## Software Architecture 6N1449

## Project Management 6N4090

## Graphical User Interface Programming 6N0736

## Object Oriented Programming 6N2108

## Relational Database 6N4785

## Report for Combined Project 2023-2024

## Student name: Serhii Sylenko

## Date: 22.03.2024

**Table of Contents**

[Software Architecture 6N1449 1](#_Toc165466234)

[Project Management 6N4090 1](#_Toc165466235)

[Graphical User Interface Programming 6N0736 1](#_Toc165466236)

[Object Oriented Programming 6N2108 1](#_Toc165466237)

[Relational Database 6N4785 1](#_Toc165466238)

[Report for Combined Project 2023-2024 1](#_Toc165466239)

[Student name: Serhii Sylenko 1](#_Toc165466240)

[Date: 22.03.2024 1](#_Toc165466241)

[Introduction 4](#_Toc165466242)

[Initial Phase 4](#_Toc165466243)

[Current System 5](#_Toc165466244)

[New System 5](#_Toc165466245)

[UML Diagrams List 6](#_Toc165466246)

[Project Development Phase 8](#_Toc165466247)

[Development Plan 8](#_Toc165466248)

[System Requirement Phase 16](#_Toc165466249)

[Use Case Diagram of Current System 16](#_Toc165466250)

[Use Case Diagram for New System 17](#_Toc165466251)

[Technical Solution Proposal 18](#_Toc165466252)

[Activity Diagram New System 18](#_Toc165466253)

[Class Diagram New System 19](#_Toc165466254)

[Object Diagram New System 20](#_Toc165466255)

[Possible Solutions for project 21](#_Toc165466256)

[Alternative Technical Solutions 22](#_Toc165466257)

[Design Document 23](#_Toc165466258)

[Problem Partition 23](#_Toc165466259)

[Table Relationships 25](#_Toc165466260)

[Table Create and insert Statements 26](#_Toc165466261)

[Queries 48](#_Toc165466262)

[Database screenshots 50](#_Toc165466263)

[Security Considerations 54](#_Toc165466264)

[Test Cases 54](#_Toc165466265)

[Test Case 1 54](#_Toc165466266)

[Test Case 2 55](#_Toc165466267)

[Test Case 3 56](#_Toc165466268)

[Test Case 4 57](#_Toc165466269)

[Test Case 5 58](#_Toc165466270)

[Project Development Plan and Design Spec Review 59](#_Toc165466271)

# Introduction

A Further Education College (FEC), offering QQI awards, has increased its student intake steadily over the past ten years. At present it has in excess of 500 students. Record-keeping and administration time overheads have increased enormously, and they are carried out on a more-or-less individual teacher basis and in an individual style. Some records are manually kept, and some are stored in spreadsheets or tables in word processing documents. Each student produces assessment material to satisfy the requirements of each module for their specific award. With the average number of modules for an award being 9 and an average of 3 assessments per module, there is a lot of assessment results to be recorded.

This project is designed to find a more effective solution for this problem. The main goal and requirement are to develop a new, digital system for the purpose of storing, searching, updating, and making other actions with the information about teachers, students, and their results. Ideally, it should be an application with the user interface, through which access to the information in the database is gained and appropriate actions are done. The documentation for using the system and with the history and details of development should be included as well.

# Initial Phase

The task was handed out on 08 January 2024.

**I have chosen the Kanban agile methodology** to develop the Further Education College Record Keeping System.

The Main elements associated with this methodology are:

1. Visual representation of the work process
2. Limitation of the number of tasks in progress
3. A steady flow of work through the system
4. Work is pulled into the system based on capacity and demand

The Main processes include:

1. Visualizing the workflow
2. Defining work items
3. Setting “work in progress” limits
4. Managing the workflow

Initial aim and goals:

* Develop a user-friendly menu-driven application for allowing the input (preferably, by selection) of assessment results for students and a class group.
* Allow for searching, editing, updating, adding, and deleting the information for both teachers and students.
* Store all the information in the database.
* Provide various types of reports based on a student and class group results.
* Write the documentation.

