

DELHI TECHNOLOGICAL UNIVERSITY



SOFTWARE ENGINEERING (MC-310)

Midterm Innovative Project

SUBMITTED BY:

Aiman Siddiqua - 2K18/MC/008

Amogh Dhar Diwan - 2K18/MC/012

SUBMITTED TO:

Jamkhongam Touthang

DTUOversight

A University Management System to manage data and records of the students and the teachers and provide other details of the University.



INDEX

INTRODUCTION	4
MOTIVATION AND IMPORTANCE	5
WATERFALL MODEL	6
SEQUENTIAL PHASES IN WATERFALL MODEL	7
REQUIREMENTS ANALYSIS: A BRIEF OVERVIEW	8
PURPOSE OF THE PROJECT	8
CURRENT SYSTEM	8
PROPOSED SYSTEM	8
FUNCTIONAL REQUIREMENT	8
NON-FUNCTIONAL REQUIREMENTS	9
OBJECT ORIENTED DESIGN	11
Inheritance	11
Encapsulation	11
Abstraction and Modularity	12
IMPLEMENTATION	13
PROJECT FEATURES	16
REFERENCES	21
APPENDIX	22

INTRODUCTION

DTUOversight is a University Management System (UMS) that is used to manage the data and records of the students and the teachers of Delhi Technological University (DTU). It also provides other relevant information about the University and has utilities like calculator and notepad.

In recent times due to Covid-19 everything has become virtual and having a unified system for college administrative purposes makes a lot of sense, we shall further elaborate this.

Some of the features offered by the System include:

- Registration
- Attendance Details
- Examination Details
- Fee Structure and Payment
- Other Utilities (Calculator, Notepad etc.)

Since the requirements of this project were specified in the beginning of the project, the **Waterfall Model** was used during this project. We also used the **Object Oriented Design** Approach during the course of this project.

MOTIVATION AND IMPORTANCE

Due to Covid 19 now more than ever due to dynamically changing times the need for a fast, efficient and more importantly contactless (i.e. virtual) system is more and more apparent.

Therefore we felt like this would be an excellent project to do as part of our coursework for this subject, because the problem is a real problem that needs to be addressed, and we feel that the core of software engineering is to enact real change for the better through software products and projects.

And this is what this project is, an attempt to make the lives of our college personnel both academic and administrative and the students more streamlined, to help eliminate redundancy and latency leading to a more pleasant college experience.

WATERFALL MODEL

The Waterfall model is the earliest Software Development Life Cycle model that was used for software development. It is also referred to as a **linear-sequential life cycle model**. It is very simple to understand and use.

The waterfall model divides the life cycle into a **set of phases**. This model considers that one phase can be started after completion of the previous phase. That is the output of one phase will be the input to the next phase. Thus the development process can be considered as a sequential flow in the waterfall. Here the phases do not overlap with each other.

The different sequential phases of the classical waterfall model are shown in the below figure:

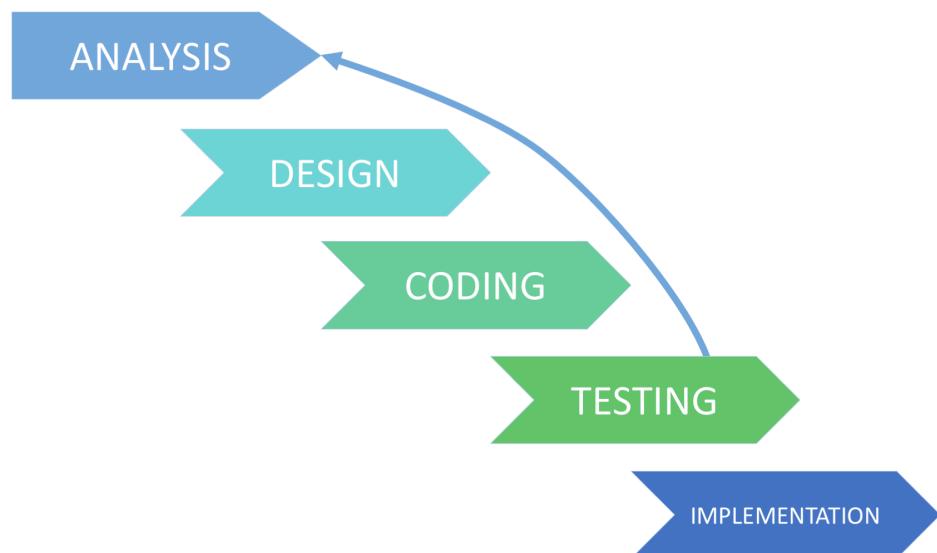


Fig1: A graphical representation of the Waterfall SDLC model

SEQUENTIAL PHASES IN WATERFALL MODEL

□ Requirement Gathering and analysis

All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.

