**Mini Project**

**ICT2132 – Object Oriented Programming Practicum**

by

TG/2020/674 – Priyamantha MPC

TG/2020/680 – Bandara KDT

TG/2020/694 – Karunarathne HARN

TG/2020/706 – Dissanayake KGS

**Lecturer in charge: Mr. P.H.P.Nuwan Laksiri**

Submission Date: 19th March 2023

Bachelor of Information and Communication Technology

Department of Information and Communication Technology

Faculty of Technology

University of Ruhuna.



Table of Contents

[1.Abstract……………………………………………………………………………………...……..3](#_Toc127727003)

[2.Introduction to University Student Management Information System……………………….……3](#_Toc127727004)

[3.Functional Requirments……………………………………………………………………………4](#_Toc127727005)

[4.Non-Functional Requirments………………………………………………………………………5](#_Toc127727009)

5.Feasibility Analysis………………………………………………………………………...………7

[6.System Analysis……………………………………………………………………………………8](#_Toc127727011)

[7. Database Design…………………………………………………………………………………...9](#_Toc127727013)

[8. Interface Design………………………………………………………………………………….1](#_Toc127727014)1

9.System Design…………………………………………………………………………………….23

10.Problems and solutions…………………………………………………………………………..24

11. Conclusion…………………………………………………………………………...………….25

12. Team member…………………………………………………………………………...………26

# **Abstract**

As the mini project of ICT2132 – Object Oriented programming practicum module, we have Developed a fully functional University Student Management Information System(USMIS) to manage the activities of a Faculty of Technology, University of Ruhuna. This project report explains the developed process, system structure and methodologies and other important importation of Mini Project.

# **Introduction to University Student Management Information System**

A University Student Management Information System (USMIS) is a software application used by universities to manage student-related data, including enrollment, grades, attendance, and other academic and administrative information. The system serves as a centralized database that allows lecturers, administrators, technical officers, and student to access and update student records in real-time, making it easier to manage and monitor student performance and progress. USMIS can help university to streamline their operations, improve communication, and enhance student learning and achievement.

No one cannot deny the importance of University Student Management Information System (USMIS) in daily life as undergraduates as well as academic staff since it handles their activities very effectively and efficiently. When these all the activities by manual methods, it is very time consuming and costly.

There are mainly 4 users in this system.

* Admin
* Lecturer
* Student
* Technical Officer

There are several modules in this program.

* User profiles (Admin, Lecturer, Student, Technical officer)
* Course Details module
* Student Marks module
* Student Attendance Module
* Notice Module
* Timetables module
* Medical details module
* Department details module

# **Functional Requirements**

**Admin Management Requirements**

* + Create and maintain all user profiles.
  + Create and maintain courses.
  + Create and maintain notices.
  + Create and maintain timetables.

**Lecturer Management requirements**

* + Lecturer Can update their profile except username and password.
  + Lecturer Can modify and add materials to courses.
  + Lecturer Can upload marks for all kind of exams.
  + Lecturer Can see student details.
  + Lecturer Can see student eligibility.
  + Lecturer Can see student marks, grades, and GPA.
  + Lecturer Can see attendance and medical records of students.
  + Lecturer Can see notices.

**Student management**

* Can update only contact details and profile picture of their profile.
* Can see attendance details.
* Can see medical details.
* Can see their course details.
* Can see their grades and GPA.
* Can see their timetables.
* Can see notices.

**Technical Officer Management requirements**

* + Can update their profile except username and password.
  + Can add and maintain attendance details of students.
  + Can add and maintain medical details of students.
  + Can see notices.
  + Can see timetables of their department.

# **Non-Functional Requirements**

* + - **Security Requirements**:

When it comes to the USIMS, security factor should have a major concern. First, we have to look on to giving access to the system. That access privilege allows to change the system data, add and remove data in the system. We have to maintain one Admin to do these changes. That admin has access to all functions in the system. Then the lectures and technical officers. We have to give them some limited access to relating to their job role. Lastly students, they need to have access only on viewing, uploading, and downloading materials.

After the distribution of access, we need to limit the access by implementing a login form. Login can prevent unauthorized access to the system. Then, only relevant people can be access to the system.

* **Performance Requirements:**

Non-functional performance requirements for a USIMS relate to the system's speed and responsiveness. These requirements ensure that the USIMS can handle its users' expectations and provide a positive user experience. The following are some examples of common non-functional performance requirements for an USIMS:

**Response time:** The USIMS should respond quickly to user requests such as page loading, form submissions, and playing media.

**Load time:** Even in high-traffic situations, the USIMS should load pages and media efficiently.

**Data storage:** The USIMS should be able to efficiently and effectively store and manage large amounts of data, such as student records and course materials.

audio without buffering or lag.

* **Portability Requirements:**

**Operating System Compatibility:** The system must be compatible with the latest Windows, Mac OS, and Linux versions.

**Platform Independence:** The system should be platform-independent and run smoothly on a variety of hardware configurations.

**Code Portability:** The USIMS designed with open-source technology and programming languages that follow industry standards. The system should be designed in a modular way that allows for easy integration with other applications.

**Data Interoperability:** The USIMS should support the import and export of data in standard formats.

* **Usability Requirements**

**User friendly:** User-friendly interface as such as clean, intuitive, and simple interface for easy navigation to use.

**Reporting and analytics:** USIMS provide meaningful and actionable data and analytics on student progress and performance.

* **Manageability Support Requirement**

Make sure the USIMS supports the file and learning formats and data collection capabilities user needs.

