



A PROJECT REPORT

ON

STUDENTS ATTENDANCE MANAGEMENT SYSTEM

DONE BY

Pornima Mahakal
(EBEON0920321676)

Under the Guidance of,
Mr. V JAYANTH
Technical Trainer
NASSCOM PROJECT



ARISE
ONCAMPUS
SOFTWARE DEVELOPER TRAINEE
Tambaram, Chennai-600045 (Tamil Nadu)
2020

ABSTRACT

Student attendance management system deals with the maintenance of the student's attendance details. It is generates the attendance of the student on basis of presence in class. It is maintained on the daily basis of their attendance. the staffs will be provided with the separate username & password to make the student's status.

The staffs handling the particular subjects responsible to make the attendance for all students. Only if the student present on that particular period, the attendance will be calculated. The students attendance reports based on weekly and consolidate will be generated..

TABLE OF CONTENTS	TITLE	PAGE NO
CHAPTER NO		
ABSTRACT		vii
LIST OF FIGURES		viii
LIST OF TABLES		ix
1 INTRODUCTION		1
1.1 Objective		1
2 SYSTEM ANALYSIS		2
2.1 Existing System		3
2.2 Proposed System		3
2.3 Feasibility Study		3
2.3.1 Economical Feasibility		4
2.3.2 Technical Feasibility		4
2.3.3 Operational Feasibility		4
3 SYSTEM SPECIFICATION		5
3.1 Hardware Specification		5
3.2 Software Specification		5
4 SOFTWARE DESCRIPTION		6
4.1 Package-Visual Studio 2010		6
4.2 jsp servlet		6
4.2.My Sql		7
4.2.2 css,javascript		7
		8
5 PROJECT DESCRIPTION		9
5.1 Problem Definition		9
5.2 Overview of The Project		9
5.3 Module Description		9
5.4 System Flow Diagram		12
5.6.2 Use case Diagram		16
5.6.3 Database Design		17
5.6.4 Input Design		20
5.6.5 Output Design		21
6 SYSTEM TESTING		22
6.1 Introduction		22
6.2 Testing Methodologies		22
6.2.1 Unit Testing		23
6.2.2 SystemTesting		23
6.2.4 Performance Testing		23
6.3 Test Cases		24
7 SYSTEM IMPLEMENTATION		28
7.1 Purpose		28
7.2. System Maintenance		29
8 CONCLUSION AND FUTURE		30

ENHANCEMENTS

8.1 Conclusion	30
8.2 Scope For Future Development	30
9 APPENDICES	37
9.1 Source code	31
9.2 Screen Shots	50
10 REFERENCES	59

CHAPTER – 1

INTRODUCTION TO THE STUDY

Introduction:

“Attendance Management System” is software developed for maintaining the attendance of the student on the daily basis in the collage. Here the staffs, who are handling the subjects, will be responsible to mark the attendance of the students. Each staff will be given with a separate username and password based on the subject they handle. An accurate report based on the student attendance is generated here. This system will also help in evaluating attendance eligibility criteria of a student. Report of the student’s attendance on weekly and monthly basis is generated.

CHAPTER 2

SYSTEM ANALYSIS

2.1 INTRODUCTION

Analysis can be defined as breaking up of any whole so as to find out their nature, function etc. It defines design as to make preliminary sketches of; to sketch a pattern or outline for plan. To plan and carry out especially by artistic arrangement or in a skillful way. System analysis and design can be characterized as a set of techniques and processes, a community of interests, a culture and an intellectual orientation.

The various tasks in the system analysis include the following.

- Understanding application.
- Planning.
- Scheduling.
- Developing candidate solution.
- Performing trade studies.
- Performing cost benefit analysis.
- Recommending alternative solutions.
- Selling of the system.
- Supervising, installing and maintaining the system.

This system manages to the analysis of the report creation and develops manual entry of the student attendance. First design the students entry form , staff allocation and time table allocation forms. This project will helps the attendance system for the department calculate percentage and reports for eligibility criteria of examination .The application attendance entry system will provide flexible report for all students.

2.2 EXISTING SYSTEM

The Existing system is a manual entry for the students. Here the attendance will be carried out in the hand written registers. It will be a tedious job to maintain the record for the user. The human effort is more here. The retrieval of the information is not as easy as the records are maintained in the hand written registers.

This application requires correct feed on input into the respective field. Suppose the wrong inputs are entered, the application resist to work. so the user find it difficult to use.

2.3 PROPOSED SYSTEM:

To overcome the drawbacks of the existing system, the proposed system has been evolved. This project aims to reduce the paper work and saving time to generate accurate results from the student's attendance. The system provides with the best user interface. The efficient reports can be generated by using this proposed system.

2.3.1 Advantages of Proposed System

- It is trouble-free to use.
- It is a relatively fast approach to enter attendance
- Is highly reliable, approximate result from user
- Best user Interface
- Efficient reports

3. FEASIBILITY STUDY:

Feasibility analysis begins once the goals are defined. It starts by generating broad possible solutions, which are possible to give an indication of what the new system should look like. This is where creativity and imagination are used. Analysts must think up new ways of doing things- generate new ideas. There is no need to go into the detailed system operation yet. The solution should provide enough information to make reasonable estimates about project cost and give users an indication of how the new system will fit into the organization. It is important not to exert considerable effort at this stage only to find out that the project is not worthwhile or that there is a need significantly change the original goal.

Feasibility of a new system means ensuring that the new system, which we are going to implement, is efficient and affordable. There are various types of feasibility to be determined. They are,

3.1 Economically Feasibility:

Development of this application is highly economically feasible. The only thing to be done is making an environment with an effective supervision.

It is cost effective in the sense that has eliminated the paper work completely. The system is also time effective because the calculations are automated which are made at the end of the month or as per the user requirement.

3.2 Technical feasibility:

The technical requirement for the system is economic and it does not use any other additional Hardware and software. Technical evaluation must also assess whether the existing systems can be upgraded to use the new technology and whether the organization has the expertise to use it.

Install all upgrades framework into the .Net package supported widows based application. this application depends on Microsoft office and intranet service ,database. Enter their attendance and generate report to excel sheet.

3.3 Operational Feasibility:

The system working is quite easy to use and learn due to its simple but attractive interface. User requires no special training for operating the system. Technical performance include issues such as determining whether the system can provide the right information for the Department personnel student details, and whether the system can be organized so that it always delivers this information at the right place and on time using intranet services.

Acceptance revolves around the current system and its personnel.

CHAPTER 3

SYSTEM SPECIFICATION

3.1 HARDWARE REQUIREMENTS (Minimum Requirement)

- **Minimum RAM:-**1GB
- **Hard Disk:-**128 GB
- **Processor:-**Intel Pentium 4(1.50 GHZ) or above

3.2 SOFTWARE REQUIREMENTS (minimum Requirement)

- **Operating system :**Windows
- **Front_End :** jsp, javascript , css
- **Front-End Language :**Visual basic
- **Back-End :** My Sql

CHAPTER 4

SOFTWARE DESCRIPTION

4.1 PACKAGE - VISUAL STUDIO 2010

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop console and graphical user interface applications along with Windows Forms or WPF applications, web sites, web applications, and web services in both native code together with managed code for all platforms supported by Microsoft Windows, Windows Mobile, Windows CE, .NET Framework, .NET Compact Framework and Microsoft Silver light.

Visual Studio supports different programming languages by means of language services, which allow the code editor and debugger to support (to varying degrees) nearly any programming language, provided a language-specific service exists.

Visual Studio also includes a web-site editor and designer that allows web pages to be authored by dragging and dropping widgets. It is used for developing VB.NET application efficiently to get input and output design easiest one. It will be run at windows application based services provide the user.

4.Jsp,servlet

JSP technology is used to create web application just like Servlet technology. It can be thought of as an extension to Servlet because it provides more functionality than servlet such as expression language, JSTL, etc.

A JSP page consists of HTML tags and JSP tags. The JSP pages are easier to maintain than Servlet because we can separate designing and development. It provides some additional features such as Expression Language, Custom Tags, etc.

Advantages of JSP over Servlet

There are many advantages of JSP over the Servlet. They are as follows:

1) Extension to Servlet

JSP technology is the extension to Servlet technology. We can use all the features of the Servlet in JSP. In addition to, we can use implicit objects, predefined tags, expression language and Custom tags in JSP, that makes JSP development easy.

2) Easy to maintain

JSP can be easily managed because we can easily separate our business logic with presentation logic. In Servlet technology, we mix our business logic with the presentation logic.

3) Fast Development: No need to recompile and redeploy

If JSP page is modified, we don't need to recompile and redeploy the project. The Servlet code needs to be updated and recompiled if we have to change the look and feel of the application.

4) Less code than Servlet

In JSP, we can use many tags such as action tags, JSTL, custom tags, etc. that reduces the code. Moreover, we can use EL, implicit objects, etc.

My Sql:

MySQL tutorial provides basic and advanced concepts of MySQL. Our MySQL tutorial is designed for beginners and professionals.

MySQL is a relational database management system based on the Structured Query Language, which is the popular language for accessing and managing the records in the database. MySQL is open-source and free software under the GNU license. It is supported by **Oracle Company**.

Our MySQL tutorial includes all topics of MySQL database that provides for how to manage database and to manipulate data with the help of various SQL queries. These queries are: insert records, update records, delete records, select records, create tables, drop tables, etc. There are also given MySQL interview questions to help you better understand the MySQL database.

Css,javascript:

CSS tutorial or CSS 3 tutorial provides basic and advanced concepts of CSS technology. Our CSS tutorial is developed for beginners and professionals. The major points of CSS are given below:

- CSS stands for Cascading Style Sheet.
 - CSS is used to design HTML tags.
 - CSS is a widely used language on the web.
 - HTML, CSS and JavaScript are used for web designing. It helps the web designers to apply style on HTML tags.
-
- JavaScript is used to create client-side dynamic pages.
 - JavaScript is *an object-based scripting language* which is lightweight and cross-platform.
 - JavaScript is not a compiled language, but it is a translated language. The JavaScript Translator (embedded in the browser) is responsible for translating the JavaScript code for the web browser.