Steps required to complete the project:

* Plan the development process of the project
* Develop the database
* Design the user interface
* Connect the database to the user interface
* Develop the application (possibly making changes to the database)
* Complete the documentation

## Current System

Current system is mostly manual with records being variously kept in paper/hardcopy format or in electronic form (Microsoft Word/Excel), or in a mixture of both. Usually, only a particular tutor knows the location of these records for student results in the modules that he/she teaches. All QQI modules and awards are listed in the QQI handbook, and all awards offered by the Further Education College are contained in its prospectus. In the end, all Further Education Colleges are required to provide this data to the Dept of Education in the form of a database detailing all class groups, student names and numbers. This database is developed using the Facility Administration application and a copy of this database is available.

Considering the fact, that the volume of records is approximately 500 per year, this system is barely suitable for this purpose. The main disadvantage is the absence of specific policy with regards to the recording of assessment results. Having a mixture of both hardcopy and electronic results may be really confusing when the results from all the modules are being gathered together into the database before they will be provided to the Dept of Education. On the other hand, this may be better for individual teachers, as they already have their own way of keeping track of results, which may be easier for them. Unfortunately, it doesn't contribute to the overall efficiency of the system, so it is still better to develop a new one in order to make the whole process more organized.

## New System

**Set of Requirements**

**User requirements:**

1. The UI should be user-friendly, with the minimum amount of input required.
2. Navigation in the application should be simple and straightforward for the user.
3. The user should be able to search an information for a particular teacher or student.
4. The user should be able to add/edit (update) the information about a teacher or student if he is allowed to do that (has login/password).
5. The user should be able to add/delete the information about a teacher or student if he is allowed to do that (has login/password).
6. The user should be able to generate a suitable type of report.

**System requirements:**

1. The system should be able to store all the relevant data in appropriate tables.
2. The system should allow for a quick and easy search, addition, deletion, and update of the information (in case of allowing the access to these actions).
3. The system should be able to provide various types of reports:
   * Full learner record for a student
   * Results for a particular module for a class group
   * Results for a particular module across all groups
   * A list of distinction students for a module
   * A full results report for a class group
   * Etc.
4. The system should be reliable enough to not allow access for operations like adding, deleting, or updating information without a login/password.
5. The system should be scalable as the amount of information changes.
6. The system should be well tested and as errorproof as possible.

## UML Diagrams List

* **UML** (Unified Model Language) is a basic visual modelling language that provides a standard way to picture a design of the system. In other words, it is a digital analog of a blueprint. The UML was developed in 1996 by the Object Management Group with the help of big tech companies, such as Microsoft, IBM, HP, and others. Its purpose is to combine all object-oriented methods into using a standard notation composed of the best features of previous notations used for this measure, as well as provide people with tools for design, analysis, code generation and other processes to make the work easier, more efficient, and understandable. In the end, the UML provides user with a ready-to-use language for creating models with the best practices, opportunity to collaborate and exchange, use frameworks, patterns, and components, while being independent of other programming languages and development processes.
* **Use Case Diagrams** are diagrams in the UML that represent users (actors) and how they interact with the system. In other words, it specifies the behaviour, but not the exact method of execution. They provide both the text and visual representation and help to design a system from the end user’s perspective; therefore, it is the most useful and efficient method of explaining and demonstrating the system to the people, who are not very familiar with the development process and its specifics. A use case diagram summarizes some of the relationships between use cases, actors, and systems, but it does not show the order in which steps are performed to achieve the goals.
* **Activity Diagrams** are diagrams in the UML that describe dynamic aspects of the system. Basically, it is an advanced version of flowchart and use case diagram combined. Activity diagrams show the flow of one activity to another within a system or process, so that even complex systems can be visualized using activity diagrams. They are used to model complex activities and workflows in operations on objects, as well as identify pre- and post-conditions (the context) for use cases.
* **Object Diagrams** are diagrams in the UML that represent a specific instance of a class diagram with a detailed description of the state of the system at a particular point in time. They are quite like class diagrams, however, while class diagrams show classes and links between them, object diagrams focus on instances of these classes and their relationships in the system. Therefore, it is often used to verify the accuracy and completeness of class diagram or to explore specific model elements and their links as more detailed examples of the classes.

