
*Green University of Bangladesh*

*Department of Computer Science and Engineering (CSE)*
*Semester: (Fall, Year: 2023), B.Sc. in CSE (Day)*

---

# GUB Fee Management System

---

*Course Title: Database System Lab*
*Course Code: CSE 210*
*Section: 213 D11*

Students Details


| Name | ID |
|---|---|
| Md. Israil Fakir | 221902125 |


*Submission Date: 03 Jan 2024*
*Course Teacher's Name: Md. Noyan Ali*

[For teachers use only: Don't write anything inside this box]

| **Lab Project Status** | |
|---|---|
| **Marks:** | **Signature:** |
| **Comments:** | **Date:** |

# Contents

# Chapter 1

# Introduction

## 1.1 Overview

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

It is a management system mark to give the safe storing of Fee details and courses. The most useful and reliable functions can be found in another system. The project is built in Java platform with the help of IDE NetBeans and MySQL database.

However, for me, second-year students with limited knowledge 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.

## 1.2 Motivation

In this course, we learn the database system and it is time to implement it in a real problem. So I chose the GUB Fee Management System where I insert the concept and experiment in our lab.

The motivations are:

- The objective of this system is to make less paperwork. And make the accounting department more reliable.
- Provide better services to students and the accounting department.
- Provide meaningful, consistent, and timely data and information to the management.

## 1.3 Objectives

The Objectives are:

- To reduce paperwork.
- To make storage of information more efficient and secure.
- To have a friendly interface.
- To operate it easily and with minimum experience.
- To save time and energy for the admin.

## 1.4 Problem Statement

The various problems faced by them are:

- A lot of manual work is to be done.
- Manual working is more error-prone.
- Easily find out the fee record of the students.

## 1.5 Problem Outline

In the beginning, we studied the problems that are faced by the accountant, like fee submission, finding out records, and general info about students.

### 1.5.1 Requirement

At first, I collected all the requirements from the account office. Also talked to the staff concerned with it. I discussed the problems and related topics to them.

### 1.5.2 Analysis

After collecting the data and information, we discussed the real needs of the Fees Management System. We analyzed the problem. And according to the needs, we started framing our project.

### 1.5.3 Designing

After analyzing the needs we decided no of the forms that we needed and required forms suitable for our project. After this, we started the designing of forms i.e. about their lookup or appearance. According to the forms required, we designed our database for storing the huge record.

### 1.5.4 Coding and Logices

According to the design of forms database tables, we applied the required logic to make the form work. First, we selected the logic and put MySQL queries in it. After this, we implemented it into the form.

### 1.5.5 Testing & cost estimation

After the coding phase i.e. applying the logic we tested each form whether there are working or not. First, we tested individually each form, then integrated each form, and then tested further.

According to the working hours and labor required, I can not calculate the cost of the project. Because I work only to learn how to database in real life.

# Chapter 2

# Requirement Analysis and Related work

## 2.1 System Analysis

System Analysis refers to the process of examining a situation with the intent of improving it through better procedures and methods. System design is the process of planning a new system to either replace or complement an existing system. But before any planning is done, the old system must be thoroughly understood and the requirements determined. System analysis is therefore the process of gathering and interpreting facts, diagnosing problems, and using the information to recommend improvements in the system. In other words, system analysis means a detailed explanation or description.

## 2.2 Identification of Need

The first step of the System Analysis process involves the Identification of need. The analyst meets with the customer and end-user. Identification of need is the starting point in the evolution of a Computer-based system. The analyst assists the customer in defining the goals of the system e.g:

- What information will be produced?
- What information is to be provided?
- What functions and performance are required?

Now the basic question that arises in the identification of need is "Why this system?"

With the advent of demanding applications and the need for applications, we need a real-time system, that can help us view information about universities online. Future Students can view all info and facilities, and current student can view their results and records online and discuss their problems with others.

## 2.3 Hardware And Software Requirements

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

### 2.3.1 Software Requirements

- **Operating System:**
  - **Windows 10** is used as the operating system as it is stable supports more features and is more user-friendly. Development tools and Programming language- Java, and NetBeans are used to write the whole code and develop graphics. [1]
- **NetBeans:**
  - **NetBeans** is an integrated development environment for Java. NetBeans allows applications to be developed from a set of modular software components called modules. [2]
- **Programming Language:**
  - **Java** is a high-level, class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible.
- **Java swing:**
  - **Swing** is a GUI widget toolkit for Java. It is part of Oracle's Java Foundation Classes – an API for providing a graphical user interface for Java programs. [3]
- **Database:**
  - **MySQL** is an open-source relational database management system (RDBMS) that uses structured query language (SQL) to manage and manipulate data. It is one of the most popular database systems and is widely used for web applications and other types of software.
- **XAMPP:**
  - **XAMPP** is a free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, **MariaDB** database, and interpreters for scripts written in the PHP and Perl programming languages. [4]

### 2.3.2 Hardware Requirements

- **Processor:**

  - A modern dual-core or quad-core processor should be more than sufficient for basic Java Swing and MySQL development. If you're working on larger projects or handling complex calculations, a more powerful processor might provide a smoother development experience. Such as at least a Celeron 300 CPU expandable to PIII 1.2 GHz.