CHAPTER 5

PROJECT DESCRIPTION

5.1 PROBLEM DEFINITION:

This system developed will reduce the manual work and avoid redundant data. By maintaining the attendance manually, then efficient reports cannot be generated. The system can generate efficient weekly, consolidate report based on the attendance. As the attendances are maintained in registers it has been a tough task for admin and staff to maintain for long time. Instead the software can keep long and retrieve the information when needed.

5.2 PROJECT OVERVIEW

Attendance Management System basically has two main modules for proper functioning

- Admin module is has rights for creating any new entry of faculty and student details.
- User has a rights of making daily attendance, generating report. Attendance report can be taken by given details of student details, date, class.

5.3 MODULE DESCRIPTION

The system should be designed in such a way that only authorized people should be allowed to access some particular modules. The records should be modified by only administrators and no one else. The user should always be in control of the application and not the vice versa.

The user interface should be consistent so that the user can handle the application with ease and speed. The application should be visually, conceptually clear.

5.3.1 ADMINISTRATOR MODULE:

☐ **Student Details:**

In this module deals with the allocation of roll no and personal details for new batch. It will generate of personal details of student and academic details of the students with the photos.

☐ **Staff Details:**

- It helps to allot the subject and the subject code to the particular staffs.
- It provides the facility to have a user name and password to the staffs .

☐ **Time table details:**

- It will retrieve the subject information from the subject database and assign time table to the staffs.
- It will help the admin, staff to make the entry of attendance based of the subject and period allotted to the respective staff.

☐ **Attendance details:**

- It will be makes to the attendance database all students. Entered attendance to stored in the database subject ,period wise into the particular date.
- It will help s to the get report of weekly and consolidate of the attendance.

Report details:

Report can be taken by daily, weekly and consolidate:

- weekly report get all hour details of attendance starting date to ending date and display the status
- Consolidate report get all student attendance details starting date to ending date status help for the eligibility criteria of the student to attend the examination.

5.3.2 STAFFS MODULE:**☐ Attendance details:**

- It assists the staff to mark attendance to the students for their subject. This will authenticate the staff before making the entry.

☐ Report details:

1. weekly report get particular hour details of attendance from starting date to ending date and display the status .
2. consolidate report get all student attendance details from starting date to ending date status help for the eligibility criteria of the student to attend the examination

5.4 SYSTEM FLOW DIAGRAM:

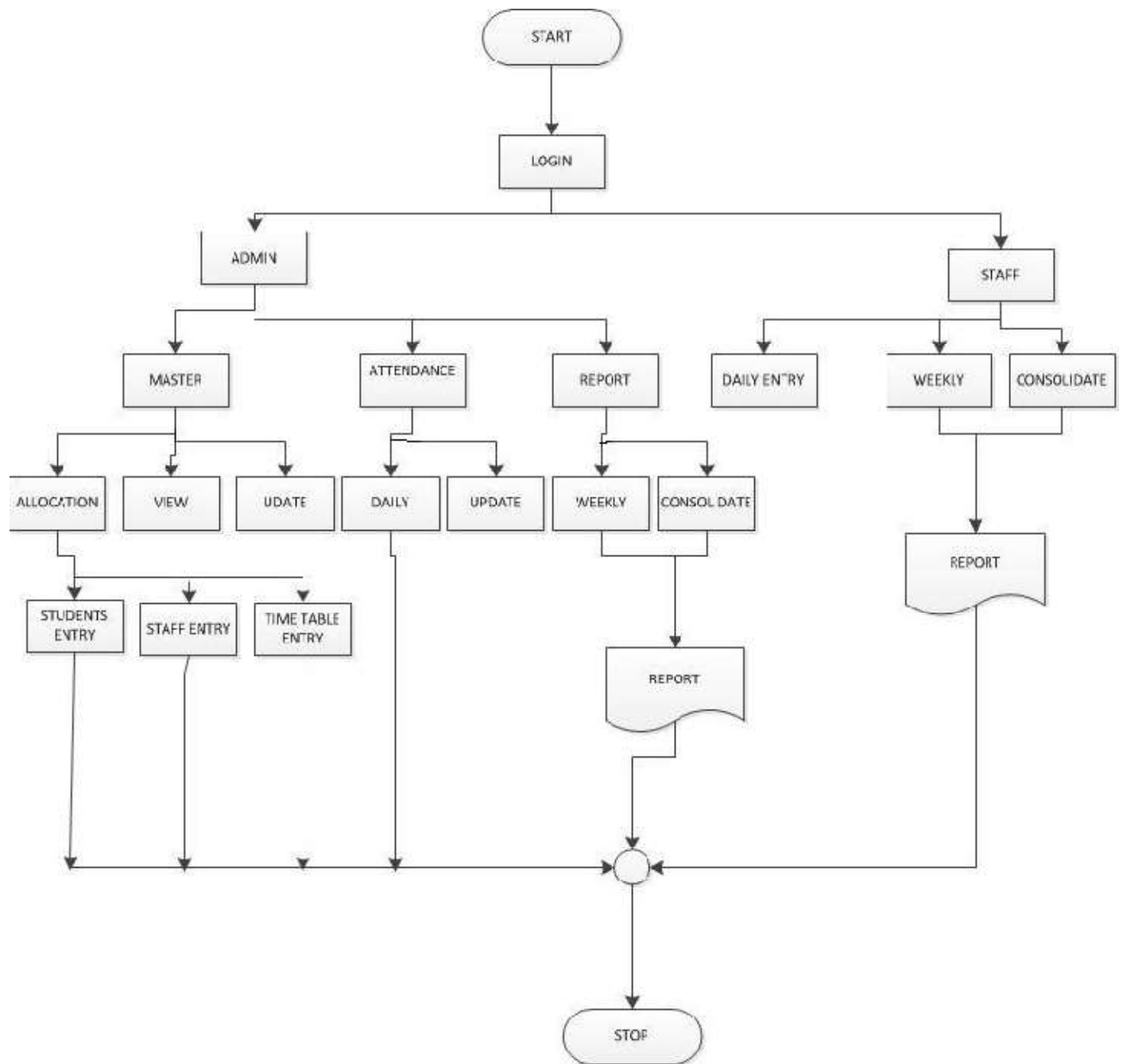


Figure 5.4-System Flow Diagram

5.6.2 Use case Diagram:

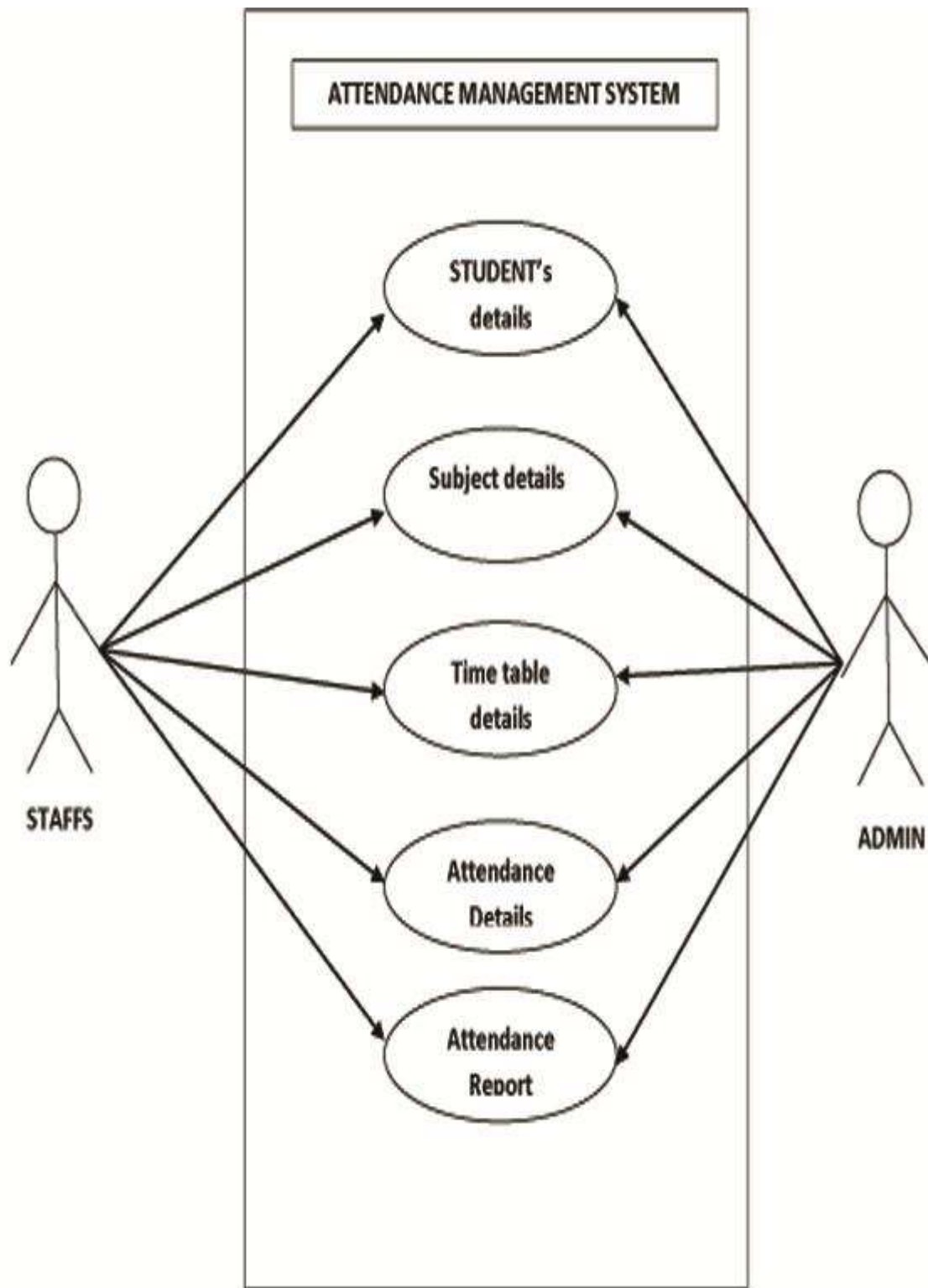


Figure:5.6.2 -Use case Diagram

CHAPTER 6

SYSTEM TESTING

6.1 Introduction

Once source code has been generated, software must be tested to uncover (and correct) as many errors as possible before delivery to customer. Our goal is to design a series of test cases that have a high likelihood of finding errors. To uncover the errors software techniques are used. These techniques provide systematic guidance for designing test that

- (1) Exercise the internal logic of software components, and
- (2) Exercise the input and output domains of the program to uncover errors in program function, behavior and performance.

