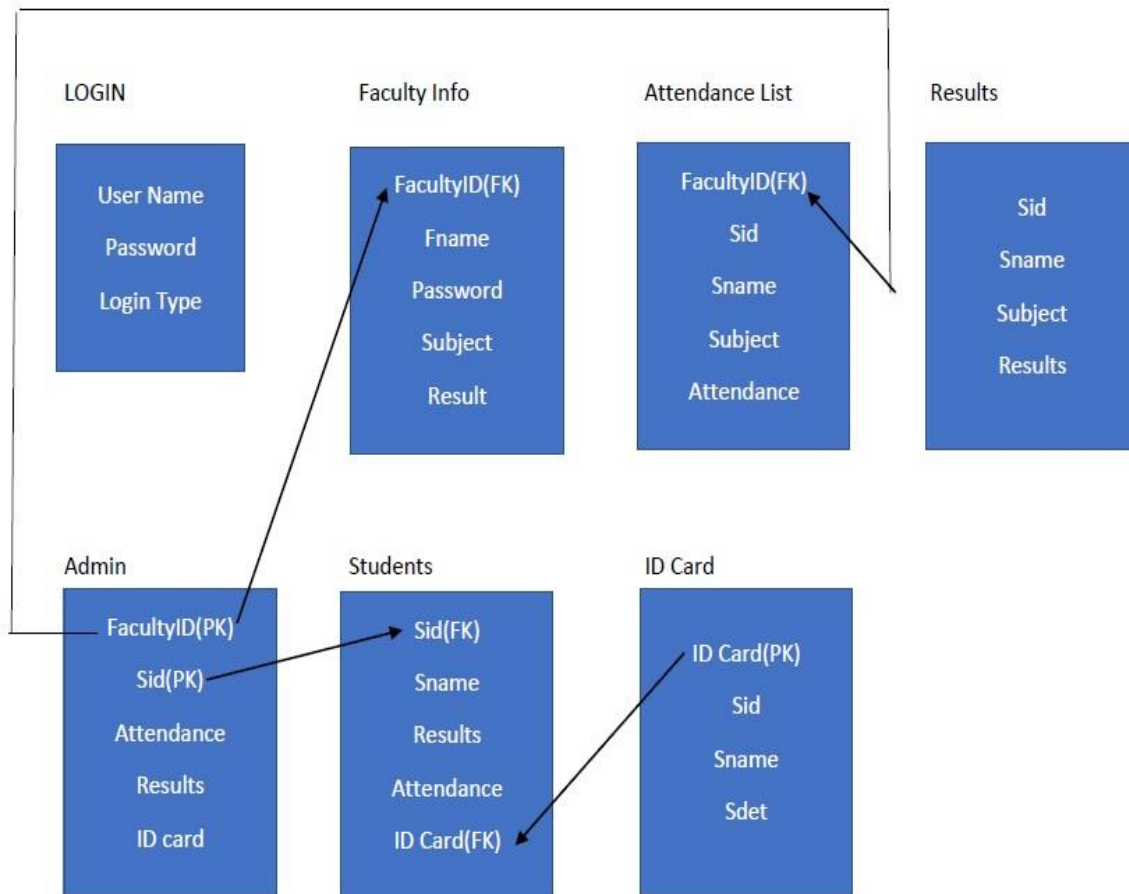


LEVEL 2:



TELESCOPE – ERP

Schema



TELESCOPE – ERP

Source code

SOURCE CODE

Markscard.java

```
package Telescope;

import java.awt.Color;

import java.awt.Font; import

java.awt.Graphics; import

java.awt.PrintJob; import

java.awt.Toolkit; import

java.sql.Connection; import

java.sql.DriverManager;

import

java.sql.PreparedStatement;

import java.sql.ResultSet;


import javax.swing.ImageIcon;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import

javax.swing.JOptionPane;

import javax.swing.JPanel;
```

```
import
javax.swing.SwingConstants;

import
javax.swing.border.LineBorder;

@SuppressWarnings("serial")

public class Markscard extends
JFrame {

    JPanelpanelid = new JPanel();

    JLabellbltitle = new JLabel("");    private
    javax.swing.JButtonbtnNewButton; String
    name; public JLabellblregno;    public
    JLabel lblname1;    public JLabel
    lblsub1_1;    public JLabel lblsub2_1;
    public JLabel lblsub3_1;    public JLabel
    lblsub4_1;    public JLabel lbltot1;
    public JLabel lblperc1;    String val;
    private JLabellblNewLabel; private JLabel
    lblNewLabel_1;    private JLabel
    lblNewLabel_2;    private JLabel
    lblNewLabel_3;
```

```
    Markscard()
```

```
{
```

TELESCOPE – ERP

```
setVisible(true);  
  
setSize(505,701);  
  
getContentPane().setLayout  
  
(null);
```

```
        btnNewButton = new JButton("Print Result");  
  
btnNewButton.setFont(new Font("Sylfaen", Font.PLAIN, 25));  
  
btnNewButton.setBounds(110, 607, 283, 47);  
  
getContentPane().add(btnNewButton);  
  
btnNewButton.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        btnNewButtonActionPerformed(evt);  
    }  
});
```

```
        panelid.setBounds(1, 0, 492, 662);  
  
getContentPane().add(panelid);  
  
panelid.setLayout(null);
```

```
        JLabellblreg = new JLabel("REGNO:");  
  
lblreg.setBounds(70, 141, 157, 39);  
  
panelid.add(lblreg);
```

```
lblreg.setHorizontalAlignment(SwingConstants.  
TRAILING); lblreg.setFont(new Font("Comic  
Sans MS", Font.PLAIN, 17));
```

```
JLabel lblname = new JLabel("Name:");  
lblname.setBounds(70, 189, 157, 39);  
panelid.add(lblname);  
lblname.setHorizontalAlignment(SwingConstants.TRAILING);  
lblname.setFont(new Font("Comic Sans MS", Font.PLAIN, 17));
```

```
JLabel lblsub1 = new JLabel("Java:\r\n");  
lblsub1.setBounds(70, 238, 157, 39);  
panelid.add(lblsub1);  
lblsub1.setHorizontalAlignment(SwingConstants.TRAILING);  
lblsub1.setFont(new Font("Comic Sans MS", Font.PLAIN, 17));
```

```
JLabel lblsub2 = new JLabel("C++:");  
lblsub2.setBounds(70, 287, 157, 39);  
panelid.add(lblsub2);  
lblsub2.setHorizontalAlignment(SwingConstants.TRAILING);  
lblsub2.setFont(new Font("Comic Sans MS", Font.PLAIN, 17));
```

```
JLabel lblsub3 = new JLabel("Python:");  
lblsub3.setBounds(70, 336, 157, 39);  
panelid.add(lblsub3);  
lblsub3.setHorizontalAlignment(SwingConstants.T
```

TELESCOPE – ERP

```
RAILING); lblsub3.setFont(new Font("Comic Sans  
MS", Font.PLAIN, 17));
```

```
JLabel lblsub4 = new JLabel("SE:");  
  
lblsub4.setBounds(70, 385, 157, 39);  
  
panelid.add(lblsub4);  
  
lblsub4.setHorizontalAlignment(SwingConstants.TRAILING);  
  
lblsub4.setFont(new Font("Comic Sans MS", Font.PLAIN, 17));
```

```
JLabel lbltot = new JLabel("Total Marks:");  
  
lbltot.setBounds(70, 434, 157, 39);          panelid.add(lbltot);  
  
lbltot.setHorizontalAlignment(SwingConstants.TRAILING);  
  
lbltot.setFont(new Font("Comic Sans MS", Font.PLAIN, 17));
```

```
JLabel lblperc = new JLabel("Percentage:");  
  
lblperc.setBounds(70, 483, 157, 39);  
  
panelid.add(lblperc);  
  
lblperc.setHorizontalAlignment(SwingConstants.TRAILING);  
  
lblperc.setFont(new Font("Comic Sans MS", Font.PLAIN, 17));
```

```
lblregno = new JLabel("");  
  
lblregno.setVerticalAlignment(SwingConstants.BOTTOM);  
  
lblregno.setBorder(new LineBorder(Color.LIGHT_GRAY));  
  
lblregno.setBackground(Color.LIGHT_GRAY); lblregno.setFont(new  
Font("Sylfaen", Font.PLAIN, 19)); lblregno.setBounds(237, 141, 136, 39);
```

```
panelid.add(lblregno);
```

```
lblname1 = new JLabel("");
```

```
lblname1.setVerticalAlignment(SwingConstants.BOTTOM);
```

```
lblname1.setBorder(new LineBorder(Color.LIGHT_GRAY));
```

```
lblname1.setBackground(Color.LIGHT_GRAY);
```

```
lblname1.setFont(new Font("Sylfaen", Font.PLAIN, 19));
```

```
lblname1.setBounds(237, 189, 136, 39);
```

```
panelid.add(lblname1);
```

```
lblsub1_1 = new JLabel("");
```

```
lblsub1_1.setVerticalAlignment(SwingConstants.BOTTOM);
```

```
lblsub1_1.setBorder(new LineBorder(Color.LIGHT_GRAY));
```

```
lblsub1_1.setBackground(Color.LIGHT_GRAY);
```

```
lblsub1_1.setFont(new Font("Sylfaen", Font.PLAIN, 19));
```

```
lblsub1_1.setBounds(237, 238, 136, 39);
```

```
panelid.add(lblsub1_1);
```

```
lblsub2_1 = new JLabel("");
```

```
lblsub2_1.setVerticalAlignment(SwingConstants.BOTTOM);
```

```
lblsub2_1.setBorder(new LineBorder(Color.LIGHT_GRAY));
```

```
lblsub2_1.setBackground(Color.LIGHT_GRAY); lblsub2_1.setFont(new  
Font("Sylfaen", Font.PLAIN, 19)); lblsub2_1.setBounds(237, 287, 136, 39);
```

```
panelid.add(lblsub2_1);
```

TELESCOPE – ERP

```
lblsub3_1 = new JLabel("");

lblsub3_1.setVerticalAlignment(SwingConstants.BOTTOM);

lblsub3_1.setBorder(new LineBorder(Color.LIGHT_GRAY));

lblsub3_1.setBackground(Color.LIGHT_GRAY);

lblsub3_1.setFont(new Font("Sylfaen", Font.PLAIN, 19));

lblsub3_1.setBounds(237, 336, 136, 39);

panelid.add(lblsub3_1);
```

```
lblsub4_1 = new JLabel("");

lblsub4_1.setVerticalAlignment(SwingConstants.BOTTOM);

lblsub4_1.setBorder(new LineBorder(Color.LIGHT_GRAY));

lblsub4_1.setBackground(Color.LIGHT_GRAY);

lblsub4_1.setFont(new Font("Sylfaen", Font.PLAIN, 19));

lblsub4_1.setBounds(237, 385, 136, 39);

panelid.add(lblsub4_1);
```

```
lbltot1 = new JLabel("");

lbltot1.setVerticalAlignment(SwingConstants.BOTTOM);

lbltot1.setBorder(new LineBorder(Color.LIGHT_GRAY));

lbltot1.setBackground(Color.LIGHT_GRAY); lbltot1.setFont(new

Font("Sylfaen", Font.PLAIN, 19)); lbltot1.setBounds(237, 434, 136, 39);

panelid.add(lbltot1);
```

```
lblperc1 = new JLabel("");

lblperc1.setVerticalAlignment(SwingConstants.BOTTOM);
```

```
lblperc1.setBorder(new LineBorder(Color.LIGHT_GRAY));

lblperc1.setBackground(Color.LIGHT_GRAY);

lblperc1.setFont(new Font("Sylfaen", Font.PLAIN, 19));

lblperc1.setBounds(237, 483, 136, 39);

panelid.add(lblperc1);

lbltitle.setForeground(Color.WHITE);

lbltitle.setFont(new Font("Sylfaen", Font.PLAIN, 39));

lbltitle.setBounds(49, 63, 365, 68);           panelid.add(lbltitle);
```

```
        lbltitle.setHorizontalAlignment(SwingConstants.CENTER);
```

```
        lblNewLabel = new JLabel("Parent's Signature");

lblNewLabel.setFont(new Font("Sylfaen", Font.PLAIN, 18));

lblNewLabel.setBounds(21, 562, 145, 32);

        panelid.add(lblNewLabel);
```

```
        lblNewLabel_1 = new JLabel("Principle's Signature");

        lblNewLabel_1.setFont(new Font("Sylfaen", Font.PLAIN, 18));

        lblNewLabel_1.setBounds(323, 565, 171, 30);

        panelid.add(lblNewLabel_1);
```

```
        lblNewLabel_2 = new JLabel("");

        lblNewLabel_2.setLabelFor(lblNewLabel_1);

        lblNewLabel_2.setIcon(new
```

