

VISVESVARAYA TECHNOLOGICAL UNIVERSITY
BELAGAVI-590018



“A MINI PROJECT REPORT”
(Subject: File Structure Laboratory with Mini Project)
(Subject Code: 18ISL67)
ON
“PLACEMENT MANAGEMENT SYSTEM”

Submitted in partial fulfillment for the requirements for the Award of Degree of

BACHELOR OF ENGINEERING
IN
INFORMATION SCIENCE AND ENGINEERING
BY

Mounika S	1EP20IS059
Poornima B	1EP20IS070
S P Sindhuja	1EP20IS078
Usha S	1EP20IS096

UNDER THE GUIDANCE OF

Prof. Tejashwini B	Prof. Teena KB
Assistant Professor	Assistant Professor
Dept. of ISE	Dept. of ISE



(Affiliated to Visvesvaraya Technological University, Belgavi)

Bangalore-560049

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the **Mini Project Work** entitled “**title of the Project**” carried out by **Ms. Mounika S, USN 1EP20IS059, Ms. Poornima B, USN 1EP20IS070, Ms. S P Sindhuja, USN 1EP20IS078, Ms. Usha S, USN 1EP20IS096**, bonafide students of **East Point College of Engineering and Technology** in partial fulfillment for the award of **Bachelor of Engineering in Information Science and Engineering** of **Visvesvaraya Technological University, Belgavi** during the year **2022-23**. It is certified that all the corrections/suggestions indicated for the Internal Assessment have been incorporated in the report deposited in the department library. The project report has been approved as it satisfies the academic requirements in respect of File Structures Laboratory with Mini Project (18ISL67) prescribed for the award of the said degree.

GUIDE
Prof. Teena KB
Asst. Professor

HOD
Prof. N Kemparaju
EPCET

PRINCIPAL
Dr. Yogesh G S
EPCET

Examiners

Name of the Examiners

Signature with date

1.

2.

ACKNOWLEDGEMENT

Any achievement, be it scholastic or otherwise does not depend solely on the individual efforts but on the guidance, encouragement and cooperation of intellectuals, elders and friends. We would like to take this opportunity to thank them all.

First and foremost we would like to express our sincere regards and thanks to **Mr. Pramod Gowda** and **Mr. Rajiv Gowda**, CEOs, East Point Group of Institutions, Bangalore, for providing necessary infrastructure and creating good environment.

We express our gratitude to **Dr. Prakash S**, Senior Vice President, EPGI and **Dr. Yogesh G S**, Principal, EPCET who has always been a great source of inspiration.

We express our sincere regards and thanks to **Prof. N Kemparaju**, Professor and Head, Department of Information Science and Engineering, EPCET, Bangalore, for his encouragement and support.

We are grateful to acknowledge the guidance and encouragement given to us by **Prof. Kavana K**, Assistant Professor and **Prof. Vaishali Sontakke**, Assistant Professor, Department of Information Science and Engineering, EPCET, Bangalore, who has rendered a valuable assistance.

We also extend our thanks to the entire faculty of the Department of **Information Science and Engineering, EPCET**, Bangalore, who have encouraged us throughout the course of the Project.

Last, but not the least, we would like to thank our family and friends for their inputs to improve the Project.

MOUNIKA S	1EP20IS059
POORNIMA B	1EP20IS070
S P SINDHUJA	1EP20IS078
USHA S	1EP20IS096

INDEX

Chapter 1 Introduction	1
1.1 Outline	1
1.2 Motivation and Scope	2
1.3 Problem Statement	2
1.4 Limitations	2
Chapter 2 Requirements Specification	3
2.1 Functional Requirements	3
2.2 Non-Functional Requirements	4
2.3 Domain Constraints	4
Chapter 3 Requirements and System Analysis	5
3.1 Overall Process of the Project	5
3.2 Components/Subsystem Design	6
Chapter 4 System Design	9
4.1 UI Logic Interface/Interaction Details	9
Chapter 5 Implementation	12
5.1 Description of Frameworks Used	12
5.2 Description of Integrated Development Environment	13
Chapter 6 Testing	14
6.1 Component Tests	14
6.2 System Tests	18
Chapter 7 Interpretation of Results	19
Conclusion	26
References	27

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE NO.
3.1	Overall System Design	5
3.1	Create File Flow Diagram	6
3.2	Display File Flow Diagram	6
3.3	Delete Flow Diagram	6
3.4	Search Flow Diagram	7
3.5	View Changes Flow Diagram	7
3.6	View Eligibility Flow Diagram	7
4.1	JButton Insert/Modify Details	9
4.2	JButton Delete Record	9
4.3	JButton Display Record	9
4.4	JButton Search Record	9
4.5	JButton View Changes	10
4.6	JButton Placement Eligibility	10
4.7	JButton Insert	10
4.8	JButton Delete	10
4.9	JButton Search	10
4.10	JButton Display	11
4.11	JButton Display (View Details)	11
4.12	JButton Display (Placement Eligibility)	11
4.13	JButton Go Back	11
7.1	Home Page	19
7.2	Insert Student Records Page	19
7.3	Delete Student Record Page	20
7.4	Display Student Details Page	20
7.5	Search Records Page	21
7.6	Search By USN Page	21
7.7	Search By Branch Page	22
7.8	Search By Company Page	22
7.9	View Changes Page	23
7.10	Placement Eligibility Page	23

LIST OF TABLES

TABLE NO.	TITLE	PAGE NO.
6.1	Home Page Module Tests	14
6.2	Insert/Modify Details Module Tests	15
6.3	Delete Record Module Tests	15
6.4	Display Record Module Tests	16
6.5	Search Record Module Tests	16
6.6	View Changes Module Tests	17
6.7	Placement Eligibility Module Tests	17
6.8	Complete System Tests	18

Chapter 1

INTRODUCTION

1.1 Outline

Placement Management System stores record system that can be used as an application for the Placement Officer of the college to manage the student information with regards to placement. So all the information will store the details of the students including their background information, educational qualification, personal details, and all the information related to their resume. This system helps Company to access the student information with regards to placement.

Typical Features include:

- **Insert/Modify Student Records** – This system helps the placement team to insert each student record once and update it timely, and store it for further usages during placements, segregating students based on their CGPA, their interests and aspirations based on the eligibility criteria.
- **Delete Student Records** – This system enables options of deleting existing student records, or wrongly entered student record.
- **Eligibility View** – As a part of final consolidated view this system provides a view of student where each student's eligibility criteria is checked and corresponding eligible companies are displayed.
- **View Changes** – We have an option to view each and every changes done to each record from the time of insertion followed by any number of modifications done to the record.
- **Searching Student record** – As with placement system will always want to categorise students and view particular set of records at a time, this system enables such features by providing variety of search categories namely search by USN, search by branch, search by company.

Placement Management System provides a clear picture of all students where they stand in terms of placement eligibility criteria and which companies they are eligibility to, it provides an efficient search feature to the user which helps the placement handling team to search records of student by their USN, or set of records by each branch, or by the company name.

1.2 Motivation and Scope

- Nowadays we all are using the internet to do multiple things like booking, academic search, and blogging, apply for any job, etc. This system can be used as an application to manage student information related to placement.
- The system handles student as well as company data and efficiently displays all this data to respective sides. This System does all work regarding placement like collecting student records and finding the eligibility of the students and is provided to Placement Officer, Company, College staff and students.
- This system can be used as an application for the Placement Officer of the college to manage the student information with regards to placements, which presently is a cumbersome task via excel sheet database.

1.3 Problem Statement

Design and develop a system for campus placement cell that manages student information in the college with regard to campus placements, which improves existing system. It has the facility of maintaining the details of the student. Placement Management System can be accessed throughout the college. Thus this stand-alone application will be used as an application for the Placement Officer of the college to manage the student information with regards to placements.

1.4 Limitations

Placement Management System uses components from the Java Swings which is pretty outdated in comparison to JFX and other GUI building technologies. There is scope of improvement in better GUI and more user interactivity options, People prefer HTML + JS over Java. We can embed AI system to categorise students based on their skills and interests and their eligibility. More over since our developed system is in the initial stages of development many functions are yet to be implemented such as,

- Data Compression
- Undo and Redo

Chapter 2**Requirements Specification****2.1 Functional Requirements**

SL No.	Description
FR1	Inserting a record: This requires the user to create a new file handle, in this unit, User can insert a student record and write the content to the file. Next, if the record already exists then the student record is added to the Ledger File with modifications if any.
FR2	Deleting an Existing Record: Here the user can delete the contents of a specific student record by entering the respective student's USN. If the record exists, then the contents of the record are deleted. Otherwise, an appropriate message is displayed.
FR3	Displaying a Record: When a user enters this option, the user can view all the updated student records in the file.
FR4	Searching a Record: Here the user can view the contents of a specific student record by entering the respective student's USN, search by branch or also search by company details. If the record exists, then the contents of the record are displayed. Otherwise, an appropriate message is displayed.
FR5	View Changes: In this unit, the user can view the ledger file of each student and thereby can be updated with all the changes of all student records.
FR6	View Eligibility: Here the user can view the placement eligibility criteria of all the students in the record along with the various categories of placements based on college eligibility criteria.