6.1.1 Steps: Software is tested from two different perspectives:

- (1) Internal program logic is exercised using —White box test case design Techniques.
- (2) Software requirements are exercised using —block box test case Design techniques.

In both cases, the intent is to find the maximum number of errors with the Minimum amount of effort and time.

6.2 Testing Methodologies:

A strategy for software testing must accommodate low-level tests that are necessary to verify that a small source code segment has been correctly implemented as well as high-level tests that validate major system functions against customer requirements. A strategy must provide guidance for the practitioner and a set of milestones for the manager. Because the steps of the test strategy occur at a time when deadline pressure begins to rise, progress must be measurable and problems must surface as early as possible. Following testing techniques are well known and the same strategy is adopted during this project testing.

6.2.1 Unit testing:

Unit testing focuses verification effort on the smallest unit of software design- the software component or module. The unit test is white-box oriented. The unit testing implemented in every module of student attendance management System. by giving correct manual input to the system ,the datas are stored in database and retrieved. If you want required module to access input or get the output from the End user. any error will accrued the time will provide handler to show what type of error will accrued .

6.2.2 System testing:

System testing is actually a series of different tests whose primary purpose is to fully exercise the computer-based system. Below we have described the two types of testing which have been taken for this project. it is to check all modules worked on input basis .if you want change any values or inputs will change all information. so specified input is must.

6.2.4 Performance Testing

Performance testing is designed to test the run-time performance of software within the context of an integrated system. Performance testing occurs throughout all steps in the testing process. Even at the unit level, the performance of an individual module may be assessed as white-box tests are conducted.

This project reduce attendance table, codes. it will generate report fast.no have extra time or waiting of results .entered correct data will show result few millisecond. just used only low memory of our system. Automatically do not getting access at another software. Get user permission and access to other applications.

CHAPTER 7

SYSTEM IMPLEMENTATION

7.1 Purpose

System implementation is the important stage of project when the theoretical design is tuned into practical system. The main stages in the implementation are as follows:

- Planning
- Training
- System testing and
- Changeover Planning

Planning is the first task in the system implementation. At the time of implementation of any system people from different departments and system analysis involve. They are confirmed to practical problem of controlling various activities of people outside their own data processing departments.

The line managers controlled through an implementation coordinating committee. The committee considers ideas, problems and complaints of user department, it must also consider:

- The implication of system environment
- Self selection and allocation for implementation tasks
- Consultation with unions and resources available
- Standby facilities and channels of communication

Student Attendance management system will implement student details ,staff handle subjects details, separate login details ,time table details. It will used to entered subject wise attendance .This application elaborate attendance table generate weekly, consolidate report provide to the End user. Mostly this application will calculate date wise attendance .To select starting date to end date generate reports at the time of activities.

7.2 SYSTEM MAINTENANCE

Software maintenance is far more than finding mistakes. Provision must be made for environment changes, which may affect either the computer, or other parts of the computer based systems. Such activity is normally called maintenance. It includes both the improvement of the system functions and the corrections of faults, which arise during the operation of a new system.

It may involve the continuing involvement of a large proportion of computer department resources. The main task may be to adapt existing systems in a changing environment.

Back up for the entire database files are taken and stored in storage devices like flash drives, pen drives and disks so that it is possible to restore the system at the earliest. If there is a breakdown or collapse, then the system gives provision to restore database files. Storing data in a separate secondary device leads to an effective and efficient maintains of the system. The nominated person has sufficient knowledge of the organization's computer passed based system to be able to judge the relevance of each proposed change.

CHAPTER 8

CONCLUSION AND FUTURE ENHANCEMENT

8.1 Conclusion

To conclude, Project Data Grid works like a component which can access all the databases and picks up different functions. It overcomes the many limitations incorporated in the attendance.

- Easy implementation Environment
- Generate report Flexibly

8.2 Scope for future development

The project has a very vast scope in future. The project can be implemented on intranet in future. Project can be updated in near future as and when requirement for the same arises, as it is very flexible in terms of expansion. With the proposed software of database Space Manager ready and fully functional the client is now able to manage and hence run the entire work in a much better, accurate and error free manner. The following are the future scope for the project.

- Discontinue of particular student eliminate potential attendance.
- Bar code Reader based attendance system.
- Individual Attendance system With photo using Student login.

APPENDIX-I

SNAPSHOTS:

USERNAME

PASSWORD

LOGIN

STUDENT DETAILS ENTRY

HOME TABLE

SELECT PROGRAM
 STARTING YEAR

BATCH YEAR
 STARTING NUMBER

DEPARTMENT
 TO

END NUMBER

* Choose all data then click

1MCA001

ENTER NAME

DATE OF BIRTH (EX:12/02/2013)

ADDRESS

MOBILE NO

E-MAIL ID

HOSTELLER/DAYSCHOLLER

Student Details

STAFFS ENTRY

HOME TABLE

STAFF ENTRY

SELECT PROGRAM

MCA

BATCH YEAR

2011

CLASS

1

SEMESTER

1

SUBJECT CODE

11MCA201

SUBJECT SHORT NAME

HCI

STAFF NAME(user name)

VANITHA

PASSWORD

123456

ADD

SUBJECT CODE

11MCA201

11MCA202

11MCA203

11MCA204

11MCA205

DFLT

REFRESH

Staff Details

ATTENDANCE ENTRY

HOME MASTER

PROGRAM DETAILS

SELECT PROGRAM

MCA

CLASS

1

BATCH

2011

SEMESTER

1

CLICK

* Choose all data then click

DATE

Friday May 10, 2013

HOUR

1

PERIOD

DS

SNO	ROLLNO	NAME	PRESENT/ABSI	ONDUTY
1	11MCA001	kalai	<input checked="" type="checkbox"/>	<input type="checkbox"/>
2	11MCA002	sam	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3	11MCA003	sanji	<input checked="" type="checkbox"/>	<input type="checkbox"/>
4	11MCA004	kousi	<input checked="" type="checkbox"/>	<input type="checkbox"/>
5	11MCA005	mani	<input checked="" type="checkbox"/>	<input type="checkbox"/>
6	11MCA006	muthu	<input checked="" type="checkbox"/>	<input type="checkbox"/>
7	11MCA007	kumar	<input type="checkbox"/>	<input type="checkbox"/>
8	11MCA008	amu	<input type="checkbox"/>	<input type="checkbox"/>
9	11MCA009	ihon	<input checked="" type="checkbox"/>	<input type="checkbox"/>
10	11MCA010	manju	<input checked="" type="checkbox"/>	<input type="checkbox"/>
11	11MCA011	bhuvi	<input checked="" type="checkbox"/>	<input type="checkbox"/>
12	11MCA012	samuel	<input type="checkbox"/>	<input type="checkbox"/>
13	11MCA013	anu	<input checked="" type="checkbox"/>	<input type="checkbox"/>

ENTRY DETAILS

☒ PRESENT
 ☐ ABSENT

SUBMIT

☒ SELECT ALL

Attendance Entry

OVER ALL WEEKLY REPORT

HOME

SELECT PROGRAM

MCA

STARTING DATE

Monday May 06

TO

Saturday May 11

BATCH

2011

CLASS

1

SEMESTER

1

CLICK

* Choose all data then click.

MCA-2011	SEMESTER-1	06-May-2013	07-May-2013	08-May-2013	09-May-2013	10-May-2013	11-May-2013
ROLLNO	STUDENT NAME	1 2 3 4 5 6 7	1 2 3 4 5 6 7	1 2 3 4 5 6 7	1 2 3 4 5 6 7	1 2 3 4 5 6 7	1 2 3 4 5 6 7
1 11MCA001	kalai	P P P P P P P	P P P P P P P	P P P P P P P	P P P P P P P	P P P P P P P	P P P P P P P
2 11MCA002	sam	P P P P P P P	P P P P P P P	P P P P P P P	P P P P P P P	P P P P P P P	P P P P P P P
3 11MCA003	sanji	P P P P P P P	P P P P P P P	P P P P P P P	P P P P P P P	P P P P P P P	P P P P P P P
4 11MCA004	kousi	P P P P P P P	P P P P P P P	P P P P P P P	P P P P P P P	P P P P P P P	P P P P P P P
5 11MCA005	mani	P P P P P P P	P P P P P P P	P P P P P P P	P P P P P P P	P P P P P P P	P P P P P P P
6 11MCA006	muthu	P P P P P P P	P P P P P P P	P P P P P P P	P P P P P P P	P P P P P P P	P P P P P P P
7 11MCA007	kumar	P P P P P P P	P P P P P P P	P P P P P P P	P P P P P P P	P P P P P P P	P P P P P P P
8 11MCA008	amu	P P P P P P P	P P P P P P P	P P P P P P P	P P P P P P P	P P P P P P P	P P P P P P P
9 11MCA009	jhon	P P P P P P P	P P P P P P P	P P P P P P P	P P P P P P P	P P P P P P P	P P P P P P P
1 11MCA010	manju	P P P P P P P	P P P P P P P	P P P P P P P	P P P P P P P	P P P P P P P	P P P P P P P
1 11MCA011	bhuvi	P P P P P P P	P P P P P P P	P P P P P P P	P P P P P P P	P P P P P P P	P P P P P P P
1 11MCA012	samuel	A A A A A A A	P P P P P P P	P P P P P P P	P P P P P P P	P P P P P P P	P P P P P P P
1 11MCA013	anu	A A A A A A A	P P P P P P P	P P P P P P P	P P P P P P P	P P P P P P P	P P P P P P P
*							

INFORMATION

* P-Present

* A-Absent

GET DOCUMENT FORMET TO EXCEL

start

allocation (Running) -...

screen shot [Compati...