TELESCOPE – ERP

```
ImageIcon("C:\\chck\\IMG_20180911_184230.jpg");
```

```
lblNewLabel_2.setBounds(322, 536, 157, 32);
```

```
panelid.add(lblNewLabel_2);
```

```
lblNewLabel_3 = new JLabel("");
```

```
lblNewLabel_3.setIcon(new ImageIcon("C:\\chck\\resultbg.jpg"));
```

```
lblNewLabel_3.setBounds(0, -42, 492, 697);
```

```
panelid.add(lblNewLabel_3);
```

```
}
```

```
//GEN-LAST:event_jButton3ActionPerformed
```

```
public void Showmarks()
```

```
{
```

```
try {
```

```
Class.forName("oracle.jdbc.driver.OracleDriver");
```

```
Connection
```

```
con
```

```
=
```

```
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl", "SYSTEM",  
"muthu405");
```

```
PreparedStatementpst = con.prepareStatement("select * from result where REGNO=?");
```

```
PreparedStatement pst1 = con.prepareStatement("select sname from reg where regno  
=?"); pst.setString(1, val);
```

```
pst1.setString(1, val);
```

```
        ResultSet rs = pst.executeQuery();

ResultSet rs2 =pst1.executeQuery();

while(rs2.next())

    {

        name = rs2.getString("sname");

System.out.println(name);

lblname1.setText(name);


        lbltitle.setText(name.toUpperCase()+"s"+" Result");

    }

    while (rs.next())

        {

            System.out.println("im here");


            String regno = rs.getString("regno");


            String sub1 = rs.getString("cpp");


            String sub2 = rs.getString("java");


            String sub3 = rs.getString("python");


            String sub4= rs.getString("se");


            lblregno.setText(regno);
```

TELESCOPE – ERP

```
        lblsub1_1.setText(sub2);

        lblsub2_1.setText(sub1);

        lblsub3_1.setText(sub3);

        lblsub4_1.setText(sub4);

        int total =
        Integer.parseInt(sub1)+Integer.parseInt(sub2)+Integer.parseInt(sub3)+Integer.parseInt(sub4);

        String

        tot=Integer.toString(total);        int per

        = (total*100)/400;

        lbltot1.setText(tot);

        String perc = Integer.toString(per);

        lblperc1.setText(perc+"%");

        System.out.println("im out here");

    }

    rs.close();

}

catch (Exception ex)

{

    System.out.println("New Error here");

    JOptionPane.showMessageDialog(null, ex.getMessage(), "Error",
    JOptionPane.ERROR_MESSAGE);
```

```
System.out.println(ex);
```

```
}
```

```
}
```

```
private void btnNewButtonActionPerformed(java.awt.event.ActionEvent evt)
```

```
{
```

```
try
```

```
{
```

```
Toolkit tkp = panelid.getToolkit();
```

```
PrintJob pjp = tkp.getPrintJob(this, null,
```

```
null); Graphics g = pjp.getGraphics();
```

```
panelid.print(g);
```

```
g.dispose();
```

```
pjp.end();
```

```
} catch
```

```
(Exception k)
```

```
{
```

```
}
```

```
}
```

```
public static void main(String[] args)
```

```
{
```

TELESCOPE – ERP

```
        try {

            for (javax.swing.UIManager.LookAndFeelInfo info :
                javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {
            java.util.logging.Logger.getLogger(Markscard.class.getName()).log(java.util.logging.
                Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {
            java.util.logging.Logger.getLogger(Markscard.class.getName()).log(java.util.logging.
                Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {
            java.util.logging.Logger.getLogger(Markscard.class.getName()).log(java.util.logging.
                Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
            java.util.logging.Logger.getLogger(Markscard.class.getName()).log(java.util.logging.
                Level.SEVERE, null, ex);
        }

        //</editor-fold>

        /* Create and display the form */

        java.awt.EventQueue.invokeLater(new Runnable() {
```

```

        public void run() {
            new
Markscard().setVisible(true);

        }

    });

}

}

```

MaqueePanel.java

```

package Telescope;

import java.awt.*;

import
java.awt.event.*;

import
javax.swing.*;

import
javax.swing.event.

*;

/**

* The MarqueePanel is used to scroll components from the right edge of the

* panel to the left edge. Scrolling is continuous. To simulate the scrolling

* of text you can simply add a JLabel to the panel.

*

* Various properties control the scrolling of the components on the panel.

* Changes to the properties are dynamic and will take effect the next time

```

TELESCOPE – ERP

* the components are scrolled.

*/

```
@SuppressWarnings("serial") public class MarqueePanel
```

```
extends JPanel          implements ActionListener,
```

```
AncestorListener, WindowListener
```

```
{
```

```
    protected
```

```
    boolean paintChildren;
```

```
    protected boolean scrollingPaused;
```

```
    protected int scrollOffset;
```

```
    protected int wrapOffset;
```

```
        private int preferredWidth = -1;
```

```
    private int scrollAmount;
```

```
        private int scrollFrequency;
```

```
        private boolean wrap =
```

```
        false; private int
```

```
        wrapAmount = 50; private
```

```
        boolean scrollWhenFocused
```

```
        = true; private Timer timer =
```

```
        new Timer(1000, this);
```

```
    /**
```

```
    * Convenience constructor that sets both the scroll frequency and
```

```
    * scroll amount to a value of 5.
```

```

    */

    public MarqueePanel()

    {

        this(5, 5);

    }

    /**

    *   Create an AnimatedIcon that will continuously cycle with the
    *   default (500ms).

    *

    *   @param component the component the icon will be painted on
    *   @param icons the Icons to be painted as part of the animation
    */

    public MarqueePanel(int scrollFrequency, int scrollAmount)

    {

        setScrollFrequency( scrollFrequency );

        setScrollAmount( scrollAmount );    setLayout(

        newBoxLayout(this, BoxLayout.X_AXIS) );

        addAncestorListener( this );

    }

    /**

    *   Translate the location of the children before they are painted so it
    *   appears they are scrolling left to right

    */

    @Override

```

TELESCOPE – ERP

```
public void paintChildren(Graphics g)
{
    // Need this so we don't see a flicker of the text before scrolling

    if (! paintChildren) return;

    // Normal painting as the components scroll right to left

    Graphics2D g2d = (Graphics2D)g;

    g2d.translate(-scrollOffset, 0);

    super.paintChildren(g);

    g2d.translate(scrollOffset, 0);

    // Repaint the start of the components on the right edge of the panel once
    // all the components are completely visible on the panel.
    // (Its like the components are in two places at the same time)

    if (isWrap())
    {
        wrapOffset    =    scrollOffset    -
super.getPreferredSize().width    - wrapAmount;

        g2d.translate(-wrapOffset, 0);

        super.paintChildren(g);

        g2d.translate(wrapOffset, 0);
    }
}
```

```
}
```

```
/*
```

- * The default preferred size will be half the size of the components added to
- * the panel. This will allow room for components to be scrolled on and off
- * the panel.

```
*
```

- * The default width can be overridden by using the `setPreferredWidth()` method.

```
*/
```

```
@Override
```

```
public Dimension getPreferredSize()
```

```
{
```

```
    Dimension d = super.getPreferredSize();
```

```
    d.width = (preferredWidth == -1) ? d.width / 2 : preferredWidth;
```

```
    return d;
```

```
}
```

```
@Override
```

```
public Dimension getMinimumSize()
```

```
{
```

```
    return getPreferredSize();
```

```
}
```

```
public int getPreferredWidth()
```

```
{
```

TELESCOPE – ERP

```
return preferredWidth;
```

```
}
```

```
/**
```

```
* Specify the preferred width on the panel. A value of -1 will cause the
```

```
* default preferred width size calculation to be used.
```

```
*
```

```
* @param preferredWidth preferred width of the panel in pixels
```

```
*/
```

```
public void setPreferredWidth(int preferredWidth)
```

```
{
```

```
    this.preferredWidth = preferredWidth;
```

```
    revalidate();
```

```
}
```

```
/**
```

```
* Get the scroll amount.
```

```
*
```

```
* @return the scroll amount in pixels
```

```
*/
```

```
public int getScrollAmount()
```

```
{
```

```
    return scrollAmount;
```

```
}
```

```
/**
```

```
* Specify the scroll amount. The number of pixels to scroll every time
```

```
* scrolling is done.
```

```
*
```

```
* @param scrollAmount scroll amount in pixels
```

```
*/
```

```
public void setScrollAmount(int scrollAmount)
```

```
{
```

```
    this.scrollAmount = scrollAmount;
```

```
}
```

```
/**
```

```
* Get the scroll frequency.
```

```
*
```

```
* @return the scroll frequency
```

```
*/
```

```
public int getScrollFrequency()
```

```
{
```

```
    return scrollFrequency;
```

```
}
```

```
/**
```

```
* Specify the scroll frequency. That is the number of times scrolling
```

```
* should be performed every second.
```

```
*
```

```
* @param scrollFrequency scroll frequency
```

```
*/
```

TELESCOPE – ERP

```
public void setScrollFrequency(int scrollFrequency)

{

    this.scrollFrequency = scrollFrequency;

    int delay = 1000 / scrollFrequency;

    timer.setInitialDelay( delay );        timer.setDelay(

delay );

    }

    /**

* Get the scroll only when visible property.

*

* @return the scroll only when visible value

*/

    public boolean isScrollWhenFocused()

    {

        return scrollWhenFocused;

    }

    /**

* Specify the scrolling property for unfocused windows.

*

* @param scrollWhenVisible when true scrolling pauses when the window
```

IDCARDGENERATINGSYSTEM.java

```
*          loses focus. Scrolling will continue when
```

```

*           the window regains focus. When false           *
           scrolling is continuous unless the window
*           is iconified.

*/

public void setScrollWhenFocused(boolean scrollWhenFocused)
{
    this.scrollWhenFocused = scrollWhenFocused;
}

/**

*           Get the wrap property.

*
*           @return the wrap value

*/

public boolean isWrap()
{
    return wrap;
}

/**

*           Specify the wrapping property. Normal scrolling is such that all the text
*           will scroll from left to right. When the last part of the text scrolls off
*           the left edge scrolling will start again from the right edge. Therefore
*           there is a time when the component is blank as nothing is displayed.
*           Wrapping implies that as the end of the text scrolls off the left edge  *
           the beginning of the text will scroll in from the right edge. So the end

```

TELESCOPE – ERP

```
*          and the start of the text is displayed at the same time.

*
*
*          @param wrap  when true the start of the text will scroll in from the right
*          edge while the end of the text is still scrolling off the left
*          edge. Otherwise the panel must be clear of text before
*          will begin again from the right edge.

*/

public void setWrap(boolean wrap)
{
    this.wrap = wrap;
}

/**

*          Get the wrap amount.

*
*          @return the wrap amount value

*/

public int getWrapAmount()
{
    return wrapAmount;
}

/**

*          Specify the wrapping amount. This specifies the space between the end
*          of the
```

```
*          text on the left edge and the start of the text from the right edge when
*
*          wrapping is turned on.
```

```

*
*          @param wrapAmount  the amount in pixels
```

```
*/
```

```
public void setWrapAmount(int wrapAmount)
```

```
{
```

```
    this.wrapAmount = wrapAmount;
```

```
}
```

```
/**
```

```
*          Start scrolling the components on the panel. Components will start
```

```
*          scrolling from the right edge towards the left edge.
```

```
*/
```

```
public void startScrolling()
```

```
{
```

```
    paintChildren = true;  scrollOffset= -
```

```
    getSize().width;
```

```
    timer.start();
```

```
}
```

```
/**
```

```
*          Stop scrolling the components on the panel. The components will be
```

```
*          cleared from the view of the panel
```

```
*/
```