2.2 Non-Functional Requirements

Performance

Performance of Placement Management System should always vary between a few hundred milliseconds. Time taken to creating records, Displaying records, Searching records, Deleting records, Viewing eligibility from within the file should be minimal.

Reliability

Placement Management System shall always provide fast and flexible file editing functionality no matter what the environment. System shall be robust enough to have high degree of fault tolerance. For example, if a user tries to open an invalid file it should display a proper error message. It shall be able to recover from hardware failures, power failures and other natural catastrophes and rollback the files to their most recent valid state.

Usability

Placement Management System shall provide an easy-to-use graphical interface similar to other existing text file editors so that the users do not have to learn a new style of interaction. Any notification or error messages generated by this system shall be clear, succinct, polite and free of jargon.

Integrity

The system must be programmed properly to prevent exploitation through buffer overflows etc. The system should be secure and could use encryption to protect the files. Users need to be authenticated before having access to any personal data.

Interoperability

The System shall minimize the effort required to couple it to another system, such as an Integrated Development Environment.

2.3 Domain Constraints

Hardware limitations: There must be a 64 MB on board memory

Control functions: The software must be very user-friendly and display appropriate error messages.

Dependencies: Requires JDK 8, Java SE and Java Swing Library. **Parallel operations:** It must support many file operations simultaneously.

Safety/security considerations: The application must be exited always normally.

Chapter 3

System/Requirements Analysis

3.1 Overall System Description

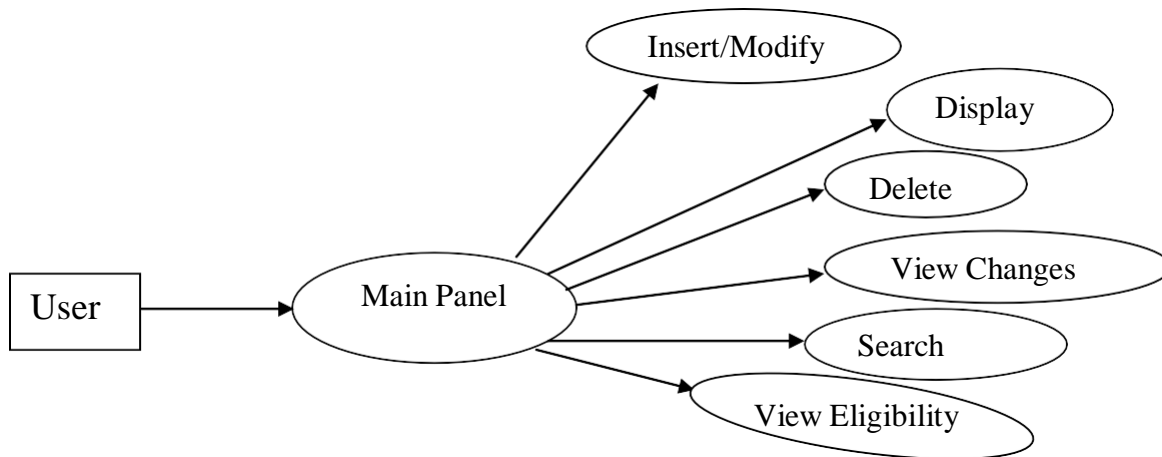


Fig. 3.1: Overall System Design

The overall description of the system is as follows:

The user is first presented with the Main Menu. Here user has a set of six options:

Insert: In this unit, User can insert a student record and write the content to the file. Next, if the record already exists then the student record is added to the Ledger File.

Display: In this unit, the user can view all the updated student records in the file.

Delete: In this unit, the user can delete the contents of a specific student record by entering the respective student's USN.

Search: Here the user can retrieve the required student's record by entering his/her USN, branch, company.

View Changes: In this unit, the user can view the ledger file of each student and thereby can be updated with all the changes of all student records.

View Eligibility: In this unit, the user can view the entire list of records and students eligibility to various companies.

3.2 Components/Subsystem Design

Create File Module

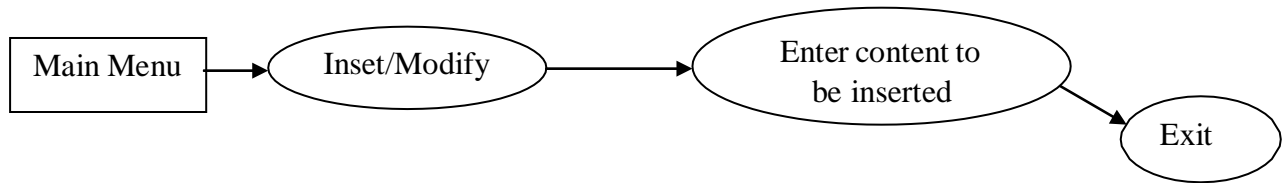


Fig. 3.2: Create File Flow Diagram

This requires the user to create a new file handle, in this unit; User can insert a student record and write the content to the file. Next, if the record already exists then the student record is added to the Ledger File.

Display Record Module

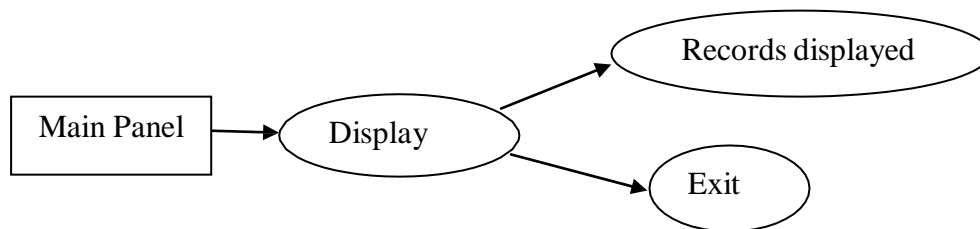


Fig. 3.3 Open File Flow Diagram

When a user enters this option, the user can view all the updated student records in the file.

Delete Record Module

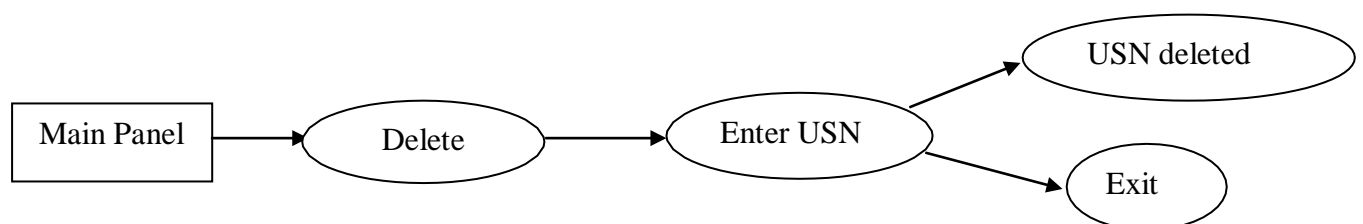


Fig. 3.4: Find and Replace Flow Diagram

Here the user can delete the contents of a specific student record by entering the respective student's USN. If the record exists, then the contents of the record are deleted. Otherwise, an appropriate message is displayed.

View Changes Module

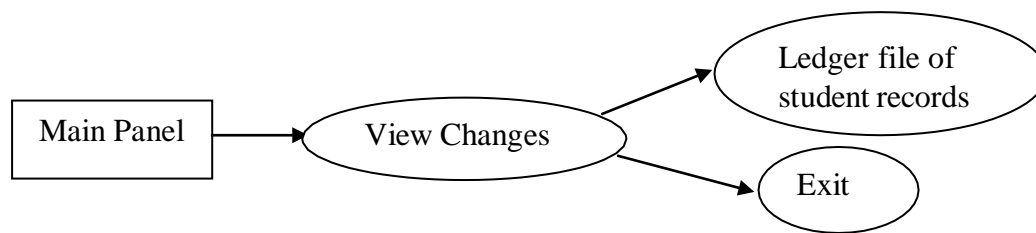


Fig. 3.5: Find and Replace Flow Diagram

In this unit, the user can view the ledger file of each student and thereby can be updated with all the changes of all student records.

View Eligibility Module

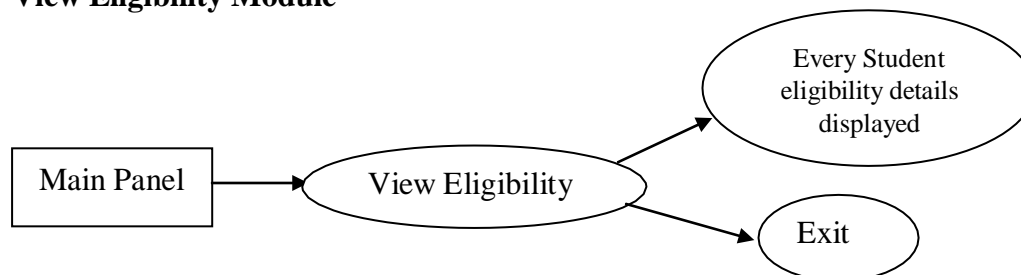


Fig. 3.6: Find and Replace Flow Diagram

Here the user can view the placement eligibility criteria of all the students in the record along with the various categories of placements.