|  |  |
| --- | --- |
| Use Case Diagram | Current System |
| Use Case Diagram | New System |
| Class Diagram | New System |
| Object Diagram | New System |
| Activity Diagram | New System |

# Project Development Phase

## Development Plan

**Communication and information channels**

Communication plays a crucial role at all levels in the life of IT projects. Most of them include teamwork, collaboration, and interaction (for example, with the clients), which might become a major complication without a well-established communication mechanism. During the development of this project, the main communication channel is face-to-face with my teachers, as the project is individual, and doesn't include much teamwork except for student's helping each other and brainstorming.

Discussing ideas and key points in project development is the main purpose of communication in this case. Clear understanding of requirements is essential, and without it no work can be done. Clarification from class tutors helps to avoid misunderstandings that can lead to potential errors and delays, as well as resolve them when needed. This is an example of a downward communication channel.

Other important information channels are email with teachers and instant messaging and chatting tools (WhatsApp & Snapchat) with groupmates. Whenever I need help or feedback, and I am not at college, I can ask them remotely. It mostly resembles the horizontal communication channel.

Finally, during the whole project lifetime, project management tools are important. I use Trello and TeamGannt as an information channel for knowing deadlines, tasks that I need to do, and get access to general planning of project whenever I need to. In this case, it is an upward communication channel as the teacher observes how the student plans the project and updates teacher on its progress.

**Trello**

Зображення, що містить текст, знімок екрана, мультимедіа, програмне забезпечення

Автоматично згенерований описЗображення, що містить текст, програмне забезпечення, мультимедіа, знімок екрана

Автоматично згенерований опис

**TeamGannt**

Зображення, що містить текст, програмне забезпечення, число, ряд

Автоматично згенерований опис Зображення, що містить знімок екрана, текст, програмне забезпечення, ряд

Автоматично згенерований опис Зображення, що містить знімок екрана, текст, програмне забезпечення, Графік

Автоматично згенерований опис Зображення, що містить текст, знімок екрана, число, програмне забезпечення

Автоматично згенерований опис

Зображення, що містить текст, знімок екрана, програмне забезпечення, число

Автоматично згенерований опис

**Project monitoring and evaluation techniques**

Project monitoring and evaluation are essential components of project management, especially in IT projects where complexities and uncertainties are common. The monitoring takes place during the whole production process (phase of the project), while evaluation takes part during the key points in the project and in the end. Usually, monitoring and evaluation are based on the elements of efficiency and effectiveness, such as quality, cost, time, relevance, and completeness.

In this project I don’t need to monitor and evaluate staff and finances, as I am working individually with the help of teachers, and software that I use is completely free in terms of money, as well as the project as whole. Therefore, the main objectives to monitor are project activities. As was mentioned before, I track project tasks, milestones, and deliverables using Kanban agile methodology in Trello with TeamGannt gannt chart addition. The assessment of quality of the product (application) is conducted using testing audit (test cases) and code reviews with teacher.

Risk assessment is definitely one of the most important parts during monitoring the project. It involves identifying, assessing, and mitigating risks throughout the project lifecycle, so that potential issues can not impact the project success drastically. There is no lack of information about risks on the internet, but sometimes this information might be very complex. Risk assessment starts during the planning phase of the project and continues until its end. When identification of risks is done, the analysis starts. First is qualitative risk analysis, which involves setting the process of priority risks for further analysis of action by assessing probability of occurrence and impact to reduce level of uncertainty and focus on high priority risks. Then it gets to performing the quantitative risk analysis - the process of numerically analysing the effect of identified risks on overall project objectives. That way it supports the decision-making to reduce project uncertainty. Overall, risk analysis helps to control and plan risk responses.