Local Disk (D:)

OVER ALL WEEKLY R...

12:56 PM

Weekly report

STAFFS ENTRY

HOME MASTER

LOGIN DETAILS

NAME: SOMU
SUBJECT CODE: 11MCA203
SUBJECT: EC

PROGRAM DETAILS

SELECT PROGRAM: MCA CLASS: 1
BATCH: 2011 SEMESTER: 1


CLICK * Choose all data then click

DATE: Friday, May 10, 2013
HOUR: Select One

SNO	ROLLNO	NAME	PRESENT/ABSI	ONDUTY
1	11MCA001	kalai	<input checked="" type="checkbox"/>	<input type="checkbox"/>
2	11MCA002	sam	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3	11MCA003	sanji	<input checked="" type="checkbox"/>	<input type="checkbox"/>
4	11MCA004	kousi	<input checked="" type="checkbox"/>	<input type="checkbox"/>
5	11MCA005	mani	<input checked="" type="checkbox"/>	<input type="checkbox"/>
6	11MCA006	muthu	<input checked="" type="checkbox"/>	<input type="checkbox"/>
7	11MCA007	kumar	<input checked="" type="checkbox"/>	<input type="checkbox"/>
8	11MCA008	amu	<input checked="" type="checkbox"/>	<input type="checkbox"/>
9	11MCA009	jhon	<input checked="" type="checkbox"/>	<input type="checkbox"/>
10	11MCA010	manju	<input checked="" type="checkbox"/>	<input type="checkbox"/>
11	11MCA011	bhuvi	<input type="checkbox"/>	<input checked="" type="checkbox"/>
12	11MCA012	samuel	<input type="checkbox"/>	<input checked="" type="checkbox"/>
13	11MCA013	anu	<input type="checkbox"/>	<input type="checkbox"/>

ENTRY DETAILS
☒ PRESENT
☐ ABSENT

SUBMIT SELECT ALL



start allocation (Running) -... screen shot [Compati... Local Disk (D:) STAFFS ENTRY 12:59 PM

Daily attendance

STAFFS WEEKLY REPORT

HOME

STUDENTS WEEKLY REPORT

LOGIN DETAILS

NAME: SOMU
SUBJECT CODE: 11MCA203
SUBJECT: EC

SELECT PROGRAM: MCA BATCH: 2011 CLASS: 1 SEMESTER: 1

STARTING DATE: Monday, May TO Friday, May

CLICK * Choose all data then click

INFORMATION

- * P-Present
- * A-Absent
- * O-Onduty

* (EX:April_10_2011_2) (MONTH_DATE_YEAR_HOUR)

SNO	ROLLNO	STUDENT NAME	06-May-2013_3	07-May-2013_2	08-May-2013_1	09-May-2013_5	09-May-2013_6	09-May-2013_7	10-May
1	11MCA001	kalai	P	P	P	P	P	P	P
2	11MCA002	sam	P	P	P	P	P	P	P
3	11MCA003	sanji	P	P	P	P	P	P	P
4	11MCA004	kousi	P	P	A	A	A	A	A
5	11MCA005	mani	P	P	P	P	P	P	P
6	11MCA006	muthu	P	P	P	P	P	P	P
7	11MCA007	kumar	P	P	P	P	P	P	P
8	11MCA008	amu	P	P	A	A	A	A	A
9	11MCA009	jhon	P	P	P	P	P	P	P
10	11MCA010	manju	P	A	A	A	A	A	A
11	11MCA011	bhuvi	P	A	A	A	A	A	A
12	11MCA012	samuel	A	P	P	P	P	P	P
13	11MCA013	anu	A	P	P	P	P	P	P

GET DOCUMENT FORMET TO EXCEL

start allocation (Running) -... screen shot [Compati... Local Disk (D:) STAFFS WEEKLY REP... 1:00 PM

Weekly report details

SOURCE CODE

COM.Email

Emailsendservlet.java

```
package com.email;

import java.io.IOException;
import java.sql.*;
import javax.print.*;
import java.io.PrintWriter;
import javax.servlet.ServletContext;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import com.attendance.DatabaseConnection;

@WebServlet("/EmailSendingServlet")
public class EmailSendingServlet extends HttpServlet {
    private String host;
    private String port;
    private String user;
    private String pass;

    public void init() {
        // reads SMTP server setting from web.xml file
        ServletContext context = getServletContext();
        host = context.getInitParameter("host");
        port = context.getInitParameter("port");
        user = context.getInitParameter("user");
        pass = context.getInitParameter("pass");
    }

    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        // reads form fields

        Connection con = null;
        Statement statement = null;
        ResultSet resultset = null;
        int student_id = Integer.parseInt(request.getParameter("student_id"));
        String email = "";
        String uname = "";
        String password = "";
        String uname_password = "";
    }
}
```

```

        try {
            con = DatabaseConnection.getConnection();
            statement = con.createStatement();
            resultset = statement.executeQuery("select email,uname,password from students
where student_id='" + student_id + "'");
            while (resultset.next()) {
                email = resultset.getString(1);
                uname = resultset.getString(2);
                password = resultset.getString(3);
                uname_password = "Hi
"+uname+", "+System.lineSeparator()+System.lineSeparator()+ "Your Student Login
Portal"+System.lineSeparator()+"User Name: "+uname + System.lineSeparator()+"Password: "+
password +System.lineSeparator()+System.lineSeparator()+"Thank
you."+System.lineSeparator()+"Student Attendance Team.";
            }
        } catch (Exception e) {
            e.printStackTrace();
        }

        String message = "Student Attendance System.";
        // String content = request.getParameter("content");
        String resultMessage = "";

        PrintWriter out = response.getWriter();
        try {
            EmailUtility.sendEmail(host, port, "Your attendance emailId here", "email
password", email, message,uname_password);
            HttpSession session=request.getSession();
            resultMessage = "Your login user name & password send successfully in your
email.";
            session.setAttribute("mail-success", resultMessage);

        } catch (Exception ex) {
            ex.printStackTrace();
            resultMessage = "There were an error: " + ex.getMessage();
        } finally {
            //request.setAttribute("Message", resultMessage);

            response.sendRedirect("send_user_name_and_password_to_student_in_mail.jsp");
        }
    }
}

```

Emailutility.java

```

package com.email;
import java.util.Date;
import java.util.Properties;

import javax.mail.Authenticator;
import javax.mail.Message;
import javax.mail.MessagingException;
import javax.mail.PasswordAuthentication;

```

```

import javax.mail.Session;
import javax.mail.Transport;
import javax.mail.internet.AddressException;
import javax.mail.internet.InternetAddress;
import javax.mail.internet.MimeMessage;

public class EmailUtility {
    public static void sendEmail(String host, String port,
        final String userName, final String password, String toAddress,
        String subject, String message) throws AddressException,
        MessagingException {

        // sets SMTP server properties
        Properties properties = new Properties();
        properties.put("mail.smtp.host", host);
        properties.put("mail.smtp.port", port);
        properties.put("mail.smtp.auth", "true");
        properties.put("mail.smtp.starttls.enable", "true");

        // creates a new session with an authenticator
        Authenticator auth = new Authenticator() {
            public PasswordAuthentication getPasswordAuthentication() {
                return new PasswordAuthentication(userName, password);
            }
        };

        Session session = Session.getInstance(properties, auth);

        // creates a new e-mail message
        Message msg = new MimeMessage(session);

        msg.setFrom(new InternetAddress(userName));
        InternetAddress[] toAddresses = { new InternetAddress(toAddress) };
        msg.setRecipients(Message.RecipientType.TO, toAddresses);
        msg.setSubject(subject);
        msg.setSentDate(new Date());
        msg.setText(message);

        // sends the e-mail
        Transport.send(msg);
    }
}

```

Com.reports

AllStudentRegisterReport.java

```

package com.reports;

import java.io.ByteArrayOutputStream;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;

```

```

import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.sql.ResultSet;
import java.text.SimpleDateFormat;
import java.util.Date;
import javax.servlet.ServletContext;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import com.attendance.DatabaseConnection;
import com.itextpdf.text.BaseColor;
import com.itextpdf.text.Document;
import com.itextpdf.text.DocumentException;
import com.itextpdf.text.Element;
import com.itextpdf.text.Font;
import com.itextpdf.text.Paragraph;
import com.itextpdf.text.Phrase;
import com.itextpdf.text.pdf.PdfPCell;
import com.itextpdf.text.pdf.PdfPTable;
import com.itextpdf.text.pdf.PdfWriter;

/**
 * Servlet implementation class UserTradingPDFReport
 */
@WebServlet("/AllStudentsRegisterReport")
public class AllStudentsRegisterReport extends HttpServlet {
    private static final long serialVersionUID = 1L;
    private static Font TIME_ROMAN = new Font(Font.FontFamily.TIMES_ROMAN, 18,
Font.BOLD);
    private static Font TIME_ROMAN_SMALL = new Font(Font.FontFamily.TIMES_ROMAN,
12, Font.BOLD);

    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        final ServletContext servletContext = request.getSession().getServletContext();
        final File tempDirectory = (File)
servletContext.getAttribute("javax.servlet.context.tempdir");
        final String temperotyFilePath = tempDirectory.getAbsolutePath();
        String fileName = "All_Students_Register_Report_" + System.currentTimeMillis() +
".pdf";

        response.setContentType("application/pdf");
        response.setHeader("Cache-Control", "no-cache");
        response.setHeader("Cache-Control", "max-age=0");
        response.setHeader("Content-disposition", "attachment; " + "filename=" + fileName);

        try {
            createPDF(temperotyFilePath + "\\ " + fileName);
            ByteArrayOutputStream baos = new ByteArrayOutputStream();