Search Record Module

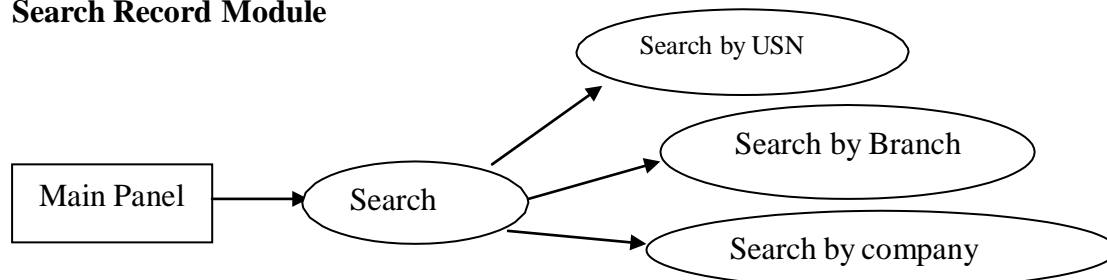


Fig. 3.7: Find and Replace Flow Diagram

Here the user can view the contents of a specific student record by entering the respective student's USN, by branch or by company details. If the record exists, then the contents of the record are displayed. Otherwise, an appropriate message is displayed.

General Ledger Management

Using General Ledger Problem, users are provided with a feature where they can see the history of all changes made to all records.

Finally, it provides information about the eligibility criteria for various packages and displays the list of all the company and packages each student is eligible for.

Using consequential matching and merging, we access all the records in both the sorted files.

The system contains:

A *journal file*, with the Student details for current semester that are ultimately to be posted to the ledger file, where the each student is marked by their USN.

A **ledger file**, containing Branch wise and Sem wise summaries of the values associated with each student.

Chapter 4

System Design

4.1 UI Logic Interface/Interaction Details

Home Page



Fig 4.1: JButton Insert/Modify Details

id: insert ; ActionEvent: go to Insert page

This Button is used to navigate from home page to insert page to inset student records.



Fig 4.2: JButton Delete Record

id: delete ; ActionEvent: go to Delete page

This Button is used to navigate from home page to delete page to delete student records.

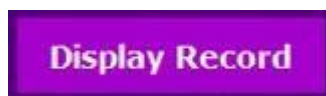


Fig 4.3: JButton Display Record

id: display ; ActionEvent: go to Display page

This Button is used to navigate from home page to display page to view all student records.

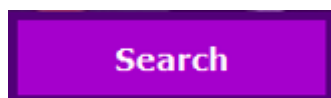


Fig 4.4: JButton Search Record

id: search ; ActionEvent: go to Search page

This Button is used to navigate from home page to search home page to select search category.

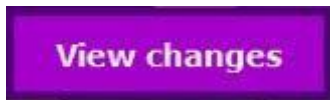


Fig 4.5: JButton View Changes

id: ledger: ActionEvent: go to View Changes page

This Button is used to navigate from home page to view ledger of student records.



Fig 4.6: JButton Placement Eligibility

id: eligible: ActionEvent: go to Placement Eligibility page

This Button is used to navigate from home page to student eligibility page.

Insert Page

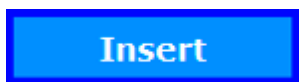


Fig 4.7: JButton Insert

id: insert ; ActionEvent: Insert

This Button is used to inset record into files after user has entered values in the inset page.

Delete Page



Fig 4.8: JButton Delete

id: delete: ActionEvent: Delete

This Button is used to delete record from files with matching credentials.

Search Page

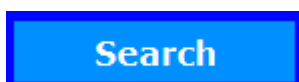


Fig 4.9: JButton Search

id: search: ActionEvent: Search

This Button is used to search record from files with matching credential

Display Page



Fig 4.10: JButton Display

id: display all records ; ActionEvent: Display

This Button is used to display record from files with matching credentials.

View Changes Page



Fig 4.11: JButton Display (View Details)

id: display all changes ; ActionEvent: Display

This Button is used to display record from ledger file to view all modification done to the record.

Placement Eligibility Page



Fig 4.1.12: JButton Display (Placement Eligibility)

id: display eligibility of all records ; ActionEvent: Display

This Button is used to display record of eligible students for placements.

Common to all Pages

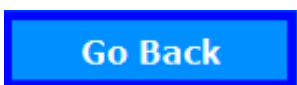


Fig 4.1.13: JButton Go Back

id: back

ActionEvent: go to Home Page

This Button is navigate back to home page from current page in UI.

Chapter 5

Implementation

5.1 Description of Frameworks Used

Java Swing

Java Swing is a GUI Framework that contains a set of classes to provide more powerful and flexible GUI components than AWT. Swing provides the look and feel of modern Java GUI. Swing library is an official Java GUI tool kit released by Sun Microsystems. It is used to create graphical user interface with Java. It defines infrastructure common to most desktop applications, making Swing applications easier to create.

Swing classes are defined in javax.swing package and its sub-packages. Features of JFC:

- Swing GUI components.
- Look and Feel support.
- Java 2D.

Swing Classes

JPanel: JPanel is Swing's version of AWT class Panel and uses the same default layout, FlowLayout. JPanel is descended directly from JComponent.

JFrame: JFrame is Swing's version of Frame and is descended directly from Frame class. The component which is added to the Frame, is referred as its Content.

JWindow: This is Swing's version of Window and has descended directly from Window class. Like Window it uses BorderLayout by default.

JLabel: JLabel has descended from JComponent, and is used to create text labels.

JButton: JButton class provides the functioning of push button. JButton allows an icon, string or both associated with a button.

JTextField: JTextField allow editing of a single line of text.

5.2 Description of Integrated Development Environment

Eclipse

Eclipse is an integrated development environment (IDE) for developing applications using the Java programming language and other programming languages such as C/C++, Python, PERL, Ruby etc.

The Eclipse platform which provides the foundation for the Eclipse IDE is composed of plug-ins and is designed to be extensible using additional plug-ins. Developed using Java, the Eclipse platform can be used to develop rich client applications, integrated development environments and other tools. Eclipse can be used as an IDE for any programming language for which a plug-in is available.

The Java Development Tools (JDT) project provides a plug-in that allows Eclipse to be used as a Java IDE, PyDev is a plugin that allows Eclipse to be used as a Python IDE, C/C++ Development Tools (CDT) is a plug-in that allows Eclipse to be used for developing application using C/C++, the Eclipse Scala plug-in allows Eclipse to be used an IDE to develop Scala applications and PHPEclipse is a plug-in to eclipse that provides complete development tool for PHP.

Among the features of the platform are:

- User interface management (e.g. menus and toolbars)
- User settings management
- Storage management (saving and loading any kind of data)
- Window management
- Wizard framework (supports step-by-step dialogs)
- NetBeans Visual Library
- Integrated development tools

Eclipse software development kit (SDK) is free and open-source software, released under the terms of the Eclipse Public License, although it is incompatible with the GNU General Public License.^[11] It was one of the first IDEs to run under GNU Classpath and it runs without problems under IcedTea.

Chapter 6

Testing

6.1 Component Test

Home Page

Table no.6.1 Home Page Tests

TEST UNIT	TEST CASE	RESULT
Insert/Modify Details	Insert/Modify Details button is pressed	The system invokes the respective function and displays the window to insert details.
Delete Record	Delete Record button is pressed.	The system invokes the respective function and displays the window to delete a record.
Display Record	Display Record button is pressed.	The system invokes the respective function and displays the window to display all records.
Search Record	Search Record button is pressed.	The system invokes the respective function and displays the window to search for a record.
View Changes	View Changes button is pressed.	The system invokes the respective function and displays the window to view changes.
Placement Eligibility	Placement Eligibility button is pressed.	The system invokes the respective function and displays the window to check placement eligibility.

Insert/Modify Details**Table no. 6.2: Insert/Modify Details Module Tests**

TEST UNIT	TEST CASE	RESULT
Insert	Insert button is pressed	The system displays a Message Dialog saying “Record Added!”
Insert	Insert button is pressed	The system displays a Message Dialog saying “A record already exists with usn <usn>” followed by another Message Dialog saying “Record Updated!” if usn already exists.
Go Back	Go Back is pressed.	The system invokes the respective function and displays the Home Page.

Delete Record**Table no. 6.3 Delete Record Module Tests**

TEST UNIT	TEST CASE	RESULT
Delete	Delete button is pressed	The system displays a Message Dialog saying “Record Deleted!”
Delete	Delete button is pressed	The system displays a Message Dialog saying “Invalid USN” if usn does not exist.
Go Back	Go Back is pressed.	The system invokes the respective function and displays the Home Page.