TELESCOPE – ERP

```
public void stopScrolling()
```

```
{
```

```
    timer.stop();
```

```
    paintChildren = false;
```

```
    repaint();
```

```
}
```

```
/**
```

```
*          The components will stop scrolling but will remain visible
```

```
*/
```

```
public void pauseScrolling()
```

```
{
```

```
    if (timer.isRunning())
```

```
    {
```

```
        timer.stop();
```

```
        scrollingPaused = true;
```

```
    }
```

```
}
```

```
/**
```

```
*          The components will resume scrolling from where scrolling was  
          stopped.
```

```
*/
```

```
public void resumeScrolling()
```

```
{
```

```

        if (scrollingPaused)
        {
            timer.restart();

            scrollingPaused = false;
        }
    }
}

```

// Implement ActionListener

```

/**
 * Adjust the offset of the components on the panel so it appears that
 * they are scrolling from right to left.
 */
@Override
public void actionPerformed(ActionEvent ae)
{
    scrollOffset = scrollOffset +
scrollAmount;          int width =
super.getPreferredSize().width;

    if (scrollOffset > width)
    {
        scrollOffset = isWrap() ? wrapOffset +
scrollAmount : - getSize().width;
    }
}

```

TELESCOPE – ERP

```
        repaint();

    }

// ImplementAncestorListener

    /**
 * Get notified when the panel is added to a Window so we can use a
 * WindowListener to automatically start the scrolling of the components.
 */

    @Override

    public void ancestorAdded(AncestorEvent e)

    {

        SwingUtilities.windowForComponent( this ).addWindowListener( this );

    }


    @Override

    public void ancestorMoved(AncestorEvent e) {}

    @Override

    public void ancestorRemoved(AncestorEvent e) {}


// ImplementWindowListener

    @Override

    public void windowActivated(WindowEvent e)

    {

        if (isScrollWhenFocused())
```

```
        resumeScrolling();  
    }  
  
    @Override  
    public void windowClosed(WindowEvent e)  
    {  
        stopScrolling();  
    }  
  
    @Override  
    public void windowClosing(WindowEvent e)  
    {  
        stopScrolling();  
    }  
  
    @Override  
    public void windowDeactivated(WindowEvent e)  
    {  
        if (isScrollWhenFocused())  
            pauseScrolling();  
    }  
  
    @Override  
    public void windowDeiconified(WindowEvent e) {  
        resumeScrolling();  
    }  
}
```

TELESCOPE – ERP

```
@Override
```

```
public void windowIconified(WindowEvent e)
```

```
{
```

```
    pauseScrolling();
```

```
}
```

```
@Override
```

```
public void windowOpened(WindowEvent e)
```

```
{
```

```
    startScrolling();
```

```
}
```

```
public void setBackground(int translucent) {
```

```
    // TODO Auto-generated method stub
```

```
}
```

```
}
```

Profile.java

```
package Telescope;
```

```
import java.awt.BorderLayout;
```

```
import java.awt.Color; import
```

```
java.awt.Dimension; import
```

```
java.awt.Font; import
```

```
java.awt.Image; import
java.awt.Toolkit; import
java.awt.event.ActionEvent;
import
java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.DriverManager;
import
java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;

import
javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import
javax.swing.JOptionPane;
import
javax.swing.JScrollPane;
import
javax.swing.JSpinner;
import javax.swing.JTable;
import
javax.swing.JTextField;
```

TELESCOPE – ERP

```
import
javax.swing.ScrollPaneCons
tants; import
javax.swing.SpinnerListMo
del; import
javax.swing.SwingConstant
s; import
javax.swing.WindowConsta
nts; import
javax.swing.table.DefaultTa
bleModel; import
javax.swing.JMenuItem;

@SuppressWarnings("serial") public class Profile
extends JFrame implements ActionListener
{
    //Declaring Components
    JLabel lbusername, lbmail, lbpass, lbcountry, lbstate ,lbphone,regnumber,lbviewname,regpic
    ;
    JTextField tfusername, tfmail, tfpass, tfcountry, tfstate, tfphone,regtext,tflast;
    JButton btnsubmit,btnclear;
    JSpinner category;

    Image img;

    String Teachit ="Teacher";
```

```
String studenthi ="Student";

JButtonresultbtn;

JButtonaddmarkbtn;

JButtontakeattenbtn;

JButtonviewattenbtn;

JButtontresult;

JButtonbtnupdate;


//Creating Constructor for initializing JFrame components

Profile()

{

    getContentPane().setForeground(Color.BLACK);

getContentPane().setFont(new Font("Arial", Font.BOLD, 12));


//Providing Title


//super("Welcome");

setIconImage(Toolkit.getDefaultToolkit().getImage("C:\\Users\\Surya\\Desktop\\pics\\
\\telescope_icon.png"));


this.setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE);
```

TELESCOPE – ERP

```
regnumber = new JLabel("Registration No:");
```

```
regnumber.setBounds(181, 66, 117, 35);
```

```
regnumber.setFont(new Font("Sylfaen", Font.BOLD, 15));
```

```
regnumber.setForeground(Color.WHITE);
```

```
regtext = new JTextField(50);
```

```
regtext.setBounds(333, 69, 158, 30);
```

```
category = new JSpinner(); category.setBounds(516, 28, 196, 30);
```

```
category.setModel(new SpinnerListModel(new String[] {"Student",  
"Teacher"}));
```

```
lbusername = new JLabel("User Name:");
```

```
lbusername.setBounds(181, 104, 117, 35);
```

```
lbusername.setFont(new Font("Sylfaen", Font.BOLD, 15));
```

```
lbusername.setForeground(Color.WHITE);
```

```
tfusername = new JTextField(50);
```

```
tfusername.setBounds(333, 112, 159, 28);
```

```
lbmail = new JLabel("Mail-id:");
```

```
lbmail.setBounds(181, 142, 117, 35);
```

```
lbmail.setFont(new Font("Sylfaen", Font.BOLD,  
15)); lbmail.setForeground(Color.WHITE);
```

```
tfmail = new JTextField(100);
```

```
tfmail.setBounds(333, 150, 159, 29);
```

```
lbpass = new JLabel("PassWord:");
```

```
lbpass.setBounds(181, 180, 117, 35);
```

```
lbpass.setFont(new Font("Sylfaen", Font.BOLD,  
15)); lbpass.setForeground(Color.WHITE);
```

```
tfpass = new JTextField(50);
```

```
tfpass.setBounds(333, 188, 159, 26);
```

```
lbcountry = new JLabel("Country:");
```

```
lbcountry.setBounds(181, 218, 117, 35);
```

```
lbcountry.setFont(new Font("Sylfaen", Font.BOLD,  
15)); lbcountry.setForeground(Color.WHITE);
```

```
tfcountry = new JTextField(50);
```

```
tfcountry.setBounds(333, 226, 159, 30); lbstate
```

```
= new JLabel("State:"); lbstate.setBounds(181,
```

```
256, 117, 35); lbstate.setFont(new
```

```
Font("Sylfaen", Font.BOLD, 15));
```

```
lbstate.setForeground(Color.WHITE);
```

```
tfstate = new JTextField(50);
```

```
tfstate.setBounds(333, 264, 159, 30); lbphone =  
new JLabel("Phone:"); lbphone.setBounds(181,
```

TELESCOPE – ERP

```
294, 117, 35); lbphone.setFont(new
```

```
Font("Sylfaen", Font.BOLD, 15));
```

```
lbphone.setForeground(Color.WHITE);
```

```
tfphone = new JTextField(50);
```

```
tfphone.setBounds(333, 302, 159, 31);
```

```
tflast = new JTextField(50);
```

```
tflast.setBounds(160, 290, 100, 20);
```

```
getContentPane().setLayout(null);
```

```
tresult = new JButton("View Marks");
```

```
tresult.addActionListener(new ActionListener() {
```

```
public void actionPerformed(ActionEvent e)
```

```
{
```

```
new Teacher();
```

```
}
```

```
});
```

```
btnupdate = new JButton("Update");
```

```
btnupdate.addActionListener(new ActionListener() {
```

```
public void actionPerformed(ActionEvent e)
```

```
{
```

```

        //JOptionPane.showMessageDialog(null, "Record not found");
int y=0;

        String regno = regtext.getText();

        String name =tfusername.getText();

        String email =tfmail.getText();

        String pass = tfpass.getText();

        String country =tfcountry.getText();

        String state =tfstate.getText();

        String phone =tfphone.getText();

        y++;

        if(category.getValue()=="Student")

        {

try

        {

            Class.forName("oracle.jdbc.driver.OracleDriver");

            Connection con =

            DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl", "SYSTEM",

            "muthu405");

            PreparedStatementps = con.prepareStatement("update reg set

            regno=?,sname=?,mail=?,pass=?,country=?,state=?,phone=? where

            regno="+regno); ps.setString(2, name); ps.setString(3,

            email); ps.setString(4, pass); ps.setString(5, country);

            ps.setString(6, state);

            ps.setString(7, phone);

            ps.setString(1, regno);

```

TELESCOPE – ERP

```
//ps.setString(4, s5);

ps.executeUpdate();

if (y > 0)

{

    JOptionPane.showMessageDialog(btnupdate, "Data updated successfully");    if
(JOptionPane.showConfirmDialog(null, "Do you want to Preview Profile?",
"Conformation",

    JOptionPane.YES_NO_OPTION) == JOptionPane.YES_OPTION)

    {

        dispose();

        Profile r =new Profile();

        r.setVisible(true);

        r.regtext.setText(regno);

        r.btnsubmit.doClick();

        tfusername.setText("");

regtext.setText("");

tfmail.setText("");

tfpass.setText("");

tfcountry.setText("");

tfstate.setText("");

tfphone.setText("");

        regpic.setText("");

    }

    else

    {
```

```

        //dispose();
    }

    }

ps.close();

    }

    catch(Exception i)

    {

        System.out.print(i);

        //JOptionPane.showMessageDialog(null,
"Record "+regno+" not found");

    }

    }

    else if(category.getValue()=="Teacher")

    {

        try

        {

            Class.forName("oracle.jdbc.driver.OracleDriver");

            Connection con =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl", "SYSTEM",
"muthu405");

            PreparedStatementps = con.prepareStatement("update regtec set
tid=?,tname=?,tmail=?,pass=?,country=?,state=?,phone=? where tid="+regno);

            ps.setString(2, name);

            ps.setString(3, email);

            ps.setString(4, pass); ps.setString(5,
country); ps.setString(6, state);

```

TELESCOPE – ERP

```
ps.setString(7, phone);

ps.setString(1, regno);

        //ps.setString(4, s5);

ps.executeUpdate();

        if (y > 0)

        {

JOptionPane.showMessageDialog(btnupdate, "Data updated successfully");    if
(JOptionPane.showConfirmDialog(null, "Do you want to Preview Profile?",
"Conformation",

JOptionPane.YES_NO_OPTION) == JOptionPane.YES_OPTION)

        {

                dispose();

                Profile r =new Profile();

                r.setVisible(true);

                r.regtext.setText(regno);

                r.category.setValue("Teacher");

                r.btnsubmit.doClick();

                tfusername.setText("");

regtext.setText("");

                tfmail.setText("");

                tfpass.setText("");

                tfcountry.setText("");

                tfstate.setText("");

                tfphone.setText("");

                regpic.setText("");
```

```

        }
        else
        {
            //dispose();
        }
    }

    ps.close();

    }

    catch(Exception i)
    {

        System.out.print(i);
        //JOptionPane.showMessageDialog(null,
"Record "+regno+" not found");

    }
}

}

});