While monitoring takes place, process evaluation is also happening. Evaluating the project’s processes and procedures helps identify areas for improvement in project execution and management. This may involve assessing the efficiency, effectiveness, and relevance of project activities and workflows. Therefore, it is closely intertwined with monitoring at this point. In terms of this project, there were a lot of changes in the management, because the flow of the project changed a little bit, and planning had to be done again as the result of me being absent for 3 weeks in a row and not being able to keep up the progress at that time. So, I had to change the priority and order of the tasks to speed up and be able to finish everything at time. Of course, the project execution also had been changed and improved as more progress was done in less time.

Gathering teacher feedback is also an important technique for the development of the project, as it provides valuable insights into areas for improvement. As it was mentioned above, clarification of the requirements and discussion of the ideas of possible implementation play a huge role in the process.

In the end, outcome evaluation is carried out. This involves assessing the project’s outcomes and impacts against the intended objectives and goals. In this case, it takes part online, where student shows the final product of the project to teachers, who observe, ask relevant questions, and assess the work. And the most important part for student except for getting marks is seeing whether “lessons were learned” and experience was helpful for the future projects.

**Contingency planning**

Contingency planning involves preparing for potential risks, emergencies, or unforeseen events that could disrupt the process of development of an IT project. It's all about thinking ahead and developing strategies to mitigate the impact of these disruptions and ensure continuity. To develop a contingency plan, it is possible to use the next steps:

1. Identify potential risks that could impact the project (technical failures, scope changes, resource constraints) and categorize them based on their likelihood and impact on the project.
2. Evaluate the potential impact of each identified risk on the project's objectives, timeline, and quality, and prioritize them based on their severity to focus on those that pose the highest threat to the project's success.
3. Develop specific strategies and action plans to address each high priority risk and define criteria for activating them. If a risk materialises, determine alternative approaches, solutions, or workarounds that can be implemented.
4. Identify the resources (financial, human, and technical) required to implement contingency plans and ensure that they are available and accessible when needed, including backup systems.
5. Document the contingency plans, including strategies, action plans, and procedures, in a clear and accessible format. Then regularly review and update the contingency plans to reflect changes in the project's scope, environment, or risks.

**Project methodology choices**

A project management methodology is a set of principles and practices forming a framework that helps to manage the project in the best way possible. Because all the projects are different, there are a lot of different project methodologies existing, as well as factors that will impact the decision of which one is right for the particular project. Key considerations are cost and budget, team size, ability to take risks, flexibility, and timeline.

The most popular and developed project methodologies are Waterfall, Agile, Scrum, Kanban, XP, Lean, and RAD. The traditional Waterfall methodology represents a linear sequence of stages, going from requirements to analysis, then design, construction, testing, and finishing with deployment and maintenance. The problem is that progress flows in one direction, which might be dangerous as expectations most likely won’t match up with reality, while there’s no going back. Agile methodology is basically “everything, but not too much”. It is a general methodology that tries to find balance between things, which created specific sub-frameworks like Scrum and Kanban. With Scrum, work is split into short cycles known as “sprints”, which usually last about 1-2 weeks, while Kanban can be applied to incrementally change the underlying process, having a visual representation of the tasks on the board. The eXtreme Programming (XP) methodology has a defined set of rules to follow, which are based on simplicity, communication, feedback, respect, and courage. Lean methodology is about applying lean principles to maximise value and minimise waste. Finally, Rapid Application Development is a type of agile project management methodology that aims to facilitate faster software development. It uses rapid prototype releases and iterations to gather feedback in a short period of time, and values that user feedback over strict planning and requirements recording.