- **Memory(RAM):**
  - Memory requirements can vary depending on the size of your application and datasets. For a basic development environment, 4GB to 8GB of RAM should be sufficient. If you're dealing with large datasets or running multiple applications concurrently, you may need more.

- **Disk Space:**
  - An **SSD** (Solid State Drive) is preferable for better read/write speeds, but a traditional **HDD** (Hard Disk Drive) should work for basic development needs.

# Chapter 3

# Implementation of the Project

## 3.1 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, there 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 concrete information which poses a problem in the case of information related to a particular search record.

Our project is very useful. The 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 does 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 workload of an organization.

In today's world, no one likes to perform calculations on a calculator or manually when the 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 a GUB Fee Management System.

## 3.2 Flowchart

In this section, I keep the database project Flowchart: [5]

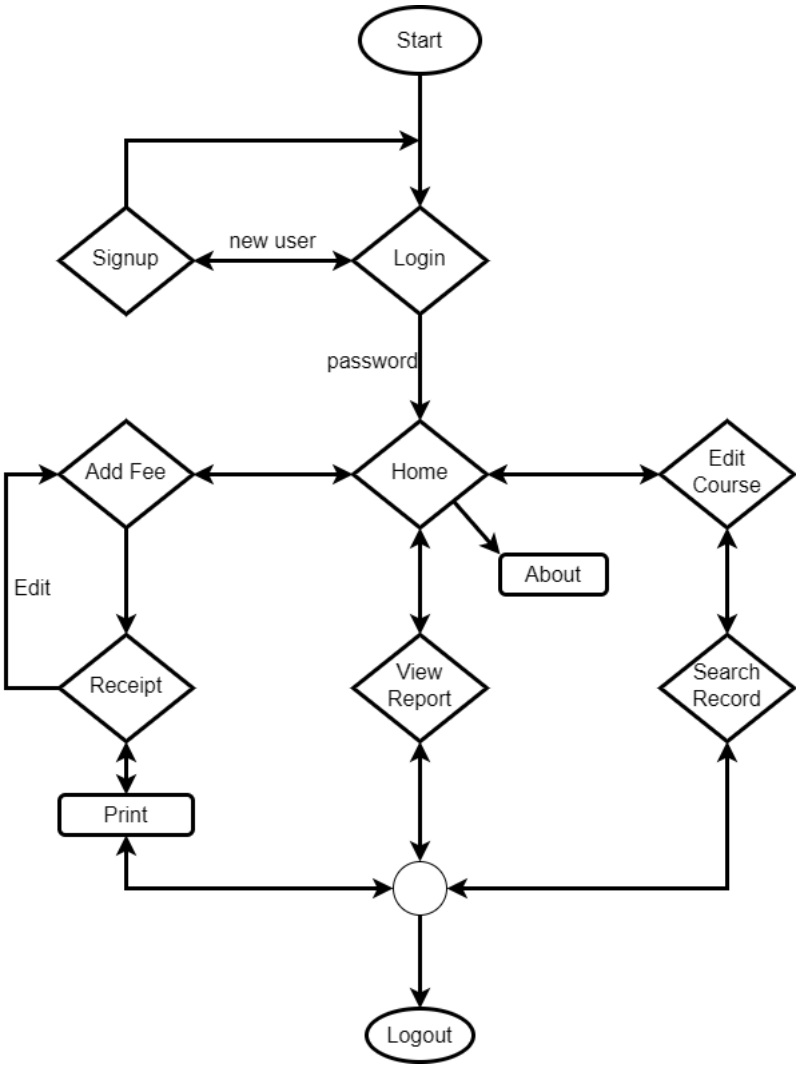


Figure 3.1: Flowchart

## 3.3 ER Diagram

In this section, I keep the database project ER Diagram:

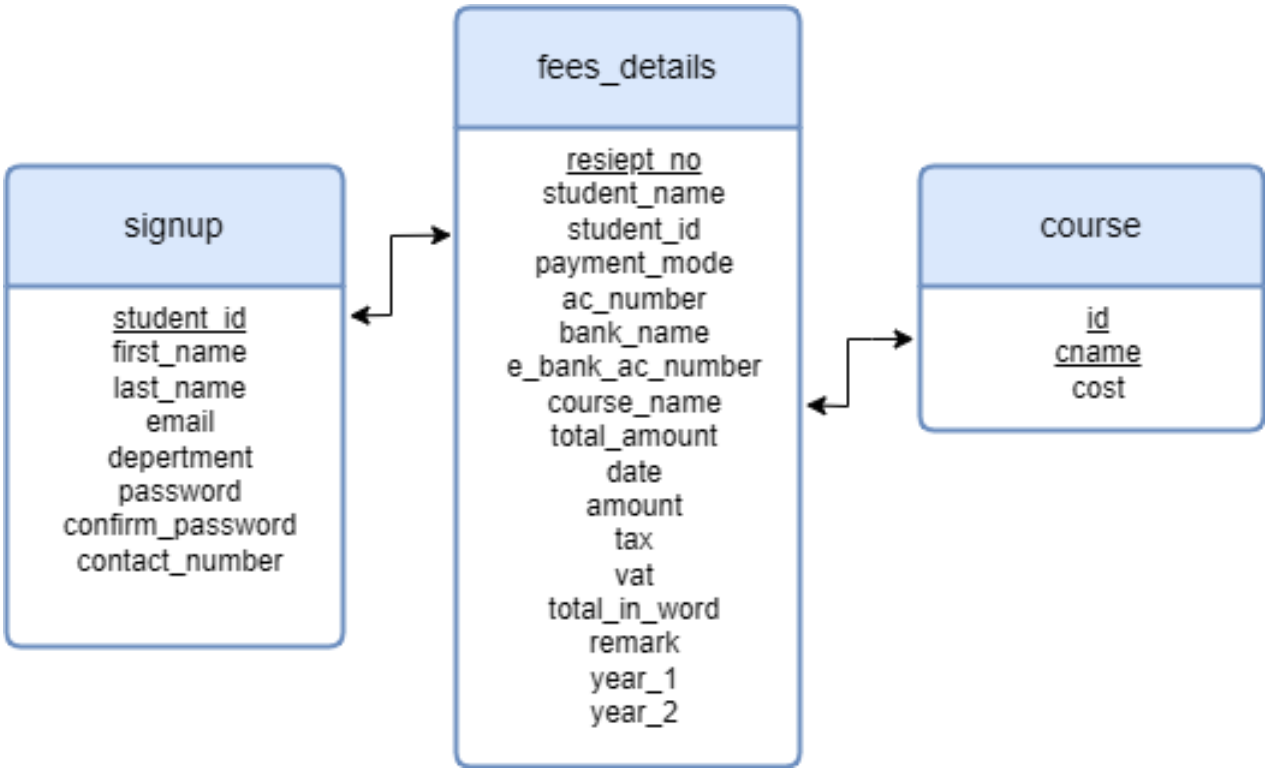


Figure 3.2: ER Diagram

## 3.4 Database Table:

In this section, I keep the database project Table:

| Field Name | Field Type |
|---|---|
| **ID** | int |
| **CName** | Varchar |
| **Cost** | int |

Figure 3.3: Course Table

| Field Name | Field Type |
|---|---|
| first_name | varchar |
| last_name | varchar |
| username | varchar |
| email | varchar |
| student_id | int |
| depertment | varchar |
| password | varchar |
| confirm_password | varchar |
| contact_number | varchar |

Figure 3.4: Signup Table

| Field Name | Field Type |
|---|---|
| resiept_no | int |
| student_name | varchar |
| student_id | int |
| payment_mode | varchar |
| ac_number | int |
| bank_name | varchar |
| e_bank_ac_number | varchar |
| course_name | varchar |
| total_amount | float |
| date | date |
| amount | float |
| tax | float |
| vat | float |
| total_in_word | varchar |
| remark | varchar |
| year_1 | int |
| year_1 | int |

Figure 3.5: Fees Details

# Chapter 4

# Performance Evaluation

## 4.1 Simulation Environment

### 4.1.1 Signup page:

This page is for only new users. some characteristics of this page. Such as :

- You cannot sign up without a gap in any field on this page
- The password must be 8 digits or more.
- The password and confirm password must match.