```

```

        fileName);
        baos = convertPDFToByteArrayOutputStream(temperotyFilePath + "\\\" +
        OutputStream os = response.getOutputStream();
        baos.writeTo(os);
        os.flush();
    } catch (Exception e1) {
        e1.printStackTrace();
    }
}

protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
}

private static ByteArrayOutputStream convertPDFToByteArrayOutputStream(String
fileName) {
    InputStream inputStream = null;
    ByteArrayOutputStream baos = new ByteArrayOutputStream();
    try {
        inputStream = new FileInputStream(fileName);

        byte[] buffer = new byte[1024];
        baos = new ByteArrayOutputStream();

        int bytesRead;
        while ((bytesRead = inputStream.read(buffer)) != -1) {
            baos.write(buffer, 0, bytesRead);
        }

    } catch (FileNotFoundException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    } finally {
        if (inputStream != null) {
            try {
                inputStream.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
    return baos;
}

public static Document createPDF(String file) {
    Document document = null;
    try {
        document = new Document();
        PdfWriter.getInstance(document, new FileOutputStream(file));
        document.open();
    }
}

```

```

        addMetaData(document);
        addTitlePage(document);
        createTable(document);
        document.close();

    } catch (FileNotFoundException e) {

        e.printStackTrace();
    } catch (DocumentException e) {
        e.printStackTrace();
    }
    return document;
}

private static void addMetaData(Document document) {
    document.addTitle("Generate PDF report");
    document.addSubject("Generate PDF report");
    document.addAuthor("Kishor Kadam");
    document.addCreator("Kishor Kadam");
}

private static void addTitlePage(Document document) throws DocumentException {

    Paragraph preface = new Paragraph();
    createEmptyLine(preface, 1);
    preface.add(new Paragraph("Students Register Report ", TIME_ROMAN));

    createEmptyLine(preface, 1);
    SimpleDateFormat simpleDateFormat = new SimpleDateFormat("MM/dd/yyyy");
    preface.add(
        new Paragraph("Students Register Report on " +
simpleDateFormat.format(new Date()), TIME_ROMAN_SMALL));

    document.add(preface);
}

private static void createEmptyLine(Paragraph paragraph, int number) {
    for (int i = 0; i < number; i++) {
        paragraph.add(new Paragraph(" "));
    }
}

private static void createTable(Document document) throws DocumentException {

    Paragraph paragraph = new Paragraph();
    createEmptyLine(paragraph, 2);
    document.add(paragraph);
    PdfPTable table = new PdfPTable(6);

    PdfPCell c1 = new PdfPCell(new Phrase("Student Id"));
    c1.setHorizontalAlignment(Element.ALIGN_CENTER);

```

```

table.addCell(c1);

c1 = new PdfPCell(new Phrase("Student Name"));
c1.setHorizontalAlignment(Element.ALIGN_CENTER);
table.addCell(c1);

c1 = new PdfPCell(new Phrase("College Name"));
c1.setHorizontalAlignment(Element.ALIGN_CENTER);
table.addCell(c1);

c1 = new PdfPCell(new Phrase("Mobile"));
c1.setHorizontalAlignment(Element.ALIGN_CENTER);
table.addCell(c1);

c1 = new PdfPCell(new Phrase("Address"));
c1.setHorizontalAlignment(Element.ALIGN_CENTER);
table.addCell(c1);

c1 = new PdfPCell(new Phrase("Email Id"));
c1.setHorizontalAlignment(Element.ALIGN_CENTER);
table.addCell(c1);

table.setHeaderRows(1);

PdfPCell[] cells = table.getRow(0).getCells();
for (int j = 0; j < cells.length; j++) {
    cells[j].setBorderColor(BaseColor.BLACK);
    cells[j].setBackgroundColor(BaseColor.GRAY);
}

try {
    PdfPCell TableCell = null;
    table.setWidthPercentage(100);
    table.getDefaultCell().setHorizontalAlignment(Element.ALIGN_CENTER);
    table.getDefaultCell().setVerticalAlignment(Element.ALIGN_MIDDLE);
    ResultSet resultset = DatabaseConnection.getResultFromSqlQuery("select *
from students ");
    while (resultset.next()) {
        String student_id = resultset.getString("student_id");
        String stud_name = resultset.getString("stud_name");
        String college_name = resultset.getString("college_name");
        String mobile = resultset.getString("mobile");
        String address = resultset.getString("address");
        String email = resultset.getString("email");
        TableCell = new PdfPCell(new Phrase(student_id));
        table.addCell(TableCell);
        TableCell = new PdfPCell(new Phrase(stud_name));
        table.addCell(TableCell);
        TableCell = new PdfPCell(new Phrase(college_name));
        table.addCell(TableCell);
        TableCell = new PdfPCell(new Phrase(mobile));
        table.addCell(TableCell);
        TableCell = new PdfPCell(new Phrase(address));

```

```

        table.addCell(TableCell);
        TableCell = new PdfPCell(new Phrase(email));
        table.addCell(TableCell);
    }
} catch (Exception e) {
    e.printStackTrace();
}
document.add(table);
}
}

```

StudentAttendanceReport.java

```

package com.reports;

import java.io.ByteArrayOutputStream;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.sql.ResultSet;
import java.text.SimpleDateFormat;
import java.util.Date;
import javax.servlet.ServletContext;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import com.attendance.DatabaseConnection;
import com.itextpdf.text.BaseColor;
import com.itextpdf.text.Document;
import com.itextpdf.text.DocumentException;
import com.itextpdf.text.Element;
import com.itextpdf.text.Font;
import com.itextpdf.text.Paragraph;
import com.itextpdf.text.Phrase;
import com.itextpdf.text.pdf.PdfPCell;
import com.itextpdf.text.pdf.PdfPTable;
import com.itextpdf.text.pdf.PdfWriter;

/**
 * Servlet implementation class UserTradingPDFReport
 */
@WebServlet("/StudentAttendanceReport")
public class StudentAttendanceReport extends HttpServlet {
    private static final long serialVersionUID = 1L;
    private static Font TIME_ROMAN = new Font(Font.FontFamily.TIMES_ROMAN, 18,
        Font.BOLD);

```



```
private static Font TIME_ROMAN_SMALL = new Font(Font.FontFamily.TIMES_ROMAN,
12, Font.BOLD);
```

```
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    final ServletContext servletContext = request.getSession().getServletContext();
    final File tempDirectory = (File)
        servletContext.getAttribute("javax.servlet.context.tempdir");
    final String temperotyFilePath = tempDirectory.getAbsolutePath();
    String fileName = "Student_Attendance_Report_" + System.currentTimeMillis() +
        ".pdf";
    response.setContentType("application/pdf");
    response.setHeader("Cache-Control", "no-cache");
    response.setHeader("Cache-Control", "max-age=0");
    response.setHeader("Content-disposition", "attachment; " + "filename=" + fileName);
```

```
        try {
            createPDF(temperotyFilePath + "\\" + fileName);
            ByteArrayOutputStream baos = new ByteArrayOutputStream();
            baos = convertPDFToByteArrayOutputStream(temperotyFilePath + "\\" +
                fileName);
            OutputStream os = response.getOutputStream();
            baos.writeTo(os);
            os.flush();
        } catch (Exception e1) {
            e1.printStackTrace();
        }
    }
```

```
protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
}
```

```
private static ByteArrayOutputStream convertPDFToByteArrayOutputStream(String
fileName) {
```

```
    InputStream inputStream = null;
    ByteArrayOutputStream baos = new ByteArrayOutputStream();
    try {
```

```
        inputStream = new FileInputStream(fileName);
```

```
        byte[] buffer = new byte[1024];
        baos = new ByteArrayOutputStream();
```

```
        int bytesRead;
        while ((bytesRead = inputStream.read(buffer)) != -1) {
            baos.write(buffer, 0, bytesRead);
        }
```

```
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    } catch (IOException e) {
```

```

        e.printStackTrace();
    } finally {
        if (inputStream != null) {
            try {
                inputStream.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
    return baos;
}

public static Document createPDF(String file) {
    Document document = null;
    try {
        document = new Document();
        PdfWriter.getInstance(document, new FileOutputStream(file));
        document.open();
        addMetaData(document);
        addTitlePage(document);
        createTable(document);
        document.close();

    } catch (FileNotFoundException e) {

        e.printStackTrace();
    } catch (DocumentException e) {
        e.printStackTrace();
    }
    return document;
}

private static void addMetaData(Document document) {
    document.addTitle("Generate PDF report");
    document.addSubject("Generate PDF report");
    document.addAuthor("Kishor Kadam");
    document.addCreator("Kishor Kadam");
}

private static void addTitlePage(Document document) throws DocumentException {

    Paragraph preface = new Paragraph();
    createEmptyLine(preface, 1);
    preface.add(new Paragraph("Student Attendance Report ", TIME_ROMAN));

    createEmptyLine(preface, 1);
    SimpleDateFormat simpleDateFormat = new SimpleDateFormat("MM/dd/yyyy");
    preface.add(
        new Paragraph("Student Attendance Report on " +
            simpleDateFormat.format(new Date()), TIME_ROMAN_SMALL));
}