Display Record**Table no. 6.4 Display Record Module Tests**

TEST UNIT	TEST CASE	RESULT
Display	Display button is pressed	The system invokes the respective function and displays all the records.
Go Back	Go Back is pressed.	The system invokes the respective function and displays the Home Page.

Search Record**Table No. 6.5 Search Record Module Tests**

TEST UNIT	TEST CASE	RESULT
Search	Search button is pressed	The system invokes the respective function and displays the search page.
Search by USN	Search by USN button is pressed	The system invokes the respective function and displays the student record.
Search by Branch	Search by Branch button is pressed	The system invokes the respective function and displays the Students in the particular branch.
Search by Company	Search by Company button is pressed	The system invokes the respective function and displays the Students aspiring for particular company.
Go Back	Go Back is pressed.	The system invokes the respective function and displays the Home Page.

View Changes

Table No. 6.1.6 View Changes Module Tests

TEST UNIT	TEST CASE	RESULT
Display	Display button is pressed	The system invokes the respective function and displays all the changes made to all records.
Go Back	Go Back is pressed.	The system invokes the respective function and displays the Home Page.

Placement Eligibility

Table No. 6.1.7 Placement Eligibility Module Tests

TEST UNIT	TEST CASE	RESULT
Display	Display button is pressed	The system invokes the respective function and displays eligibility status of all the records.
Go Back	Go Back is pressed.	The system invokes the respective function and displays the Home Page.

6.2 System Testing

Table No. 6.2.1: Complete System Test

TEST UNIT	TEST CASE	RESULT
Creating a file	Click on Insert button	Creates a new file onto the disk and stores a record
Inserting a record	Click on Insert button	Opens a previously created file from disk and inserts record.
Modifying a record	Click on Insert button	Opens a previously created file from disk and modifies a record.
Deleting a record	Click on Delete button	Opens a previously created file from disk and deletes a record.
Displaying records	Click on Display button	Opens a previously created file from disk and traverses through the file.
Searching a record	Click on Search button	Opens a previously created file from disk and finds a record.

Chapter 7

Interpretation of Result

- Home page of the application which helps navigating to different functionalities.

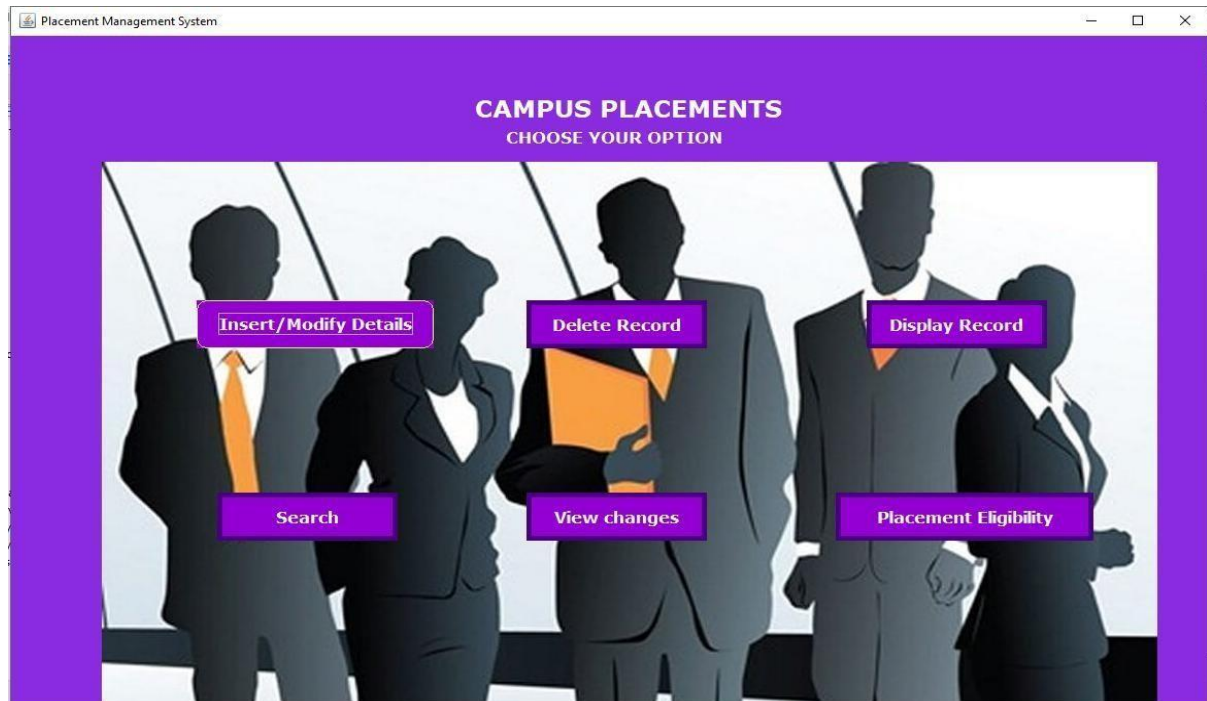


Fig 7.1: Main Panel of Placement Management System

- Insert/Modify page takes all user inputs of student records and checks for duplicates.

Fig 7.2: Insert Page

- Delete record page of the application which takes USN as user input and on the press of delete button the corresponding record is deleted.

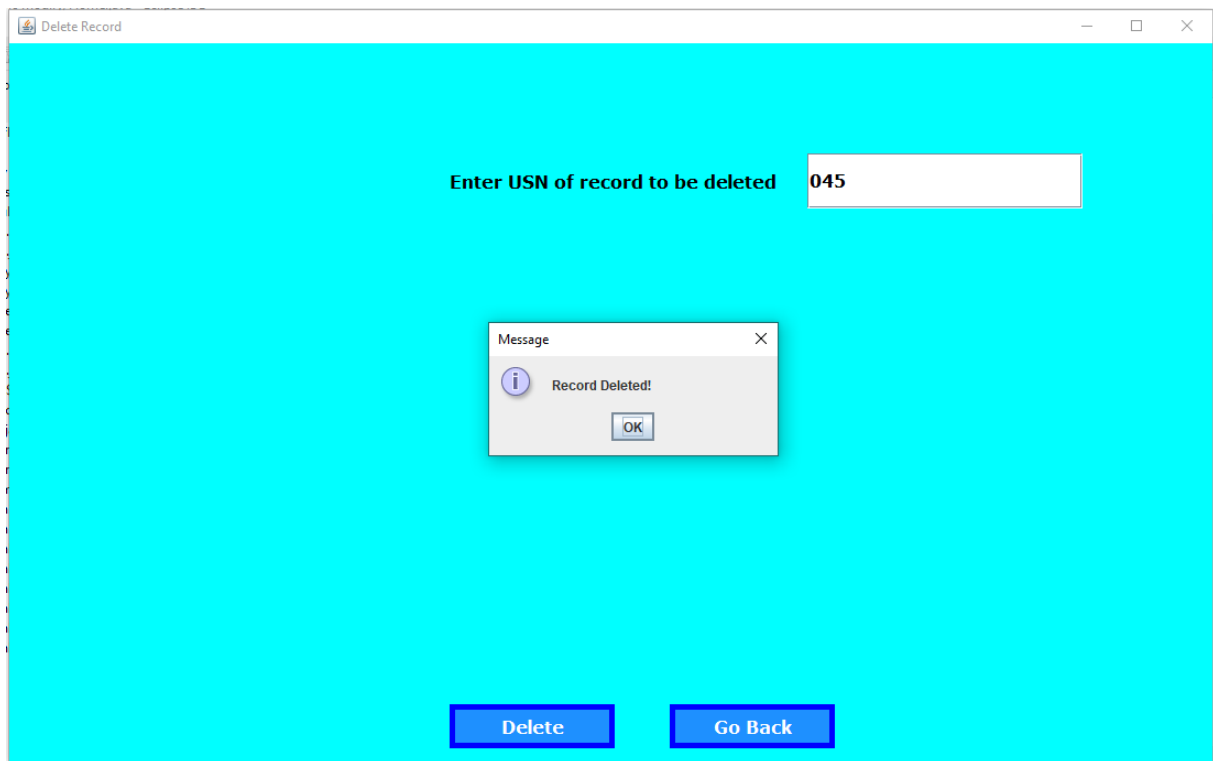


Fig 7.3: Delete Page

- Display record page of the application, when the Display button is pressed all the records in the file are displayed, Go back button navigates to home page.

