PROJECT SPECIFICATION

INS2073

A. Topic Requirements

1. Student Management System

- *Classes and Relationships:* Define classes for ***Student, Course,*** and ***Enrollment***. Relationships should clearly indicate how students can enroll in multiple courses.
- *Basic Algorithms:* Implement algorithms for searching (by student ID, name, course), inserting (new student records), updating (student details, course enrollment), and deleting (student records, course enrollments) in arrays or linked lists.

2. Library Management System

- *Classes and Relationships:* Include classes for ***Book, Author, Patron,*** and ***Loan***. Detail relationships to show how books can be loaned to patrons.
- *Functionality:* Ensure features for adding new books/authors/patrons, removing them, updating their details, and searching (by book title, author, patron ID) are present.

3. Hospital Management System

- *Classes and Relationships:* Develop classes for ***Patient, Doctor, Appointment,*** and ***Treatment***. Illustrate how patients are assigned to doctors and receive treatments.
- *Operations:* Implement add, remove, update, and search functionalities for patient records, doctor assignments, and treatment records.

4. Staff Management System

- *Classes and Relationships:* Create classes for ***Employee, Department***, and ***Schedule***. Describe the association between employees and departments, and how schedules are managed.
- *Core Features:* Include capabilities to add, delete, update, and search for employee records, department details, and scheduling information.

5. Warehouse Management System

- *Classes and Relationships:* Formulate classes for ***Item, Warehouse***, and ***Inventory***. Show how items are stored in different warehouses and managed through an inventory system.

- *Functional Requirements:* Facilitate adding, removing, updating, and searching for items, warehouse locations, and inventory levels.

B. Report Requirements

1. Member Contribution

- *Roles and Responsibilities:* Clearly define the roles and responsibilities of each team member involved in the project. This could include who was responsible for design, implementation, testing, documentation, etc.
- *Contribution Overview:* Provide a summary of each member's contributions to the project. This might include specific features or components they worked on, challenges they addressed, or innovative solutions they implemented.

2. Program Overview

- *Objectives:* Define the specific goals or outcomes that the project aims to achieve.
- *Technologies:* Introduce tool, programming language and other technical means used to build and implement the project

3. Program Design

- *Architecture:* Present the overall architecture of your system, including major components and their interactions.
- *Class Diagrams:* Provide UML class diagrams showing classes, relationships, and key methods.

4. Program Implementation

- *Coding Standards:* Detail the coding standards and conventions followed (e.g., naming conventions, commenting guidelines).
- *Key Implementations:* Highlight the implementation of critical features such as the algorithms for adding, removing, updating, and searching records. Include code snippets where relevant.
- *File Handling:* Explain how files are used for record storage, including file format, read/write operations, and error handling.
- *Program Screenshots:* Attach some screenshots of console application

5. Program Evaluation

- *Advantages/Strengths:* List some outstanding features of the system

- *Disadvantages/Weaknesses:* Clarify incomplete features, remained errors
- *Future/Possible improvements:* Suggest improvements for a better version in future

C. Project Requirements

- Develop a console application using C++ language
- Define all the necessary classes and their relationships such as abstract classes, interfaces, etc.
- Use file to store records
- Students are expected to write the code into classes
- Students are required to master coding skills for classes, methods, arrays, file, etc.
- ***Copying code is strictly prohibited***
- All students must submit their work in a ZIP file, including complete codes and comprehensive report covering the above sections, to Microsoft Teams

D. Presentation Requirements

- Summarize your work using MS PowerPoint
- Each group has about 10-15 minutes to present and demo project
- All team members must be prepared to answer questions given by lecturers