A

Project Report

On

"Fees Management System"

Submitted for the partial fulfillment for the award of the

Diploma

In

Computer Science and Engineering

Submitted By:

Niten Kumawat

Sarwan Ram

Jitendra Gehlot

Guide:

Er. Naveen Sankhla

(Lecturer, CSE Department)

Department of Computer Science and Engineering

GOVERNMENT POLYTECHNIC COLLEGE, JODHPUR

BOARD OF TECHNICAL EDUCATION, JODHPUR(RAJASTHAN)

(Session 2020-21)

CANDIDATE'S DECLARATION

I hereby declare that the work, which is being presented in Project, entitled **"Fees management system"** in partial fulfillment for the award of Degree of "Diploma in Computer science" and submitted to the Department of Computer science Engineering, Government Polytechnic College, Jodhpur is a record of my own developed under the guidance of Shri Naveen Sankhla, lecturer, Department of Computer Science and Engineering of Government Polytechnic College Jodhpur.

I have not submitted the matter presented in this report anywhere for the award of any other work.

1.Niten Kumawat

Enroll.no:CS2019L002/009

2.Sarwan Ram

Enroll.no:CS2019L005/009

3.Jitendra Gehlot

Enroll.no:CS20180010/009

ACKNOWLEDGMENT

The project report on **FEES MANAGEMENT SYSTEM** is an outcome of guidance, moral support and devotion bestowed on me through my work. For this let me acknowledge and express our performance sense of gratitude and thanks to everybody who have been source of inspiration during the project preparation.

First and foremost, I offer my sincere phrases of thanks with innate humility to **Dr. AJAY MATHUR (H.O.D)** Computer Science, Govt. Polytechnic College, Jodhpur and my project guide for providing help whenever needed.

The consistence guidance and support provided by **Er. Naveen Sankhla lecturer,** Govt. Polytechnic College, Jodhpur. So, we are very thankful to acknowledged for key role played by him in providing me with his precious ideas, suggestion and help that enabled in shaping project work.

If I can say in words, I must at the outset tender my intimate and receipt of affectionate care to Govt. Polytechnic College, Jodhpur for providing such stimulating atmosphere & work environment.

I acknowledge here our debt to those who contributed significantly to one or more steps. I take full responsibility for any remaining sins of omission and commission.

Niten Kumawat

Sarwan Ram

Jitendra Gehlot

INDEX

List of Figures

ABSTRACT

The "**Fees Management System"** is a desktop application with specialized majorities in the field of Fees management. Allow to admin to includes the information of Courses and Fees details. Allow customers to check, modify the information, and print data, fand verify the information.

It is a management system aiming to give the safe storing of Fees details and courses. The most useful and reliable functions which can't be found in another system. The project is built in Java platform with the help of IDE NetBeans and JDBC Derby database.

In order to make the best product possible, insight in languages like. Java, SQL for database, and another one is inevitable.

However, all of us final year students with limited knowledge and to be adept in all those languages is never an easy deal. Since then, although we do try our best, there are still some small malfunctions in our product.

INTRODUCTION

In the existing system, most of the records are maintained on paper. It becomes very inconvenient to modify the data. In the existing system, here is a possibility that the same data in different registers may have different values which means the entries of the same data do not match. This inconsistent state does not supply the concrete information which poses a problem in the case information related to particular search record.

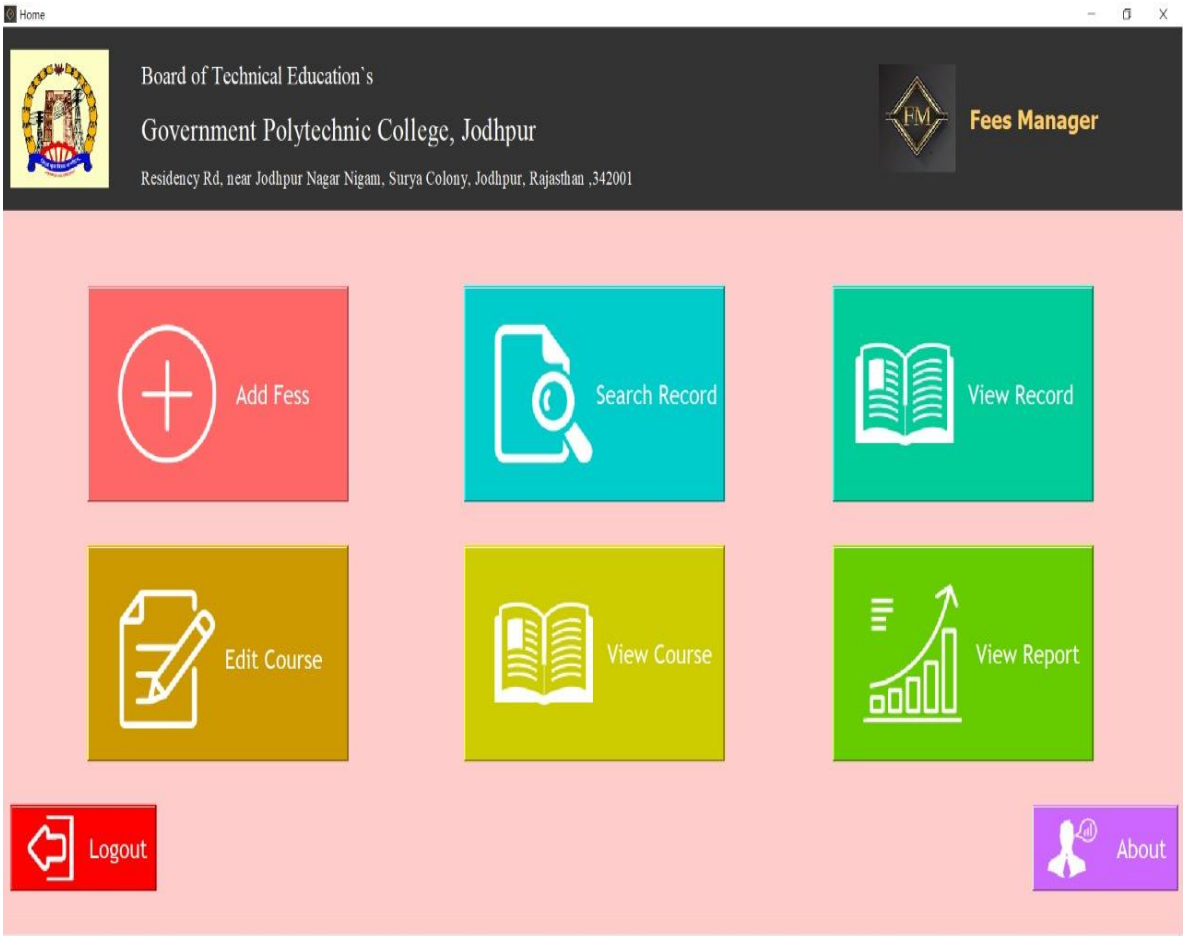


Figure 1.1: Home page of Fees manager.

Our project is very useful. User is no longer required to check his register in search of records, as now it can be searched over the software by choosing some options. The user need not to type in most of the information. He/she is just required to enter the desired options. On the whole it liberates the user from keeping lengthy manual records. In a nutshell, it abates the work load of an organization.

In today's world, no one likes to perform calculations on calculator or manually when computer is there. Everyone wants his/her work to be done by computer automatically and displaying the result for further manipulations. So, this project is about providing convenience regarding fee management system.

The program can run as following flow diagram: -

When we run the program then first of all login page is display to screen, if admin is new then he must be signup first and after that he can login.

After login home page screen display .and admin can go with choice.

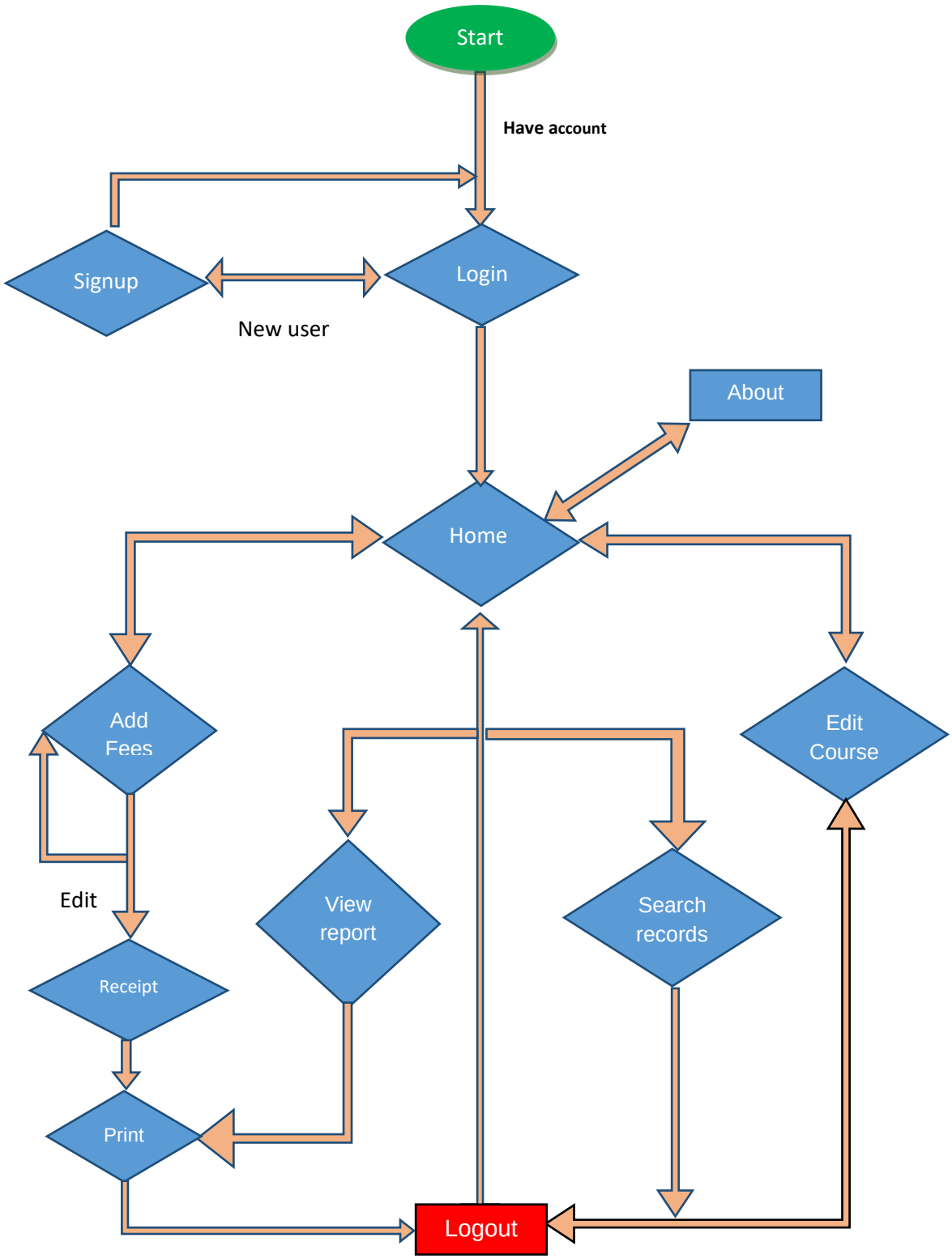


Figure 1.2: Flow diagram of Fees manager.