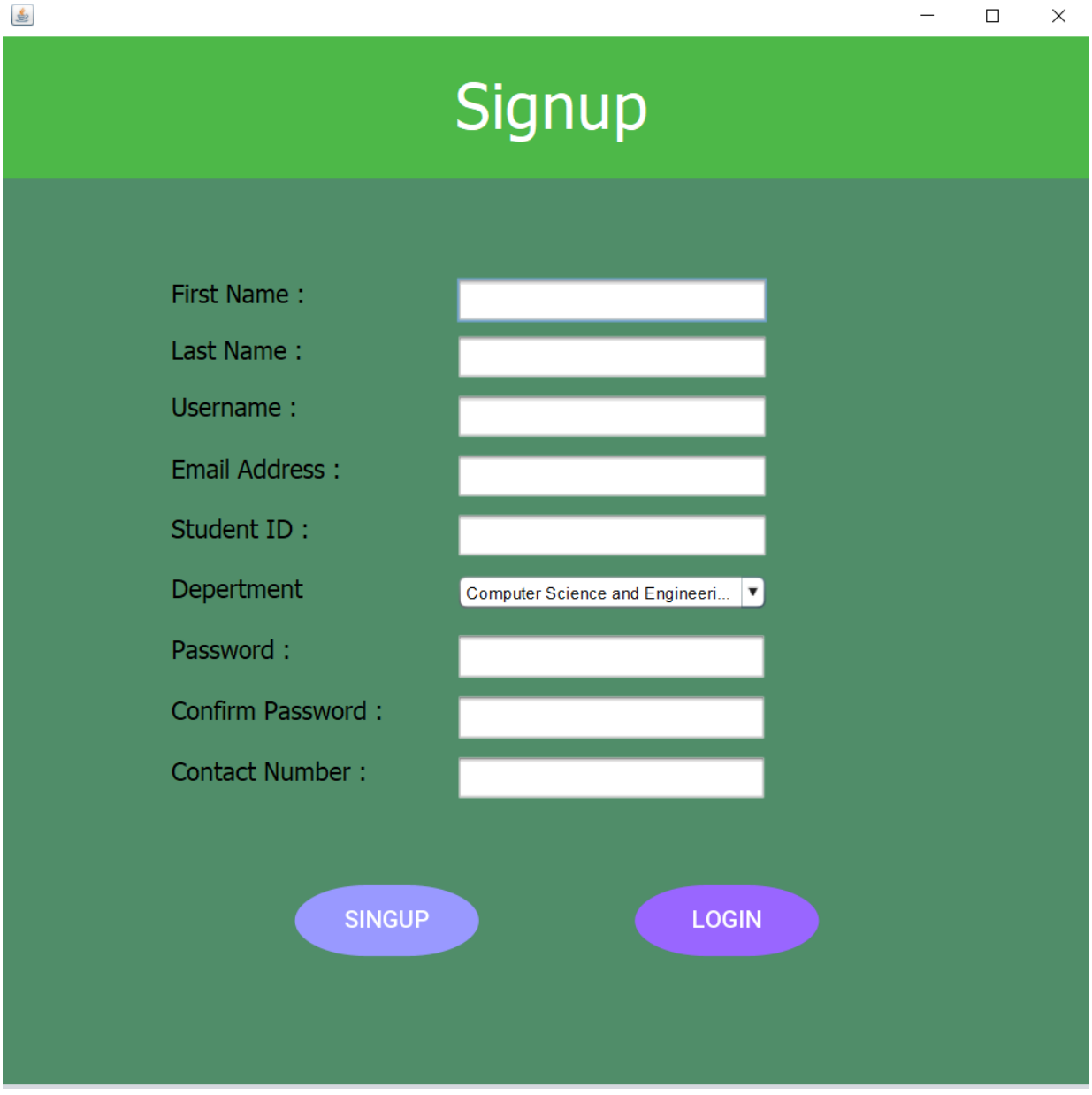


Figure 4.1: Signup page

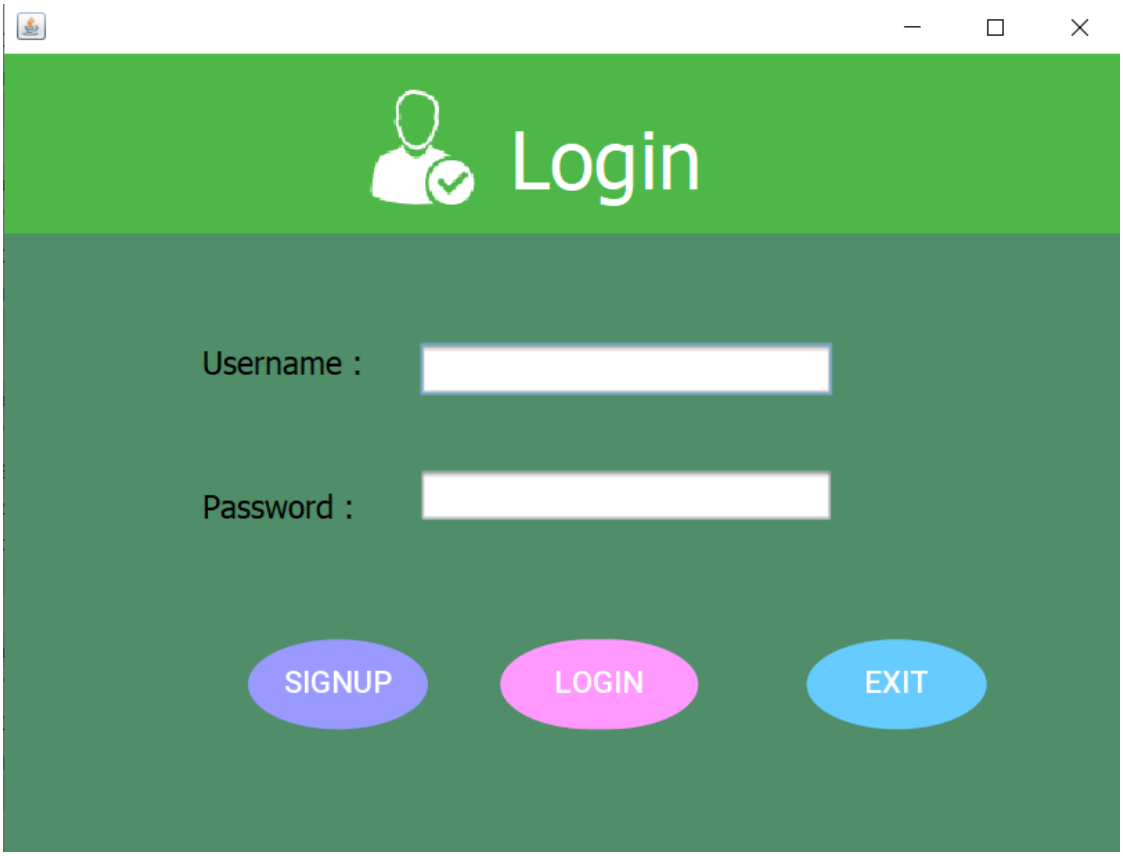


Figure 4.2: Lonin Page

### 4.1.2 Lonin Page

[h] This page is for only existing users else we need to create a new user ID.

### 4.1.3 Home page:

On this page, we face multiple options to access a variety of page


Figure 4.3: Home page

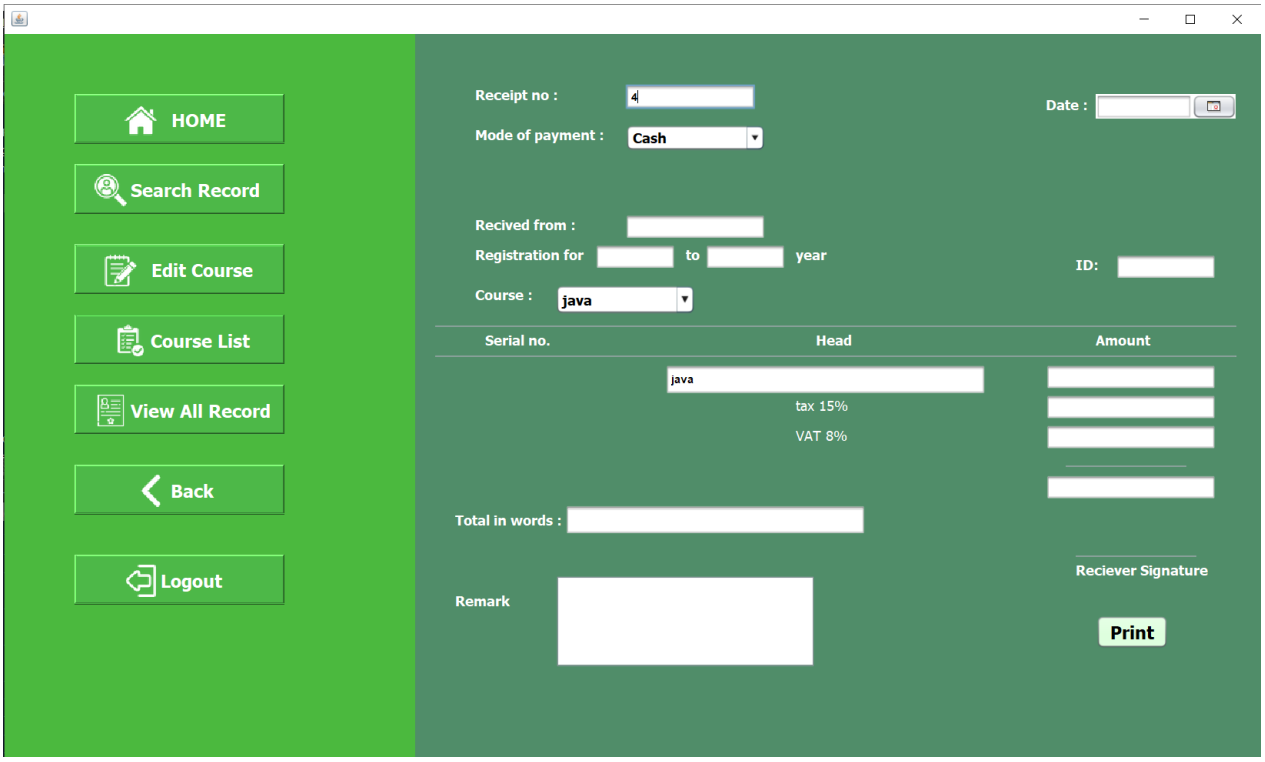


Figure 4.4: Add Fee page

### 4.1.4 Add Fee:

On this page, an admin can add data for new students. this page has some characteristics of this page.

Such as :

- The receipt number is automatically generated.
- By clicking on any given Date.
- Payment mode can be selected. such as Cash, Card and etc.
- If we select the course then it is automatically added to the head part.
- We only generate the vat and tax automatically on the amount and also generate the Total in word part.
- By clicking the print button we can print the fee details.

### 4.1.5 Search Record:

On this page, an admin can search student's records. this page has some characteristics of this page. Admin can search any value like with-

- By using a Student first name or last name.
- By using a Student ID or Receipt number.
- By using course name
- By using Payment mode.