□ System Design

The requirement specifications from the first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.

□ Implementation

With inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.

□ Integration and Testing

All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.

□ Deployment of system

Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.

□ Maintenance

There are some issues which come up in the client environment. To fix those issues, patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

REQUIREMENTS ANALYSIS: A BRIEF OVERVIEW

PURPOSE OF THE PROJECT

To provide a safe, efficient and virtual administrative panel for the management of DTU, facilitating a smooth student management process.

CURRENT SYSTEM

A traditional counter based system relying on physical presence for getting tasks done.

PROPOSED SYSTEM

An administrative system consisting of a backend server, with a frontend GUI based desktop application for the administrative staff of DTU.

FUNCTIONAL REQUIREMENT

Since the project centers around the administrative management of the student populace, the functional requirements have been structured as such, if new information comes to light as to the administrative protocols of the department, the requirements shall be updated to reflect that, incorporating the changes made into the project.

As of now the functional requirements are as follows:

- **Registration:** A utility to add new students to the college database for admissions purposes
- **Detail Management:** A system emulating a CRUD API for the purposes of Creating, Reading, Updating and Deleting Details.
 - Details Display
 - Updation of Details
 - Deletion of Details
- **Examination Details:** A utility to manage examination and academic related details of the students, like updating marks and calculating CGPA and SGPA
- **Attendance:** A utility to update and record student attendance subject wise
- **General Utilities:** Utilities like calculator and notepad for generic use purposes.

NON-FUNCTIONAL REQUIREMENTS

Non functional requirements are those which aren't concerned with what is being done but are more concerned with how it is being done, i.e. these are operational requirements that are concerned with smooth workflow and efficient software products.

The non functional requirements for this project are as follows:

- **Scalable:** The system should be scalable for short term loads, due to high traffic during admissions and subject selection
- **Reliable:** Student data is highly sensitive and subject to semi-regular updates, therefore the data should be secure, with backup tools like RAID in place for real time backup of data.
- **Little/No Downtime:** A lot of times, the administrative tasks follow time sensitive guidelines, in such cases if the system goes down, it leads to delays and possible mishaps.
- **Secure:** Student data contains a lot of personal information which is sensitive in nature and can be misused by malicious individuals, therefore the product must be secure.

OBJECT ORIENTED DESIGN

In the object-oriented design method, the system is viewed as a **collection of objects** (i.e., entities). The state is distributed among the objects, and each object handles its state data. The tasks defined for one purpose cannot refer or change data of other objects. Objects have their internal data which represent their state. Similar objects create a **class**. In other words, each object is a member of some class. Classes may inherit features from the **superclass**.

Inheritance

A core component of object oriented design, inheritance is the reuse of code components in a hierarchical fashion, allowing the inheriting class to reuse the code of the inherited class. More appropriately, inheritance is the passing down of component properties of certain pieces of code.

Here inheritance has been used in almost every class, to display almost every component the frame class has been inherited by multiple components in the project.

Encapsulation

Encapsulation or data hiding, is essential as it allows users to interact with data without modifying unless they have the suitable permission.

Here we have only given the necessary permissions to the administrative staff that they obtain after logging in.

Abstraction and Modularity

Abstraction refers to the hiding of the core component logic and only uncovering the functionality required, whereas modularity is the ability of the code to be reused independently as and when needed.

This has been employed throughout the code in the form of functions and classes, which are highly modular in nature, capable of being reused independently.

IMPLEMENTATION

The project had the following special features:

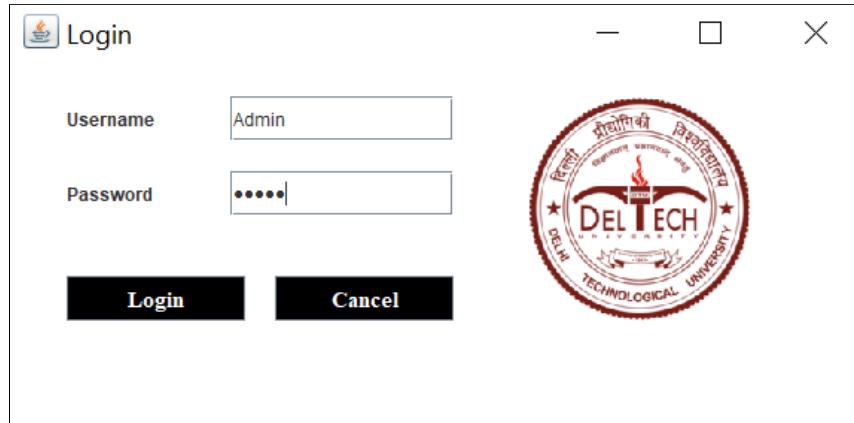
- The entire University Management System (UMS) is specific for a DTU ERP System. This includes the fee structure and the courses.
- All result features were modeled on the basis of our own 3rd year subjects.
- The entire Attendance Feature was planned and implemented by us.
- The UMS includes a feature that can open your Browser, Calculator and Notepad.

It was made using the following tech stack:

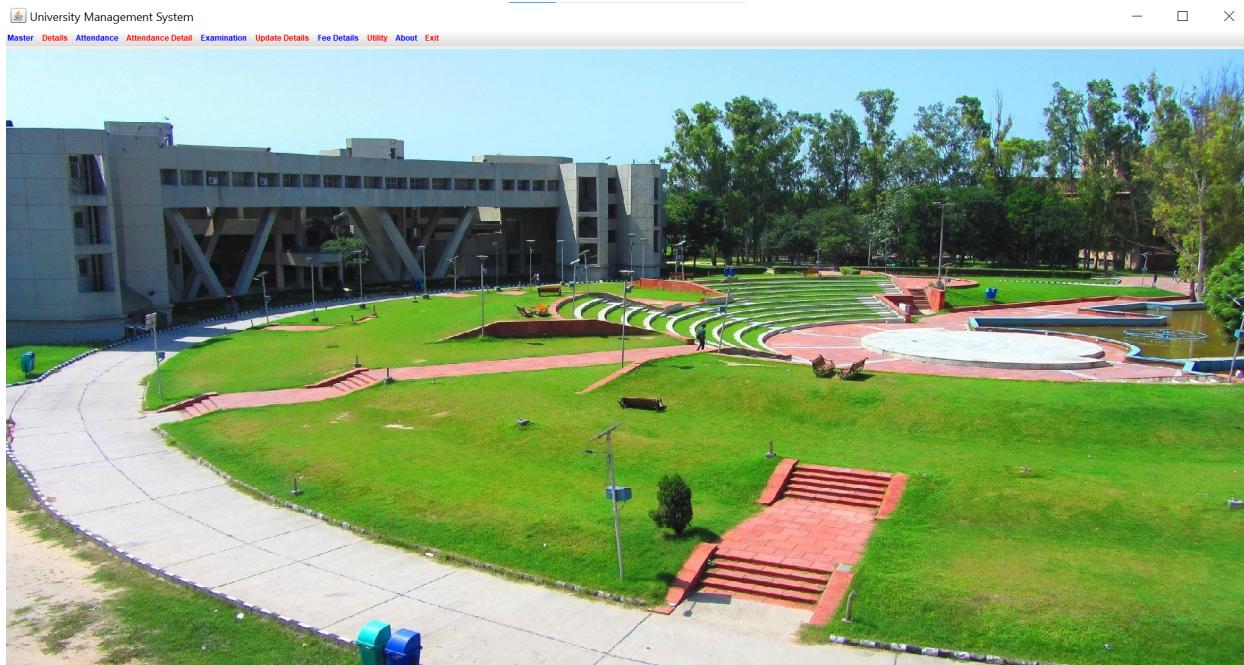
- Java
- Java Swing (GUI)
- MySQL (Backend)