```

```

        document.add(preface);

    }

    private static void creteEmptyLine(Paragraph paragraph, int number) {
        for (int i = 0; i < number; i++) {
            paragraph.add(new Paragraph(" "));
        }
    }

    private static void createTable(Document document) throws DocumentException {

        Paragraph paragraph = new Paragraph();
        creteEmptyLine(paragraph, 2);
        document.add(paragraph);
        PdfPTable table = new PdfPTable(5);

        PdfPCell c1 = new PdfPCell(new Phrase("Student Id"));
        c1.setHorizontalAlignment(Element.ALIGN_CENTER);
        table.addCell(c1);

        c1 = new PdfPCell(new Phrase("Student Name"));
        c1.setHorizontalAlignment(Element.ALIGN_CENTER);
        table.addCell(c1);

        c1 = new PdfPCell(new Phrase("Total Present Days"));
        c1.setHorizontalAlignment(Element.ALIGN_CENTER);
        table.addCell(c1);

        c1 = new PdfPCell(new Phrase("Total Absent Days"));
        c1.setHorizontalAlignment(Element.ALIGN_CENTER);
        table.addCell(c1);

        c1 = new PdfPCell(new Phrase("Total Days"));
        c1.setHorizontalAlignment(Element.ALIGN_CENTER);
        table.addCell(c1);

        table.setHeaderRows(1);

        PdfPCell[] cells = table.getRow(0).getCells();
        for (int j = 0; j < cells.length; j++) {
            cells[j].setBorderColor(BaseColor.BLACK);
            cells[j].setBackgroundColor(BaseColor.GRAY);
        }

        try {
            PdfPCell TableCell = null;
            table.setWidthPercentage(100);
            table.getDefaultCell().setHorizontalAlignment(Element.ALIGN_CENTER);
            table.getDefaultCell().setVerticalAlignment(Element.ALIGN_MIDDLE);
            ResultSet resultset = DatabaseConnection.getResultFromSqlQuery(
                "select
student_attendance.student_id,students.stud_name,count(case when attendance ='present' then 1 end)

```

```

as Present_Count,count(case when attendance ='absent' then 1 end) as Absent_Count,count(distinct
system_date) as Total_Count from student_attendance INNER JOIN students ON
students.student_id=student_attendance.student_id group by student_id ");
        while (resultset.next()) {
            String student_id = resultset.getString("student_id");
            String stud_name = resultset.getString("stud_name");
            String Present_Count = resultset.getString("Present_Count");
            String Absent_Count = resultset.getString("Absent_Count");
            String Total_Count = resultset.getString("Total_Count");
            TableCell = new PdfPCell(new Phrase(student_id));
            table.addCell(TableCell);
            TableCell = new PdfPCell(new Phrase(stud_name));
            table.addCell(TableCell);
            TableCell = new PdfPCell(new Phrase(Present_Count));
            table.addCell(TableCell);
            TableCell = new PdfPCell(new Phrase(Absent_Count));
            table.addCell(TableCell);
            TableCell = new PdfPCell(new Phrase(Total_Count));
            table.addCell(TableCell);
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
    document.add(table);
}
}

```

StudentActiveorinactive.java

```
package com.reports;
```

```

import java.io.ByteArrayOutputStream;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.sql.ResultSet;
import java.text.SimpleDateFormat;
import java.util.Date;
import javax.servlet.ServletContext;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import com.attendance.DatabaseConnection;
import com.itextpdf.text.BaseColor;
import com.itextpdf.text.Document;
import com.itextpdf.text.DocumentException;
import com.itextpdf.text.Element;
import com.itextpdf.text.Font;
import com.itextpdf.text.Paragraph;
import com.itextpdf.text.Phrase;
import com.itextpdf.text.pdf.PdfPCell;
import com.itextpdf.text.pdf.PdfPTable;

```

```

import com.itextpdf.text.pdf.PdfWriter;

/**
 * Servlet implementation class UserTradingPDFReport
 */
@WebServlet("/StudentsActiveOrInactive")
public class StudentsActiveOrInactive extends HttpServlet {
    private static final long serialVersionUID = 1L;
    private static Font TIME_ROMAN = new Font(Font.FontFamily.TIMES_ROMAN, 18, Font.BOLD);
    private static Font TIME_ROMAN_SMALL = new Font(Font.FontFamily.TIMES_ROMAN, 12,
Font.BOLD);

    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        final ServletContext servletContext = request.getSession().getServletContext();
        final File tempDirectory = (File)
servletContext.getAttribute("javax.servlet.context.tempdir");
        final String temperotyFilePath = tempDirectory.getAbsolutePath();
        String fileName = "All_Students_Active_Or_Inactive_Report_" +
System.currentTimeMillis() + ".pdf";
        response.setContentType("application/pdf");
        response.setHeader("Cache-Control", "no-cache");
        response.setHeader("Cache-Control", "max-age=0");
        response.setHeader("Content-disposition", "attachment; " + "filename=" +
fileName);

        try {
            createPDF(temperotyFilePath + "\\\" + fileName);
            ByteArrayOutputStream baos = new ByteArrayOutputStream();
            baos = convertPDFToByteArrayOutputStream(temperotyFilePath + "\\\" +
fileName);

            OutputStream os = response.getOutputStream();
            baos.writeTo(os);
            os.flush();
        } catch (Exception e1) {
            e1.printStackTrace();
        }
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
    }

    private static ByteArrayOutputStream convertPDFToByteArrayOutputStream(String fileName)
    {

        InputStream inputStream = null;
        ByteArrayOutputStream baos = new ByteArrayOutputStream();
        try {

            inputStream = new FileInputStream(fileName);

            byte[] buffer = new byte[1024];
            baos = new ByteArrayOutputStream();

            int bytesRead;
            while ((bytesRead = inputStream.read(buffer)) != -1) {
                baos.write(buffer, 0, bytesRead);
            }

        } catch (FileNotFoundException e) {
            e.printStackTrace();
        } catch (IOException e) {

```

```

        e.printStackTrace();
    } finally {
        if (inputStream != null) {
            try {
                inputStream.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
    return baos;
}

public static Document createPDF(String file) {
    Document document = null;
    try {
        document = new Document();
        PdfWriter.getInstance(document, new FileOutputStream(file));
        document.open();
        addMetaData(document);
        addTitlePage(document);
        createTable(document);
        document.close();

    } catch (FileNotFoundException e) {

        e.printStackTrace();
    } catch (DocumentException e) {
        e.printStackTrace();
    }
    return document;
}

private static void addMetaData(Document document) {
    document.addTitle("Generate PDF report");
    document.addSubject("Generate PDF report");
    document.addAuthor("Kishor Kadam");
    document.addCreator("Kishor Kadam");
}

private static void addTitlePage(Document document) throws DocumentException {

    Paragraph preface = new Paragraph();
    creteEmptyLine(preface, 1);
    preface.add(new Paragraph("Students Active Or Inactive Report ", TIME_ROMAN));

    creteEmptyLine(preface, 1);
    SimpleDateFormat simpleDateFormat = new SimpleDateFormat("MM/dd/yyyy");
    preface.add(new Paragraph("Students Active Or Inactive Report on " +
simpleDateFormat.format(new Date()), TIME_ROMAN_SMALL));

    document.add(preface);
}

private static void creteEmptyLine(Paragraph paragraph, int number) {
    for (int i = 0; i < number; i++) {
        paragraph.add(new Paragraph(" "));
    }
}

private static void createTable(Document document) throws DocumentException {

```

```

Paragraph paragraph = new Paragraph();
createEmptyLine(paragraph, 2);
document.add(paragraph);
PdfPTable table = new PdfPTable(6);

PdfPCell c1 = new PdfPCell(new Phrase("Student Id"));
c1.setHorizontalAlignment(Element.ALIGN_CENTER);
table.addCell(c1);

c1 = new PdfPCell(new Phrase("Student Name"));
c1.setHorizontalAlignment(Element.ALIGN_CENTER);
table.addCell(c1);

c1 = new PdfPCell(new Phrase("College Name"));
c1.setHorizontalAlignment(Element.ALIGN_CENTER);
table.addCell(c1);

c1 = new PdfPCell(new Phrase("Mobile"));
c1.setHorizontalAlignment(Element.ALIGN_CENTER);
table.addCell(c1);

c1 = new PdfPCell(new Phrase("Address"));
c1.setHorizontalAlignment(Element.ALIGN_CENTER);
table.addCell(c1);

c1 = new PdfPCell(new Phrase("Status"));
c1.setHorizontalAlignment(Element.ALIGN_CENTER);
table.addCell(c1);

table.setHeaderRows(1);

PdfPCell[] cells = table.getRow(0).getCells();
for (int j = 0; j < cells.length; j++) {
    cells[j].setBorderColor(BaseColor.BLACK);
    cells[j].setBackgroundColor(BaseColor.GRAY);
}

try {
    PdfPCell TableCell = null;
    table.setWidthPercentage(100);
    table.getDefaultCell().setHorizontalAlignment(Element.ALIGN_CENTER);
    table.getDefaultCell().setVerticalAlignment(Element.ALIGN_MIDDLE);
    ResultSet resultset = DatabaseConnection.getResultFromSqlQuery("select *
from students ");
    while (resultset.next()) {
        String student_id = resultset.getString("student_id");
        String stud_name = resultset.getString("stud_name");
        String college_name = resultset.getString("college_name");
        String mobile = resultset.getString("mobile");
        String address = resultset.getString("address");
        String status = resultset.getString("status");
        TableCell = new PdfPCell(new Phrase(student_id));
        table.addCell(TableCell);
        TableCell = new PdfPCell(new Phrase(stud_name));
        table.addCell(TableCell);
        TableCell = new PdfPCell(new Phrase(college_name));
        table.addCell(TableCell);
        TableCell = new PdfPCell(new Phrase(mobile));
        table.addCell(TableCell);
        TableCell = new PdfPCell(new Phrase(address));
        table.addCell(TableCell);
        TableCell = new PdfPCell(new Phrase(status));
    }
}

```

```

        table.addCell(TableCell);
    }
} catch (Exception e) {
    e.printStackTrace();
}
document.add(table);
}
}

