**Introduction**

This assessment comprises enhancing an existing codebase to create an application that addresses a practical problem. It will assess the student’s understanding of software development and implement relational database and the ability to document the development process.

The objective is to enhance this codebase by incorporating supplementary features and functionalities that contribute to the overall functionality and usability of the application

**Assignment Description:**

This assessment task focuses on the process of improving and refining an existing codebase that has been provided. The codebase serves as the ground work for a school management application. The application will be developed using an OOP (Object Oriented Programming) Language (Python) that supports CRUD (Create, Read, Update, and Delete) operations.

1. **Artefact [50 Marks]**

Student will be provided with codebase for a School Management Application. The student task is to perform the following steps by utilising the provided code to:

* Create an application in an OOP (Object Oriented Programming) Language (Python) that supports CRUD (Create, Read, Update, and Delete) operations for each of the database tables using Stored Procedures.
* Adding more features and functionalities to align with a problem for a given scenario.
* Build a simple database using a DBMS (DataBase Management System) application such as MSSQL, MySQL, and Oracle. The application must successfully include a connected database that host tables.
* Understand the design of your application with the help of ERD (Entity Relational Diagrams), Conceptual, Logical and physical diagrams for the completed database.
* The database need to be normalised to 3NF (Normal Form), at minimum.
* Your application should use sorting and searching algorithms to filter and sort loaded data.
* Software Development Life Cycles (SDLC)
* Requirement Elicitation and Engineering
* Object Oriented Programming (OOP) Principles
* Programming Constructs
* Database Design (ERD, Normalisation, SQL..)
* Testing
* Consideration of the ethical issues in software design and data storage

**Required improvement to the codebase:**

|  |  |
| --- | --- |
| **Required Improvement Num.** | **Description** |
| **R1** | Adding code comments to explain the code. |
| **R2** | Better organize the code into different files/classes and folders (make sure your solution is a proper Object Oriented solution including the use of inheritance, encapsulation and any other applicable OOP principles). |
| **R3** | Implement a well-designed database using your proposed ERD including Students, Teachers, Courses and Registrations entities, as well as adding necessary attributes with accurate datatypes. Also, your database should include constraints such as Primary Keys, Foreign keys, referential integrity constraints, Unique, Index, Default, and Check Constraints if applicable. Your ERD should be normalised to the third normal form, at minimum. |
| **R4** | CRUD functionalities must be fully implemented to manage students, teachers, and courses records, as well as managing student registrations for different courses. Which means, all functionalities (add, update, delete, and view all records) must be working properly within your application so the data in your database can be easily manipulated. |
| **R5** | Use sorting and searching algorithms to filter and sort loaded data. (for example your application should allow search for a particular record (e.g student record) by id or name...etc) |
| **R6** | Error and Exception handling (try, except) |
| **R7** | Implementing input validation to ensure your program receives valid input. |
| **R8** | Follow PEP 8 Standards |
| **R9** | Implementing loops, so the program should run until the user decides to quit the program. (the program should not close after executing the first operation, also invalid input should promote the user to input a valid choice until the user choose to exit the program). |