Between all of them I have chosen the Kanban Agile project management methodology as an appropriate for this project. The main reason is visualisation of the tasks which is really important for such a big project, especially for the first time, and which makes it easier to use. Kanban board is available on Trello and has many free basic functions that are useful for the project management process. Because there is not much time for the project (but not little as well), I want to be able to have a working model as soon as possible and have a detailed specification that outlines all functional and non-functional requirements. Kanban is perfectly suited for this role, as it gives an immediate visual overview of where each piece of work stands at any given time. Work is pulled from the predefined backlog on a continuous basis and moved through the columns on the board, with each column representing a stage of the process. It also helps to see where tasks are gathering, so that attention is paid to them. Work in progress limits are also useful as they restrict the number of tasks in play at any given time. Comparing to RAD methodology, which also takes place when there is a quite small-time limit, Kanban is more organised, and easily allows for changes unlike Waterfall. Other Agile methodologies doesn’t have many major differences, so Kanban is winning in terms of details.

In this project, I have also divided the Kanban board into sections, that represent modules (Project Management, Software Architecture, GUI, OOP, and Relational Database). This way it is easier for me to navigate through the tasks. In other words, it is backlog divided into sub-backlogs. First, I plan the tasks for a week, moving them into the column called “To-Do”, and after I start one of them, it moves to the “Doing” column. After it is done, it goes into “Done” column. Concurrently, all the progress is tracked in Gannt chart.

**Managing changes in a project**

Managing change in a project when taking corrective action to meet agreed goals involves several key steps. When regularly monitoring the project progress and performance, there are times and cases when there might be a need for change. That is the identification of it. Before implementing a change, it is necessary to assess the potential impact on the project. In other words, to understand whether the change is worth it. To develop a plan, first step is to outline the specific changes required to address the identified issues. This plan should include clear objectives, strategies for implementation, timelines, and resource requirements. Then it is time to implement the change using the produced plan. Especially at the early stage of this process, it is important to continuously monitor the impact of the changes on project performance and progress toward the goals. If needed, adjusting the approach based on feedback and new insights takes place. Again, if the change doesn’t work as planned, or complicates the process, it is a good time and idea to reconsider it as early as possible.

In this particular project changes are more than likely to happen, especially because of lack of experience of working on such a big project. Therefore, it is really relevant and important to think twice and ahead before implementing any, as it might ruin everything. To prevent this from happening, it is best to have a plan beforehand, and follow it directly at least for the first couple of time.

**Financial planning and budget models**

For this project, there is no particular budget, as everything that is used is free of money cost. All the software has a free version and access that is more than enough for an effective work. For project management, these are Trello and Gannt chart, which both have premium versions, but they are not needed, while NetBeans IDE for Java programming, MySQL Workbench for SQL database, and draw.io for diagrams are completely free. Therefore, it comes down to time and resources constraints, such as time taken for each part of the project during both inside and outside of college time, help availability, and internet information resources. The easiest part is to plan the mechanism of finding and using information resources in internet – it is possible to do it beforehand. For example, to find and save official documentations for Java, javax.swing, MySQL, NetBeans, MySQL Workbench, Trello, etc. Then, during the active phase of development, save other information resources that proved to be useful. In terms of help availability, it is impossible to fully plan it, as there are life conditions when something might happen. But it is good to know and remember the schedule of lessons, as well as about possibility of organisation of one-to-one help sessions, so that the work is going to be more effective. About time, it is important to plan as much as possible in advance, but more importantly, stick to it. That way it will be more likely to finish the project qualitatively and on time.

**A summary of my experience**

**Phase 1 - Understand Brief**

When I received the brief, I immediately began to research the requirements for the application. From the internet sources, I got to know the structure of Further Education College in terms of education. I understood wrote out connections between courses, modules, assessments, teachers and students, class groups, which later provided me with the basis for database planning and development, as well as understanding of the system as a whole. I visited [www.gti.ie](http://www.gti.ie) and [www.qqi.ie](http://www.qqi.ie) websites for this purpose.

After this, I analysed the old system in terms of its advantages and disadvantages, so that I would know how to improve it. Then I combined my research results with requirements, what laid the foundation for the new system design.