2

SOFWARE DEVLOPMENT METHODOLOGY

The establishment and use of sound engineering principles in order to obtain economically developed software that is reliable and works efficiently on real machines is called *software engineering.*

Software engineering is the discipline whose aim is:

1. Production of quality software.
2. Software that is delivered on time.
3. Cost within the budget.
4. Satisfies all requirements.

Software process is the way in which we produce the software. Apart from hiring smart, Knowledge able engineers and buying the latest development tools, effective software development process is also needed, so that engineers can systematically use the best technical and managerial practices to successfully complete their projects.

A **software life cycle** is the series of identifiable stages that a software product undergoes during its lifetime. A software lifecycle model is a descriptive and diagrammatic representation of the software life cycle. A life cycle model represents all the activities required to make a software product transit through its lifecycle phases. It also captures the order in which these activities are to be taken.

2.*1 LIFE CYCLE MODELS*: There are various life cycle models to improve the software processes.

1. WATERFALL MODEL
2. PROTOTYPE MODEL
3. ITERATIVE ENHANCEMENT MODEL
4. EVOLUTIONARY MODEL
5. SPIRAL MODEL

This model contains 6 phases:

1. **Feasibility study:** The feasibility study activity involves the analysis of the problem and collection of the relevant information relating to the product. The main aim of the feasibility study is to determine whether it would be financially and technically feasible to develop the product.

2. **Requirement analysis and specification:** The goal of this phase is to understand the exact requirements of the customer and to document them properly. (SRS)

3. **Design:** The goal of this phase is to transform the requirement specification into a structure that is suitable for implementation in some programming language.

4. **Implementation and unit testing:** During this phase the design is implemented. Initially small modules are tested in isolation from rest of the software product.

5. **Integration and system testing:** In this all the modules are integrated and then tested altogether.

6. **Operation and maintenance**: Release of software inaugurates the operation and life cycle phase of the operation. The phases always occur in this order and do not overlap.

2.2 EVOLUTIONARY MODEL

In the project, **EVOLUTIONARY MODEL** is followed.

Evolutionary model is a combination of Iterative and Incremental model of software development life cycle. Delivering your system in a big bang release, delivering it in incremental process over time is the action done in this model. Some initial requirements and architecture envisioning need to be done.
It is better for software products that have their feature sets redefined during development because of user feedback and other factors. The Evolutionary development model divides the development cycle into smaller, incremental waterfall models in which users are able to get access to the product at the end of each cycle.

Feedback is provided by the users on the product for the planning stage of the next cycle and the development team responds, often by changing the product, plan or process. Therefore, the software product evolves with time. All the models have the disadvantage that the duration of time from start of the project to the delivery time of a solution is very high. Evolutionary model solves this problem in a different approach.

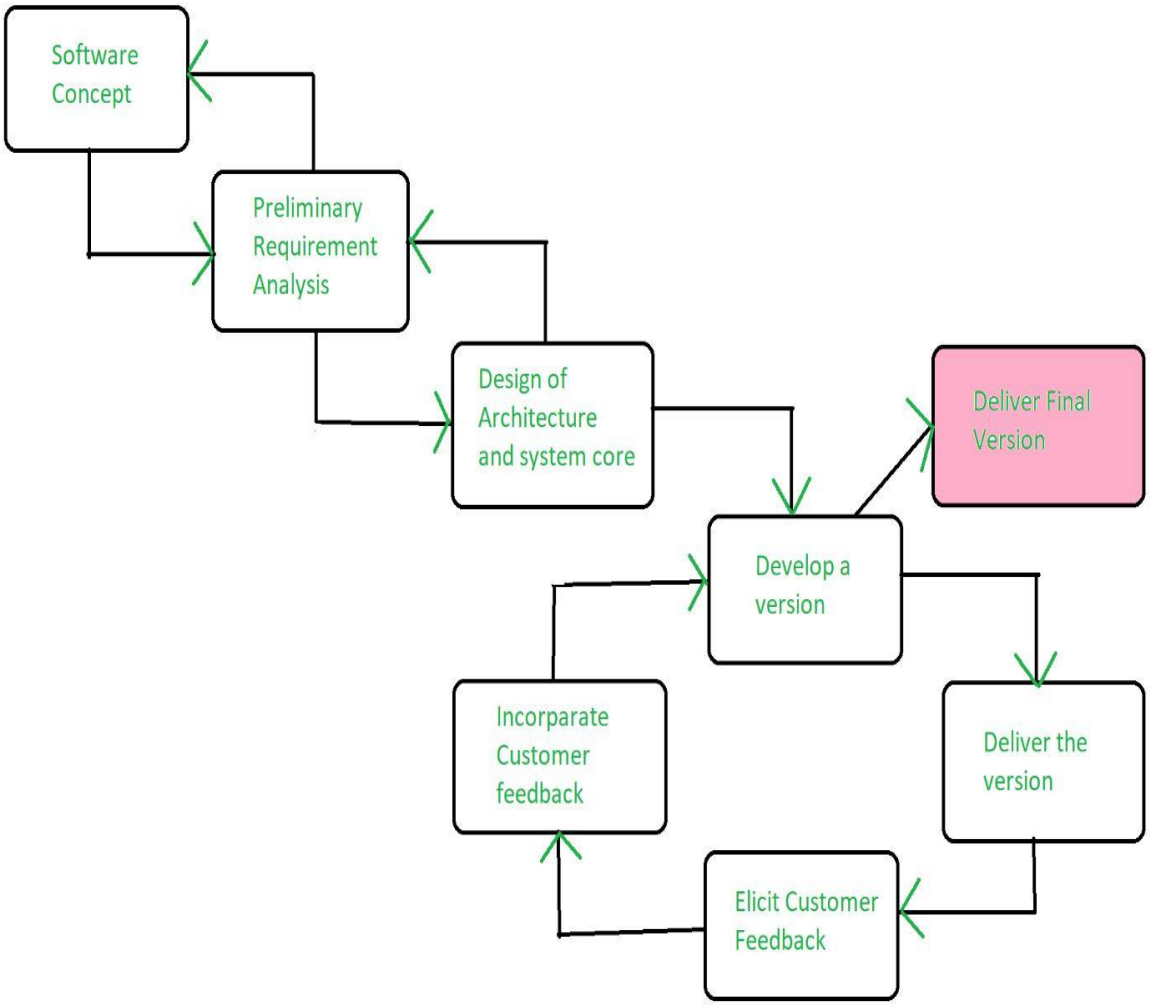


Figure 2.1: Evolutionary model approaches.

3

SYSTEM ANALYSIS

We will discuss and analysis about the developing process of Fees Management System including software requirement specification (SRS) and comparison between existing and proposed system. The functional and non-functional requirements are included in SRS part to provide complete description and overview of system requirement before the developing process is carried out. Besides that, existing vs proposed provides a view of how the proposed system will be more efficient than the existing one.

3.1 SOFTWARE REQUIREMENT SPECIFICATION (SRS)

3.1.1 INTRODUCTION:

The following subsections of the SRS document provide an overview of the entire SRS.

i) **Purpose:** The purpose of the project is to deposit in to reduce paperwork, make storage of information more efficient and secure, have a user-friendly interface, operate it easily and with minimum experience and to save time and energy of the admin and students.

ii) **Scope:** Specifically designed for an individual College. Inserting new student records are not possible. It is based on the Desktop application. Not including fees other than academics like bus fees and etc. There are only limited number of modules for fee management.

iii) **Benefits:** The Fee Management System is a desktop system aimed at maintaining student records and their fee details. It also generates records like i.e., Fees paid, dues, and etc. The system requires small amount of time to generate reports needed to manage the fees of the students.

iv) Abbreviations:

(a)Java.

(b)NetBeans 12.3.

v) Reference:

(a)IEEE Recommended Practice for Software Requirements Specification- IEEE Std 830-1993.

vi) Overview: The rest of this SRS document describes the various system requirements, interfaces, features and functionalities in detail.

3.2 Features

1. Upload thousands of numbers of student Fees details in system.
2. Download your Fees manager to Admin PC.
3. Automatic generate fees deposit receipts and print.
4. Fees manager can include the same person's name by roll no.
5. Fees manager is providing Add new courses for collecting fees.
6. In the fees manager admin can see report of Fees deposit And can print data on papers.
7. In the fees manager admin can search data by specific name and detail.
8. In the fees manager admin can view course details, Records of fees deposit and other detail.

3.3 SOFTWARE AND HARDWARE REQUIREMENTS

This section describes the software and hardware requirements of the system.

3.3.1 SOFTWARE REQUIREMENTS

Operating system- Windows 10 is used as the operating system as it is stable and supports more features and is more user friendly. Development tools and Programming language- java, NetBeans is used to write the whole code and develop graphics.

3.3.2 HARDWARE REQUIREMENTS

Amd Ryzen 5 ,3rd generation is used as a processor because it is fast than other processors and provide reliable and stable and we can run our pc for long time. By using this processor, we can keep on developing our project without any worries.

RAM 8GB is used as it will provide fast reading and writing capabilities and will in turn support in processing.

Nvidia 1650 4gb HD Graphics is used as a graphics design. It shown better graphics quality in High Definition.

3.4 SOFTWARE TOOLS USED

NetBeans was used for developed to the whole project.

NetBeans IDE is a free and open-source integrated development environment for **application** development on Windows, Mac, Linux, and Solaris operating systems. The IDE simplifies the development of web, enterprise, desktop, and mobile **applications** that **use** the Java and HTML5 platforms.

NetBeans is coded in Java and runs on most operating systems with a Java Virtual Machine (JVM), including Solaris, Mac OS, and Linux.

NetBeans manages the following platform features and components:

- User settings
- Windows (placement, appearance, etc.)
- NetBeans Visual Library
- Storage
- Integrated development tools
- Framework wizard

NetBeans uses components, also known as modules, to enable software development. NetBeans dynamically installs modules and allows users to download updated features and digitally authenticated upgrades.

NetBeans IDE modules include NetBeans Profiler, a Graphical User Interface (GUI) design tool, and NetBeans JavaScript Editor.

NetBeans framework reusability simplifies Java Swing desktop application development, which provides platform extension capabilities to third-party developers.

4.1 What is Java?

Java is a **programming language** and a **platform**. Java is a high level, robust, object-oriented and secure programming language.

Java was developed by *Sun Microsystems* (which is now the subsidiary of Oracle) in the year 1995. *James Gosling* is known as the father of Java. Before Java, its name was *Oak*. Since Oak was already a registered company, so James Gosling and his team changed the Oak name to Java.

Platform: Any hardware or software environment in which a program runs, is known as a platform. Since Java has a runtime environment (JRE) and API, it is called a platform.

4.1.1 Types of java Applications

There are mainly 4 types of applications that can be created using Java programming:

1 Standalone Application

Standalone applications are also known as desktop applications or window-based applications. These are traditional software that we need

to install on every machine. Examples of standalone application are Media player, antivirus, etc. AWT and Swing are used in Java for creating standalone applications.

2 Web Application

An application that runs on the server side and creates a dynamic page is called a web application.
Currently, Servlet, JSP, Struts, Spring, Hibernate, JSF,etc.
Technologies are used for creating web applications in Java.