Docs

PDFs

Standard image formats (png, jpg, gif)

* **External interface requirements**

When we have to build the connection between software and the user, we need to implement a user interface. Using that interface, the user can do what they desire simply. This includes screen layouts, buttons, functions on screen.

1. **Feasibility Analysis.**

A University Student Information Management System (USIMS) is a software application designed to manage and organize student information, including personal details, academic performance, attendance records, and more. A feasibility analysis of a SIMS involves determining whether it is possible and practical to develop and implement such a system.

Here are some factors to consider in a feasibility analysis of a USIMS:

**Technical Feasibility**: The first consideration is whether it is technically feasible to develop a USIMS. This involves evaluating the resources required, such as hardware, software, and personnel. It is essential to ensure that the USIMS can handle the expected number of users and data without crashing or slowing down.

**Technologies use in Project**:

* As the main language, JAVA Runtime Environment
* As database management systems, MySQL
* Web servers: Apache
* IDE, Apache NetBeans
* Collaborative tool: GitHub

1. **System Analysis**

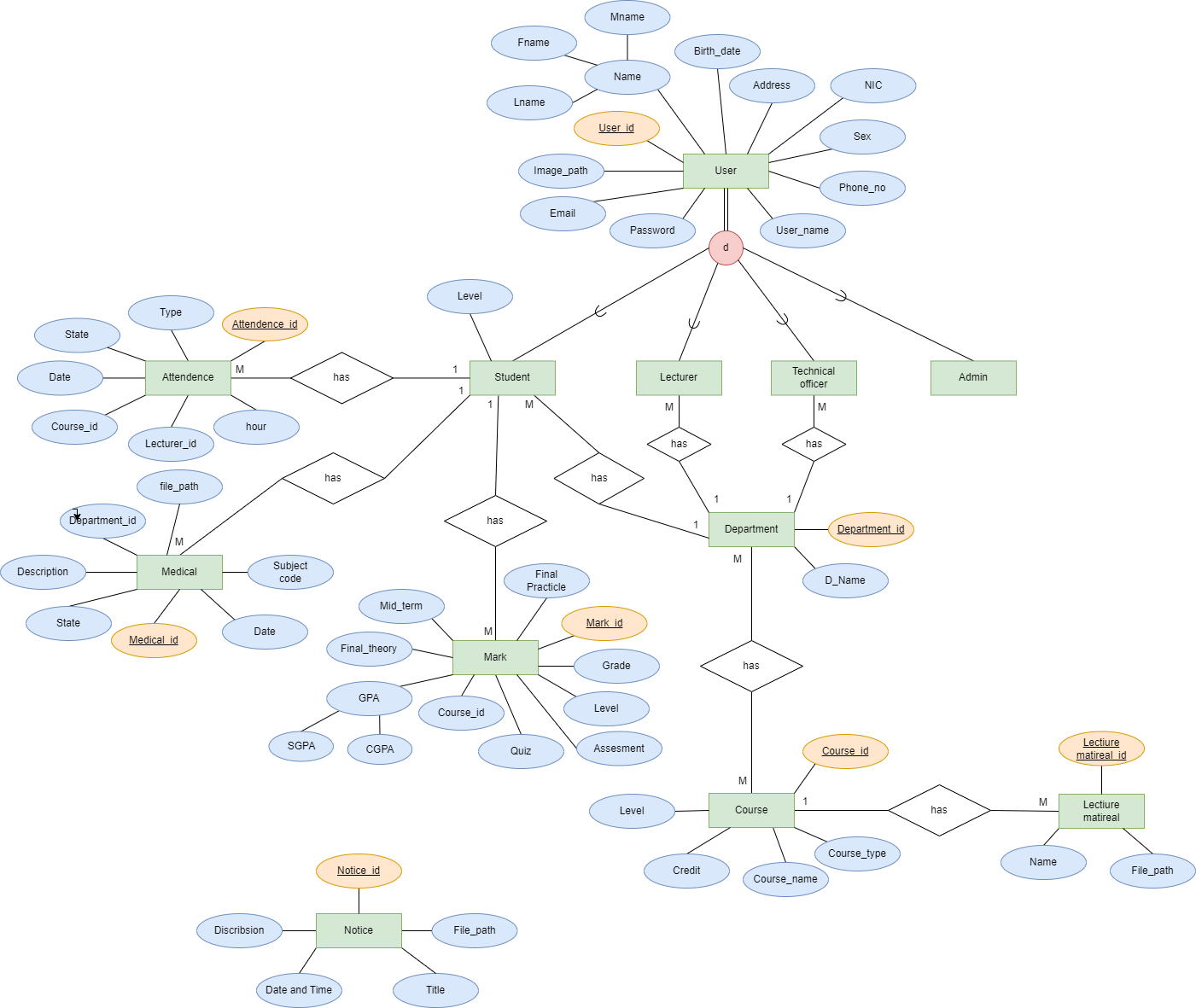
**UML class diagram**

Diagram

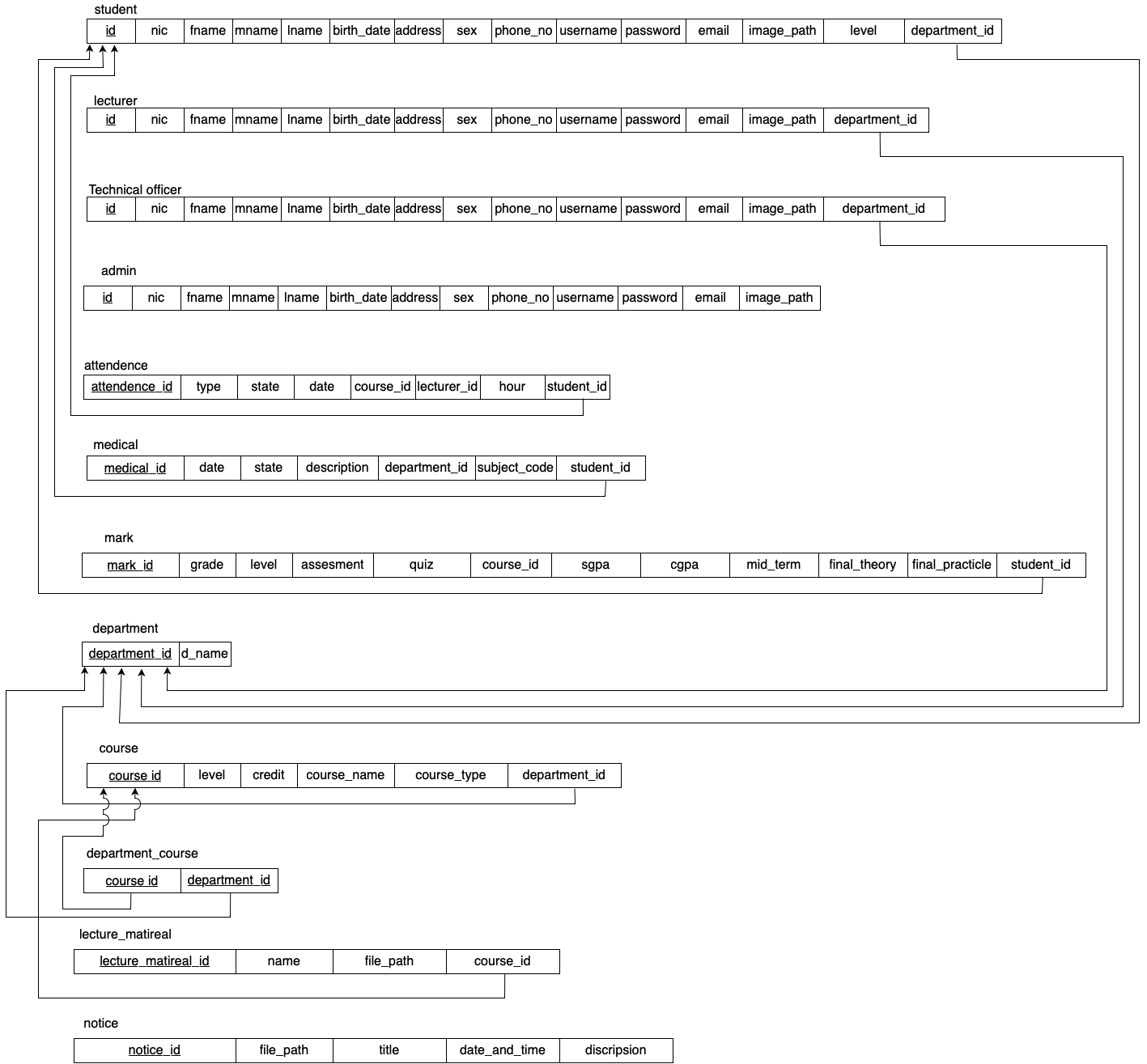