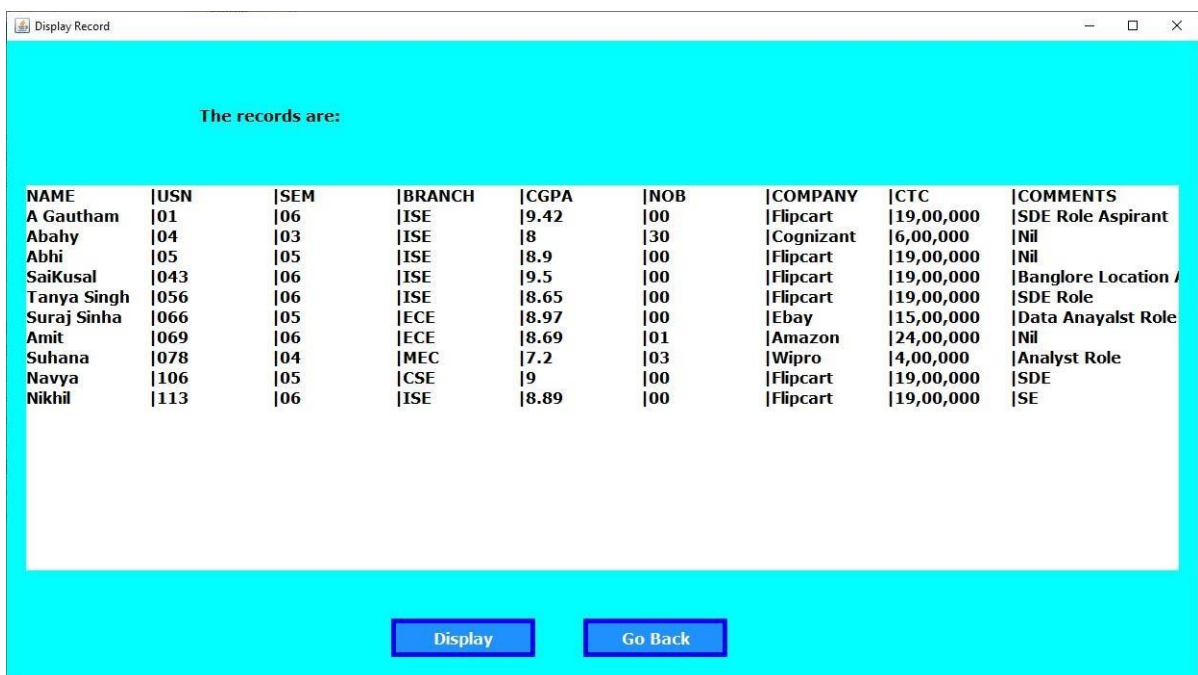


Fig 7.4: Display Page

- Search record page of the application, provides us with different categories of search and navigates us to corresponding search page.

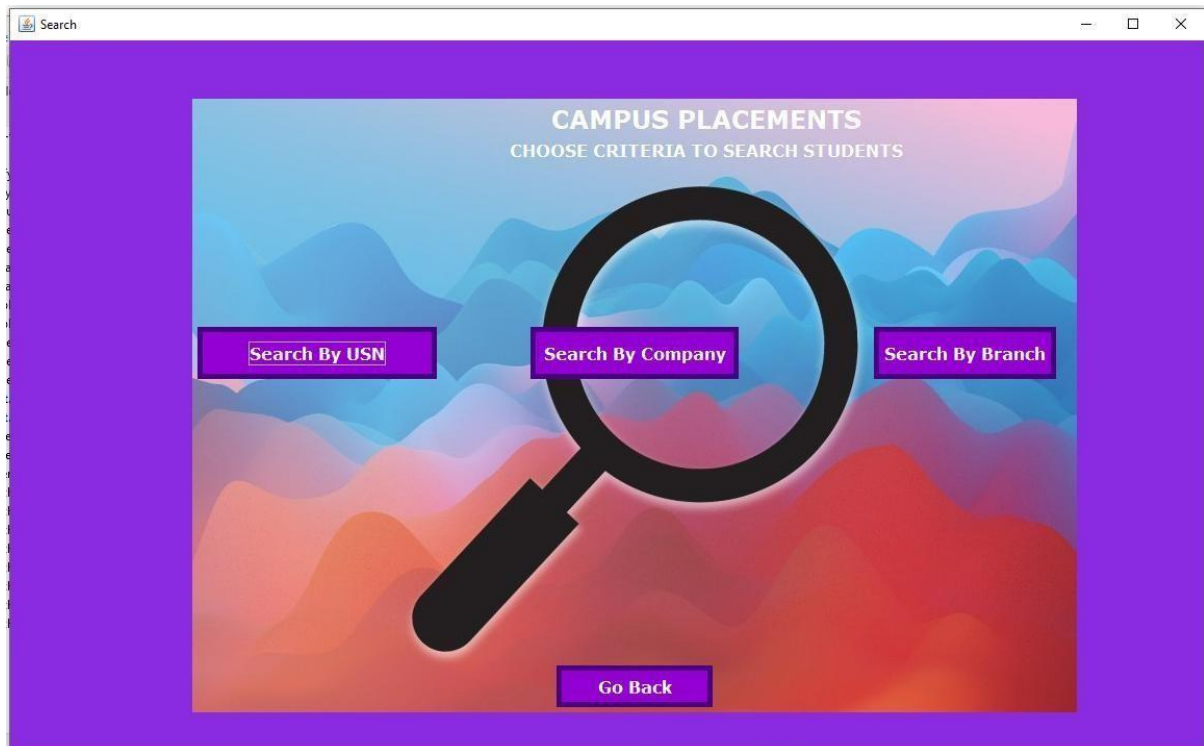


Fig 7.5: Search Page

- Search by USN page of the application, when the USN is entered & button is pressed the record matching with USN and if found in file then it is displayed.

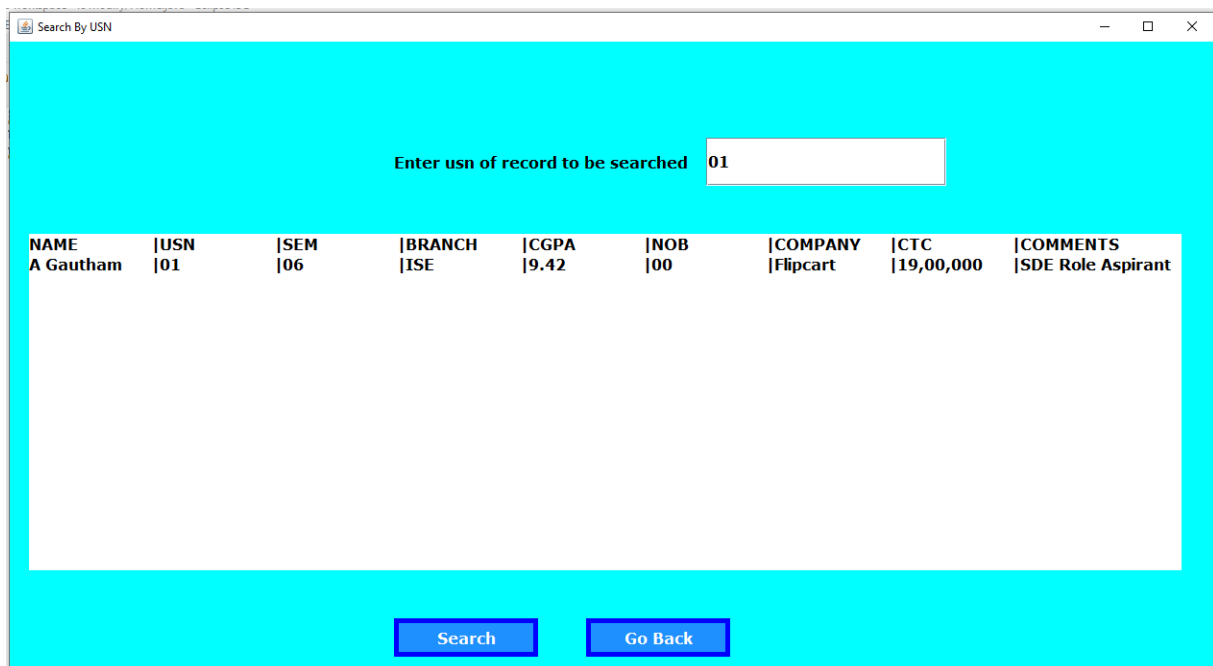


Fig 7.6: Search By USN Page (Match Found)

- Search by Branch page of the application, when the branch is selected from dropdown is entered & button is pressed all the records matching with branch selected, and if found in file then it is displayed.

Search By Branch

Enter Branch of record to be searched

ISE

NAME	USN	SEM	BRANCH	CGPA	NOB	CTC	COMMENTS
A Gautham	01	06	ISE	9.42	00	19,00,000	SDE Role Aspirant
Abahy	04	03	ISE	8	30	000	Nil
Abhi	05	05	ISE	8.9	00	000	Nil
SaiKusal	043	06	ISE	9.5	00	000	Banglore Location /
Tanya Singh	056	06	ISE	8.65	00	19,00,000	SDE Role
Nikhil	113	06	ISE	8.89	00	19,00,000	SE

Search Go Back

Fig 7.7: Search By Branch

- Search by Company page of the application, when the branch is selected from dropdown is entered button is pressed all the records matching with company selected, and if found in file then it is displayed.