3 Enterprise Application

An application that is distributed in nature, such as banking applications, etc. is called enterprise application. It has advantages of the high-level security, load balancing, and clustering. In Java, EJB is used for creating enterprise applications.

4 Mobile Application

An application which is created for mobile devices is called a mobile application. Currently, Android and Java ME are used for creating mobile applications.

4.1.2 Java Platform/Editions

There are 4 platforms or editions of Java:

1 Java SE (Java Standard Edition)

It is a Java programming platform. It includes Java programming APIs such as java.lang, java.io, java.net, java.util, java.sql, java.math etc. It includes core topics like OOPs, String, Regex, Exception, Inner classes, Multithreading, I/O Stream, Networking, AWT, Swing, Reflection, Collection, etc.

2 Java EE (Java Enterprise Edition)

It is an enterprise platform which is mainly used to develop web and enterprise applications. It is built on the top of the Java SE platform. It includes topics like Servlet, JSP, Web Services, EJB, JPA, etc.

3 Java ME (Java Micro Edition)

It is a micro platform which is mainly used to develop mobile applications. It is used to develop rich internet applications. It uses a light-weight user interface API.

4.2 Coding Standards for Components: It is recommended to write components name by its purpose. This approach improves the readability and maintainability of code.

4.3 Coding Standards for Classes: Usually class name should be noun starting with uppercase letter. If it contains multiple word than every inner word should start with uppercase.

Eg: String, StringBuffer, Dog.

4.4 Coding Standards for Interface: Usually interface name should be adjective starting with uppercase letter. If it contains multiple word than every inner word should start with uppercase.

Eg: Runnable, Serializable, Comparable.

4.5 Coding Standards for Methods: Usually method name should either be verb or verb noun combination starting with lower letter. If it contains multiple word than every inner word should start with uppercase.

Eg: print(), sleep(), setSalary().

4.6 Coding Standards for Variables: Usually variable name should be noun starting with lowercase letter. If it contains multiple word than every inner word should start with uppercase.

Eg: name, age. mobileNumber.

4.7 Coding Standards for Constants: Usually constant name should be noun. It should contain only uppercase If it contains multiple word than words are separated with (_) underscore symbol. Usually, we declare constants with public static and final modifiers.

Java Bean Coding Standards: A Java Bean is a simple java class with private properties and public getter and setter methods.

4.8 Getter Methods: -

1. It should be public method.
2. Method name should be prefixed with “get”.
3. It should not take any argument.

4.9 Setter Methods: -

1. It should be public method.
2. Return Type should be void.
3. Method name should be prefixed with “set”.
4. It should take some argument.

5.1 What is Apache Derby?

Apache Derby is a **R**elational **D**atabase **M**anagement **S**ystem which is fully based on (written/implemented in) Java programming language. It is an open-source database developed by Apache Software Foundation.

Oracle released the equivalent of Apache Derby with the name Java DB.

5.1.1Features of Apache Derby

Following are the notable features of Derby database −

- **Platform independent** − Derby uses on-disc database format where the databases in it are stored in a file in the disc within the directory with the same name as the database.
- **No modifying data** − Because of this, you can move derby databases to other machines without modifying the data.
- **Transactional support** − Derby provides complete support for transactions ensuring data integrity.
- **Including databases** − You can include pre-build/existing databases into your current derby applications.
- **Less space** − Derby database has a small footprint, i.e., it occupies less space and it is easy to use and deploy it.
- **Embed with Java Application** − Derby provides an embedded database engine which can be embedded in to Java applications and it will be run in the same JVM as the application. Simply loading the driver starts the database and it stops with the applications.

In the following Figure, view of Fees management system`s database details in NetBeans: -

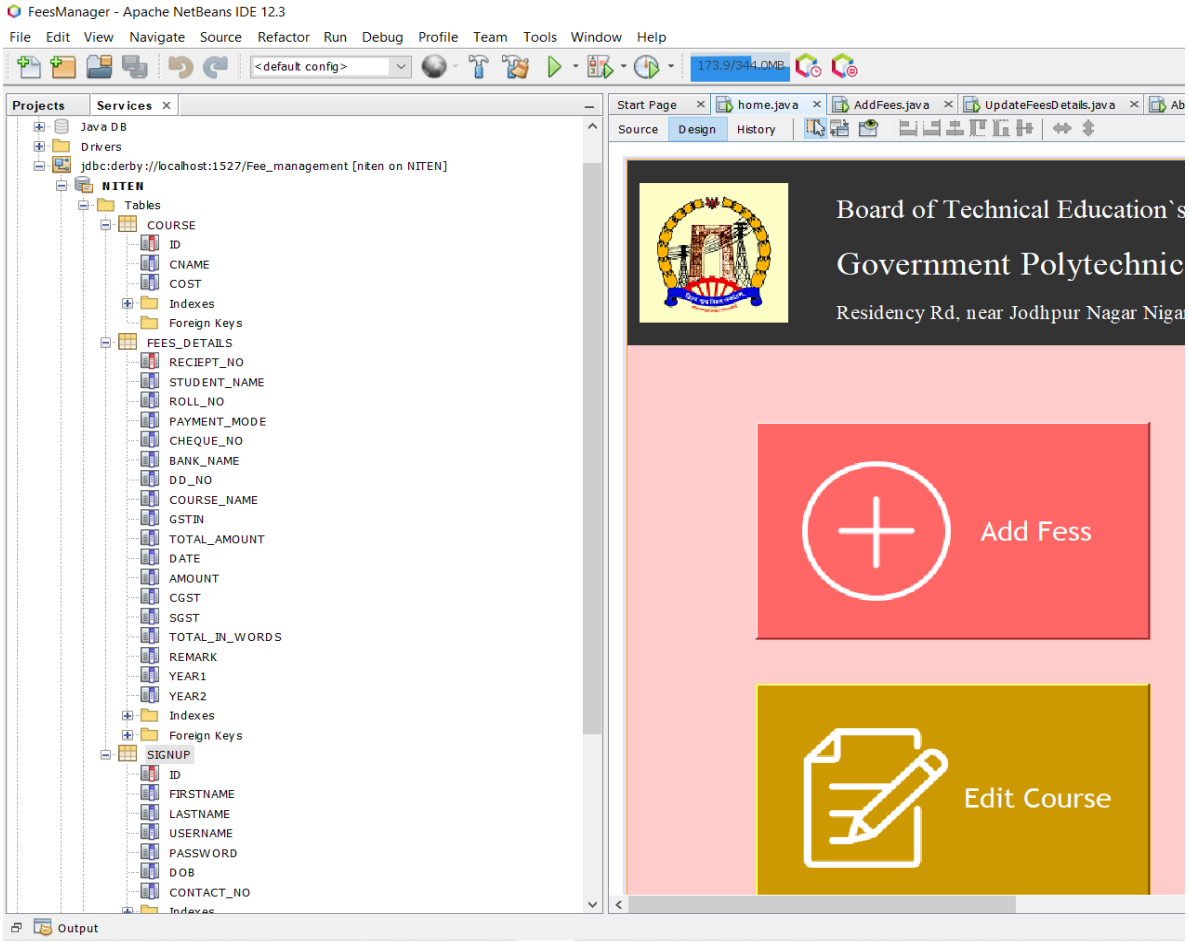


Figure 5.1: Fee_management database files in NetBeans.

5.2 Limitations of Apache Derby

Following are the limitations of Apache Derby −

- Derby does not support indexes for datatypes such as BLOB and LONGVARCHAR.
- If Derby does not have enough disc space, it will shut down immediately.

5.3 DATA FLOW DIAGRAM

A DFD also known as 'bubble chart', has the purpose of clarifying system requirements and identifying major transformations. It shows the flow of data through a system. It is a graphical tool because it presents a picture. The DFD may be partitioned into levels that represent increasing information flow and functional detail. Four simple notations are used to complete a DFD. These notations are given below: -

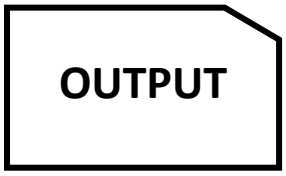
DATA FLOW: - The data flow is used to describe the movement of information from one part of the system to another part. Flows represent data in motion. It is a pipe line through which information flows. Data flow is represented by an arrow.

PROCESS: - A circle or bubble represents a process that transforms incoming data to outgoing data. Process shows a part of the system that transform inputs to outputs.

EXTERNAL ENTITY: - A square defines a source or destination of system data. External entities represent any entity that supplies or receive information from the system but is not a part of the system.

DATA STORE: - The data store represents a logical file. A logical file can represent either a data store symbol which can represent either a data structure or a physical file on disk. The data store is used to collect data at rest or a temporary repository of data. It is represented by open rectangle.

OUTPUT: -The output symbol is used when a hard copy is produced and the user of the copies cannot be clearly specified or there are several users of the output.



ER-DIAGRAM

An **Entity–relationship model (ER model)** describes the structure of a database with the help of a diagram, which is known as **Entity Relationship Diagram (ER Diagram)**. An ER model is a design or blueprint of a database that can later be implemented as a database. The main components of E-R model are: entity set and relationship set.

An ER diagram shows the relationship among entity sets. An entity set is a group of similar entities and these entities can have attributes. In terms of DBMS, an entity is a table or attribute of a table in database, so by showing relationship among tables and their attributes, ER diagram shows the complete logical structure of a database. Let's have a look at a simple ER diagram of project to understand this concept.

Rectangle: Represents Entity sets.
Ellipses: Attributes.
Diamonds: Relationship Set.
Lines: They link attributes to Entity Sets and Entity sets to Relationship Set.
Double Ellipses: Multivalued Attributes.
Dashed Ellipses: Derived Attributes.
Double Rectangles: Weak Entity Sets.
Double Lines: Total participation of an entity in a relationship set.

1 Login Form

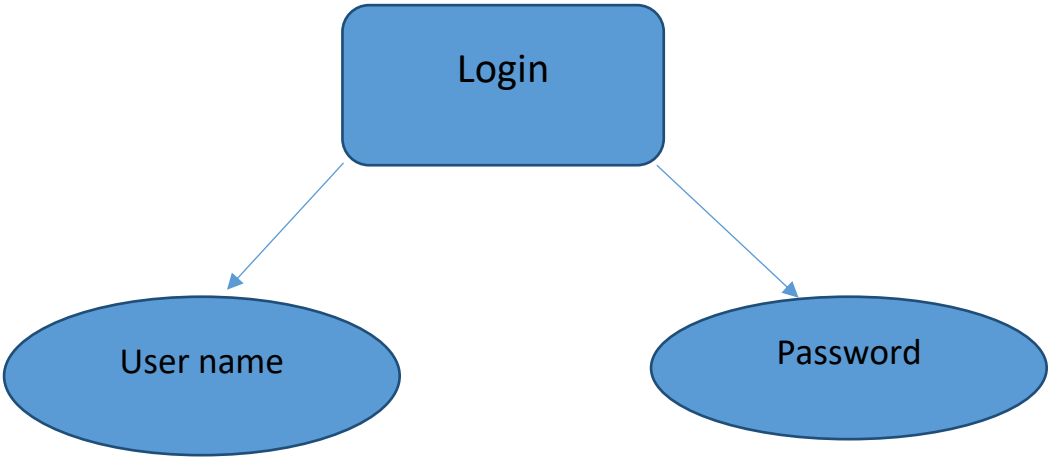


Figure 6.1: ER-diagram of Login.