Description automatically generated

1. **Database Designing**

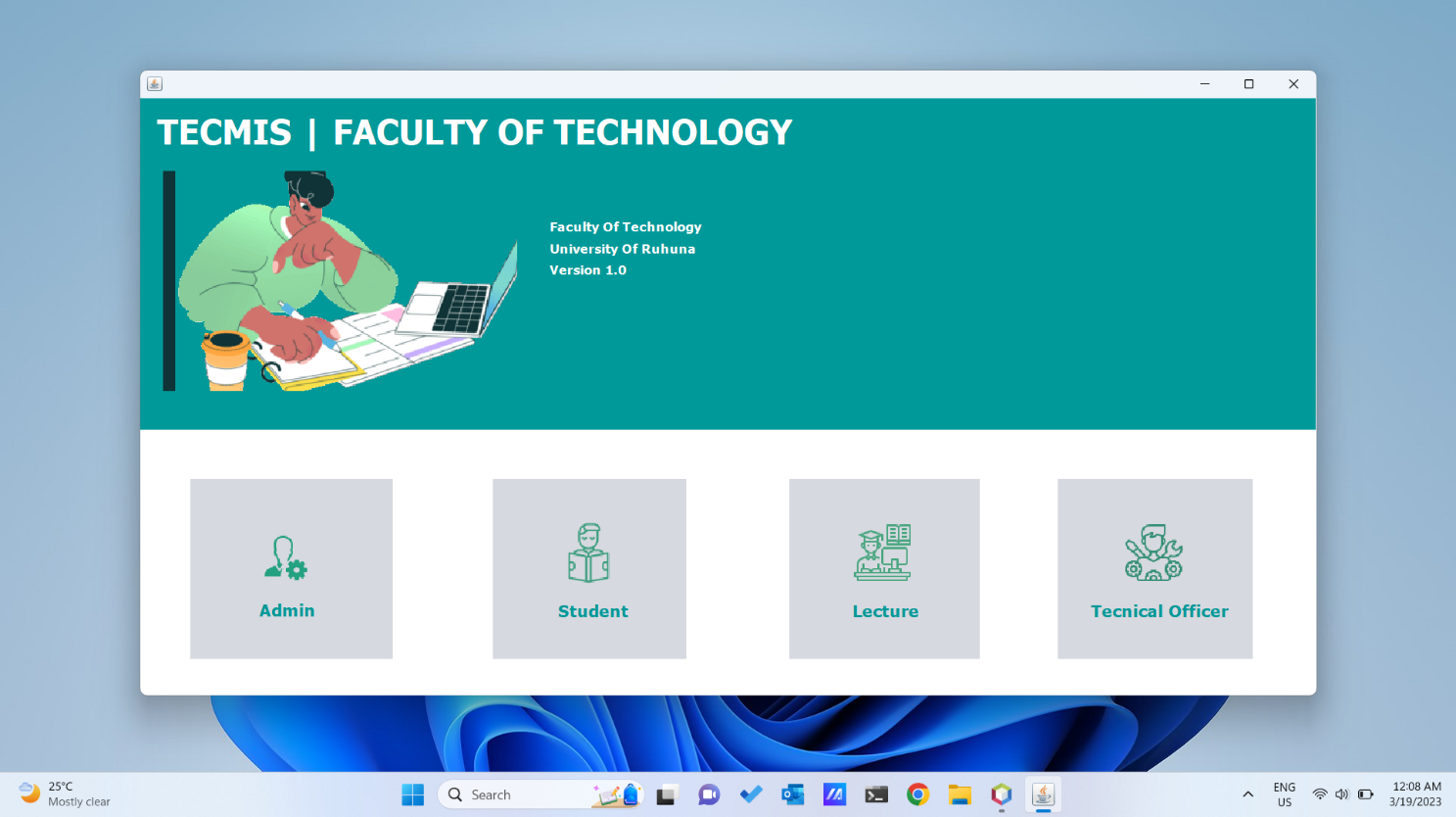
**Entity Relationship Model**



**Relational Schema**

****

1. **User interfaces**
2. Loading interface of the system

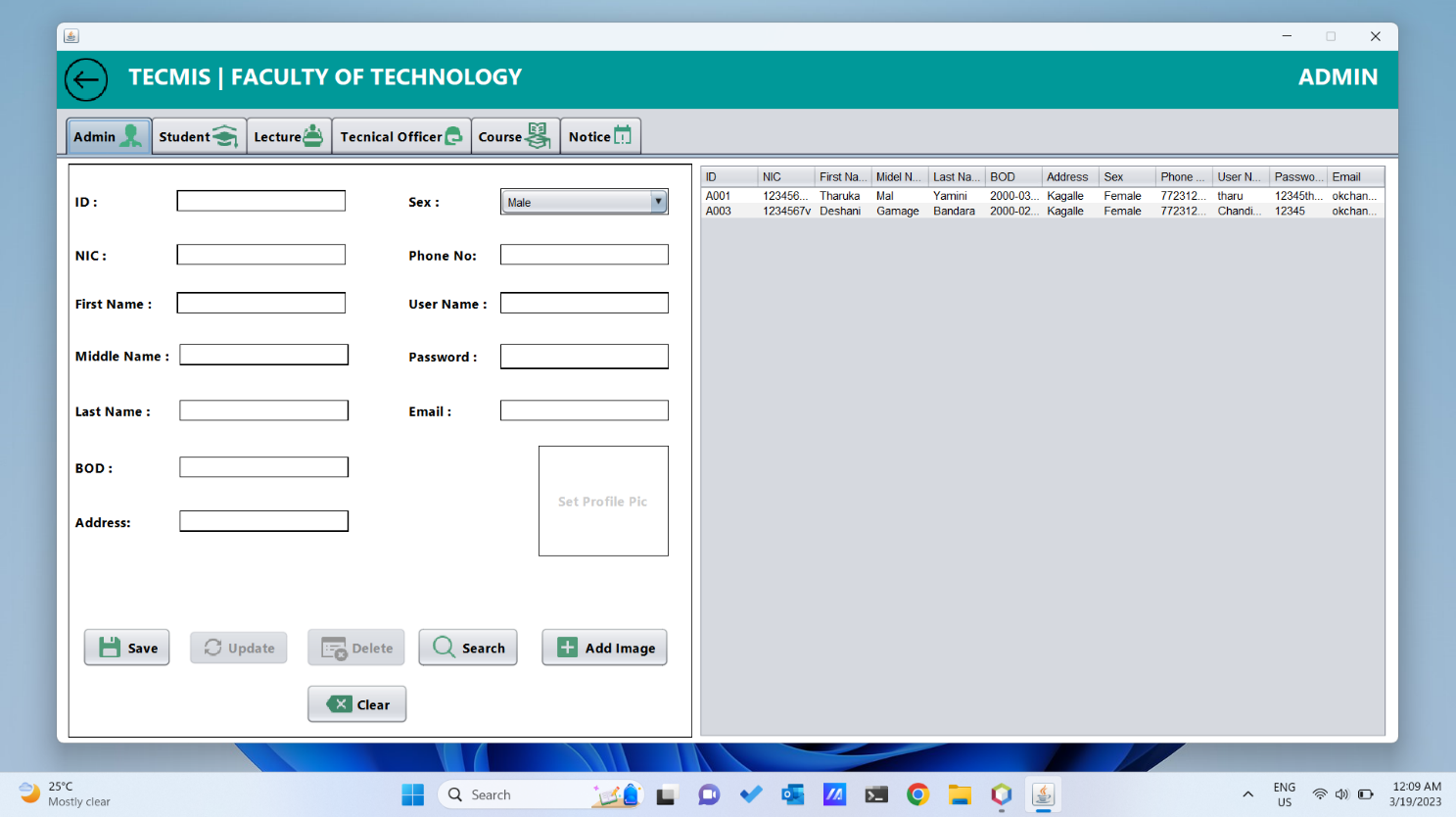


1. Login interface

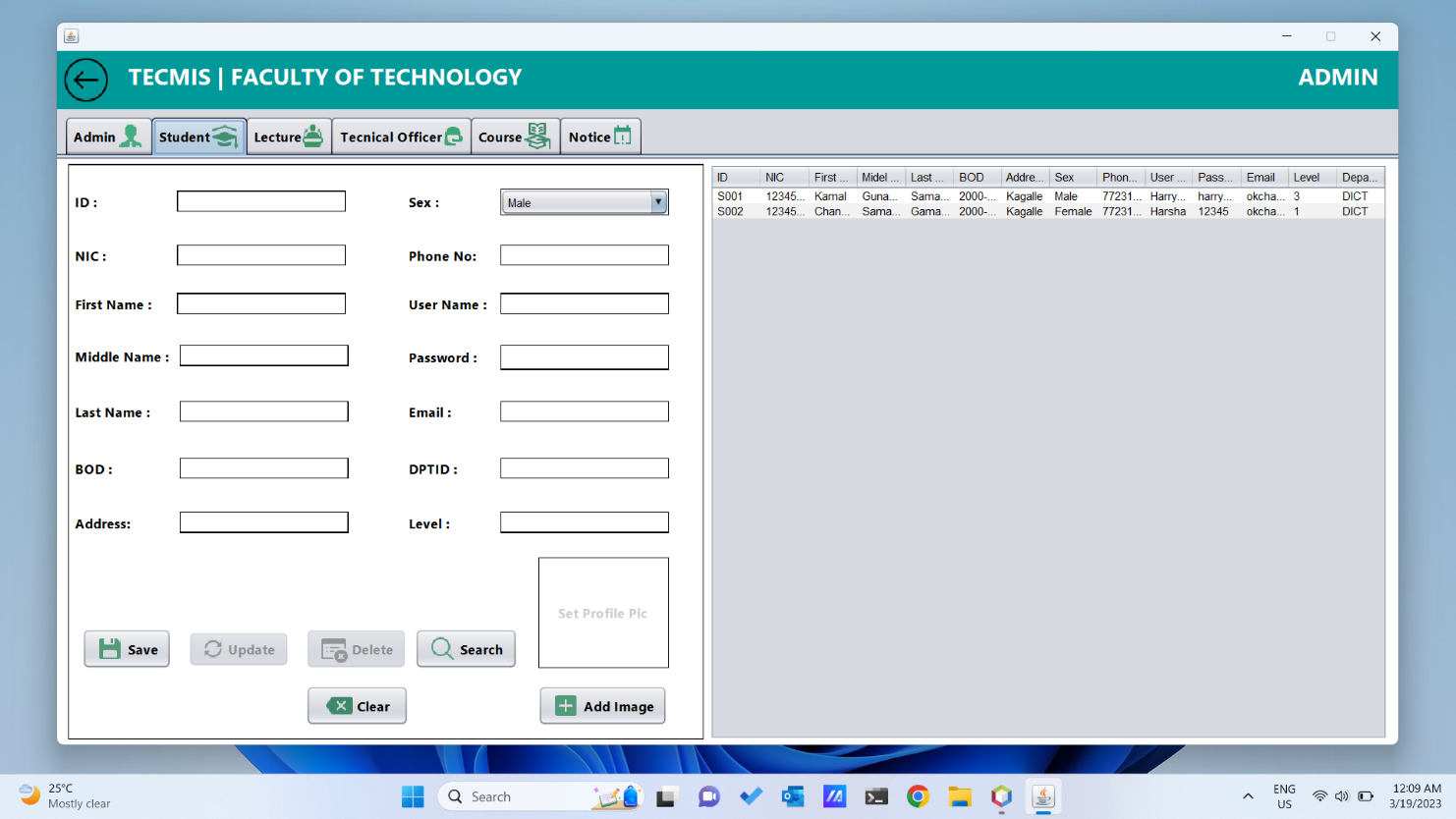
Graphical user interface, application

Description automatically generated

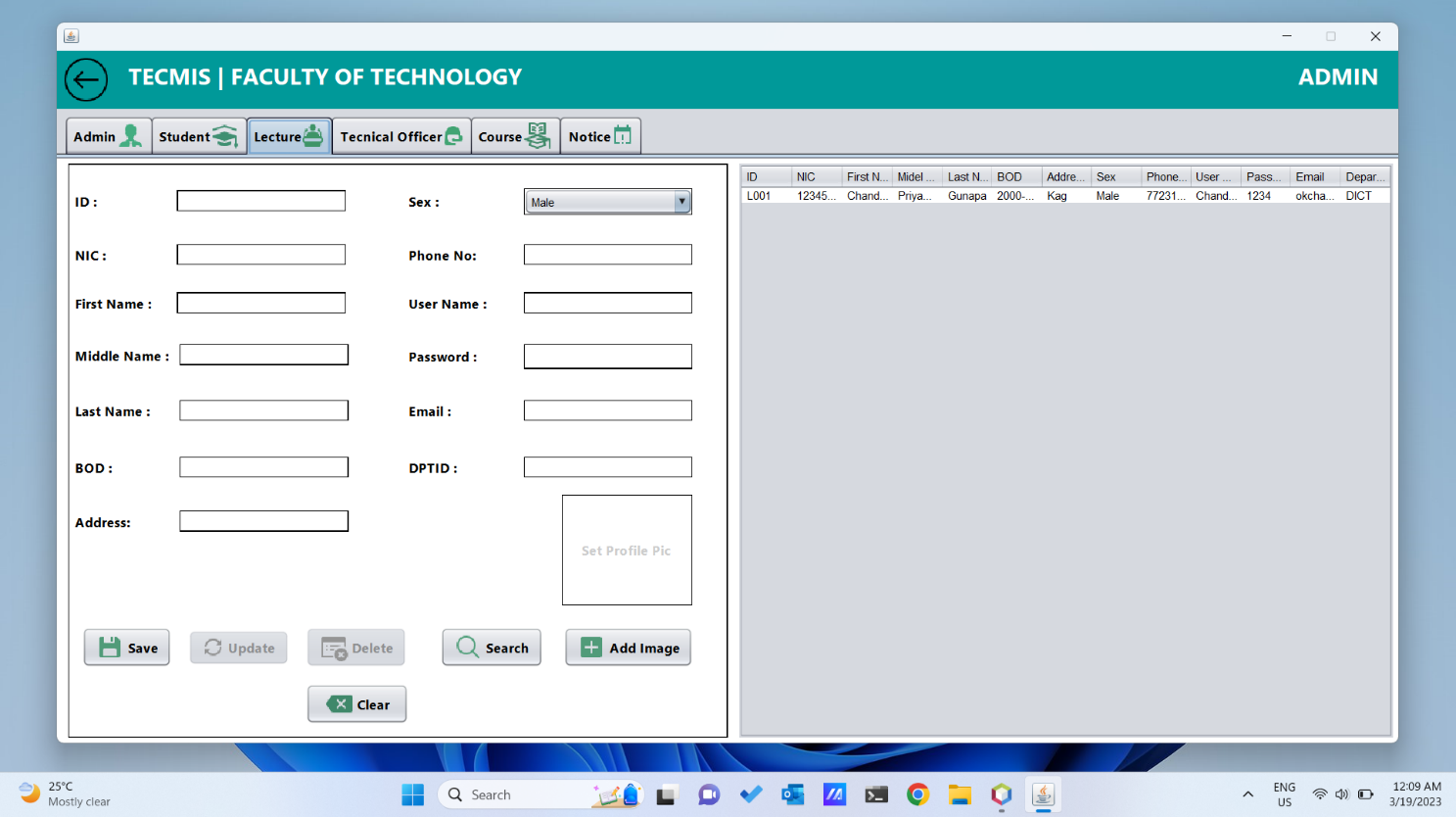
1. **Create and maintain all user profile (Admin)**