Search By Company

Enter Company of record to be searched

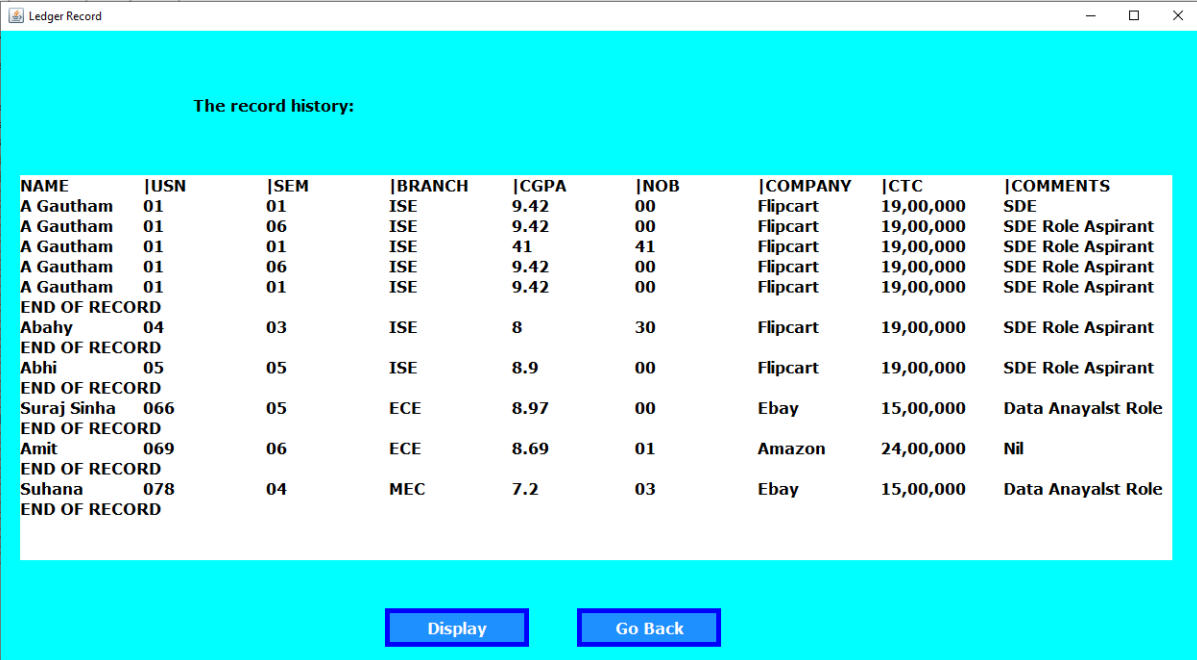
Flipcart

NAME	USN	SEM	BRANCH	CGPA	NOB	COMPANY	CTC	COMMENTS
A Gautham	01	06	ISE	9.42	00	Flipcart	19,00,000	SDE Role Aspirant
Abhi	05	05	ISE	8.9	00	Flipcart	19,00,000	Nil
SaiKusal	043	06	ISE	9.5	00	Flipcart	19,00,000	Banglore Location /
Tanya Singh	056	06	ISE	8.65	00	Flipcart	19,00,000	SDE Role
Navya	106	05	CSE	9	00	Flipcart	19,00,000	SDE
Nikhil	113	06	ISE	8.89	00	Flipcart	19,00,000	SE

Search Go Back

Fig 7.8: Search By Company

- View Changes page acts as a ledger file and displays all the changes done to a particular record and keeps a log of all activities performed.



The record history:

NAME	USN	SEM	BRANCH	CGPA	NOB	COMPANY	CTC	COMMENTS
A Gautham	01	01	ISE	9.42	00	Flipcart	19,00,000	SDE
A Gautham	01	06	ISE	9.42	00	Flipcart	19,00,000	SDE Role Aspirant
A Gautham	01	01	ISE	41	41	Flipcart	19,00,000	SDE Role Aspirant
A Gautham	01	06	ISE	9.42	00	Flipcart	19,00,000	SDE Role Aspirant
A Gautham	01	01	ISE	9.42	00	Flipcart	19,00,000	SDE Role Aspirant
END OF RECORD								
Abahy	04	03	ISE	8	30	Flipcart	19,00,000	SDE Role Aspirant
END OF RECORD								
Abhi	05	05	ISE	8.9	00	Flipcart	19,00,000	SDE Role Aspirant
END OF RECORD								
Suraj Sinha	066	05	ECE	8.97	00	Ebay	15,00,000	Data Anayalst Role
END OF RECORD								
Amit	069	06	ECE	8.69	01	Amazon	24,00,000	Nil
END OF RECORD								
Suhana	078	04	MEC	7.2	03	Ebay	15,00,000	Data Anayalst Role
END OF RECORD								

Display Go Back

Fig 7.9: View Changes Page

- Placement Eligibility page displays records of all students and their eligibility to companies based on the credentials entered while inserting record.



PLACEMENT ELIGIBILITY

PLATINUM PACKAGE: CGPA>=8 AND NO BACKLOGS ALLOWED
 GOLD PACKAGE: CGPA>=7 AND NO BACKLOGS ALLOWED
 SILVER PACKAGE: CGPA>=6

A Gautham : Candidate Eligible For Platinum,Gold, Silver Packages
 Eligible companies: Google (24LPA), Amazon (19LPA), Adobe (11LPA), Flipkart (18LPA), Ebay (6LPA), Infosys (4LPA)

Abahy : Candidate Eligible For only Silver Package
 Eligible companies: Cognizant (6LPA), Wipro (4LPA)

Abhi : Candidate Eligible For Platinum,Gold, Silver Packages
 Eligible companies: Google (24LPA), Amazon (19LPA), Adobe (11LPA), Flipkart (18LPA), Ebay (6LPA), Infosys (4LPA)

SaiKusal : Candidate Eligible For Platinum,Gold, Silver Packages
 Eligible companies: Google (24LPA), Amazon (19LPA), Adobe (11LPA), Flipkart (18LPA), Ebay (6LPA), Infosys (4LPA)

Tanya Singh : Candidate Eligible For Platinum,Gold, Silver Packages
 Eligible companies: Google (24LPA), Amazon (19LPA), Adobe (11LPA), Flipkart (18LPA), Ebay (6LPA), Infosys (4LPA)

Suraj Sinha : Candidate Eligible For Platinum,Gold, Silver Packages
 Eligible companies: Google (24LPA), Amazon (19LPA), Adobe (11LPA), Flipkart (18LPA), Ebay (6LPA), Infosys (4LPA)

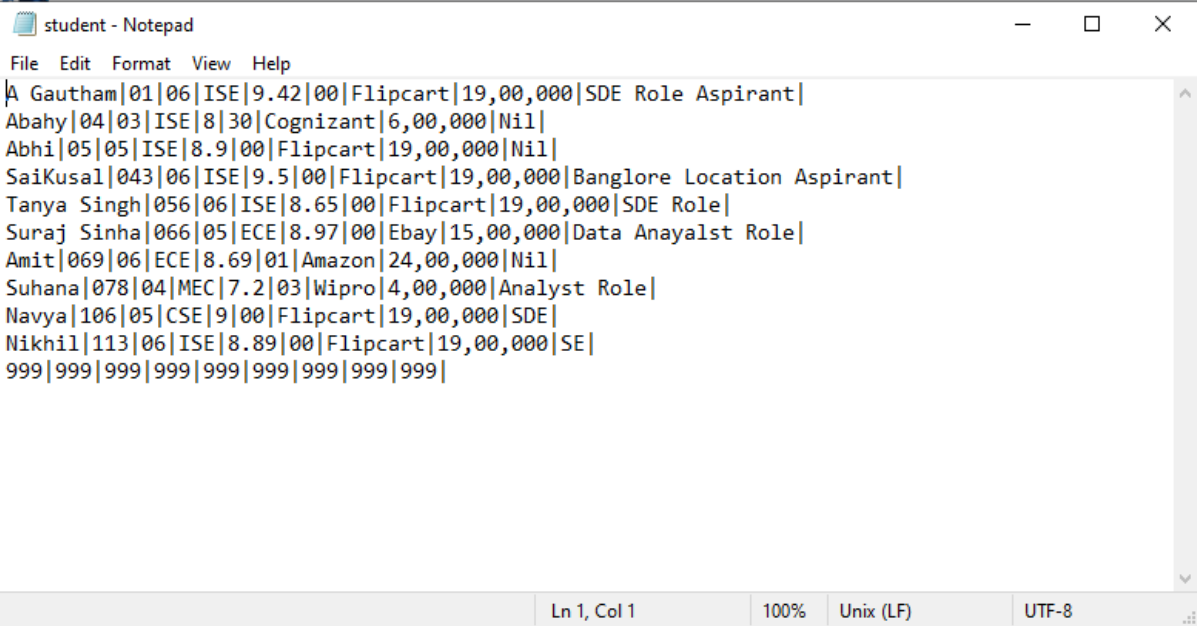
Amit : Candidate Eligible For only Silver Package
 Eligible companies: Cognizant (6LPA), Wipro (4LPA)

Suhana : Candidate Eligible For only Silver Package
 Eligible companies: Cognizant (6LPA), Wipro (4LPA)

Display Go Back

Fig 7.10: Placement Eligibility Page

- Student.txt file stored on disk containing all student record and terminated by default 999 record.