**Phase 2 - Database Planning & Development**

From the beginning it was decided that I am going to use SQL programming language in MySQL Workbench system software, because that was what we were studying. Planning began from identifying what data would be collected. Afterwards, a list of the entities and their attributes with suitable data types and primary keys was designed and put into a form of EER diagram. When this was done, the time for establishing relationships has come. In case of many-to-many relationships, I tried to minimise them as much as possible, and if they were still necessary, I created a separate table for establishing them.

As soon as EER diagram was ready, I forward engineered it and created the database with all of the appropriate tables. Then it became a time to populate them for test purposes and to see if each attribute and its datatype, primary/foreign keys were created right. I tested the database using sample queries of selecting, updating, and deleting the information.

Refactoring took place during the development phase of the project. Some other relationships between tables were added (many-to-many between staff and student), datatypes changed (date of birth), and sizes of attributes as well. All changes were implemented successfully.

**Phase 3 - GUI Planning & Development**

For Graphical User Interface, the Use Case Diagram helped me to identify individual interaction with the system. Activity diagram helped to properly design the correct flow of each interaction, and class diagram with object diagram helped to identify the main objects needed to deliver on the functionality required. After this, it was up to my imagination to visualise the final look of the application. As the main requirement was to make it user-friendly, I designed an easy navigation, and preferred selection to typing in implementation. I also decided to add some tooltips and icons, so that people wouldn’t be confused as much. I used bright colours on the main buttons for users to find them quickly and made it error-proof in case if someone would accidentally (or not) try to use or insert something in the wrong way. When I allowed some options for the user, I made sure that all the appropriate data will be chosen correctly.

**Phase 4 – Integration**

Integration is a crucial aspect of software development, involving the combination of various subsystems within a product. Within the Waterfall project management methodology, integration occurs at a late stage in the project, often posing challenges due to mismatches between subsystems. Modern methodologies, however, practice early integration of subsystems from the project's start.

Within this project, integration is implemented with merging the graphical user interface (GUI) with the database (DB). For this purpose, a connection to the database is included in the Java code on GUI side, and all the queries are transferred to the database through it. JDBC connect requirements were considered, as well as making sure that all of the SQL queries in Java were aligned with the requirements of the DB.

Furthermore, this project emphasized prioritizing the development of the DB over the GUI, deferring the latter for later stages. This facilitated a successful integration of the two sub-systems, which is crucial for achieving the success with the project.

# System Requirement Phase

## Use Case Diagram of Current System

The following image is a Use Case Diagram of my impression of the current system:

Зображення, що містить текст, схема, знімок екрана, ряд

Автоматично згенерований опис

## Use Case Diagram for New System

The following image is part of a use case diagram of my new proposed system regarding students and tutors:

Зображення, що містить текст, ескіз, малюнок, схема

Автоматично згенерований опис

# Technical Solution Proposal

## Activity Diagram New System

The following image is part of an activity diagram of my new proposed system regarding students and tutors:

Зображення, що містить текст, схема, знімок екрана, ряд

Автоматично згенерований опис

## Class Diagram New System

The following image is part of an activity diagram of my new proposed system regarding students and tutors:

Зображення, що містить текст, знімок екрана, програмне забезпечення, Паралель

Автоматично згенерований опис

Code engineering involves the systematic approach to designing, developing, and maintaining software systems. Class diagrams and Use Case Diagrams are two essential tools for design specification.

Class diagrams serve as a blueprint for implementation. They help in identifying the main classes of the system based on requirements. Once classes are identified, attributes and methods for each class are defined. Relationships between classes (like inheritance, association, etc.) are established too. Class diagrams provide a visual representation of the structure of the system, making it easier for developers to understand the architecture. Their main benefit is making the system more adaptable to future modifications and changes by providing a clear structure.