2 Signup Form

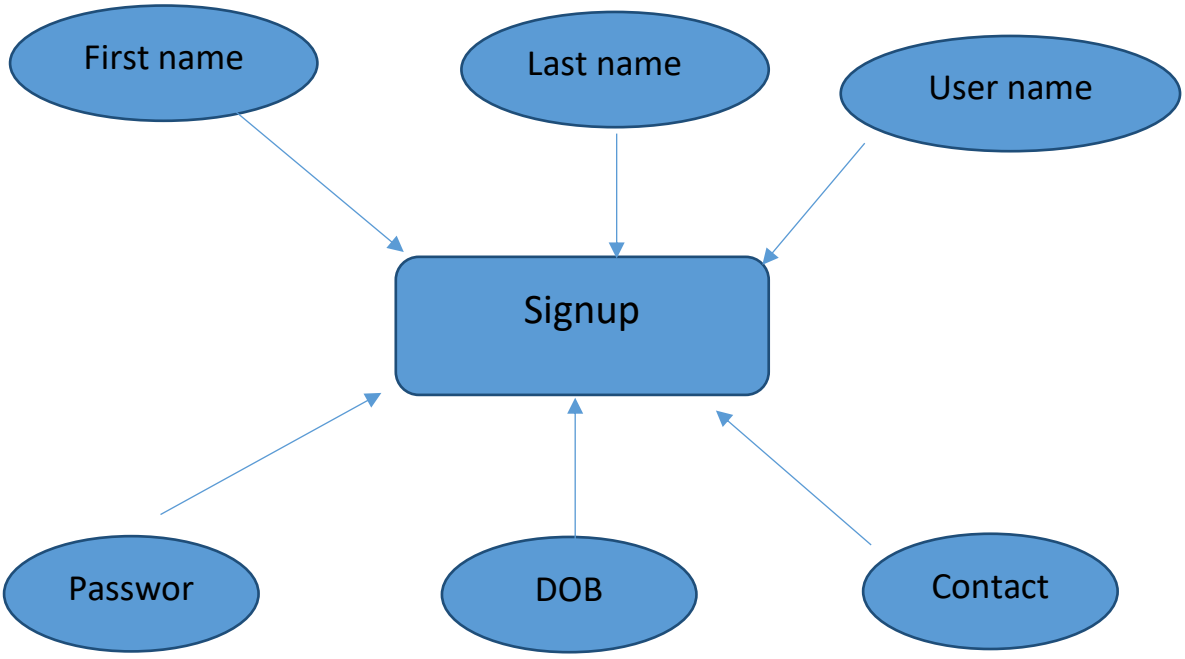


Figure 6.2: ER-diagram of Signup.

3 Add Fees Form

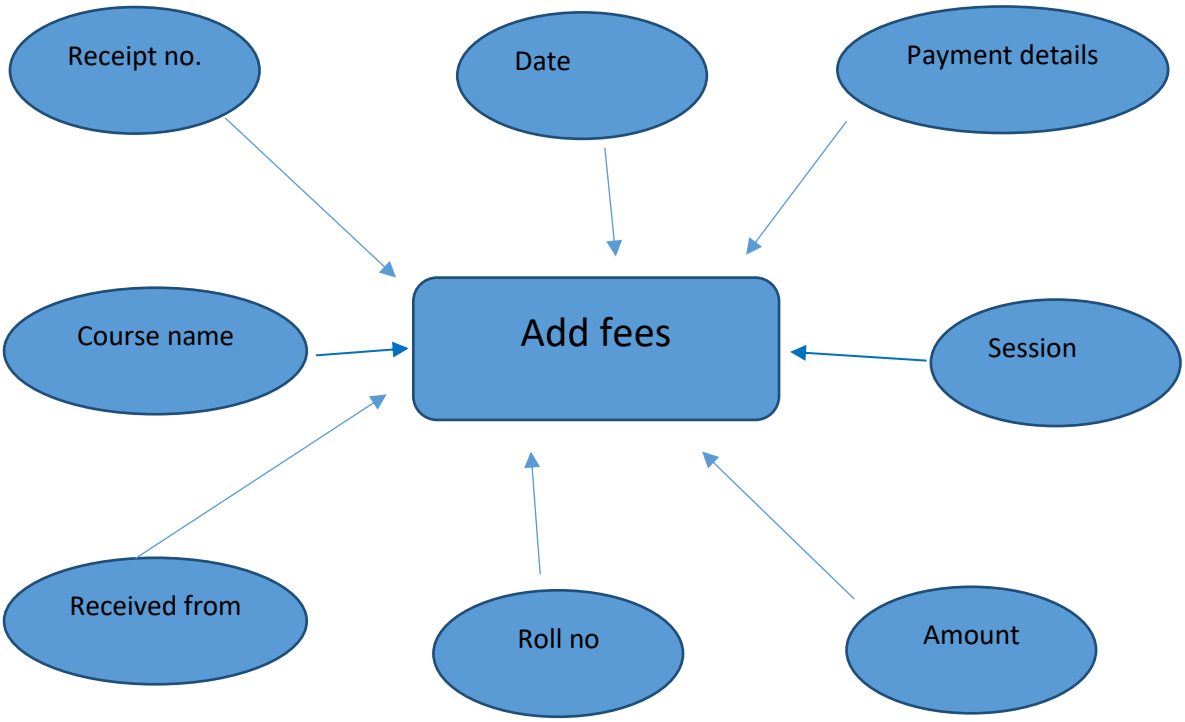


Figure 6.3: ER-diagram for Add fees.

4 Edit Course Form

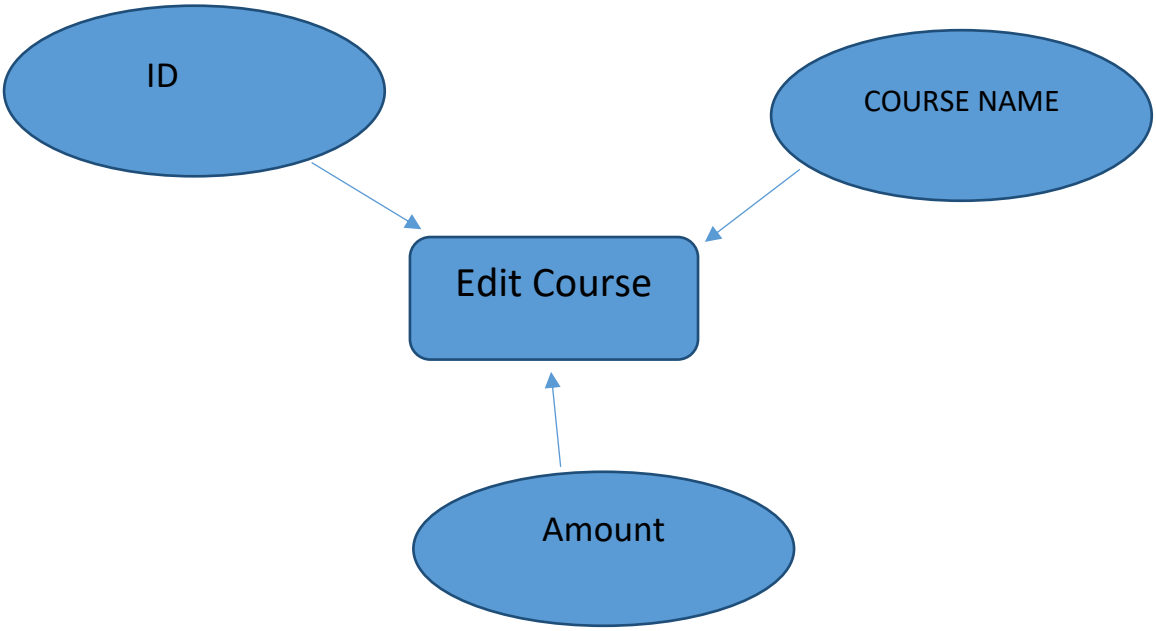


Figure 6.4: ER-diagram for Edit Course.

SYSTEM IMPLEMENTATION

When a user install setup, and run software, then first of all program displaying login page, as figure 7.1.

1. Login

User can login with using password and user name. Only when he has an account.

But if user not have an account, then login error prompt coming.

If user do not have an account, then first of all click on signup and go to the signup page.

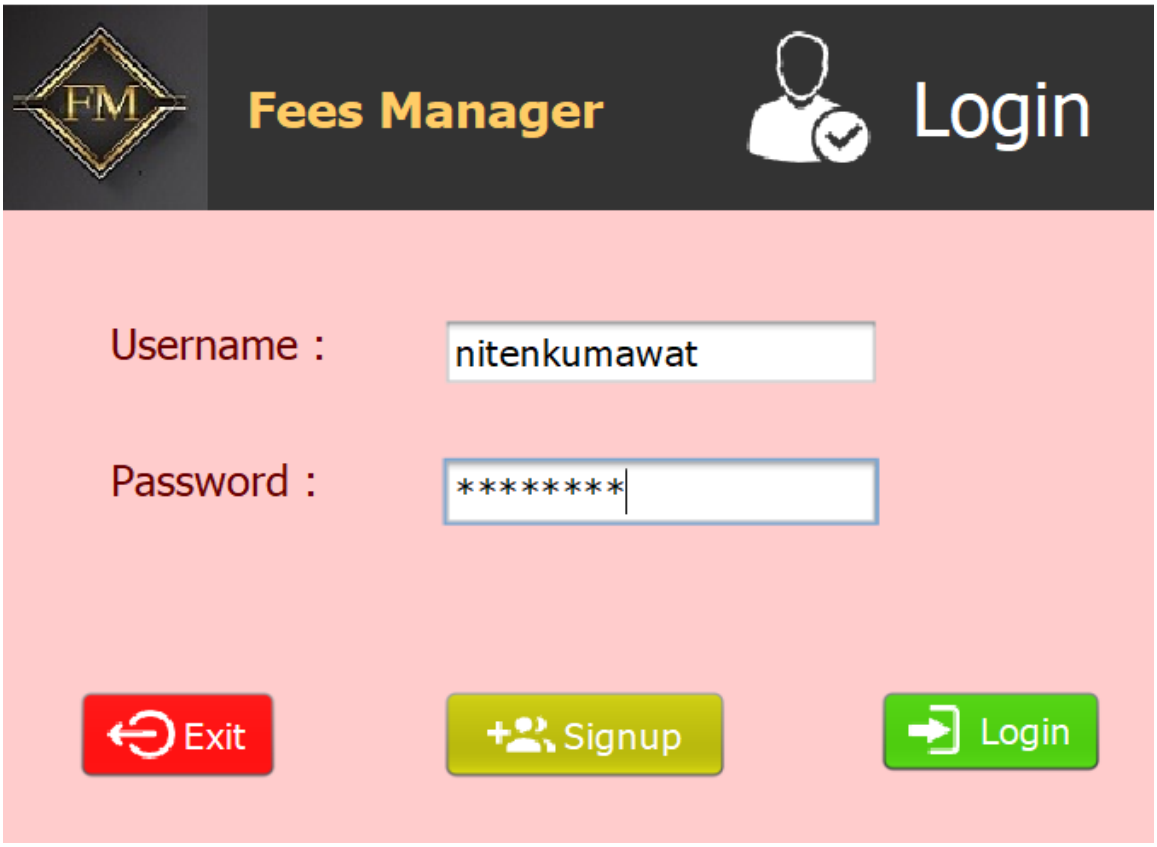


Figure 7.1: Login page.