```

```

JMenuItem mntmNewMenuItem = new JMenuItem("View
Marks"); mntmNewMenuItem.addActionListener(new
ActionListener() { public void actionPerformed(ActionEvent e)

```

```

{

    new Teacher();

}

});

```

```

mntmNewMenuItem.setSelectedIcon(new

```

TELESCOPE – ERP

```
ImageIcon("C:\\Users\\Surya\\Desktop\\bg\\idlogo.png"));

mntmNewMenuItem.setHorizontalAlignment(SwingConstants.CEN
TER); mntmNewMenuItem.setFont(new Font("Sylfaen",
Font.PLAIN, 17));

mntmNewMenuItem.setForeground(Color.BLACK);

mntmNewMenuItem.setBounds(0, 0, 122, 24);

getContentPane().add(mntmNewMenuItem); btnupdate.setFont(new
Font("Sylfaen", Font.PLAIN, 17)); btnupdate.setBounds(483, 353,
115, 30); getContentPane().add(btnupdate); tresult.setFont(new
Font("Sylfaen", Font.PLAIN, 18)); tresult.setBounds(516, 71, 196,
43); getContentPane().add(tresult);

btnsubmit = new JButton("Submit");

btnsubmit.setFont(new Font("Sylfaen", Font.PLAIN,
17)); btnsubmit.setBounds(183, 351, 115, 30);

getContentPane().add(btnsubmit);

btnsubmit.addActionListener(this);

btnclear = new JButton("Clear");

btnclear.setFont(new Font("Sylfaen",
Font.PLAIN, 17)); btnclear.setBounds(336, 352,
115, 30); getContentPane().add(btnclear);

btnclear.addActionListener(this); regpic = new
JLabel(""); regpic.setBounds(6, 75, 171, 245);

getContentPane().add(regpic);    //Add
components to the JFrame
```

```
getContentPane().add(regnumber);  
getContentPane().add(regtext);  
getContentPane().add(lbusername);  
getContentPane().add(tfusername);  
getContentPane().add(lbmail);  
getContentPane().add(tfmail);  
getContentPane().add(lbpass);  
getContentPane().add(tfpas);  
getContentPane().add(lbcountry);  
getContentPane().add(tfcountry);  
getContentPane().add(lbstate);  
getContentPane().add(tfstate);  
getContentPane().add(lbphone);  
getContentPane().add(tfphone);  
getContentPane().add(btnsubmit);  
getContentPane().add(btnclear);  
getContentPane().add(category);
```

```
setSize(756, 523);
```

```
//Set TextField Editable False
```

```
tfusername.setEditable(true);
```

TELESCOPE – ERP

```
resultbtn = new JButton("View Marks");

resultbtn.setBounds(516, 75, 196, 39);

resultbtn.addActionListener(new ActionListener() {

    @Override                public void

    actionPerformed(ActionEvent e)

    {

        Markscard r = new Markscard();

        r.lblregno.setText(regtext.getText());

        r.val =regtext.getText();

        r.Showmarks();

        if(r.lblsub1_1.getText().isEmpty()&&r.lblsub2_1.getText().isEmpty()

&&r.lblsub3_1.getText().isEmpty()&&r.lblsub4_1.getText().isEmpty())

        {

            JOptionPane.showMessageDialog(null, "NO RECORD FOUND"); }

        r.setVisible(true);

    }

});

resultbtn.setFont(new Font("Sylfaen", Font.PLAIN, 17));

getContentPane().add(resultbtn);

lblviewname = new JLabel("");

lblviewname.setForeground(Color.WHITE);

lblviewname.setHorizontalAlignment(SwingConstants.CENTER);
```

```
lblviewname.setFont(new Font("Sylfaen", Font.PLAIN, 16));
```

```
lblviewname.setBounds(183, 13, 308, 43);
```

```
getContentPane().add(lblviewname);
```

```
addmarkbtn = new JButton("Update Marks");
```

```
addmarkbtn.setBounds(516, 119, 196, 39);
```

```
addmarkbtn.addActionListener(new ActionListener() {
```

```
    public void actionPerformed(ActionEvent e)
```

```
    {
```

```
        Marks mark = new Marks();
```

```
mark.setVisible(true);
```

```
    }
```

```
});
```

```
addmarkbtn.setFont(new Font("Sylfaen", Font.PLAIN, 17));
```

```
getContentPane().add(addmarkbtn);
```

```
takeattenbtn = new JButton("Update Attendance");
```

```
takeattenbtn.setBounds(516, 168, 196, 40);
```

```
takeattenbtn.addActionListener(new ActionListener() {
```

```
    public void actionPerformed(ActionEvent e)
```

```
    {
```

```
        Attendance att =new Attendance();
```

```
att.startUI();
```

```
    }
```

```
});
```

TELESCOPE – ERP

```
takeattenbtn.setFont(new Font("Sylfaen", Font.PLAIN, 17));

getContentPane().add(takeattenbtn);


viewattenbtn = new JButton("View Attendance");

viewattenbtn.setBounds(516, 218, 196, 36);

viewattenbtn.addActionListener(new ActionListener() {

    public void actionPerformed(ActionEvent e)

        {

            int reg = 0;

            JFrame frame1;

            JTable table;

            String[] columnNames = {"RegNo", "Student Name", "JAVA", "DCN"};


            frame1 = new JFrame("Student Record");


            frame1.setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE);


            frame1.getContentPane().setLayout(new BorderLayout());


            DefaultTableModel model = new DefaultTableModel();


            model.setColumnIdentifiers(columnNames);
```

```
table = new JTable();
```

```
table.setModel(model);
```

```
table.setAutoResizeMode(JTable.AUTO_RESIZE_ALL_COLUMNS);
```

```
table.setFillsViewportHeight(true);
```

```
JScrollPane scroll = new JScrollPane(table);
```

```
scroll.setHorizontalScrollBarPolicy(  
ScrollPaneConstants.HORIZONTAL_SCROLLBAR_AS_NEEDED);
```

```
scroll.setVerticalScrollBarPolicy(  

```

```
ScrollPaneConstants.VERTICAL_SCROLLBAR_AS_NEEDED);
```

```
if(true)
```

```
{
```

```
try
```

```
{
```

```
//Connecting to Database.
```

```
Class.forName("oracle.jdbc.driver.OracleDriver");
```

```
Connection
```

```
con
```

```
=
```

```
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl", "SYSTEM",  
"muthu405");
```

```
PreparedStatement st3 = con.prepareStatement("SELECT * FROM (SELECT
```

TELESCOPE – ERP

REGNO FROM Attendance ORDER BY REGNO DESC) WHERE ROWNUM

= 1"); ResultSet r = st3.executeQuery();

if(true)

{

while(r.next())

{

System.out.println("im in while");

String test = r.getString(1);

reg = Integer.parseInt(test);

System.out.println(reg);

}

}

st3.close();

r.close();

}

catch(Exception a)

{

System.out.println(a);

}

for(int i = 100;i<=reg;i++)

{

```
String regno ="";
```

```
String sname ="";
```

```
String java ="";
```

```
String dcn ="";
```

```
try {
```

```
        PreparedStatementpst;
```

```
        Connection con =
```

```
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl", "SYSTEM",  
"muthu405");        pst = con.prepareStatement("select * from Attendance where
```

```
REGNO=" + i +
```

```
""");
```

```
        ResultSet rs = pst.executeQuery();
```

```
        int
```

```
        j = 0;
```

```
        if(true)
```

```
        {
```

```
            while (rs.next())
```

```
            {
```

```
                regno = rs.getString(1);
```

TELESCOPE – ERP

```
sname = rs.getString(2);
```

```
java = rs.getString(3);
```

```
dcn = rs.getString(4);
```

```
model.addRow(new Object[]{regno,sname,java, dcn});
```

```
j++;
```

```
        }
    }

    if (j < 1)
    {
        //JOptionPane.showMessageDialog(null, "Record " + reg+" not
found");
    }

    if (j == 1)
    {
        System.out.println(i + " Record Found");
    }
    else
    {
        System.out.println(i + " Records Found");
    }
    pst.close(); rs.close();
}

catch (Exception ex)
{
```

TELESCOPE – ERP

TELESCOPE – RP

```
JOptionPane.showMessageDialog(null, ex.getMessage(),
JOptionPane.ERROR_MESSAGE);

                                "Error",

                                }

                                frame1.getContentPane().add(scroll);

                                frame1.setVisible(true);

                                frame1.setSize(700, 500);

                                }

                                }

                                }

                                });

viewattenbtn.setFont(new Font("Sylfaen", Font.PLAIN,
17)); getContentPane().add(viewattenbtn);

JLabellblNewLabel = new JLabel("");

lblNewLabel.setIcon(new
ImageIcon("C:\\chck\\profile.jpg"));

lblNewLabel.setBounds(0, 22, 742, 464);

getContentPane().add(lblNewLabel);
```

```

        int x = (screenSize.width-getSize().width)/2;

int y = (screenSize.height-getSize().height)/2;

setLocation(x,y);


        // Make JWindow visible setVisible(true);


    }


@Override public void
actionPerformed(ActionEvent e)
{

    if (e.getSource() == btnsubmit)
    {
        try{
            if(category.getValue()=="Student")
            {
                Class.forName("oracle.jdbc.driver.OracleDriver");

                Connection con =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl", "SYSTEM",
"muthu405");

                Statement st = con.createStatement();

ResultSetrs = st.executeQuery("select * from reg where regno = '"+regtext.getText()+"'");

                if(rs.next())

```

```

    {

byte[] img = rs.getBytes("Image");


//Resize TheImageIcon

ImageIcon image = new ImageIcon(img);

    Image im = image.getImage();

    Image myImg = im.getScaledInstance(regpic.getWidth(),
regpic.getHeight(),Image.SCALE_SMOOTH); ImageIconnewImage = new
ImageIcon(myImg); regpic.setIcon(newImage);

    }


con.close();

rs.close(); st.close();

    }

else

if(Integer.parseInt(regtext.getText())>=1000&&category.getValue()=="Teacher")

    {

        if(category.getValue()=="Student")

            {

JOptionPane.showMessageDialog(null, "Data Not Found");

            }

        else

            {

                Class.forName("oracle.jdbc.driver.OracleDriver");

```

TELESCOPE – ERP

```
Connection con =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl", "SYSTEM",
"muthu405");

Statement st = con.createStatement();

ResultSet rs = st.executeQuery("select * from regtec where tid =
"+regtext.getText()+"");

if(rs.next()) { byte[] img =
rs.getBytes("Image");

//Resize TheImageIcon

ImageIcon image = new ImageIcon(img);

Image im = image.getImage();

Image myImg = im.getScaledInstance(regpic.getWidth(),
regpic.getHeight(),Image.SCALE_SMOOTH); ImageIconnewImage = new
ImageIcon(myImg); regpic.setIcon(newImage);

}

con.close();

rs.close(); st.close();

}

}
```

TELESCOPE – ERP

```
}catch(Exception ex)

    {

ex.printStackTrace();

    }

try

    {

        Class.forName("oracle.jdbc.driver.OracleDriver");

        Connection con =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl", "SYSTEM",
"muthu405");

PreparedStatementst = con.prepareStatement("select * from reg where regno=?");

PreparedStatement tec1=con.prepareStatement("select * from regtec where TID=?");


        String str = regtext.getText();


st.setString(1, str);

        tec1.setString(1, str); //Executing

Query if((String)
category.getValue()==studenthi)

    {

ResultSet rs1 = st.executeQuery();

System.out.println("fetching students");

while (rs1.next())

    {

        String regno = rs1.getString(7);
```

TELESCOPE – ERP

```
String sname = rs1.getString(1);
```

```
String mail = rs1.getString(2);
```

```
String pass = rs1.getString(3);
```

```
String country = rs1.getString(4);
```

```
String state = rs1.getString(5);
```

```
String phone = rs1.getString(6);
```

```
//Sets Records in TextFields.
```

```
//lblviewname.setVisible(true); lblviewname.setText(sname.toUpperCase()
```

```
+ ""s" + " Profile");
```

```
regtext.setText(regno);
```