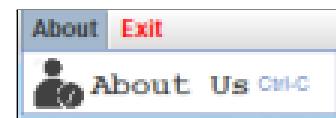
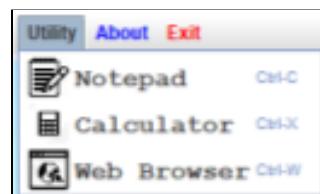
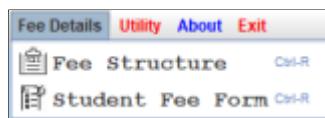
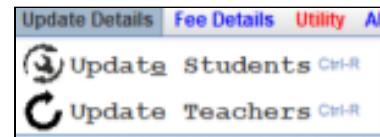
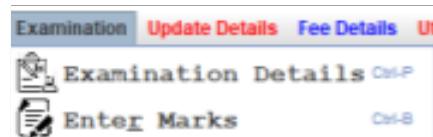
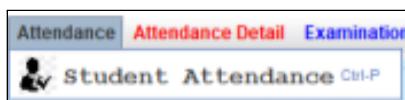
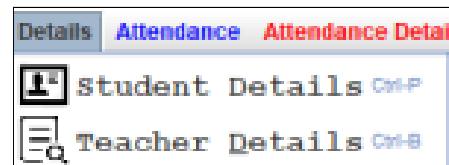
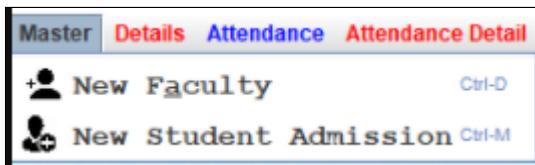
Before the main project can be accessed we need to login using a Username and Password which are verified in the database.



After this the main operating screen appears which provides the various features offered by the project.



Master Details Attendance Attendance Detail Examination Update Details Fee Details Utility About Exit



PROJECT FEATURES

The Project offers the following features:

- **Registration** - This includes registration of a new student or a new teacher. It has fields for Name, Father's Name, Age, Date of Birth, Address, Aadhar No, Phone No., Roll No. or Employee ID etc.

The image shows two windows side-by-side. The left window is titled 'Add Teacher' and has a header 'New Teacher Details'. It contains fields for Name, Father's Name, Age, DOB, Address, Phone, Email Id, Aadhar No, Designation (with a dropdown for Professor), Department (with a dropdown for COE), and Emp id. The right window is titled 'Add Student' and has a header 'New Student Details'. It contains fields for Name, Father's Name, Age, DOB, Address, Phone, Email Id, Class X(%), Class XII(%), Aadhar No, Roll No, Course (with a dropdown for B.Tech FT), Branch (with a dropdown for COE), and JEE Mains Rank. Both windows have 'Submit' and 'Cancel' buttons at the bottom.

The tables for the student and teacher

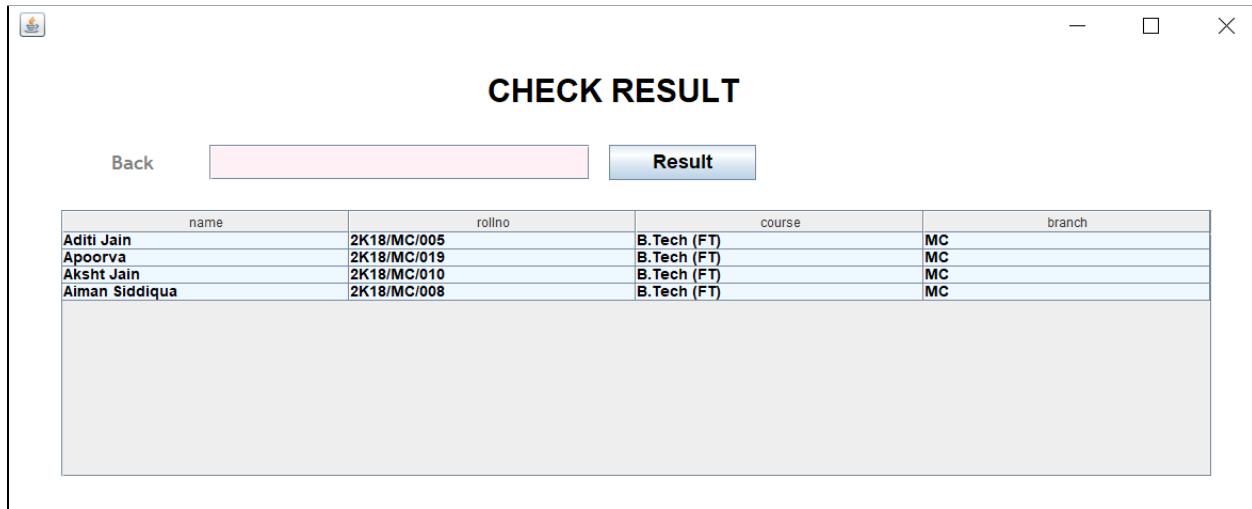
```
mysql> describe student;
+-----+-----+-----+-----+-----+
| Field | Type  | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| name  | varchar(28) | YES | NULL |          |       |
| fathers_name | varchar(28) | YES | NULL |          |       |
| age   | varchar(5)  | YES | NULL |          |       |
| dob   | varchar(20) | YES | NULL |          |       |
| address | varchar(50) | YES | NULL |          |       |
| phone  | varchar(15) | YES | NULL |          |       |
| email   | varchar(50) | YES | NULL |          |       |
| class_x | varchar(10) | YES | NULL |          |       |
| class_xii | varchar(10) | YES | NULL |          |       |
| aadhar | varchar(15) | YES | NULL |          |       |
| rollno | varchar(15) | NO  | PRI | NULL    |       |
| course  | varchar(15) | YES | NULL |          |       |
| branch  | varchar(20) | YES | NULL |          |       |
| jee_rank | varchar(7)  | YES | NULL |          |       |
+-----+-----+-----+-----+-----+
14 rows in set (2.25 sec)
```

```
mysql> describe teacher;
+-----+-----+-----+-----+-----+
| Field | Type  | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| name  | varchar(20) | YES | NULL |          |       |
| fathers_name | varchar(20) | YES | NULL |          |       |
| age   | varchar(5)  | YES | NULL |          |       |
| dob   | varchar(20) | YES | NULL |          |       |
| address | varchar(30) | YES | NULL |          |       |
| phone  | varchar(15) | YES | NULL |          |       |
| email   | varchar(25) | YES | NULL |          |       |
| aadhar | varchar(15) | YES | NULL |          |       |
| emp_id | varchar(15) | NO  | PRI | NULL    |       |
| dept   | varchar(20) | YES | NULL |          |       |
| designation | varchar(30) | YES | NULL |          |       |
+-----+-----+-----+-----+-----+
11 rows in set (0.13 sec)
```