2. Signup

If you are new to fees manager, then first signup. For signup user need to type first, last name and username. Password must be 8 characters.

Choose d.o.b. and contact numbers. And go for signup.

If user have an account by same username, then prompt will display.

After signup user can login and go for home panel.

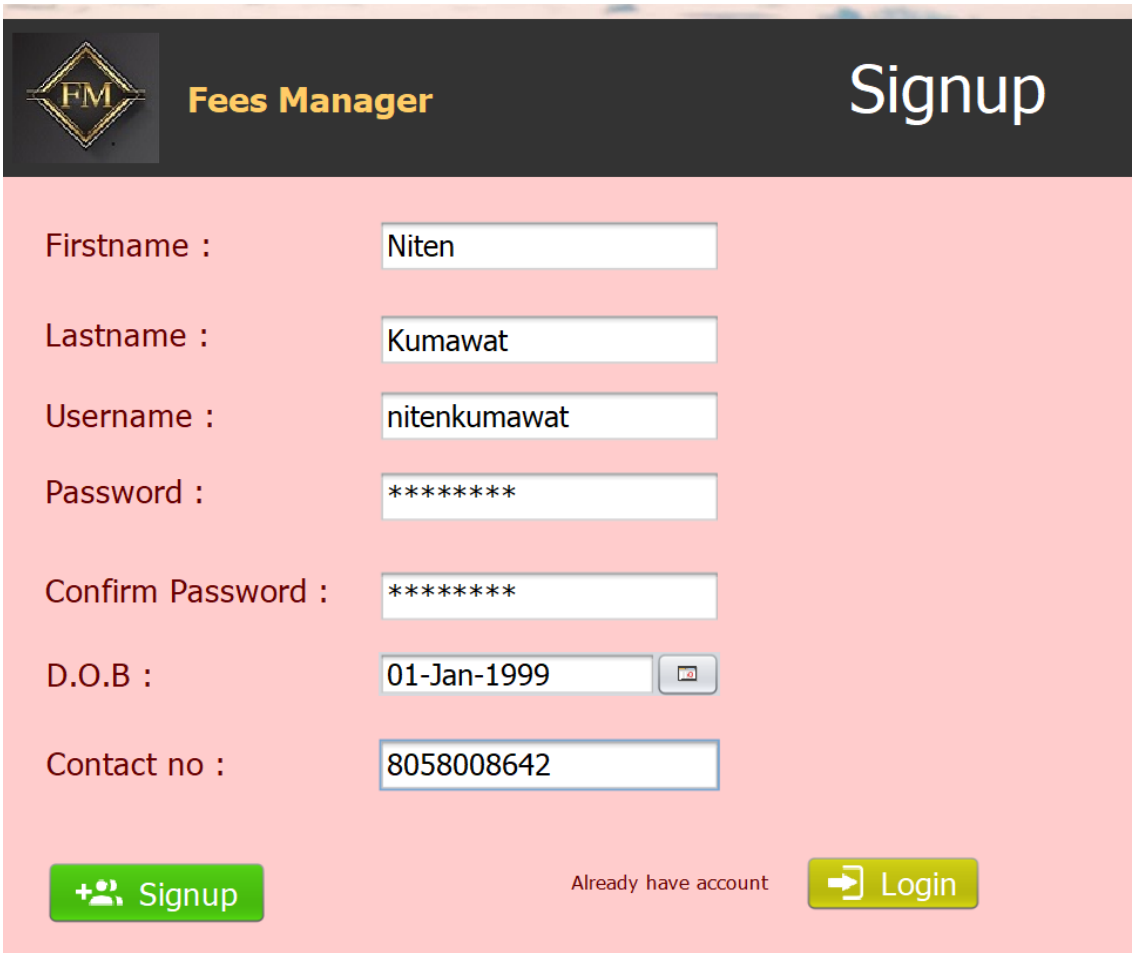


Figure 7.2: Signup page.

3. Home

When user get login successfully then home page will display on the screen. In this page user can add new fees, search records of fees deposits, view fees records, view courses edit courses and view report of deposits.

On top of the home page college and system's logo, and name is added you can see in figure 7.3.

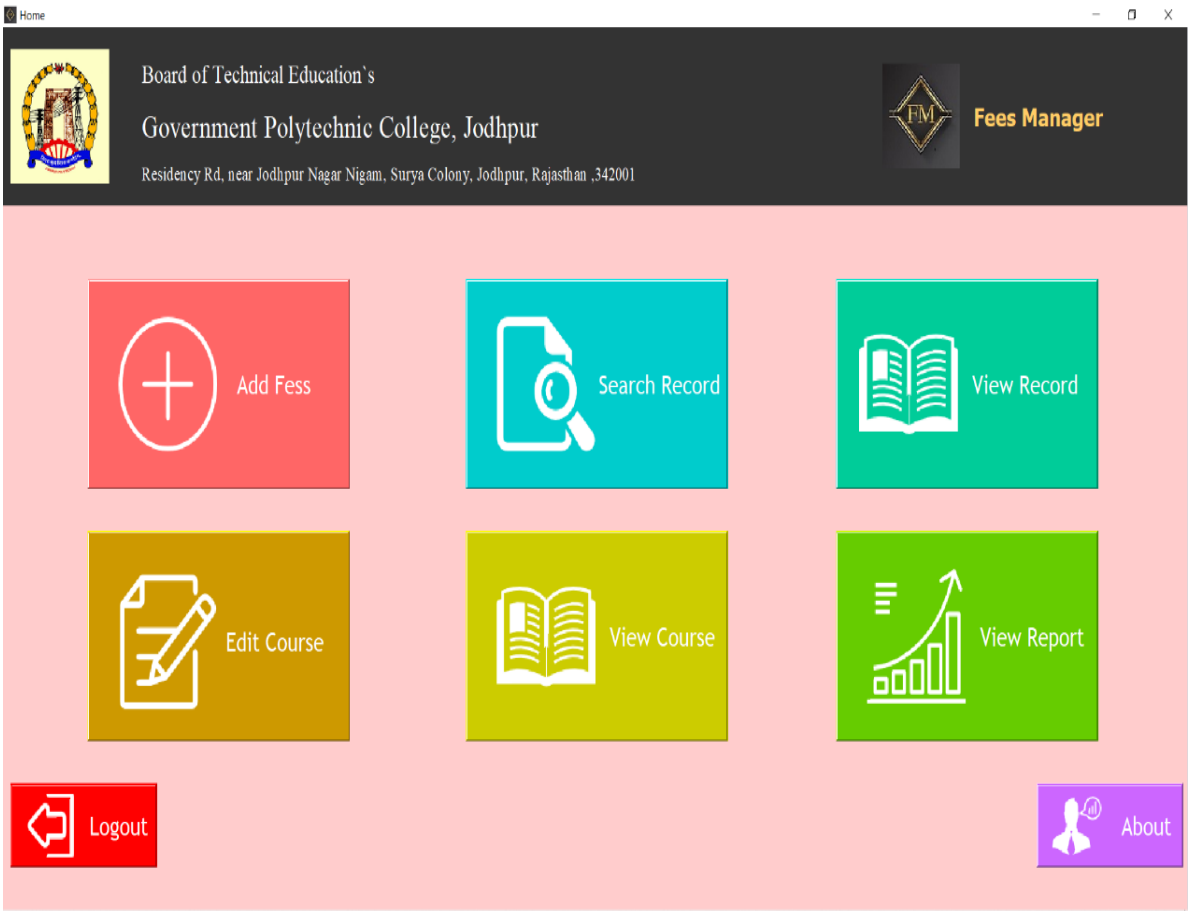


Figure 7.3: Home page.

4. About (Copyright Page)

In this about page you can see copyright and developer's information.

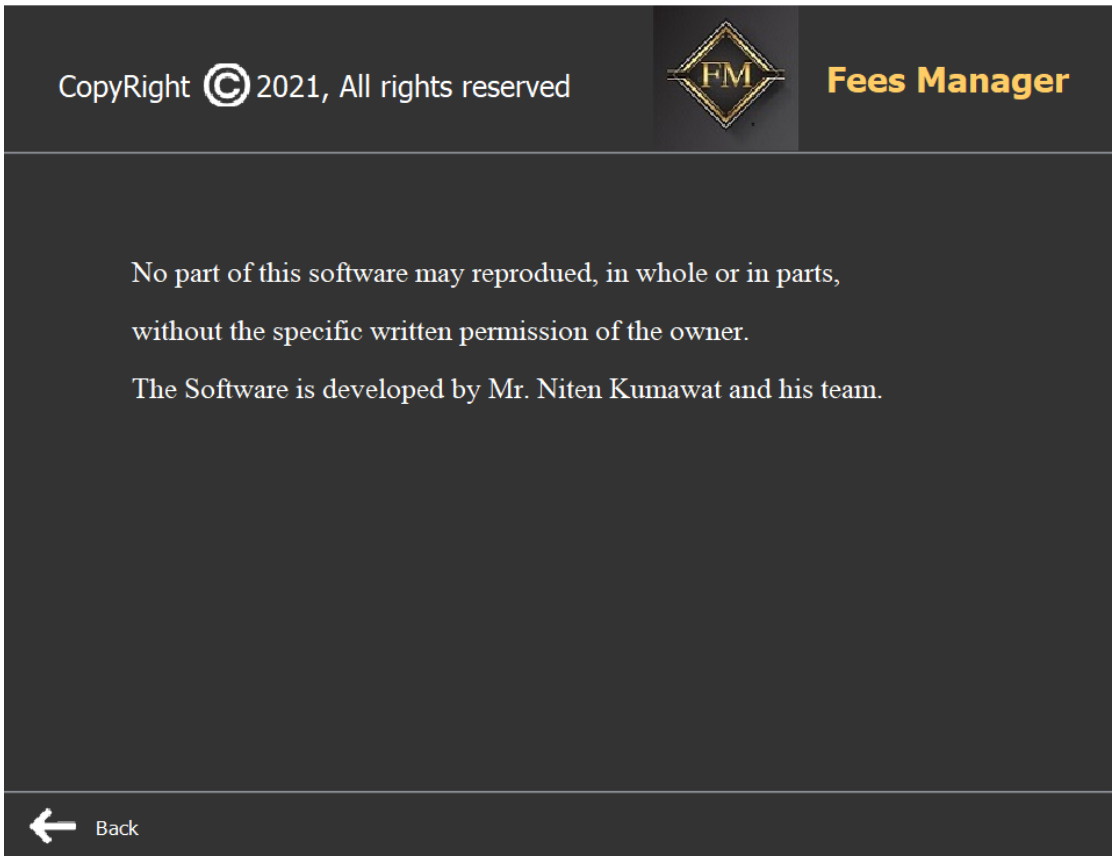


Figure 7.4 About page.

5. Add Fees page

In this page user can add student fees details. Student deposit his fees in four modes (Cash, Card, Cheque and DD); by choose a mode in combo box.

After enter course amount press enter program will calculate total with CGST and SGST, like figure 7.5

In this section user can also add a remark for future like fist ensoulment, and for save click on print button and go for print receipt page.