TELESCOPE – ERP

TELESCOPE – ERP

```
tfusername.setText(sname);
```

```
tfmail.setText(mail);
```

```
tfpass.setText(pass);
```

```
tfcountry.setText(country);
```

```
tfstate.setText(state);
```

```
tfphone.setText(phone);
```

```
    }
```

```
con.close();
```

```
    rs1.close();
```

```
    }
```

```
        //Create Exception Handler        else
```

```
if((String) category.getValue() == Teachit)
```

```
{
```

```
System.out.println("fetching teachers");
```

TELESCOPE – ERP

```
ResultSet rs3 = tec1.executeQuery();
```

```
    while (rs3.next())
```

```
    {
```

```
        String teacherid = rs3.getString(7);
```

```
        String tname = rs3.getString(1);
```

```
        String tmail = rs3.getString(2);
```

```
        String tpass = rs3.getString(3);
```

```
        String tcountry = rs3.getString(4);
```

```
        String tstate = rs3.getString(5);
```

```
        String tphone = rs3.getString(6);
```

```
        //Sets Records in TextFields.
```

```
        //lblviewname.setVisible(true); lblviewname.setText(tname.toUpperCase()
```

```
+ "'s" + " Profile");
```

```
regtext.setText(teacherid);
```

```
tfusername.setText(tname);
```

```
tfmail.setText(tmail);
```

TELESCOPE – ERP

```
tfpass.setText(tpass);
```

```
tfcountry.setText(tcountry);
```

```
tfstate.setText(tstate);
```

```
tfphone.setText(tphone);
```

```
    }
```

```
rs3.close();
```

```
    }
```

```
st.close();
```

```
tec1.close();
```

```
    }
```

```
        catch (Exception ex)
```

```
        {
```

```
System.out.println(ex);
```

```
    }
```

```
    }
```

```
else
```

TELESCOPE – ERP

```
{

    regtext.setText("");

    tfusername.setText("");

    tfmail.setText("");

    tfpass.setText("");

    tfcountry.setText("");

    tfstate.setText("");

    tfphone.setText(""); regpic.setText("");
}

}

//Running Constructor public static
void main(String args[])

{

    try {

        for (javax.swing.UIManager.LookAndFeelInfo info :
            javax.swing.UIManager.getInstalledLookAndFeels()) {

            if ("Nimbus".equals(info.getName())) {

                javax.swing.UIManager.setLookAndFeel(info.getClassName());

                break;

            }

        }

    }

    catch (ClassNotFoundException ex) {

        java.util.logging.Logger.getLogger(Main.class).log(Level.SEVERE, null, ex);

    }

    catch (InstantiationException ex) {

        java.util.logging.Logger.getLogger(Main.class).log(Level.SEVERE, null, ex);

    }

    catch (IllegalAccessException ex) {

        java.util.logging.Logger.getLogger(Main.class).log(Level.SEVERE, null, ex);

    }

    finally {

    }

}
```

TELESCOPE – ERP

```
    }

    }

    } catch (ClassNotFoundException ex) {
java.util.logging.Logger.getLogger(Profile.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

    } catch (InstantiationException ex) {
java.util.logging.Logger.getLogger(Profile.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

    } catch (IllegalAccessException ex) {
java.util.logging.Logger.getLogger(Profile.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
java.util.logging.Logger.getLogger(Profile.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

    }

//</editor-fold>

/* Create and display the form */ java.awt.EventQueue.invokeLater(new
Runnable() {

    public void run() {

        new Profile().setVisible(true);

    }

});

;

}
```

TELESCOPE – ERP

WELCOME.java

```
package Telescope; import

java.awt.Color; import

java.awt.Dimension; import

java.awt.Font; import java.awt.Image;

import java.awt.SystemColor; import

java.awt.Toolkit; import

java.awt.event.ActionEvent; import

java.awt.event.ActionListener; import

java.util.Timer; import

java.util.TimerTask; import

javax.swing.ImageIcon; import

javax.swing.JButton; import

javax.swing.JFrame; import

javax.swing.JLabel; import

javax.swing.JOptionPane; import

javax.swing.WindowConstants; import

java.awt.event.MouseAdapter; import

java.awt.event.MouseEvent;

@SuppressWarnings("serial") public

class Welcome extends JFrame

{

    static JButton btnNewButton = new JButton("New button");

    @SuppressWarnings({ "unused" })
```

TELESCOPE – ERP

Welcome()

```
{  
  
    setTitle("TeleScope ERP");  
  
    setIconImage(Toolkit.getDefaultToolkit().getImage("C:\\Users\\Surya\\Desktop\\bg\\telescope.png"));  
  
  
    getContentPane().setBackground(Color.WHITE);  
  
    Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();  
  
    int x= (int) screenSize.getWidth();   int y = (int) screenSize.getHeight();  
  
  
    setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE);  
  
  
    Timer timer;  
  
    timer = new Timer();  
  
    MarqueePanelprov =new MarqueePanel();  
  
    getContentPane().setLayout(null); MarqueePanel m =new MarqueePanel();  
  
  
        m.setBounds(132, 17, 502, 53);  
  
    getContentPane().add(m);  
  
  
  
  
  
  
  
  
  
        m.setVisible(true);  
  
  
  
  
  
  
        m.setWrap(true);  
  
        m.setBackground(Color.TRANSLUCENT);  
  
        m.setScrollFrequency(25);
```

TELESCOPE – ERP

```
JLabelMlabel = new JLabel("TELESCOPE ERP-Join the VISION");

Mlabel.setBackground(SystemColor.activeCaption);

        m.add(Mlabel);

        Mlabel.setFont(new Font("Sylfaen", Font.PLAIN, 22));

        //Mlabel.setBackground(Color.GRAY);

Mlabel.setForeground(new Color(220, 20, 60));


        JLabel lblNewLabel_1 = new JLabel("");

lblNewLabel_1.setBounds(250, 80, 275, 92);

getContentPane().add(lblNewLabel_1);        lblNewLabel_1.setIcon(new

ImageIcon("C:\\Users\\Surya\\Desktop\\bg\\logo.png"));

        JButtonloginbtn = new JButton("LOGIN");

        loginbtn.addMouseListener(new MouseAdapter() {

            @Override

            public void mouseEntered(MouseEvent e) {

                loginbtn.setForeground(Color.RED);

            }

            @Override

            public void mouseExited(MouseEvent e)

            {

                loginbtn.setForeground(Color.black);

            }

        });
```

TELESCOPE – ERP

```
loginbtn.setBounds(70, 285, 186, 45);

getContentPane().add(loginbtn);          loginbtn.setForeground(new
Color(0, 0, 0));          loginbtn.setFont(new Font("Comic Sans MS",
Font.PLAIN, 18));

loginbtn.setBackground(SystemColor.activeCaption);
```

```
//Action Performed on Login Button.
```

```
loginbtn.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e)

    {

        Login log =new Login();

        log.setVisible(true);

    }

});
```

```
//Creating labels and buttons
```

```
JButtonRegbtn = new JButton("REGISTRATION ");

Regbtn.addMouseListener(new MouseAdapter() {

    @Override

    public void mouseEntered(MouseEvent e)

    {

        Regbtn.setForeground(Color.RED);

    }

});
```

TELESCOPE – ERP

@Override

```
public void mouseExited(MouseEvent e) {
```

```
    Regbtn.setForeground(Color.black);
```

```
}
```

```
});
```

```
Regbtn.setBounds(70, 197, 186, 53);
```

```
getContentPane().add(Regbtn); Regbtn.setForeground(new Color(0,  
0, 0));
```

```
Regbtn.setFont(new Font("Comic Sans MS", Font.PLAIN, 18));
```

```
Regbtn.setIcon(null);
```

```
Regbtn.setBackground(SystemColor.activeCaption);
```

```
//Action Performed on Registration Button.
```

```
Regbtn.addActionListener(new ActionListener() {
```

```
    public void actionPerformed(ActionEvent e)
```

```
    {
```

```
        Registration reg =new Registration();
```

```
        reg.setVisible(true);
```

TELESCOPE – ERP

```
}

});

JButtonAboutbtn = new JButton("ABOUT\r\n");

Aboutbtn.addMouseListener(new MouseAdapter() {

    @Override

    public void mouseEntered(MouseEvent e) {

        Aboutbtn.setForeground(Color.RED);

    }

    @Override

    public void mouseExited(MouseEvent e)

    {

        Aboutbtn.setForeground(Color.black);

    }

});

Aboutbtn.setBounds(70, 373, 186, 45);

getContentPane().add(Aboutbtn);

Aboutbtn.setForeground(new Color(0, 0, 0));

Aboutbtn.setFont(new Font("Comic Sans MS", Font.PLAIN, 18));

Aboutbtn.setBackground(SystemColor.activeCaption);

//Action Performed on Profile Button.

Aboutbtn.addActionListener(new ActionListener() {

    public void actionPerformed(ActionEvent e)

    {

        JOptionPane.showMessageDialog(Aboutbtn,"This Project is
```

TELESCOPE – ERP

Developed by S.Surya, Sasithar M and Prerana for the purpose of ease access to Educational ERP");

```
}
```

```
});
```

```
JLabellblNewLabel = new JLabel("New label");
```

```
lblNewLabel.setIcon(new  
ImageIcon("C:\\Users\\Surya\\Desktop\\bg\\Prov.png"));
```

```
lblNewLabel.setBounds(406, 182, 363, 278);
```

```
getContentPane().add(lblNewLabel);
```

```
JLabelprv = new JLabel("<html>An investment in knowledge<br> pays the  
best interest !</html> ");
```

```
prv.setBackground(Color.white);
```

```
prv.setForeground(Color.black);          prv.setFont(new Font("Comic Sans MS",
```

```
Font.BOLD | Font.ITALIC, 16 ));          prv.setBounds(465, 227, 275, 198);
```

```
getContentPane().add(prv);          btnNewButton.setVisible(false);
```

```
/*  btnNewButton.addActionListener(new ActionListener() {
```

```
int clicks = 0;
```

```
public void actionPerformed(ActionEvent e)
```

```
{
```

TELESCOPE – ERP

```
        }

    });*/

    //Slideshow Quote.

    Timer timer1 = new Timer();

    TimerTaskmyTask = new TimerTask() {

        public void run() {

            btnNewButton.setVisible(false);

            btnNewButton.doClick();

        }

    };

    timer.schedule(myTask, 5000, 3000);

    JLabel lblNewLabel_3 = new JLabel("New label");

    lblNewLabel_3.setFont(new Font("Comic Sans MS", Font.PLAIN, 18));

    lblNewLabel_3.setIcon(new ImageIcon("D:\\Downloads\\pexels-johannesplenio-1103970.jpg"));

    lblNewLabel_3.setBounds(0, 0, 855, 666);

    getContentPane().add(lblNewLabel_3);

    btnNewButton.addActionListener(new ActionListener() {
```

TELESCOPE – ERP

```
private int counter = 0;           int clicks=1;

public void actionPerformed(ActionEvent e)

    {

        counter++;

        switch (++clicks)

            {

                case 1:

prv.setText("<html>Live as if you were to die tomorrow. <br>Learn as if you were to live
forever." This inspirational quote from Gandhi");

                break;

                case 2:

prv.setText("Learning never exhausts the mind_Davinci");

                break;

                case 3:

prv.setText("<html>I would rather entertain and hope that people learned something <br>
than educate people and hope they were entertained._Walt Disney");

                break;

                case 4:

prv.setText("surya");

                break;

                case 5:

prv.setText("vicky");

                break;

                case 6:
```

TELESCOPE – ERP

```
prv.setText("hi");
```

```
    break;
```

```
case 7:
```

```
prv.setText("hello");
```

```
    clicks=1;
```

```
    break;
```

```
    }
```

```
    }
```

```
});
```

```
    btnNewButton.setBounds(350, 306, 85, 30);
```

```
getContentPane().add(btnNewButton);
```

```
    /*Registration registration = new Registration();
```

```
registration.setBounds(245, 98, 200, 50);
```

```
getContentPane().add(registration);*/
```

```
setVisible(true);
```

```
setSize(869,703);
```

```
}
```

TELESCOPE – ERP

Image image;

@SuppressWarnings("unused")

public void run()

{

try

{

// Get the width of the screen, so that the frame has to go till end!

int width=Toolkit.getDefaultToolkit().getScreenSize().width; int

height=Toolkit.getDefaultToolkit().getScreenSize().height;

// Start the loop for(int

i=0;i<=height;i++)

{

// If i value equals width (i.e. reaches end of screen), start from starting of the screen
(i.e. i=0) so looped forever

if(i==height) i=0;

// Set location, x-coordinate changes but y-coordinate is constant setLocation(55,i);

// Get the effect!