1. **Create and maintain all user profile (Student)**

****

1. **Create and maintain all user profile (Lecturer)**



1. **Create and maintain all user profile (Technical Officer)**

Graphical user interface, application

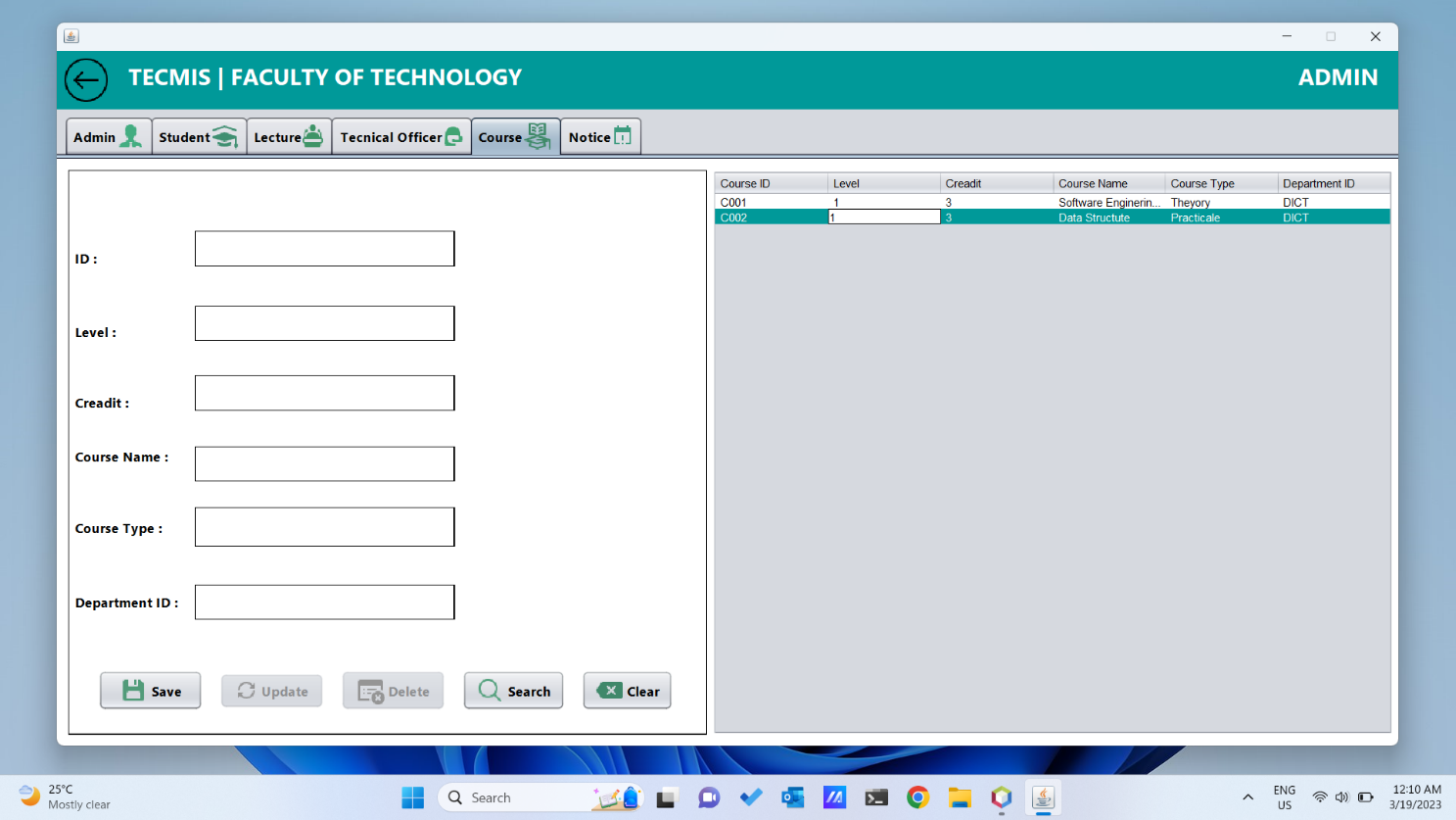
Description automatically generated

1. **Admin can maintain notices.**

Graphical user interface

Description automatically generated

1. **Admin can maintain course details.**



1. Can update their profile except username and password (Lecturer)

Graphical user interface

Description automatically generated

1. Can see student details. (Lecturer)

Graphical user interface

Description automatically generated

1. Can see attendance and medical records of students. (Lecturer)

Graphical user interface

Description automatically generated

1. Can see notices. (Lecturer)

Graphical user interface

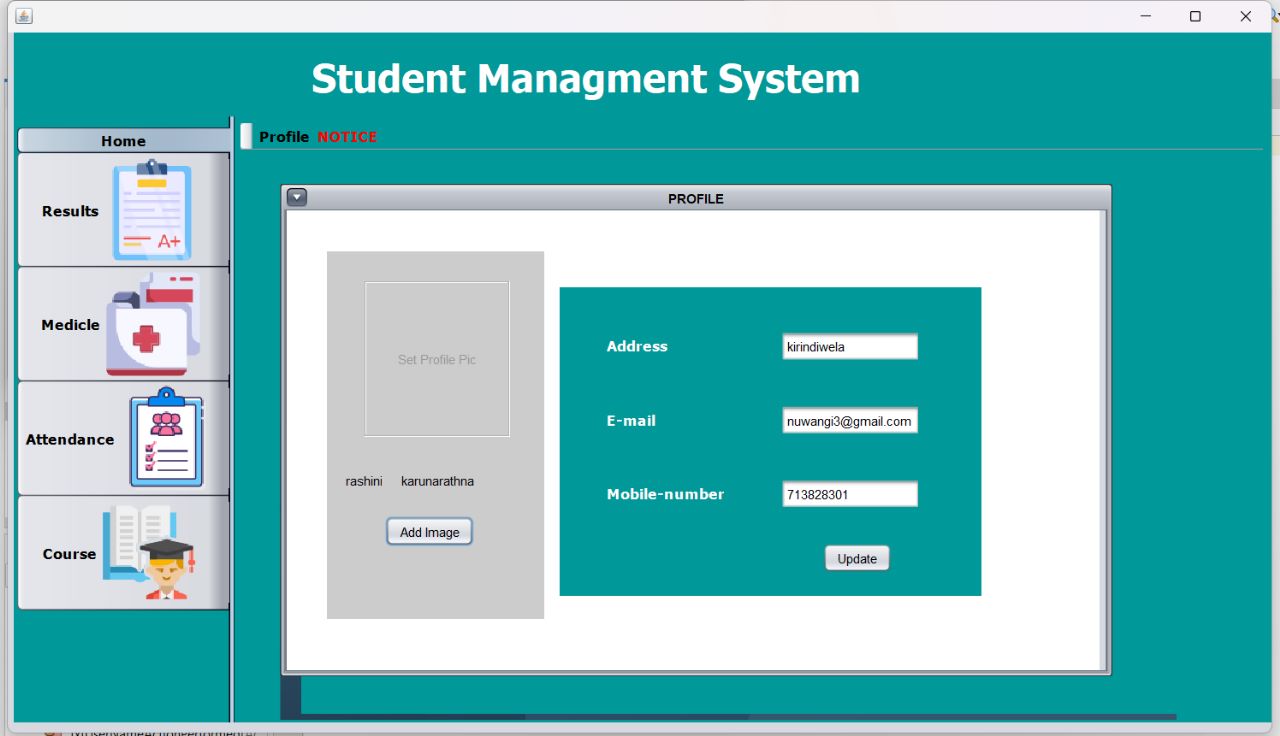
Description automatically generated

1. Student Dashboard

Graphical user interface, website

Description automatically generated

1. Can update only contact details and profile picture of their profile(Student)

****

1. Can see attendance details (Student)

Graphical user interface

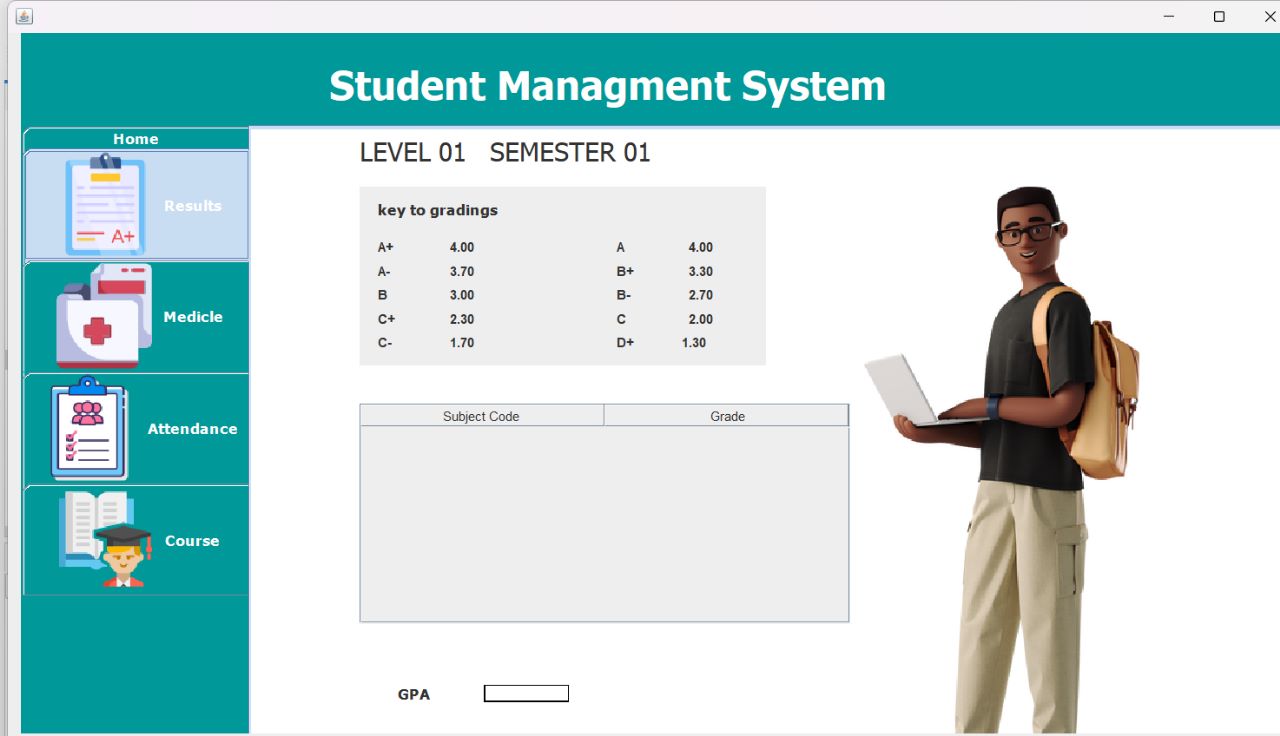
Description automatically generated

1. Can see medical details (Student)

Graphical user interface, application

Description automatically generated

1. Can see their grades and GPA (Student)



1. Can see Course material. (Student)

Graphical user interface, application

Description automatically generated

1. **Technical officer dashboard**

Graphical user interface, diagram

Description automatically generated

1. **Technical officer login form**

Graphical user interface, application, website

Description automatically generated

1. Can update their profile except username and password (technical officer)