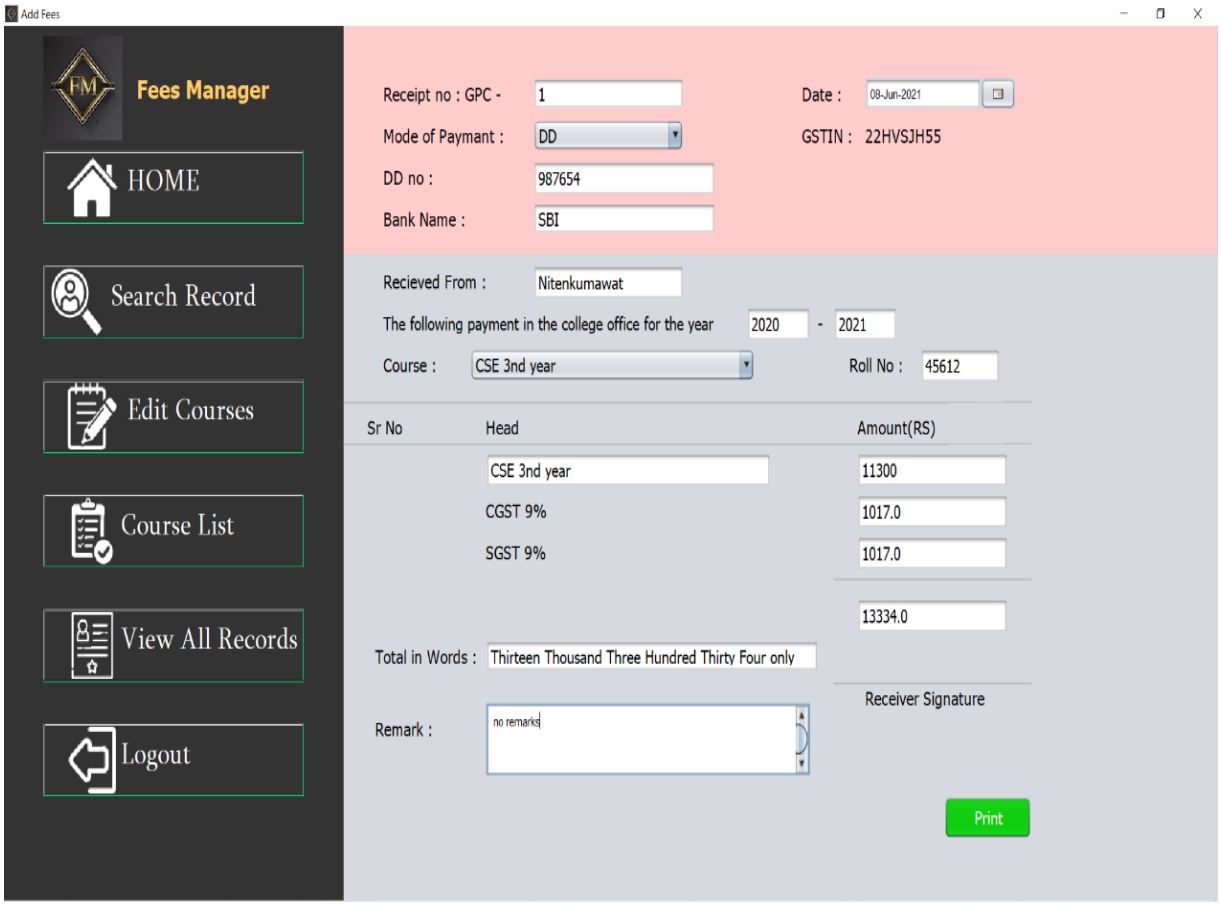


Figure 7.5: Add fees page.

6. Print Receipt page

If all of details are correct then click in print and print the receipt.

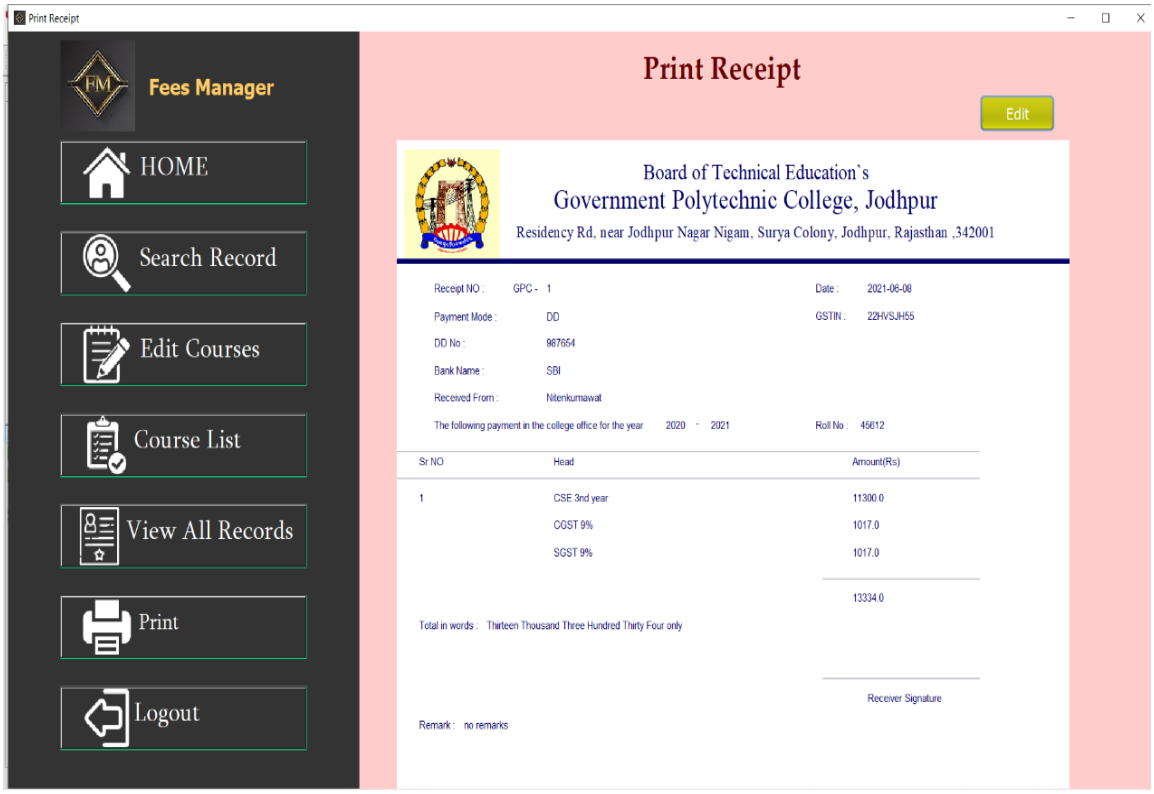


Figure 7.6: Print receipt page.

And if you want edit this receipt then click on edit and go for update receipt page.

7. Update Fees Details

In update fees details page user can make changes in fees details and go for print receipt page.

In the figure 7.6 we view some mistakes, so now we change them like bank name and roll no etc like figure 7.7

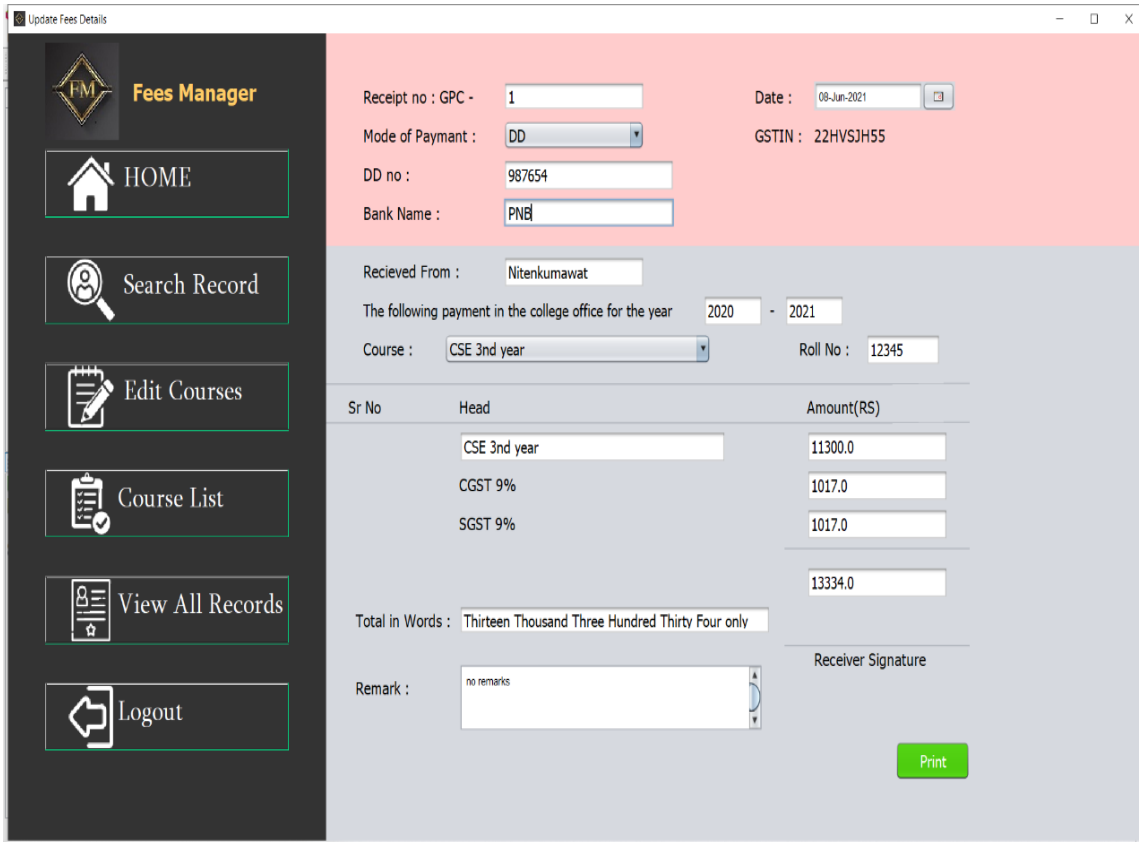


Figure 7.7: Update fees details page.

8. Updated print receipt pages

After the add and edit feed details admin comes on print receipt page. for print receipt click on print button and save and print of receipt. Like figure 7.8.