Thread.sleep(1);

}

TELESCOPE – ERP

```
} catch(Exception e){} finally {  
  
    }  
  
}
```

```
public static void main(String[] args)
```

```
{
```

```
    try {  
        for (javax.swing.UIManager.LookAndFeelInfo info :  
            javax.swing.UIManager.getInstalledLookAndFeels()) {  
            if ("Nimbus".equals(info.getName())) {  
                javax.swing.UIManager.setLookAndFeel(info.getClassName());  
                break;  
            }  
        }  
    } catch (ClassNotFoundException ex) {  
        java.util.logging.Logger.getLogger(Welcome.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);  
    } catch (InstantiationException ex) {  
        java.util.logging.Logger.getLogger(Welcome.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);  
    } catch (IllegalAccessException ex) {  
        java.util.logging.Logger.getLogger(Welcome.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);  
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {  
        java.util.logging.Logger.getLogger(Welcome.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);  
    }  
}
```

TELESCOPE – ERP

//</editor-fold>

```
/* Create and display the form */ java.awt.EventQueue.invokeLater(new  
Runnable() {  
    public void run() {  
        new Welcome().setVisible(true);  
    }  
});  
}  
}
```

Testing

SYSTEM TESTING

TESTING

Testing goes through the various stages, during testing the program to be tested has to be executed with a set of test cases, and the output of the program for the test case is evaluated to determine if the program is performing as expected. Due to its approach dynamic testing only ascertains the presence of error in the program. The exact nature of error is not usually decided by testing. Testing form is the first in determining error in the program.

Once the programs are tested individually then the system as a whole needs to be tested. During testing, the system is used experimentally to ensure that the software does not fail i.e. it will run according to its specification. The programs executed to check for any syntax or logical error. The error is corrected and test is made to determine whether the program is doing what it is supposed to do.

Various types of testing Unit testing

Each component of the system is tested individually. The programmer does the testing. Testing is restrictive in nature i.e. programmer should try to test all individual conditions and see if the program breaks under any circumstance.

System testing

This is an integrated form of testing, which focuses on functionality and interface between units and team in a controlled environment does it.

Acceptance Testing

This is system testing done by the user of the application the only emphasis is functionally testing as the user is not aware of the technical aspect of the system. The testing is also done in

TELESCOPE – ERP

a controlled environment with logging of all error based on the error found in the system, the user has to accept or reject the system.

Module Testing

This is an optional form of testing, which is done only for large system, which has a large number of modules.

Security Testing

Security testing will be done as a specialized form of testing if there is a high risk exposure in that area. If the risk exposure is not very high, then it can be done as part of the system testing. Typically, security testing would involve trying to break in to the system, trying to execute transactions not allowed to person; to access areas on disk where the user is not allowed.

Testing is vital to the success of the system. If it is done successfully, this shows that the parts of the system are working correctly and all the goals are achieved.

IMPLEMENTATION

Implementation is used here to mean the process of converting a new or revised system design into operational one; conversion is one aspect of implementation. The other aspect is post implementation review and software and maintenance.

There are three types of implementation:

1. Implementation of a computer system
2. Implementation of a new computer system
3. Implementation of a modified application.

MAINTAINANCE

After the system has successfully been implemented maintenance activity may require continuous involvement of the developers. Provision must be made for environmental changes, which may affect either the computer, or other parts of the computer based system:

TELESCOPE – ERP

such activity is normally called maintenance. It includes both the improvement of the system functions and the correction of faults that arise during the operation of a system.

Maintenance activity may require the continuing involvement of a large proportion of computer department resources. Maintenance works may arise due to two reasons:

1. Errors that creep up during normal running
2. Request for changes by the service providers. As part of the normal running of the system when errors are found.

This maintenance work will help to ensure that the system works smoothly as predicted in the open environment. Whenever a maintenance work arises, the work has to be properly carried out with proper documentation. This is to avoid any form of changes in the structure of the system.

For every maintenance work an amendment notification is to be issued. This notification will have required changes and also authenticated. On the receipt of the amendment notification the amendment

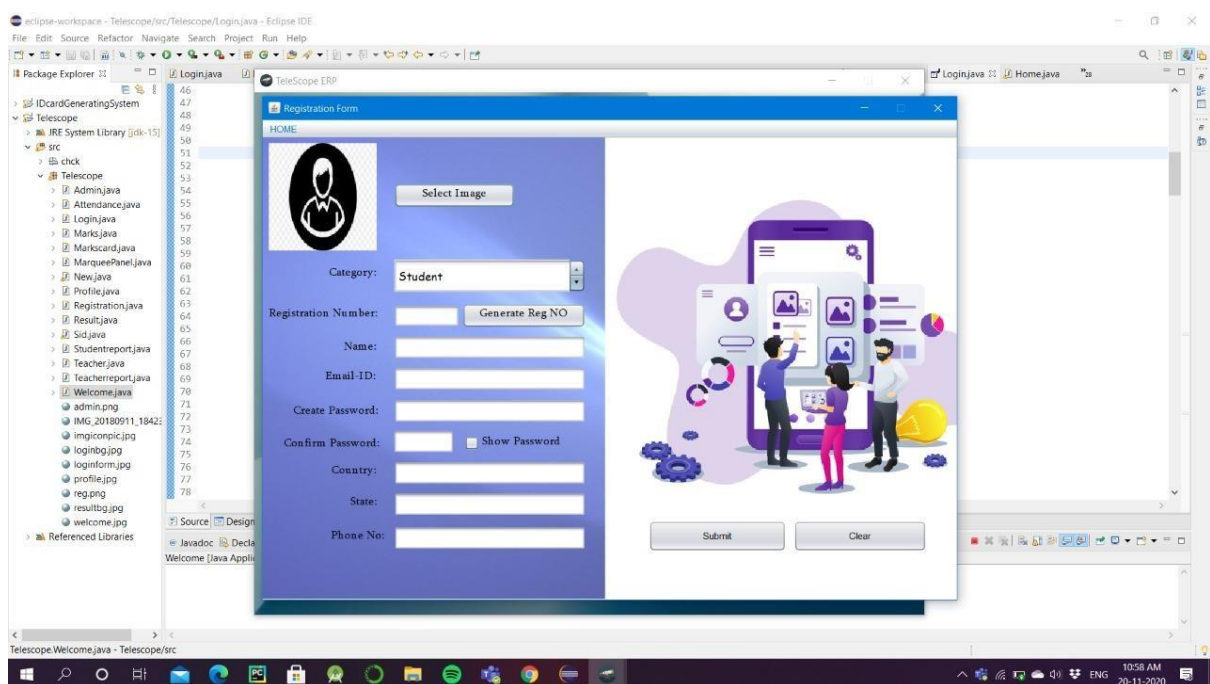
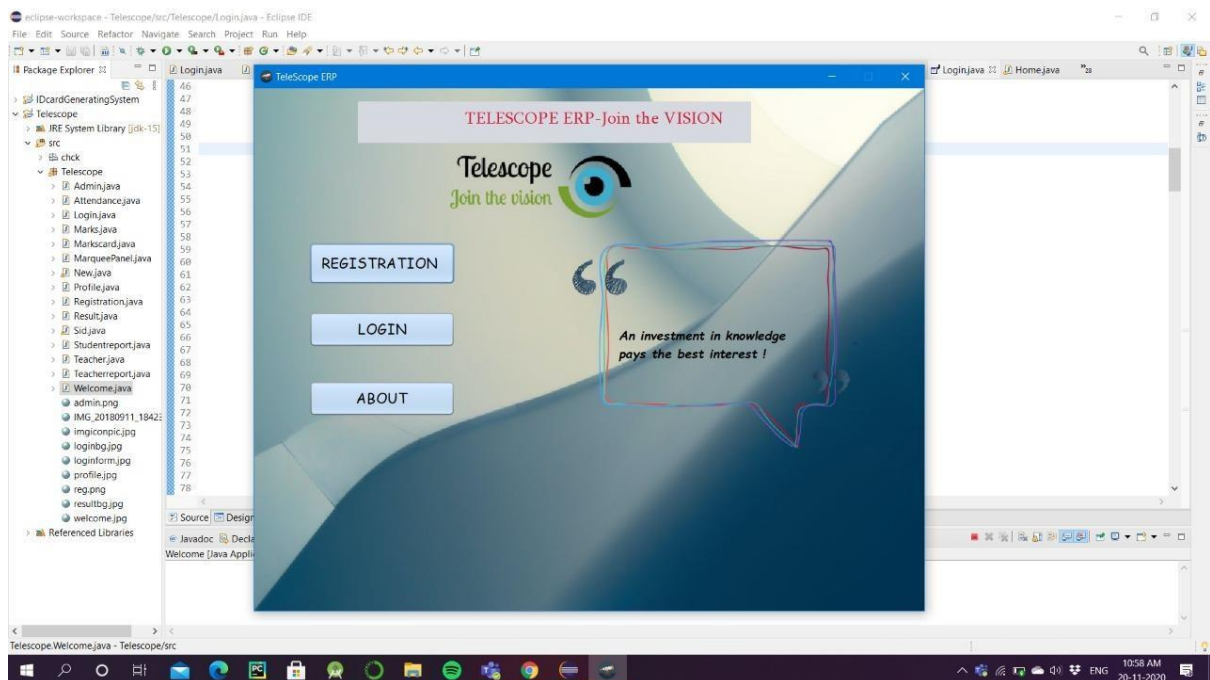
Log is prepared which records these courses of action that has been planned to be taken. It also records the estimated and the actual completion of each activity.