Graphical user interface

Description automatically generated

1. Can add and maintain attendance details of students (technical officer)

Graphical user interface, application

Description automatically generated

1. Can add and maintain medical details of students (technical officer)

Graphical user interface

Description automatically generated

1. Can see notices (technical officer)Graphical user interface

   Description automatically generated
2. **System Design**

Used Technologies

* Java 19
* MYSQL Workbench
* WAMP server
* phpMyAdmin 5.0.2
* NetBeans 17 IDE
* GITHUB Version Control System

1. **Problem and solution**
2. **Technical Difficulties**: Java programming can be complex and challenging, especially for students who are new to the language. Technical difficulties can arise, including coding errors and compatibility issues.

**Solution**: To address technical difficulties, teams can establish coding standards and best practices. Team members can also work collaboratively to identify and troubleshoot technical issues. Consider assigning a team member to serve as a technical lead, who can provide guidance and support as needed.

1. **Time Management**: Group projects can be time-consuming, and team members may struggle to balance their project work with other commitments.

**Solution:** To manage time effectively, teams can establish a clear timeline for the project and create a schedule that outlines each team member's tasks and deadlines. Encourage team members to set aside dedicated time for project work and to communicate any scheduling conflicts as early as possible.

1. **Communication Issues**: One of the biggest problems that can arise during Java group projects is a lack of communication. Team members may not be able to communicate effectively due to language barriers, different time zones, or other factors.