```

StudentFeedbackReport.java

```

package com.reports;

import java.io.ByteArrayOutputStream;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.sql.ResultSet;
import java.text.SimpleDateFormat;
import java.util.Date;
import javax.servlet.ServletContext;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import com.attendance.DatabaseConnection;
import com.itextpdf.text.BaseColor;
import com.itextpdf.text.Document;
import com.itextpdf.text.DocumentException;
import com.itextpdf.text.Element;
import com.itextpdf.text.Font;
import com.itextpdf.text.Paragraph;
import com.itextpdf.text.Phrase;
import com.itextpdf.text.pdf.PdfPCell;
import com.itextpdf.text.pdf.PdfPTable;
import com.itextpdf.text.pdf.PdfWriter;

/**
 * Servlet implementation class UserTradingPDFReport
 */
@WebServlet("/StudentsFeedbackReport")
public class StudentsFeedbackReport extends HttpServlet {
    private static final long serialVersionUID = 1L;
    private static Font TIME_ROMAN = new Font(Font.FontFamily.TIMES_ROMAN, 18, Font.BOLD);
    private static Font TIME_ROMAN_SMALL = new Font(Font.FontFamily.TIMES_ROMAN, 12,
Font.BOLD);

    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        final ServletContext servletContext = request.getSession().getServletContext();
        final File tempDirectory = (File)
servletContext.getAttribute("javax.servlet.context.tempdir");
        final String temperotyFilePath = tempDirectory.getAbsolutePath();
        String fileName = "Students_Feedback_Report_" + System.currentTimeMillis() +
".pdf";

        response.setContentType("application/pdf");
        response.setHeader("Cache-Control", "no-cache");
        response.setHeader("Cache-Control", "max-age=0");
    }
}

```



```

        response.setHeader("Content-disposition", "attachment; " + "filename=" +
fileName);

        try {
            createPDF(temperotyFilePath + "\\ " + fileName);
            ByteArrayOutputStream baos = new ByteArrayOutputStream();
            baos = convertPDFToByteArrayOutputStream(temperotyFilePath + "\\ " +
fileName);

            OutputStream os = response.getOutputStream();
            baos.writeTo(os);
            os.flush();
        } catch (Exception e1) {
            e1.printStackTrace();
        }
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
    }

    private static ByteArrayOutputStream convertPDFToByteArrayOutputStream(String fileName)
    {

        InputStream inputStream = null;
        ByteArrayOutputStream baos = new ByteArrayOutputStream();
        try {

            inputStream = new FileInputStream(fileName);

            byte[] buffer = new byte[1024];
            baos = new ByteArrayOutputStream();

            int bytesRead;
            while ((bytesRead = inputStream.read(buffer)) != -1) {
                baos.write(buffer, 0, bytesRead);
            }

        } catch (FileNotFoundException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        } finally {
            if (inputStream != null) {
                try {
                    inputStream.close();
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        }
        return baos;
    }

    public static Document createPDF(String file) {
        Document document = null;
        try {
            document = new Document();
            PdfWriter.getInstance(document, new FileOutputStream(file));
            document.open();
            addMetaData(document);
            addTitlePage(document);
            createTable(document);
            document.close();

```

```

        } catch (FileNotFoundException e) {

            e.printStackTrace();
        } catch (DocumentException e) {
            e.printStackTrace();
        }
        return document;
    }

    private static void addMetaData(Document document) {
        document.addTitle("Generate PDF report");
        document.addSubject("Generate PDF report");
        document.addAuthor("Kishor Kadam");
        document.addCreator("Kishor Kadam");
    }

    private static void addTitlePage(Document document) throws DocumentException {

        Paragraph preface = new Paragraph();
        createEmptyLine(preface, 1);
        preface.add(new Paragraph("Students Feedback Report ", TIME_ROMAN));

        createEmptyLine(preface, 1);
        SimpleDateFormat simpleDateFormat = new SimpleDateFormat("MM/dd/yyyy");
        preface.add(new Paragraph("Students Feedback Report on " +
simpleDateFormat.format(new Date()), TIME_ROMAN_SMALL));

        document.add(preface);
    }

    private static void createEmptyLine(Paragraph paragraph, int number) {
        for (int i = 0; i < number; i++) {
            paragraph.add(new Paragraph(" "));
        }
    }

    private static void createTable(Document document) throws DocumentException {

        Paragraph paragraph = new Paragraph();
        createEmptyLine(paragraph, 2);
        document.add(paragraph);
        PdfPTable table = new PdfPTable(4);

        PdfPCell c1 = new PdfPCell(new Phrase("Student Id"));
        c1.setHorizontalAlignment(Element.ALIGN_CENTER);
        table.addCell(c1);

        c1 = new PdfPCell(new Phrase("Student Name"));
        c1.setHorizontalAlignment(Element.ALIGN_CENTER);
        table.addCell(c1);

        c1 = new PdfPCell(new Phrase("Mobile"));
        c1.setHorizontalAlignment(Element.ALIGN_CENTER);
        table.addCell(c1);

        c1 = new PdfPCell(new Phrase("Feedback"));
        c1.setHorizontalAlignment(Element.ALIGN_CENTER);
        table.addCell(c1);
    }

```

```

        table.setHeaderRows(1);

        PdfPCell[] cells = table.getRow(0).getCells();
        for (int j = 0; j < cells.length; j++) {
            cells[j].setBorderColor(BaseColor.BLACK);
            cells[j].setBackgroundColor(BaseColor.GRAY);
        }

        try {
            PdfPCell TableCell = null;
            table.setWidthPercentage(100);
            table.getDefaultCell().setHorizontalAlignment(Element.ALIGN_CENTER);
            table.getDefaultCell().setVerticalAlignment(Element.ALIGN_MIDDLE);
            ResultSet resultset = DatabaseConnection.getResultFromSqlQuery("select *
from student_feedback ");
            while (resultset.next()) {
                String fid = resultset.getString("fid");
                String full_name = resultset.getString("full_name");
                String mobile = resultset.getString("mobile");
                String feedback = resultset.getString("feedback");
                TableCell = new PdfPCell(new Phrase(fid));
                table.addCell(TableCell);
                TableCell = new PdfPCell(new Phrase(full_name));
                table.addCell(TableCell);
                TableCell = new PdfPCell(new Phrase(mobile));
                table.addCell(TableCell);
                TableCell = new PdfPCell(new Phrase(feedback));
                table.addCell(TableCell);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
        document.add(table);
    }
}

```

StudentleavedApproved.java

```

package com.reports;

import java.io.ByteArrayOutputStream;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.sql.ResultSet;
import java.text.SimpleDateFormat;
import java.util.Date;
import javax.servlet.ServletContext;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import com.attendance.DatabaseConnection;
import com.itextpdf.text.BaseColor;
import com.itextpdf.text.Document;
import com.itextpdf.text.DocumentException;

```

```

import com.itextpdf.text.Element;
import com.itextpdf.text.Font;
import com.itextpdf.text.Paragraph;
import com.itextpdf.text.Phrase;
import com.itextpdf.text.pdf.PdfPCell;
import com.itextpdf.text.pdf.PdfPTable;
import com.itextpdf.text.pdf.PdfWriter;

/**
 * Servlet implementation class UserTradingPDFReport
 */
@WebServlet("/StudentsLeaveApproved")
public class StudentsLeaveApproved extends HttpServlet {
    private static final long serialVersionUID = 1L;
    private static Font TIME_ROMAN = new Font(Font.FontFamily.TIMES_ROMAN, 18, Font.BOLD);
    private static Font TIME_ROMAN_SMALL = new Font(Font.FontFamily.TIMES_ROMAN, 12,
Font.BOLD);

    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        final ServletContext servletContext = request.getSession().getServletContext();
        final File tempDirectory = (File)
servletContext.getAttribute("javax.servlet.context.tempdir");
        final String temperotyFilePath = tempDirectory.getAbsolutePath();
        String fileName = "Students_Leave_Approved_Report_" + System.currentTimeMillis() +
".pdf";

        response.setContentType("application/pdf");
        response.setHeader("Cache-Control", "no-cache");
        response.setHeader("Cache-Control", "max-age=0");
        response.setHeader("Content-disposition", "attachment; " + "filename=" +
fileName);

        try {
            createPDF(temperotyFilePath + "\\\" + fileName);
            ByteArrayOutputStream baos = new ByteArrayOutputStream();
            baos = convertPDFToByteArrayOutputStream(temperotyFilePath + "\\\" +
fileName);

            OutputStream os = response.getOutputStream();
            baos.writeTo(os);
            os.flush();
        } catch (Exception e1) {
            e1.printStackTrace();
        }
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
    }

    private static ByteArrayOutputStream convertPDFToByteArrayOutputStream(String fileName)
{
        InputStream inputStream = null;
        ByteArrayOutputStream baos = new ByteArrayOutputStream();
        try {

            inputStream = new FileInputStream(fileName);

            byte[] buffer = new byte[1024];
            baos = new ByteArrayOutputStream();

            int bytesRead;
            while ((bytesRead = inputStream.read(buffer)) != -1) {

```

```

        baos.write(buffer, 0, bytesRead);
    }

    } catch (FileNotFoundException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    } finally {
        if (inputStream != null) {
            try {
                inputStream.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
    return baos;
}

public static Document createPDF(String file) {
    Document document = null;
    try {
        document = new Document();
        PdfWriter.getInstance(document, new FileOutputStream(file));
        document.open();
        addMetaData(document);
        addTitlePage(document);
        createTable(document);
        document.close();

    } catch (FileNotFoundException e) {

        e.printStackTrace();
    } catch (DocumentException e) {
        e.printStackTrace();
    }
    return document;
}

private static void addMetaData(Document document) {
    document.addTitle("Generate PDF report");
    document.addSubject("Generate PDF report");
    document.addAuthor("Kishor Kadam");
    document.addCreator("Kishor Kadam");
}

private static void addTitlePage(Document document) throws DocumentException {

    Paragraph preface = new Paragraph();
    creteEmptyLine(preface, 1);
    preface.add(new Paragraph("Students Leave Approved Report ", TIME_ROMAN));

    creteEmptyLine(preface, 1);
    SimpleDateFormat simpleDateFormat = new SimpleDateFormat("MM/dd/yyyy");
    preface.add(new Paragraph("Students Leave Approved Report on " +
simpleDateFormat.format(new Date()), TIME_ROMAN_SMALL));

    document.add(preface);
}

private static void creteEmptyLine(Paragraph paragraph, int number) {