Sample Test Cases:

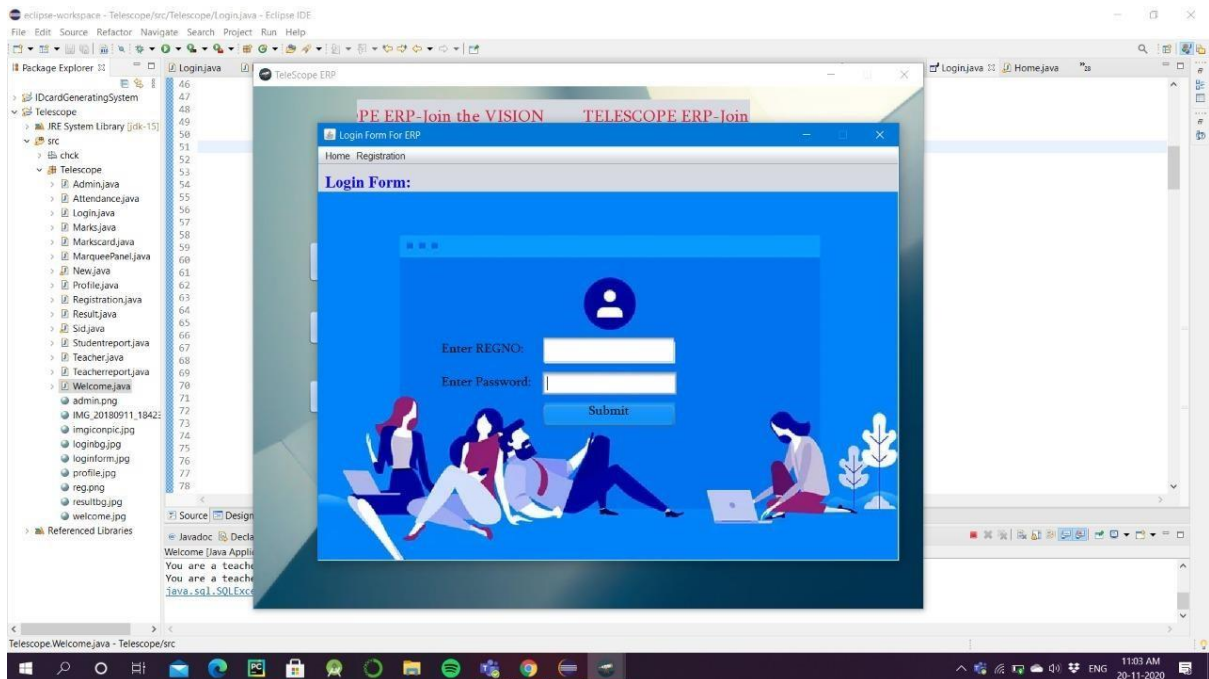
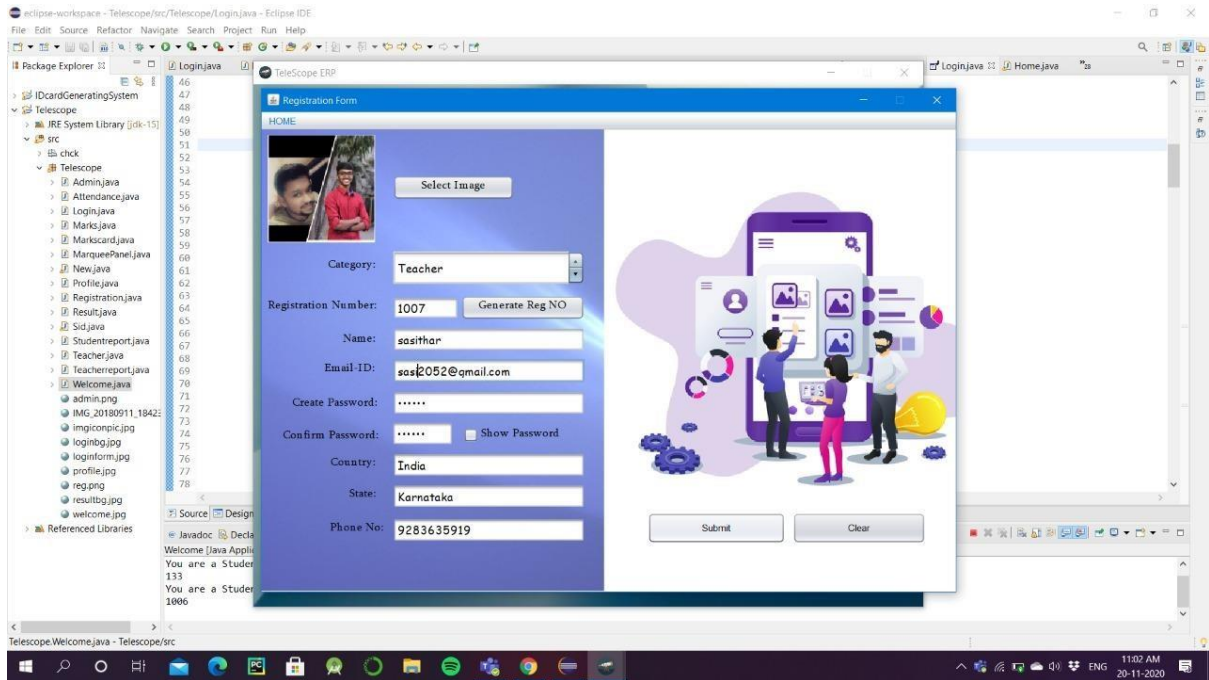
- Invalid Login credentials are tested and appropriate error messages are displayed.
- The data capture part is thoroughly tested for entry of correct data in acceptable type and size.
- Appropriate interfaces are created to link various modules.

Snapshots

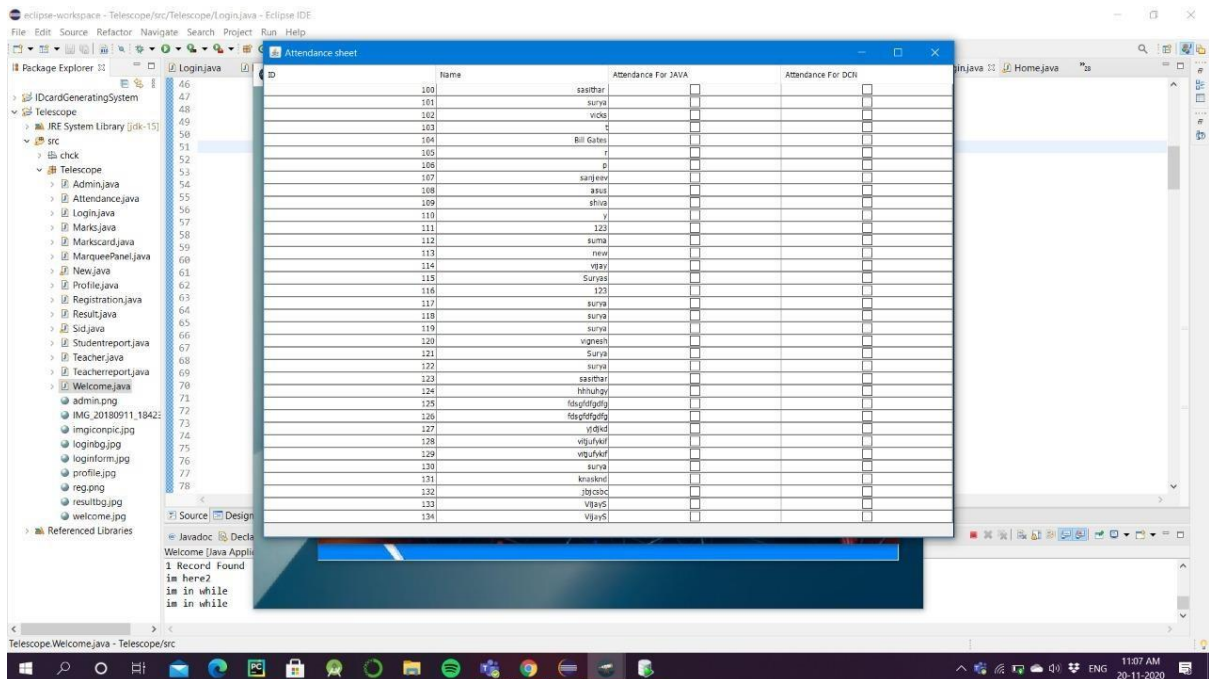
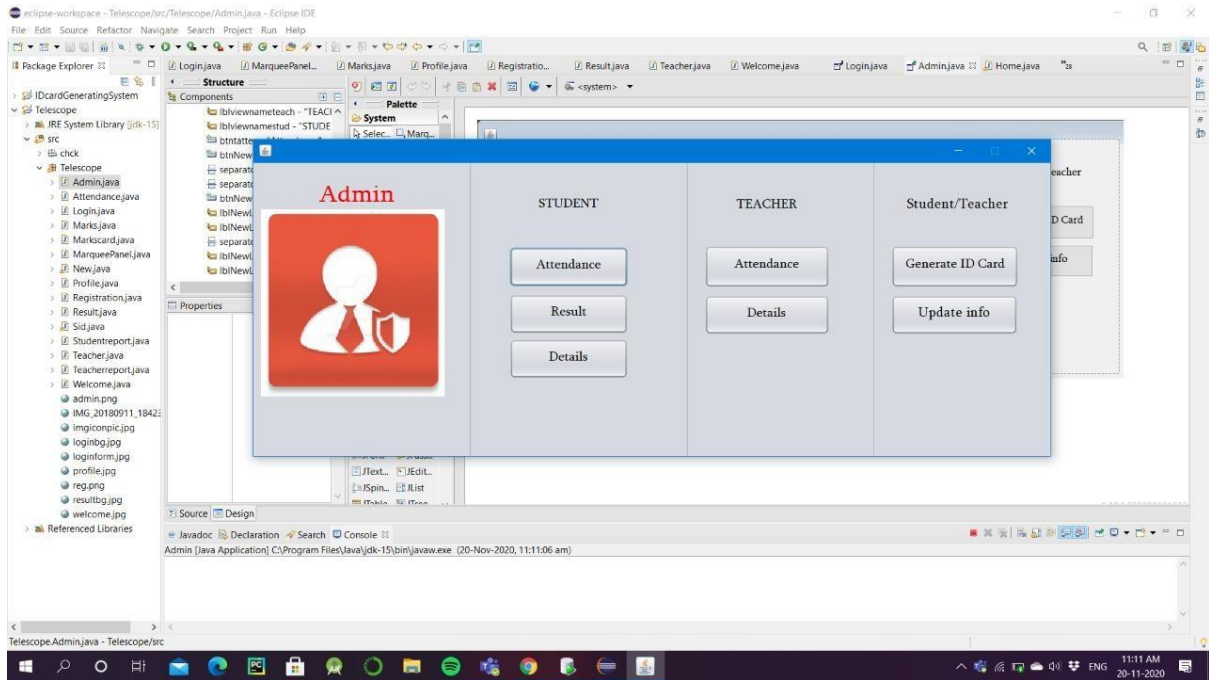
TELESCOPE – ERP