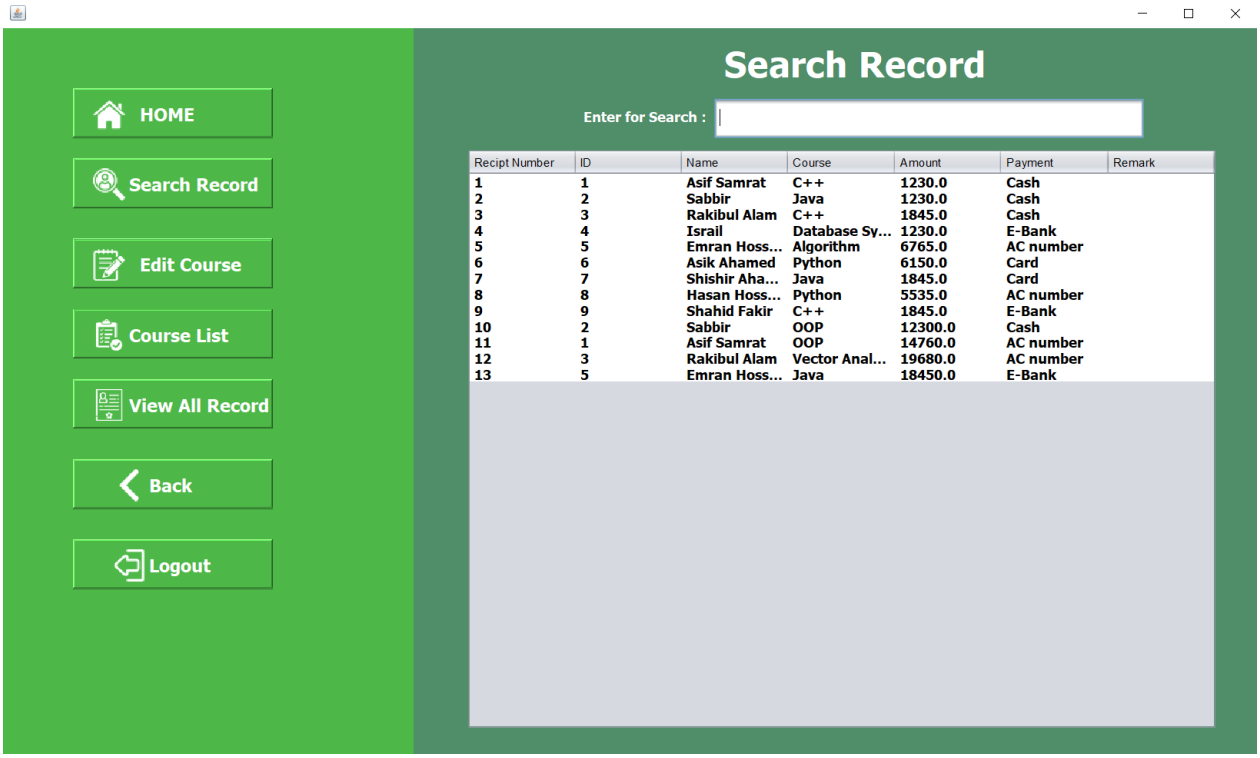


Figure 4.5: Search Record

### 4.1.6 View Recodes

On this page, we can see all the students that already enrolled in the course.


Figure 4.6: View All Recodes

### 4.1.7 Edit Course

On this page, an admin can add new course, update any course and delete any course.