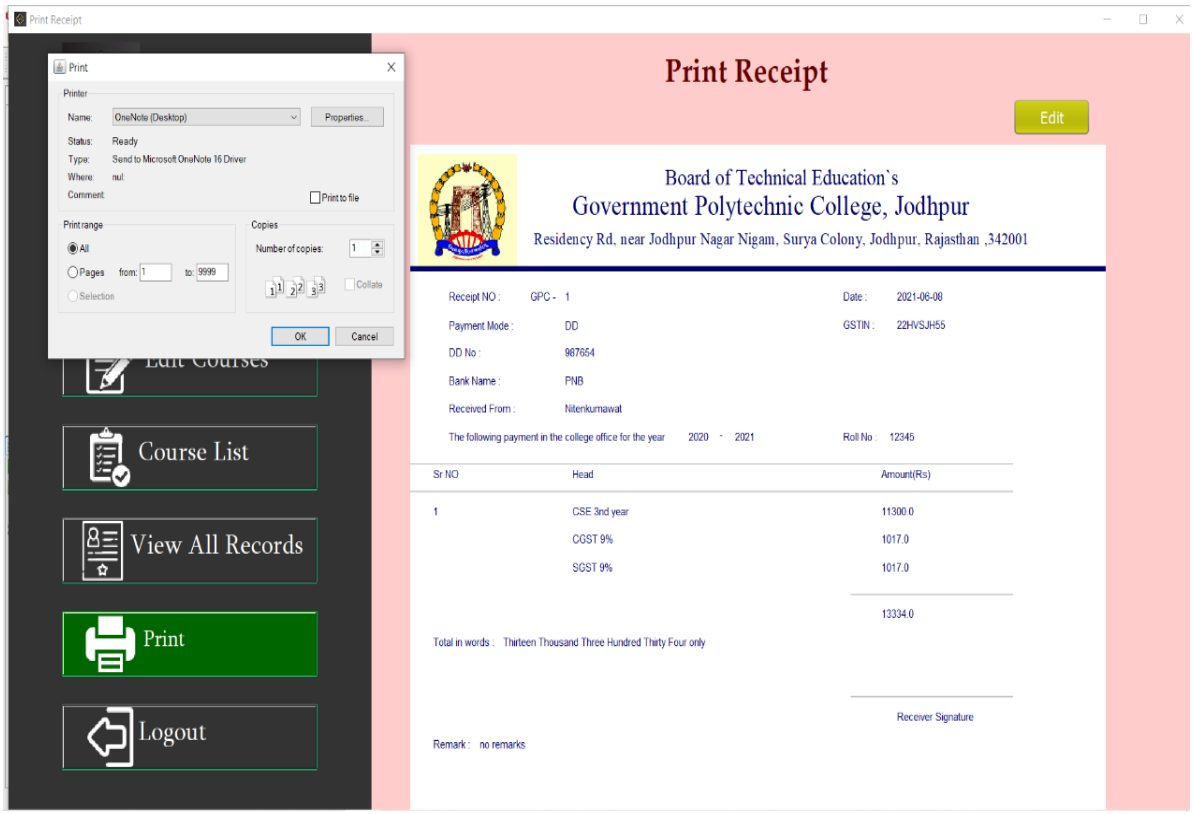


Figure 7.8: Print receipt page.

9. Edit Course Details

In college types of many courses are available, so if admin wants add a new course, update an existing course and delete courses; That's why edit course details page was developed.in figure 7.9 you can view edit course page of fees manager.

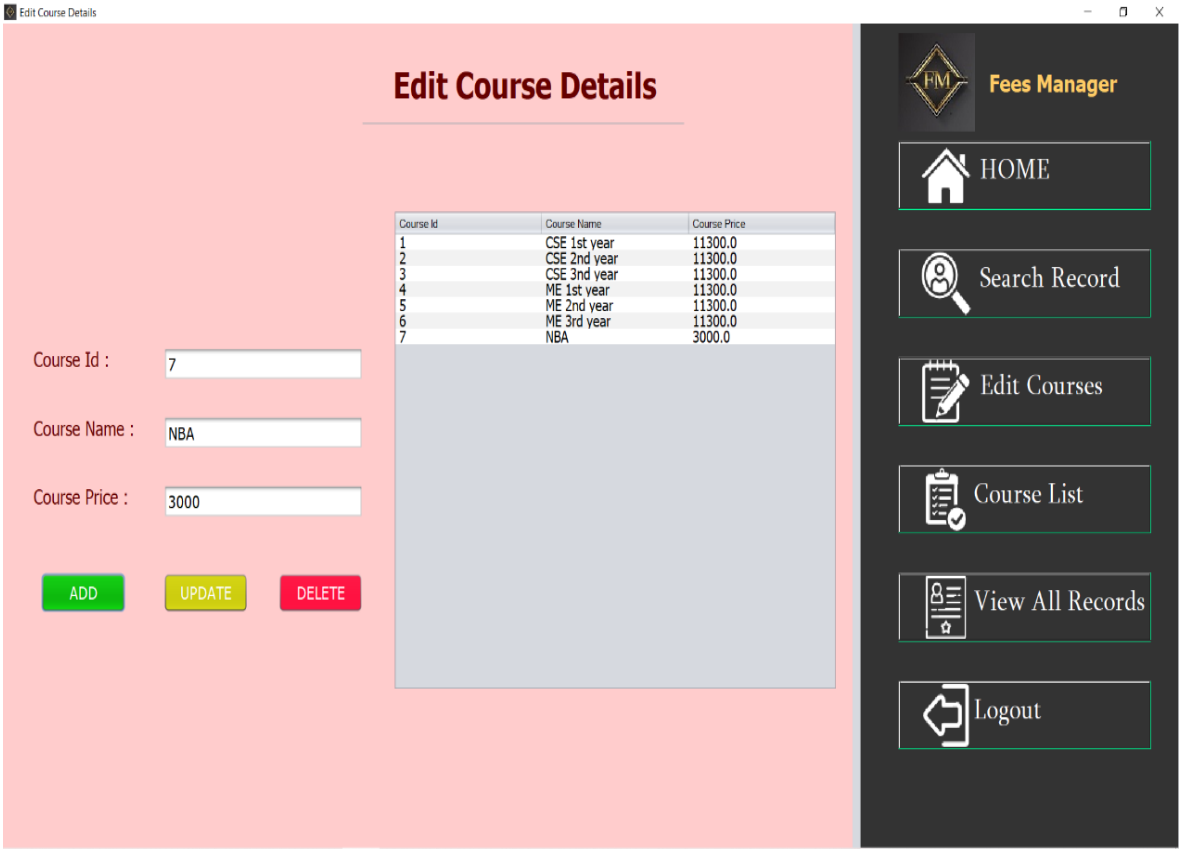


Figure 7.9: Edit course details page.

10. View Course Details

If admin wants to view course price and name, he can see all course details by click view course details in home page and course List in any page side bar.

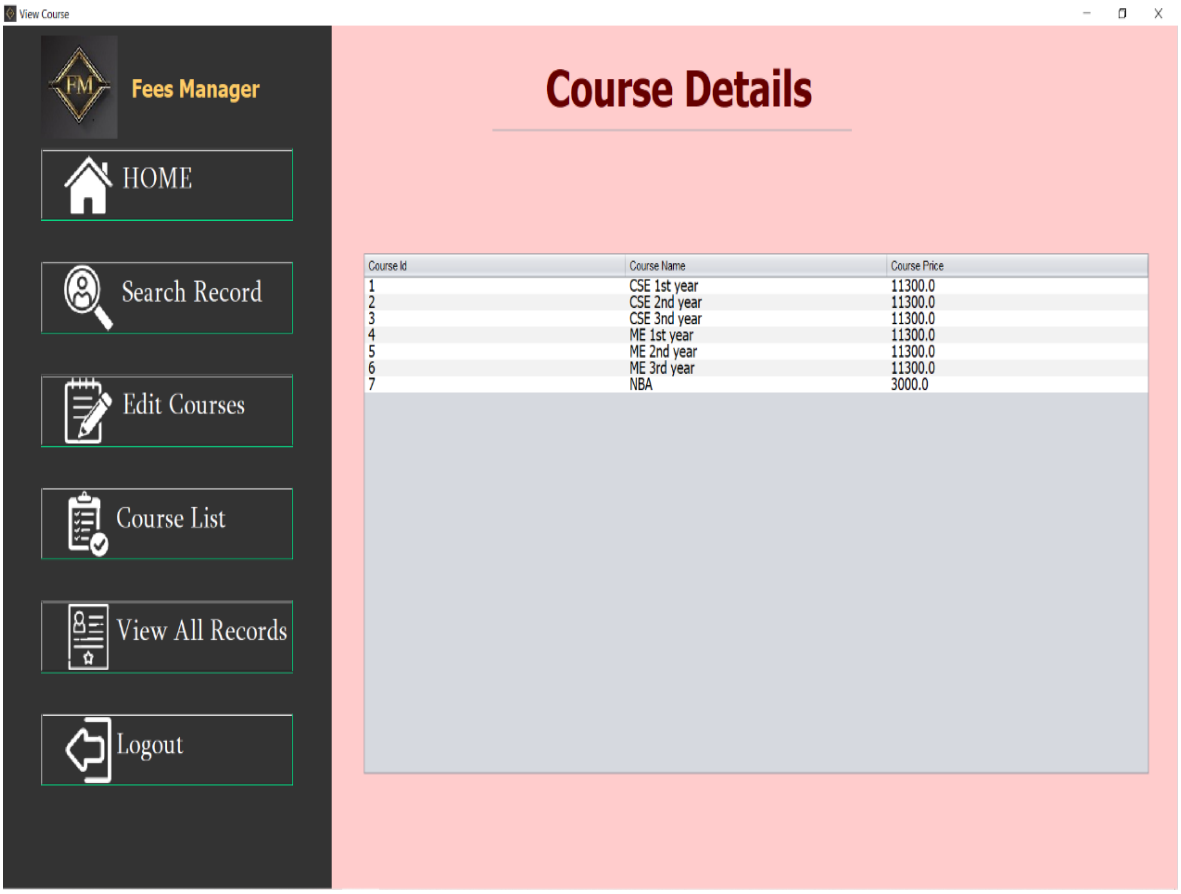


Figure 7.10: View course details

11. View Records

If admin wants to view records of fees deposits and other details; he can see all records by click view records in home page and in any page side bar.

In this page a table will display. In this table you can see receipt no, roll no, student name, course name total amount mode of payment date of deposit and remarks like figure 7.11.

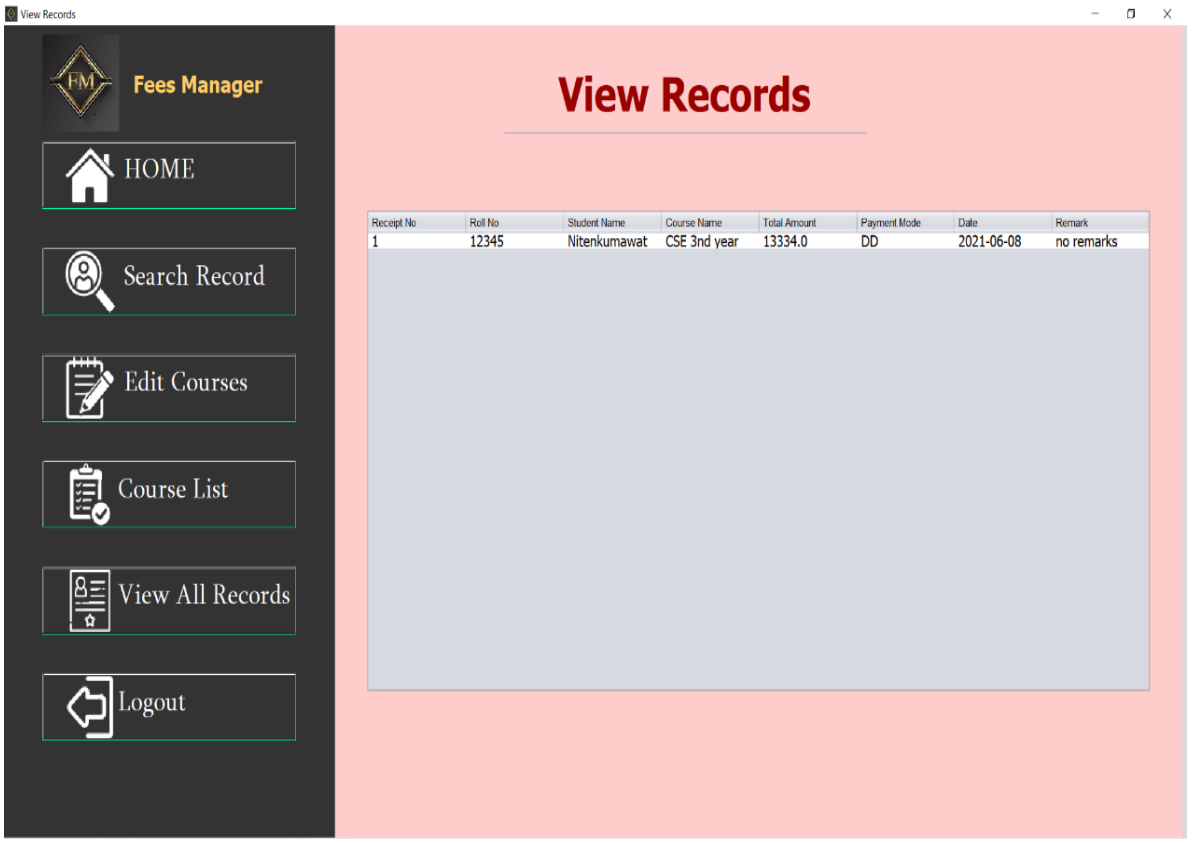


Figure 7.11: View records page.