```

```

        for (int i = 0; i < number; i++) {
            paragraph.add(new Paragraph(" "));
        }
    }

    private static void createTable(Document document) throws DocumentException {

        Paragraph paragraph = new Paragraph();
        createEmptyLine(paragraph, 2);
        document.add(paragraph);
        PdfPTable table = new PdfPTable(5);

        PdfPCell c1 = new PdfPCell(new Phrase("Student Id"));
        c1.setHorizontalAlignment(Element.ALIGN_CENTER);
        table.addCell(c1);

        c1 = new PdfPCell(new Phrase("Student Name"));
        c1.setHorizontalAlignment(Element.ALIGN_CENTER);
        table.addCell(c1);

        c1 = new PdfPCell(new Phrase("Leave Reason"));
        c1.setHorizontalAlignment(Element.ALIGN_CENTER);
        table.addCell(c1);

        c1 = new PdfPCell(new Phrase("No of Days"));
        c1.setHorizontalAlignment(Element.ALIGN_CENTER);
        table.addCell(c1);

        c1 = new PdfPCell(new Phrase("Leave Status"));
        c1.setHorizontalAlignment(Element.ALIGN_CENTER);
        table.addCell(c1);

        table.setHeaderRows(1);

        PdfPCell[] cells = table.getRow(0).getCells();
        for (int j = 0; j < cells.length; j++) {
            cells[j].setBorderColor(BaseColor.BLACK);
            cells[j].setBackgroundColor(BaseColor.GRAY);
        }

        try {
            PdfPCell TableCell = null;
            table.setWidthPercentage(100);
            table.getDefaultCell().setHorizontalAlignment(Element.ALIGN_CENTER);
            table.getDefaultCell().setVerticalAlignment(Element.ALIGN_MIDDLE);
            ResultSet resultset = DatabaseConnection.getResultFromSqlQuery("select *
from student_leave where leave_status='Approved'");
            while (resultset.next()) {
                String id = resultset.getString("id");
                String name = resultset.getString("name");
                String leave_reasone = resultset.getString("leave_reasone");
                String no_of_days = resultset.getString("no_of_days");
                String leave_status = resultset.getString("leave_status");
                TableCell = new PdfPCell(new Phrase(id));
                table.addCell(TableCell);
                TableCell = new PdfPCell(new Phrase(name));
                table.addCell(TableCell);
                TableCell = new PdfPCell(new Phrase(leave_reasone));
                table.addCell(TableCell);
                TableCell = new PdfPCell(new Phrase(no_of_days));
                table.addCell(TableCell);
                TableCell = new PdfPCell(new Phrase(leave_status));
            }
        }
    }
}

```

```

        table.addCell(TableCell);
    }
    } catch (Exception e) {
        e.printStackTrace();
    }
    document.add(table);
}
}

```

StudentLeavepending.java

```

package com.reports;

import java.io.ByteArrayOutputStream;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.sql.ResultSet;
import java.text.SimpleDateFormat;
import java.util.Date;

import javax.servlet.ServletContext;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import com.attendance.DatabaseConnection;
import com.itextpdf.text.BaseColor;
import com.itextpdf.text.Document;
import com.itextpdf.text.DocumentException;
import com.itextpdf.text.Element;
import com.itextpdf.text.Font;
import com.itextpdf.text.Paragraph;
import com.itextpdf.text.Phrase;
import com.itextpdf.text.pdf.PdfPCell;
import com.itextpdf.text.pdf.PdfPTable;
import com.itextpdf.text.pdf.PdfWriter;

/**
 * Servlet implementation class StudentsLeavePending
 */
@WebServlet("/StudentsLeavePending")
public class StudentsLeavePending extends HttpServlet {
    private static final long serialVersionUID = 1L;
    private static Font TIME_ROMAN = new Font(Font.FontFamily.TIMES_ROMAN, 18, Font.BOLD);
    private static Font TIME_ROMAN_SMALL = new Font(Font.FontFamily.TIMES_ROMAN, 12,
Font.BOLD);

    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        final ServletContext servletContext = request.getSession().getServletContext();
        final File tempDirectory = (File)
servletContext.getAttribute("javax.servlet.context.tempdir");
        final String temperotyFilePath = tempDirectory.getAbsolutePath();
        String fileName = "Students-Leave-Pending-Report-" + System.currentTimeMillis() +
".pdf";

        response.setContentType("application/pdf");
        response.setHeader("Cache-Control", "no-cache");
    }
}

```

```

        response.setHeader("Cache-Control", "max-age=0");
        response.setHeader("Content-disposition", "attachment; " + "filename=" +
fileName);

        try {
            createPDF(temperotyFilePath + "\\ " + fileName);
            ByteArrayOutputStream baos = new ByteArrayOutputStream();
            baos = convertPDFToByteArrayOutputStream(temperotyFilePath + "\\ " +
fileName);

            OutputStream os = response.getOutputStream();
            baos.writeTo(os);
            os.flush();
        } catch (Exception e1) {
            e1.printStackTrace();
        }
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
    }

    private static ByteArrayOutputStream convertPDFToByteArrayOutputStream(String fileName)
    {

        InputStream inputStream = null;
        ByteArrayOutputStream baos = new ByteArrayOutputStream();
        try {

            inputStream = new FileInputStream(fileName);

            byte[] buffer = new byte[1024];
            baos = new ByteArrayOutputStream();

            int bytesRead;
            while ((bytesRead = inputStream.read(buffer)) != -1) {
                baos.write(buffer, 0, bytesRead);
            }

        } catch (FileNotFoundException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        } finally {
            if (inputStream != null) {
                try {
                    inputStream.close();
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        }
        return baos;
    }

    public static Document createPDF(String file) {
        Document document = null;
        try {
            document = new Document();
            PdfWriter.getInstance(document, new FileOutputStream(file));
            document.open();
            addMetaData(document);
            addTitlePage(document);
            createTable(document);

```



```

        document.close();

    } catch (FileNotFoundException e) {

        e.printStackTrace();
    } catch (DocumentException e) {
        e.printStackTrace();
    }
    return document;
}

private static void addMetaData(Document document) {
    document.addTitle("Generate PDF report");
    document.addSubject("Generate PDF report");
    document.addAuthor("Kishor Kadam");
    document.addCreator("Kishor Kadam");
}

private static void addTitlePage(Document document) throws DocumentException {

    Paragraph preface = new Paragraph();
    createEmptyLine(preface, 1);
    preface.add(new Paragraph("Students Leave Pending Report ", TIME_ROMAN));

    createEmptyLine(preface, 1);
    SimpleDateFormat simpleDateFormat = new SimpleDateFormat("MM/dd/yyyy");
    preface.add(new Paragraph("Students Leave Pending Report on " +
simpleDateFormat.format(new Date()), TIME_ROMAN_SMALL));

    document.add(preface);
}

private static void createEmptyLine(Paragraph paragraph, int number) {
    for (int i = 0; i < number; i++) {
        paragraph.add(new Paragraph(" "));
    }
}

private static void createTable(Document document) throws DocumentException {

    Paragraph paragraph = new Paragraph();
    createEmptyLine(paragraph, 2);
    document.add(paragraph);
    PdfPTable table = new PdfPTable(5);

    PdfPCell c1 = new PdfPCell(new Phrase("Student Id"));
    c1.setHorizontalAlignment(Element.ALIGN_CENTER);
    table.addCell(c1);

    c1 = new PdfPCell(new Phrase("Student Name"));
    c1.setHorizontalAlignment(Element.ALIGN_CENTER);
    table.addCell(c1);

    c1 = new PdfPCell(new Phrase("Leave Reason"));
    c1.setHorizontalAlignment(Element.ALIGN_CENTER);
    table.addCell(c1);

    c1 = new PdfPCell(new Phrase("No of Days"));
    c1.setHorizontalAlignment(Element.ALIGN_CENTER);
    table.addCell(c1);
}

```

```

c1 = new PdfPCell(new Phrase("Leave Status"));
c1.setHorizontalAlignment(Element.ALIGN_CENTER);
table.addCell(c1);

table.setHeaderRows(1);

PdfPCell[] cells = table.getRow(0).getCells();
for (int j = 0; j < cells.length; j++) {
    cells[j].setBorderColor(BaseColor.BLACK);
    cells[j].setBackgroundColor(BaseColor.GRAY);
}

try {
    PdfPCell TableCell = null;
    table.setWidthPercentage(100);
    table.getDefaultCell().setHorizontalAlignment(Element.ALIGN_CENTER);
    table.getDefaultCell().setVerticalAlignment(Element.ALIGN_MIDDLE);
    ResultSet resultset = DatabaseConnection.getResultFromSqlQuery("select *
from student_leave where leave_status='Pending'");
    while (resultset.next()) {
        String id = resultset.getString("id");
        String name = resultset.getString("name");
        String leave_reasone = resultset.getString("leave_reasone");
        String no_of_days = resultset.getString("no_of_days");
        String leave_status = resultset.getString("leave_status");
        TableCell = new PdfPCell(new Phrase(id));
        table.addCell(TableCell);
        TableCell = new PdfPCell(new Phrase(name));
        table.addCell(TableCell);
        TableCell = new PdfPCell(new Phrase(leave_reasone));
        table.addCell(TableCell);
        TableCell = new PdfPCell(new Phrase(no_of_days));
        table.addCell(TableCell);
        TableCell = new PdfPCell(new Phrase(leave_status));
        table.addCell(TableCell);
    }
} catch (Exception e) {
    e.printStackTrace();
}
document.add(table);
}
}

```