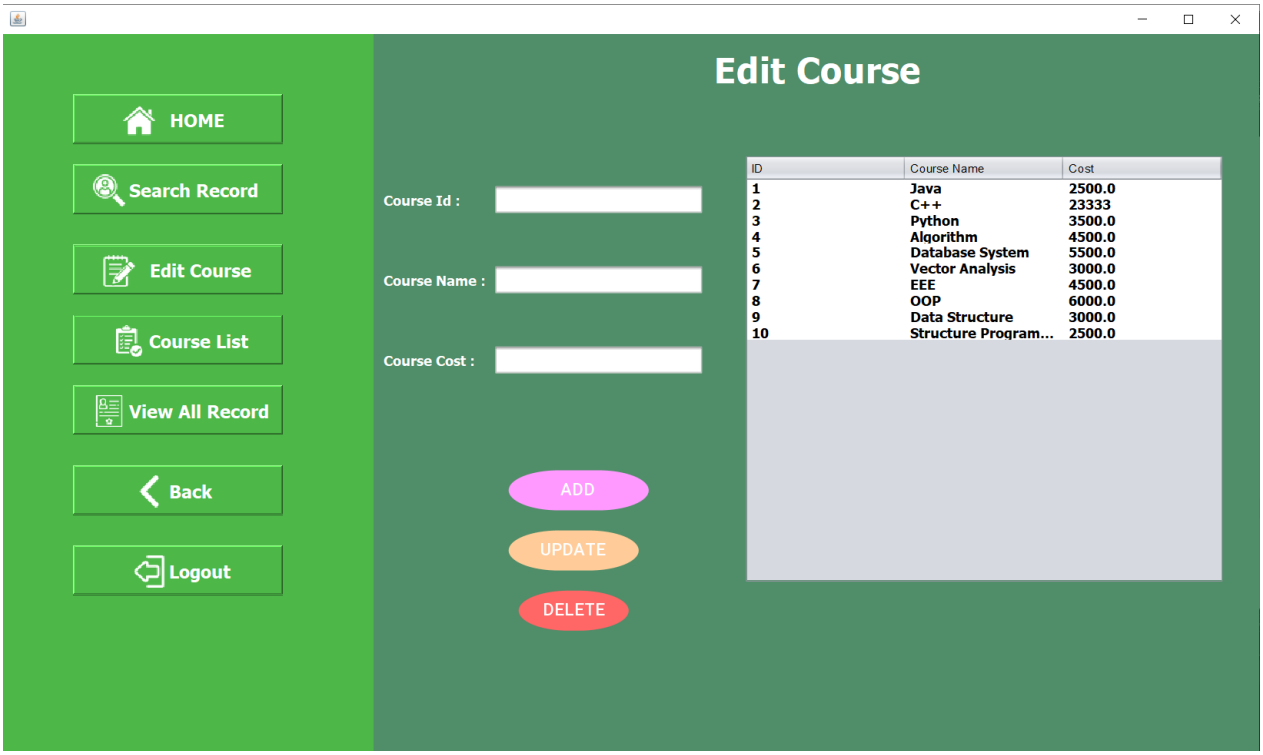


Figure 4.7: Edit Course Page

### 4.1.8 View Course Details

On this page, an admin can see the course details like id, course name, and cost.

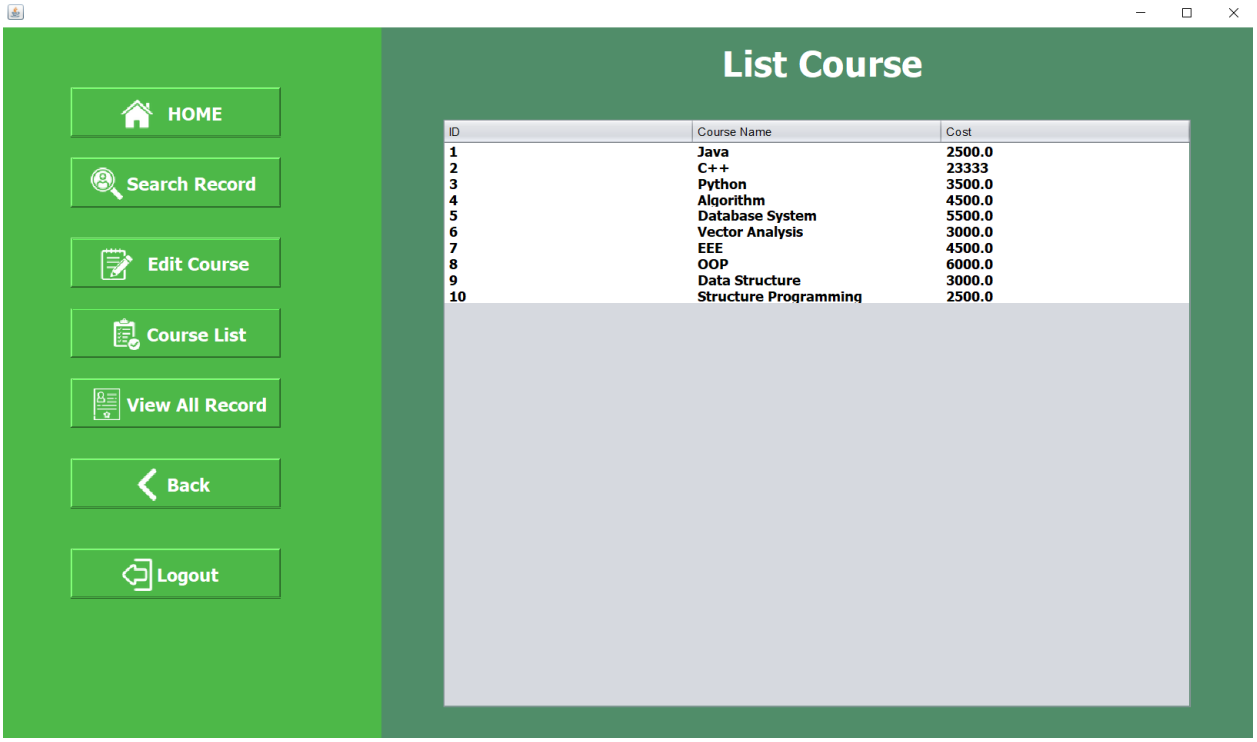


Figure 4.8: View all Course

### 4.1.9 View Report

On this page, an admin can generate student records from one date to another date and select a course. this page has some characteristics of this page. such as

- The user can see the total amount for select course on this page.
- Generate report by using from date to another date and also with course.
- After create the Report user can print out with Print button.
- After create the Report user can construct a Excel file with Path location.