- **Displaying Details** - It displays all the details of the student or the teacher. It also provides options for Adding new student/teacher details, Updating Details and Deleting a particular record.

- **Attendance** - It is used to update the attendance of students in a particular subject and display the attendance of all the students.

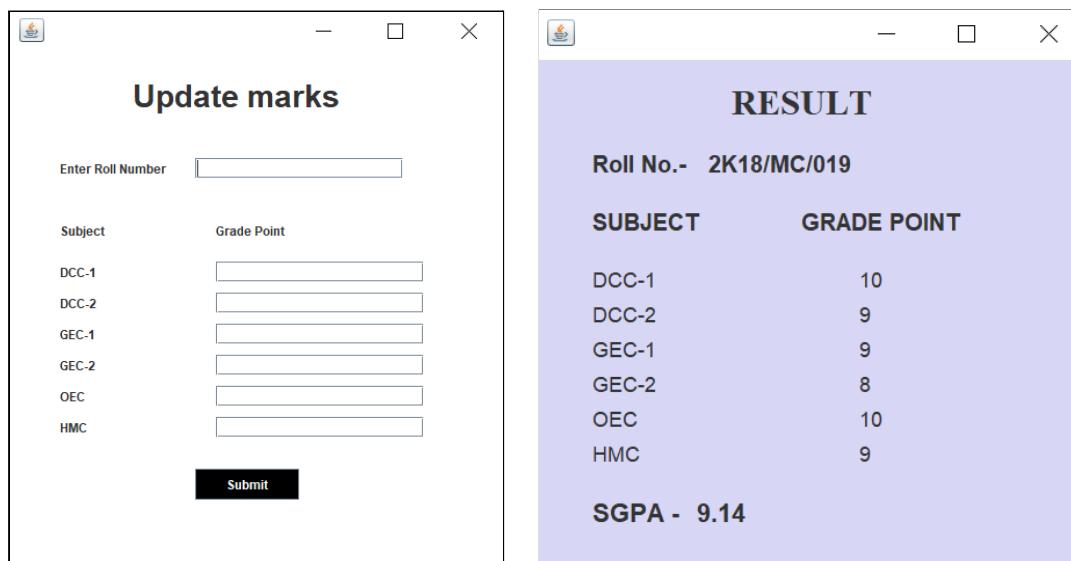
- **Examination** - It is used to enter marks of various subjects for a particular student and display the result. The SGPA for the result is calculated by the program as per DTU credit system.