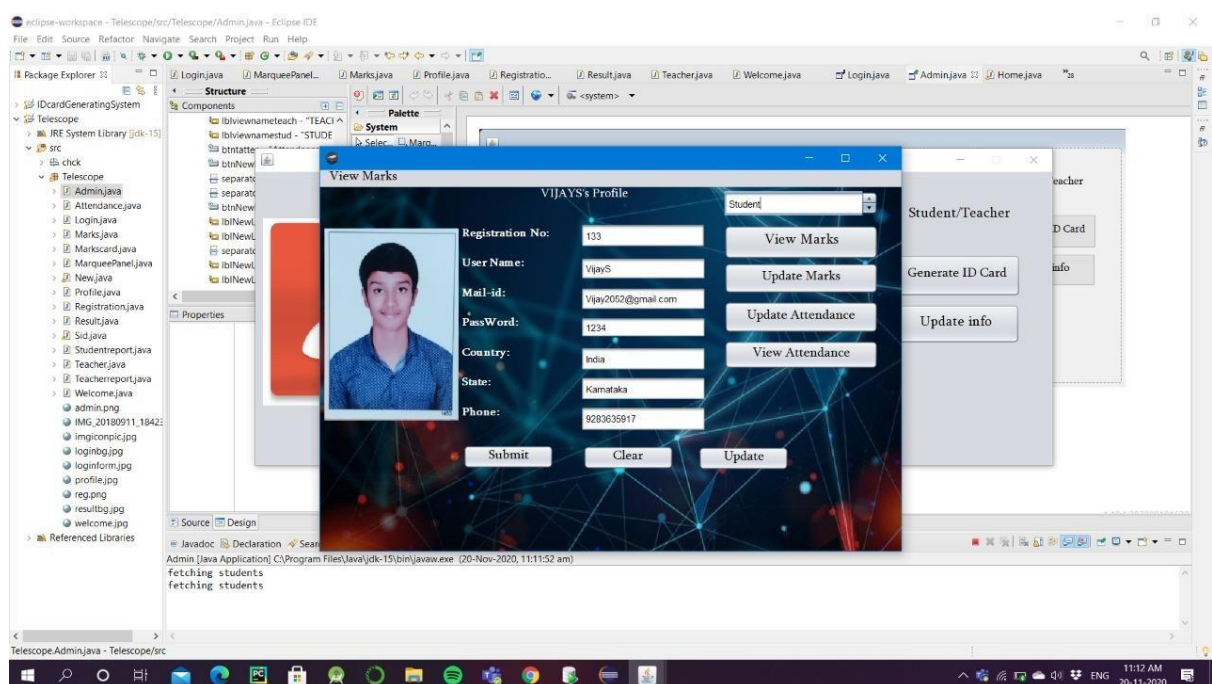
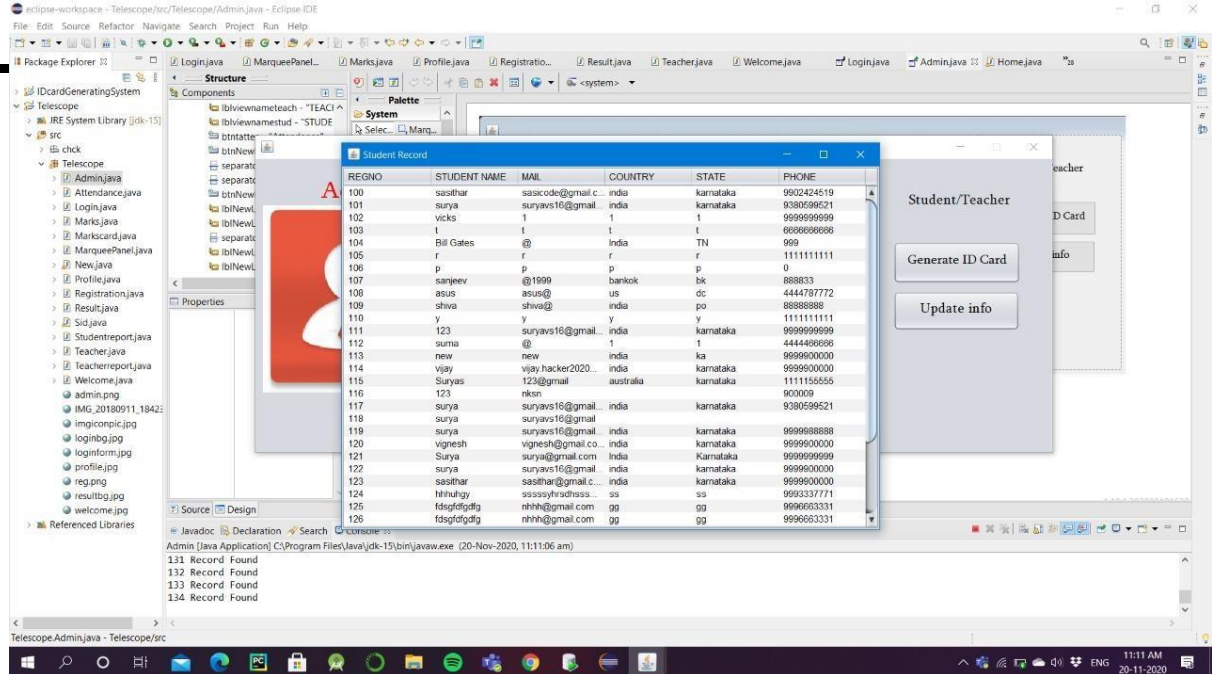


TELESCOPE – ERP

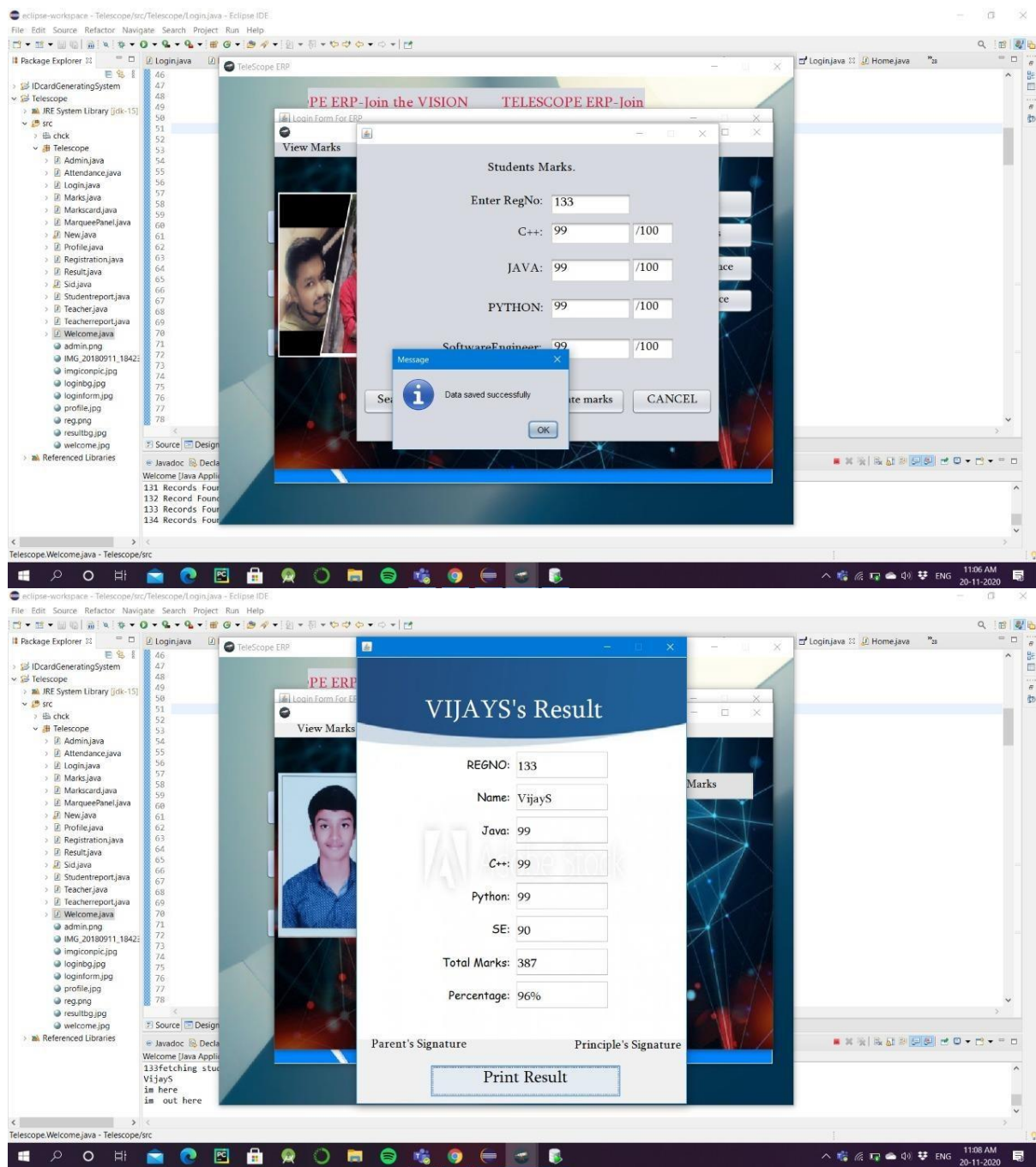


TELESCOPE – ERP

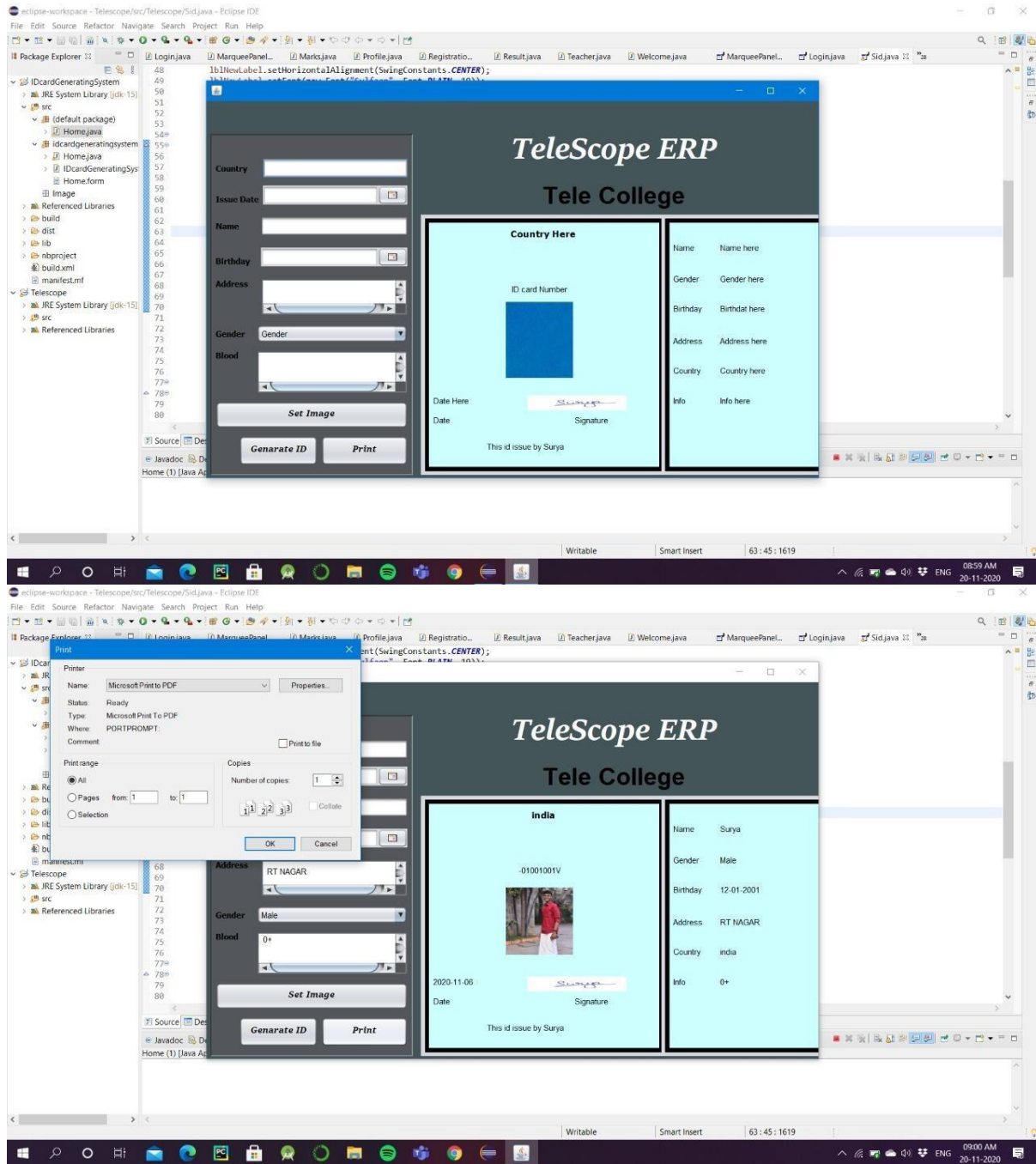




TELESCOPE – ERP



TELESCOPE – ERP



Future Enhancements

FUTURE ENHANCEMENTS

As per the user's needs new modules can be integrated without any modification to the existing application and now it is maintaining only few modules hence we can try to include more features like creating graphs and charts for analysing result visually so that it will give better understanding.

MERITS

- This project gives detailed overview of user's details securely .
- Increase in efficiency of revenue within institution.

TELESCOPE – ERP

- It improves accuracy of the work of operators.
- It provides safety and security.
- We have provided user-friendly interface to help the user to operate the system.
- ID CARDS and marks card can be printed easily.
- Reports can be generated at any time without any errors.

Conclusion

TELESCOPE – ERP

CONCLUSION

This project can highly increase the productivity of students, faculties, and teachers. Its advanced security and precise information can be of great use. This ERP can highly reduce the incurred expenditure of an institution. It can store a lot of information precisely on the database. Editiom, updatation and deletion of the data are not tedious anymore. Further, advancements like displaying the information in the form of graphics and charts can also be implemented.

Bibliography

TELESCOPE – ERP

BIBLIOGRAPHY

- Head First Java: A Brain-Friendly Guide, 2nd Edition
 - **Bert Bates**
- Effective Java
 - **Joshua Bloch**
- Java for Dummies
 - **Barry Burd**
- <https://www.tutorialspoint.com/java>
- <https://www.w3schools.com/java/>
- <https://www.javatpoint.com/java-tutorial>