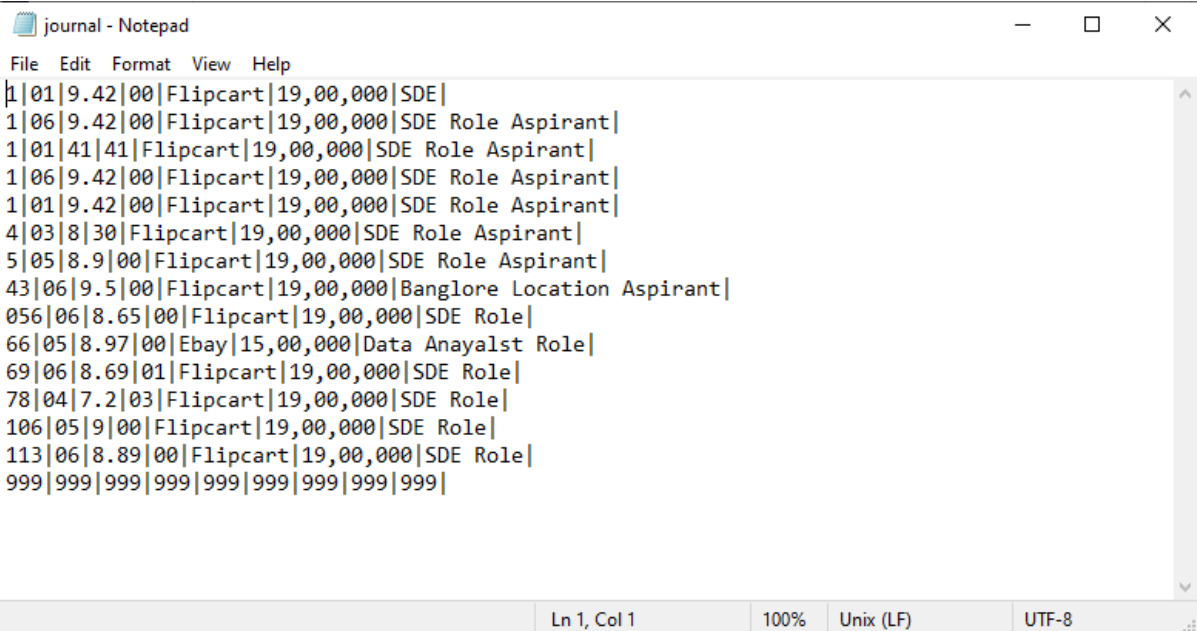
```

student - Notepad
File Edit Format View Help
A Gautham|01|06|ISE|9.42|00|Flipcart|19,00,000|SDE Role Aspirant|
Abahy|04|03|ISE|8|30|Cognizant|6,00,000|Nil|
Abhi|05|05|ISE|8.9|00|Flipcart|19,00,000|Nil|
SaiKusal|043|06|ISE|9.5|00|Flipcart|19,00,000|Banglore Location Aspirant|
Tanya Singh|056|06|ISE|8.65|00|Flipcart|19,00,000|SDE Role|
Suraj Sinha|066|05|ECE|8.97|00|Ebay|15,00,000|Data Anayalst Role|
Amit|069|06|ECE|8.69|01|Amazon|24,00,000|Nil|
Suhana|078|04|MEC|7.2|03|Wipro|4,00,000|Analyst Role|
Navya|106|05|CSE|9|00|Flipcart|19,00,000|SDE|
Nikhil|113|06|ISE|8.89|00|Flipcart|19,00,000|SE|
999|999|999|999|999|999|999|999|
Ln 1, Col 1 100% Unix (LF) UTF-8

```

Fig 7.11: Student Records File

- journal.txt file stored on disk containing all student record and the modifications done on them.



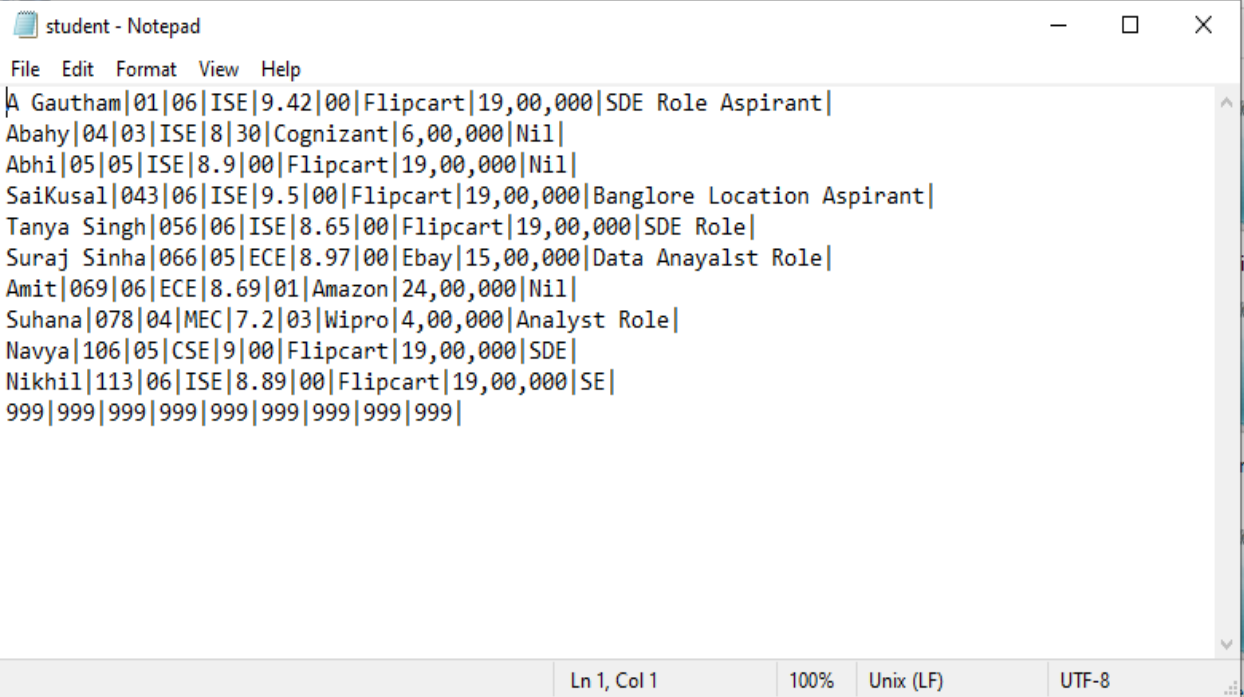
```

journal - Notepad
File Edit Format View Help
1|01|9.42|00|Flipcart|19,00,000|SDE|
1|06|9.42|00|Flipcart|19,00,000|SDE Role Aspirant|
1|01|41|41|Flipcart|19,00,000|SDE Role Aspirant|
1|06|9.42|00|Flipcart|19,00,000|SDE Role Aspirant|
1|01|9.42|00|Flipcart|19,00,000|SDE Role Aspirant|
4|03|8|30|Flipcart|19,00,000|SDE Role Aspirant|
5|05|8.9|00|Flipcart|19,00,000|SDE Role Aspirant|
43|06|9.5|00|Flipcart|19,00,000|Banglore Location Aspirant|
056|06|8.65|00|Flipcart|19,00,000|SDE Role|
66|05|8.97|00|Ebay|15,00,000|Data Anayalst Role|
69|06|8.69|01|Flipcart|19,00,000|SDE Role|
78|04|7.2|03|Flipcart|19,00,000|SDE Role|
106|05|9|00|Flipcart|19,00,000|SDE Role|
113|06|8.89|00|Flipcart|19,00,000|SDE Role|
999|999|999|999|999|999|999|999|
Ln 1, Col 1 100% Unix (LF) UTF-8

```

Fig 7.12: Ledger/View Changes File

- temp.txt file stored on disk containing all student record used for consequential merging and matching process.



```

student - Notepad
File Edit Format View Help
A Gautham|01|06|ISE|9.42|00|Flipcart|19,00,000|SDE Role Aspirant|
Abahy|04|03|ISE|8|30|Cognizant|6,00,000|Nil|
Abhi|05|05|ISE|8.9|00|Flipcart|19,00,000|Nil|
SaiKusal|043|06|ISE|9.5|00|Flipcart|19,00,000|Bangalore Location Aspirant|
Tanya Singh|056|06|ISE|8.65|00|Flipcart|19,00,000|SDE Role|
Suraj Sinha|066|05|ECE|8.97|00|Ebay|15,00,000|Data Anayalst Role|
Amit|069|06|ECE|8.69|01|Amazon|24,00,000|Nil|
Suhana|078|04|MEC|7.2|03|Wipro|4,00,000|Analyst Role|
Navya|106|05|CSE|9|00|Flipcart|19,00,000|SDE|
Nikhil|113|06|ISE|8.89|00|Flipcart|19,00,000|SE|
999|999|999|999|999|999|999|999|999|
Ln 1, Col 1 100% Unix (LF) UTF-8

```

Fig 7.13: Student Temp File.