12. Search Records

If admin wants to search fees deposits records by a specific data; he can go for search records page and search any detail by specific data.

Like figure 7.12 you can see search by roll no data.

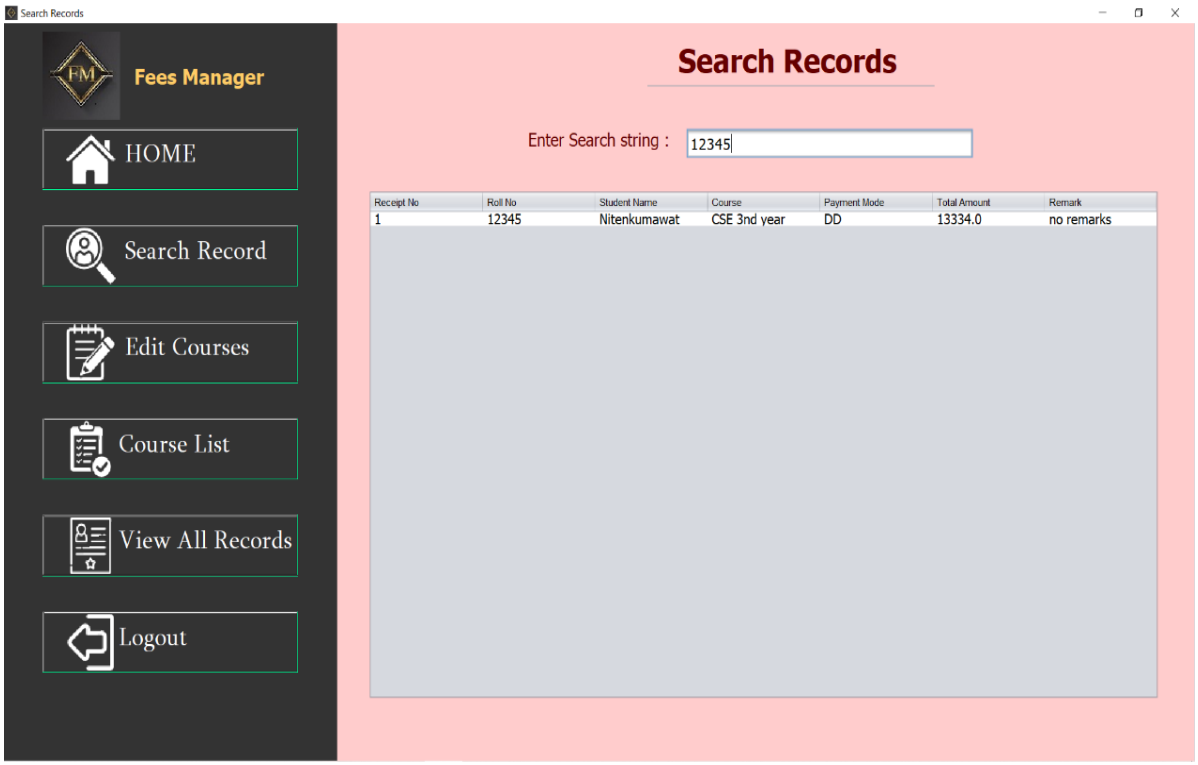


Figure 7.12: Search records page.

13. View Report

The fees manager provides a report of fees collection between specific dates.

Select course choose dates and submit. Now you can see all report with details. You can also print table by click print button.

If admin want to export report in excel file browse for path and enter name of file .and click to export to excel button and you see a prompt like figure 7.13.

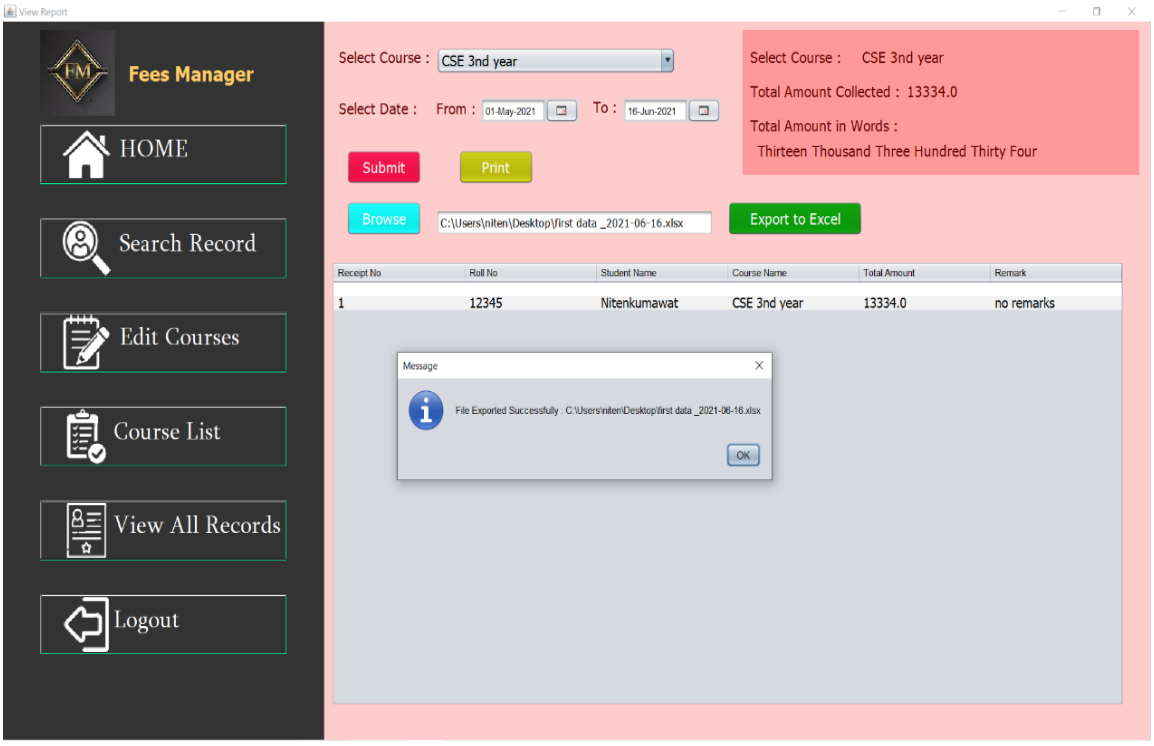


Figure 7.13: View report page.

8

RESULTS

The main objective of the Project on Fees Management System is to manage the Fees details of Student, College Courses, Deposit. It manages all the information about Fees. The project is totally built at administrative end and thus only the administrator is guaranteed the access. The purpose of the project is to build an application program to reduce the manual work for managing the Student Deposits of Fees. It tracks all the details about the Course and Deposits.

Key points in my project: -

1. Only admin can login in system.
2. Admin can add fees details and edit them before print.
3. Admin can search details by any specific data.
4. User can view reports by any dates and export data to excel and print it.
5. Admin can view Course and charges, fees details.
6. Students can add fees by DD, cheque, cash and card.

LIMITATIONS

Although I have put my best efforts to make the software flexible, easy to operate but limitations cannot be ruled out even by me.

Though the software presents a broad range of options to its users some intricate options could not be covered into it; partly because of logistic and partly due to lack of sophistication.

Paucity of time was also major constraint; thus, it was not possible to make the software fool proof and dynamic. Lack of time also compelled me to ignore some part such as Due payments and student details etc.

Considerable efforts have made the software easy to operate even for the people not related to the field of computers but it is acknowledged that a layman may find it a bit problematic at the first instance. The user is provided help at each step for his convenience in working with the software.

List of Limitations: -

It can run only in Windows operating system. It is not support mac and android operating system.

In this project, we cannot add student details and cannot generate due report.

10

FUTURE ENHANCENENTS

We can give more advance software for Fees Management System including more facilities.

In future, this project can run on any OS. We can add authentication by USER ID in this project.

The above-mentioned points are the enhancements, which can be done to increase the applicability and usage of this project.

Here we can maintain the records of Student fees deposits. Also. as it can be seen that now-a-days the players are versatile. i.e., so, there is a scope for introducing a method to maintain the Fees Management System.

Enhancements can be done to maintain all the student and college.

We have left all the options open so that if there is any other future requirement in the system by the user for the enhancement of the system then it is possible to implement them.

In the last we would like to thanks all the persons involved in the development of the system directly or indirectly. We hope that the project will serve its purpose for which it is develop there by underlining success of process.

CONCLUSION

This application provides a computerized version of Fees management system which will benefit the students as well as the staff of the library.

It makes entire process online where admin can search records, user can generate reports and do Fee's transactions.

Make functioning of Fees managing faster.
To minimize the loss of time and money of admin and students.

Note: - This software is presently working in 32-bit architecture windows operating system. Hence, we are expanding this project also in 64-bit architecture windows.

REFERENCES

1. www.quora.com (31/05/2021)
2. www.stackoverflow.com (05/06/2021)
3. www.cplusplus.com (13/06/2021)
4. www.geekforgeeks.org (10/06/2021)
5. www.sololearn.com (14/06/2021)
6. www.includehelp.com (13/06/2021)

Database and NetBeans

1. https://db.apache.org/derby
2. https://netbeans.apache.org

BIBLIOGRAPHY

1. Fundamental of Software Engineering Rajib Mall, PHI

(ISBN 9789388028028)

2. An Integrated approach to Software Engineering Pankaj Jalote, Springer

(ISBN 978-1-4757-3857-5)

3. Internet and Java Programing R.Krishnamurthy,S. Prabhu New age

(ISBN: 8122413528)

4. The Complete Reference Java 2 Herbert Schildt, (TMH)

(ISBN 0072119764)

5. Java 2.0 Programming E.Balaguruswami, (TMH)

(ISBN 978-954-400-773-7)