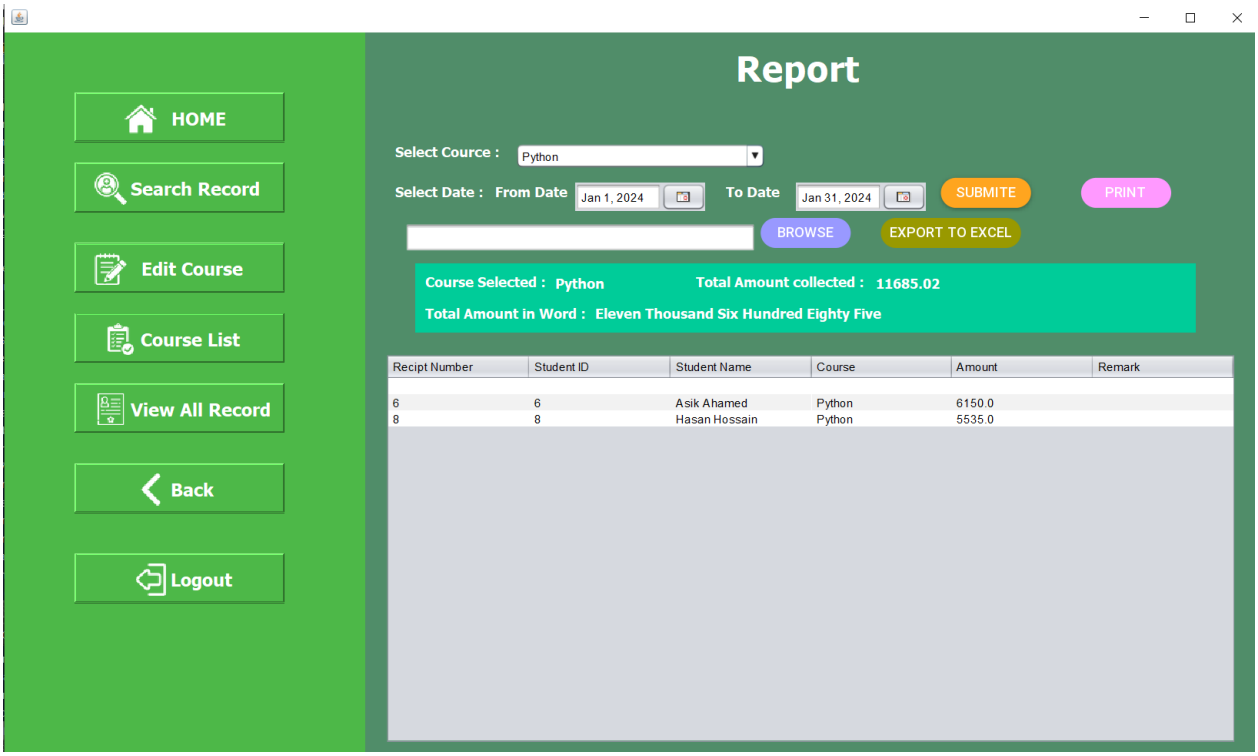


Figure 4.9: View Report

## 4.2 Result

The main objective of the Project on Fees Management System is to manage the Fees details of Students, Versity Courses, and Deposits. It manages all the information about Fees. The project is totally built at the administrative end and thus only the administrator is guaranteed 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.

## 4.3 Key points in my project:

- Only the admin can log in to the system.
- Admin can add fee details and edit them before printing.
- Admin can search details by any specific data.
- User can view reports by any date and export data to **Excel** and **print** it.
- Admin can view Course and charges, fees details.
- Students can add fees by Card, cash, and card.

# Chapter 5

# Conclusion

## 5.1 Discussion

This application provides a computerized version of the GUB Fee Management System which will benefit the students as well as the staff of the University. It makes the entire process online where the admin can search records, the user can generate reports, and do Fee's transactions. Make functioning of Fee management faster. To minimize the loss of time and money of admin and students.

## 5.2 Limitations

Although I have put my best efforts into making the software flexible, and easy to operate 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; partly because of logistics and partly due to a lack of sophistication. The paucity of time was also a major constraint; thus, it was not possible to make the software foolproof and dynamic. Lack of time also compelled me to ignore some parts such as Due payments and student details etc. Considerable efforts have made the software easy to operate even for 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.

It can run only on the Windows operating system. It does not support Mac and Android operating systems.

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

## 5.3 Scope of Future project

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

In the future, this project can run on any OS and also on it will an online version. 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 fee deposits. Also. Enhancements can be made to maintain all the students and the university. ln the last we would like to thank all the persons involved in the development of the system directly or indirectly. We hope that the project will serve the purpose for which it is developed thereby underlining the success of the process. [6]

# References


[1] Windows. https://www.microsoft.com/en-us/software-download/windows10.

[2] Netbeans IDE. https://netbeans.apache.org/front/main/.

[3] Programing Language. https://www.java.com/en.

[4] . https://www.apachefriends.org/.

[5] Diagram maker on online. https://app.diagrams.net/.

[6] All icon collect from online. https://www.flaticon.com/.