CHECK RESULT

Back **Result**

name	rollno	course	branch
Aditi Jain	2K18/MC/005	B.Tech (FT)	MC
Apoorva	2K18/MC/019	B.Tech (FT)	MC
Aksht Jain	2K18/MC/010	B.Tech (FT)	MC
Aiman Siddiqua	2K18/MC/008	B.Tech (FT)	MC



Update marks

Enter Roll Number

Subject	Grade Point
DCC-1	<input type="text"/>
DCC-2	<input type="text"/>
GEC-1	<input type="text"/>
GEC-2	<input type="text"/>
OEC	<input type="text"/>
HMC	<input type="text"/>

RESULT

Roll No.- 2K18/MC/019

SUBJECT	GRADE POINT
DCC-1	10
DCC-2	9
GEC-1	9
GEC-2	8
OEC	10
HMC	9

SGPA - 9.14

- **Update Details** - It is used to update the details of an existing student or teacher using the Roll No. or Employee ID.

Update Student details

Update Student Details:

Enter roll number to update the data of student

Name	<input type="text"/>	Father's Name	<input type="text"/>
Age	<input type="text"/>	DOB (dd/mm/yyyy)	<input type="text"/>
Address	<input type="text"/>	Phone	<input type="text"/>
Email Id	<input type="text"/>	Class X(%)	<input type="text"/>
Class XII(%)	<input type="text"/>	Aadhar No	<input type="text"/>
Roll No	<input type="text"/>	Course	<input type="text"/>
Branch	<input type="text"/>	Jee Mains Rank	<input type="text"/>

Update Teacher details

Update Teacher Details:

Enter employee id to update the data of teacher

Name	<input type="text"/>	Father's Name	<input type="text"/>
Age	<input type="text"/>	DOB (dd/mm/yyyy)	<input type="text"/>
Address	<input type="text"/>	Phone	<input type="text"/>
Email Id	<input type="text"/>	Aadhar No	<input type="text"/>
Employee Id	<input type="text"/>	Designation	<input type="text"/>
Department	<input type="text"/>		

- **Fee Structure and Payment** -The Fee Structure displays the fees for the various courses offered by DTU (B.Tech, M.Tech etc..) for different years.

Student Fee Form

Fee-Form

Select Roll No	<input type="text" value="2K18/MC/008"/>
Name	<input type="text" value="Aiman Siddiqua"/>
Father's Name	<input type="text" value="Nematullah"/>
Course	<input type="text" value="B.Tech (FT)"/>
Branch	<input type="text" value="MC"/>
Year	<input type="text" value="3rd"/>
Total Payable	<input type="text" value="190000"/>

```
mysql> describe fees;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| RollNo | varchar(15) | NO | PRI | NULL |
| Name | varchar(25) | YES | | NULL |
| Fathers_Name | varchar(25) | YES | | NULL |
| Course | varchar(15) | YES | | NULL |
| Branch | varchar(5) | YES | | NULL |
| Year | int(11) | YES | | NULL |
| Paid | decimal(10,0) | NO | | NULL |
+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)
```

FEE STRUCTURE

Course	B.Tech (FT)	B.Tech (LE)	B.Tech Evening	B.Des	BBA	M.Tech (FT)	MBA
AY 19-20	Rs 166000	Rs 166000	Rs 130500	Rs 166000	Rs 82000	Rs 166000	Rs 166000
AY 20-21	Rs 190000	Rs 190000	Rs 140000	Rs 190000	Rs 90000	Rs 166000	Rs 190000
AY 21-22	Rs 206000	Rs 206000	Rs 148500	Rs 206000	Rs 93500		
AY 22-23	Rs 219000		Rs 150000	Rs 219000			

- **About Us** - It displays the Name and Roll No. of the developers of this project along with other relevant information.



- **Utilities** - It includes facilities to open Calculator, Notepad and Chrome.
- **Exit** - It is used to exit the system.

REFERENCES

- https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm
- <https://www.geeksforgeeks.org/software-engineering-classical-waterfall-model/>
- <https://www.javatpoint.com/software-engineering-object-oriented-design>
- <https://www.cs.fsu.edu/~lacher/courses/COP3331/rad.html>
- <https://docs.oracle.com/javase/tutorial/>
- <https://docs.oracle.com/javase/7/docs/api/javax/swing/package-summary.html>
- <https://dev.mysql.com/doc/>
- <https://docs.phalcon.io/3.4/en/webserver-wamp>
- **Software Engineering KK Aggarwal Notes**

APPENDIX

The project was developed using the **Apache Netbeans** development environment. **Wampserver** was used to integrate the backend of the project using MySQL. **Java Database Connectivity (JDBC)**, a Java API, was used to connect and execute queries with the database.

The source code for the project can be found on the following link:

<https://github.com/AMAN-SIDDIQUA/DTUOversight>