Chapter 7

CODE

```
import java.awt.*;
import javax.swing.*;
import javax.swing.border.Border;

import java.awt.event.*;
import static javax.swing.JOptionPane.showMessageDialog;
import java.util.*;
import java.io.*;

public class Home extends JFrame implements ActionListener
{
    JLabel welcome, heading;
    private JButton insert, delete, display, search, ledger, eligible,search1;

    Home()
    {
        super("Placement Management System");
        Container con = getContentPane();
        con.setLayout(null);
        Color lightBlue = new Color(138,43,226);
        con.setBackground(lightBlue);
        con.setVisible(true);

        heading = new JLabel("CAMPUS PLACEMENTS");
        heading.setBounds(450,0, 400,150);
        con.add(heading);

        Font fonth = new Font("Verdana", Font.BOLD, 24);
        heading.setFont(fonth);
        heading.setForeground(Color.WHITE);

        welcome = new JLabel("CHOOSE YOUR OPTION");
        welcome.setBounds(480,30, 400,150);
```

```
con.add(welcome);
```

```
Font fontw = new Font("Verdana", Font.BOLD, 16);  
welcome.setFont(fontw);  
welcome.setForeground(Color.WHITE);
```

```
Color blue = new Color(148,0,211);  
insert = new JButton("Insert/Modify Details");  
insert.addActionListener(this);  
insert.setBounds(180,275,230, 50);  
Font font1 = new Font("Verdana", Font.BOLD, 16);  
insert.setFont(font1);  
Color pul = new Color(75,0,130);  
Border bored = BorderFactory.createLineBorder(pul,5);  
insert.setBorder(bored);  
insert.setBorder(new RoundBtn(20));  
insert.setForeground(Color.WHITE);  
insert.setBackground(blue);
```

```
delete = new JButton("Delete Record");  
delete.addActionListener(this);  
delete.setBounds(500,275, 175, 50);  
Font font2 = new Font("Verdana", Font.BOLD, 16);  
delete.setFont(font2);  
delete.setBorder(bored);  
delete.setForeground(Color.WHITE);  
delete.setBackground(blue);
```

```
display = new JButton("Display Record");  
display.addActionListener(this);
```

```
display.setBounds(830,275, 175, 50);
```

```
Font font3 = new Font("Verdana", Font.BOLD, 16);  
display.setFont(font3);  
display.setBorder(bored);  
display.setForeground(Color.WHITE);
```

```
display.setBackground(blue);
```

```
search1 = new JButton("Search");
    search1.addActionListener(this);
    search1.setBounds(200,475, 175, 50);
    Font font7 = new Font("Verdana", Font.BOLD, 16);
search1.setFont(font7);
search1.setBorder(bored);
search1.setForeground(Color.WHITE);
search1.setBackground(blue);
```

```
    ledger = new JButton("View changes");
    ledger.addActionListener(this);
    ledger.setBounds(500,475, 175, 50);
    Font font5 = new Font("Verdana", Font.BOLD, 16);
ledger.setFont(font5);
ledger.setBorder(bored);
ledger.setForeground(Color.WHITE);
ledger.setBackground(blue);
```

```
    eligible = new JButton("Placement Eligibility");
    eligible.addActionListener(this);
    eligible.setBounds(800,475, 250, 50);
    Font font6 = new Font("Verdana", Font.BOLD, 16);
eligible.setFont(font6);
eligible.setBorder(bored);
eligible.setForeground(Color.WHITE);
    eligible.setBackground(blue);
    con.add(insert);
    con.add(delete);
    con.add(display);
    con.add(search1);
    con.add(ledger);
    con.add(eligible);
```

```
JLabel background;
con.setSize(1200,700);
```

```
con.setLayout(null);
```

```
ImageIcon img=new
```

```
ImageIcon("C:\\Users\\MCHOME\\Downloads\\fsminipresent\\fsminipresent\\placement.jpg"
);
```

```
    background = new JLabel("",img,JLabel.CENTER);
```

```
background.setBounds(0,0,1200,850);
```

```
add(background);
```

```
    }
```

```
public void actionPerformed(ActionEvent ae)
```

```
{
```

```
    if(ae.getSource()==insert)
```

```
    {
```

```
        this.dispose();
```

```
        Insert in=new Insert();
```

```
        in.setSize(2300,790);
```

```
        in.setVisible(true);
```

```
    }
```

```
if(ae.getSource()==delete)
```

```
{
```

```
    this.dispose();
```

```
    Delete del=new Delete();
```

```
    del.setSize(2300,790);
```

```
    del.setVisible(true);
```

```
}
```

```
if(ae.getSource()==display)
```

```
{
```

```
    this.dispose();
```

```
    Display dis=new Display();
```

```
    dis.setSize(2300,790);
```

```
    dis.setVisible(true);
```

```
}
```

```
if(ae.getSource()==search1)
```

```
{
```

```

        this.dispose();
        Search1 ser=new Search1();
        ser.setSize(2300,790);
        ser.setVisible(true);
    }

    if(ae.getSource()==ledger)
    {
        this.dispose();
        Ledger led=new Ledger();
        led.setSize(2300,790);
        led.setVisible(true);
    }

    if(ae.getSource()==eligible)
    {
        this.dispose();
        Eligible eli=new Eligible();
        eli.setSize(2300,790);
        eli.setVisible(true);
    }
}

class RoundBtn implements Border
{
    private int r;
    RoundBtn(int r) {
        this.r = r;
    }
    public Insets getBorderInsets(Component c) {
        return new Insets(this.r+1, this.r+1, this.r+2, this.r);
    }
    public boolean isBorderOpaque() {
        return true;
    }
    public void paintBorder(Component c, Graphics g, int x, int y,
        int width, int height) {
        g.drawRoundRect(x, y, width-1, height-1, r, r);

```

```
    }  
}  
  
public static void main(String args[])  
{  
    Home h=new Home();  
        h.setSize(2300,790);  
        h.setVisible(true);  
    }  
}
```

Conclusion

Once this project is completed it offers users the following functionalities:

Allows users to easily create new student records and also allows them to modify it whenever there is a change in CGPA, number of backs, etc.

It allows the users to delete records if needed.

All records with the latest details can be displayed whenever the user needs it.

Users can search for a particular record as well which displays the requested record with the latest details and offers categories of search like search by USN, search by branch and search by company.

Using General Ledger Problem, users are provided with a feature where they can see the history of all changes made to all records.

Finally, it provides information about the eligibility criteria for various packages and displays the list of all the company and packages each student is eligible for.

Overall, it provides the necessary features for managing a placement database in the form of files. This application is a Cross Platform and Architecture independent and can run efficiently on Windows, Mac or Linux as long as JAVA installation files are present on the system

References

- [1] Michael J. Folk, Bill Zoellick, Greg Riccardi: File Structures-An Object Oriented Approach with C++, 3rd Edition, Pearson Education, 1998.
- [2] Herbert Schildt: JAVA the Complete Reference, 7th/9th Edition, Tata McGraw Hill, 2007.
- [3] Jim Keogh: J2EE-The Complete Reference, McGraw Hill, 2007.
- [4] StackOverflow: www.stackoverflow.com
- [5] Codeproject: www.codeproject.com
- [6] Javapoint: www.javapoint.com

ABSTRACT

Placement Management System is built using Java Swing Library, it is a fast-flexible file computer system that helps manage the information related to college student placement database effectively. They manage the data related to all student academic details concerned with campus placements for the campus placement drive which is conducted annually.

Placement Management system is a software application built for campus placement and training department for accessing student details and making further placement process easier.

Placement Management System offers the following functionalities:

- **Adding and Modifying new Student Records:** New student records can be added and modified.
- **Deleting Existing Records:** Existing student records can be deleted.
- **Search:** Allows users to search existing student records by various categories like search student details by USN, Branch, Company.
- **Display:** All records in the system can be displayed.
- **Ledger:** Using General Ledger Problem, all modifications made are displayed
- **Eligibility:** Placement eligibility of all students can be viewed where each student details are displayed according to their eligibility criteria and which company they are eligible to.

Literature Survey

- This gives the introduction of the placement management system and all the features that system contains, problems statement and the limitations of the system.
- This specifies the functional and non-functional requirements of the Placement Management System which includes requirements like performance, reliability, usability, integrity.
- This depicts about all System/Requirement Analysis and the functions that user can perform in Placement Management System.
- This gives the entire information of UI details that user can view and also shows about the functionalities that user can perform.
- This gives the entire information of how the user can implement Placement Management System and frameworks and IDE required to implement the project.
- This gives the results of each component test and System testing after each component is integrated.
- This gives the results of all the functionalities that user can perform and depicts the screenshots of each function.