**Solution:** To avoid communication issues, it is essential to establish clear lines of communication early on in the project. Teams can use collaboration tools like Slack, Trello, or Zoom to communicate and share ideas. Regular meetings and check-ins can help ensure that everyone is on the same page.

1. **Database connection between team members:**
2. **Conclusion**

As the mini project of ICT2132 – Object Oriented programming practicum module, we have Developed a Fully functional Student Management System to manage the activities of a Faculty of Technology. In this report has explained the About the system, Functional Requirements, feasibility analysis, System Design, Database Design, Program Modules, User Interfaces, Problems arrived, and how we managed them properly.

1. **Team members**

As a team we discussed ER diagram and UML diagram for University Student Information Management System and followings are the individual contributions of our team members.

|  |  |
| --- | --- |
| Team member | Individual contribution |
| TG/2020/674 -Priyamantha MPC | Loading Screen  Login Panel  Admin Dashboard  All functionalities of Admin (  Create and maintain all User Profiles, Create and maintain Courses,  Create and maintain Notices, Create and maintain Timetables) |
| TG/2020/680-Bandara KDT | Functionalities of Lecturer  (Update lecturer details  View Student Details  Maintain course materials.  Maintain notices) |
| TG/2020/694- Karunarathne HARN | Edit student Profile.  Functionalities of student (view notices, course details, marks, attendance, medical) |
| TG/2020/706 -Dissanayake KGS | Edit Technical officer Profile.  Technical officer dashboard  Technical officer Functionalities (view notices and timetables, add and maintain attendance details, create and maintain medical details)  Creating DataBase and Relational Schema |

\*\*\* The End \*\*\*