## Object Diagram New System

The following image is part of an object diagram of my new proposed system regarding students and tutors:

Зображення, що містить текст, знімок екрана, документ, Шрифт

Автоматично згенерований опис

# Possible Solutions for project

**Main target systems:**

* **MySQL**: An open-source relational database management system used to create, design, and develop a database through MySQL Workbench tool using SQL programming language.
* **Java**: A programming language used to develop a system using object-oriented programming concepts.
* **NetBeans**: An Integrated Development Environment for running Java and creating a Graphical User Interface through javax.swing for the system.

**Feasibility:**

* **Technical**: Technical Feasibility includes the assessment of the technical capabilities required to develop and maintain the system, including hardware and software, as well as evaluation of compatibility with existing systems and technologies used within the college. In this case, hardware and software provided from college and the ones that I have myself are suitable and compatible, so technical capabilities are sufficient. The question of determining whether I have sufficient knowledge and skills to develop this project is also included here. I am confident that I am capable of completing this project and creating the qualitative product. Even if I don’t know enough at the moment, I am ready to learn new things if necessary.
* **Commercial**: Commercial Feasibility includes the analysis of demand for the system and potential competition, as well whether the college is ready to invest in the development and implementation of the system. In this case, there is definitely a demand for this system from Further Education College, which implies that it is also ready to invest and participate in this project. Competition is also present between students, as only the best system might be chosen for this purpose.
* **Financial**: Financial Feasibility is usually about estimating the costs associated with system development, including software development, hardware acquisition, and ongoing maintenance; comparing the estimated costs with potential benefits such as increased efficiency, and conducting a cost-benefit analysis to determine if the investment in the system is financially justified. But in this project, I don’t have a budget or the need to operate with finances, as I have everything that I need for completing it. The most important thing is that the project is part of a college assignment, so I won’t receive money for this, only marks.
* **Organisational**: Organizational Feasibility is about evaluating the impact of the system on existing organisational processes and structures; assessing the readiness of the organisation to adopt and integrate the new system into its operations, and identifying potential risks and challenges related to organisational change management. In this project the impact on the process will be positive, as the main goal of it is to develop a new, efficient system of record keeping, so organisation is ready to adopt and integrate when it is ready, as it is important for them to solve this problem. The risks and challenge, of course, are that the implementation of the project might fail, or it might be more difficult to implement if people in organisation won’t be fully satisfied with the product.

## Alternative Technical Solutions

The client side can be developed using HTML, CSS, and JS as a website application instead of desktop. It will still be possible to connect it to the same database, but will require more knowledge for the connection purposes, as well as a decent knowledge of languages mentioned above. Also, in this case it will be better to use JavaScript related database and frameworks, such as MongoDB and React with Node.js.

# Design Document

## Problem Partition

**Graphical User Interface**

The biggest challenge of developing GUI was a planning and design phase. It was hard to imagine the look of the new system having so many requirements and a big difference with the previous one. It all started from the decision on IDE – NetBeans, and then the library – Swing. Before, I had only limited knowledge of Swing library, so I had to keep the line between fantasy and reality of what I can do using it. In the process, I still had to learn how other unknown to me before controls work, but I think I managed to do well. The hardest part was to decide on design as the look of the whole application depended on it. At the same time, I always had to remember to keep design user-friendly, and implement selection instead of typing where possible. Navigation was the key point as well, but arguably the easiest between all of them in terms of decision.

Thanks to NetBeans and Swing, I did not have to use much of Java coding, as the creation of controls was done automatically. Therefore, it all came down to knowing properties and events of controls, which I practiced along the development of GUI. Sometimes it was challenging, but generally went more or less easy. Sometimes I had to change controls and refactor the code, but it was mostly due to my desire and ability to make it even better. In the end, in my opinion everything went pretty well, and I am happy with the result.

**Database**

At first, database was the most challenging part for me. Mostly, because I only started learning SQL, and haven’t done similar projects before. It was quite hard to understand all the concepts of database design and development from the first try. For example, I had to come back to EER diagram and change it 3 times during the development phase of the project, as some attributes and relationships were not done correctly. That was a headache because I had to regenerate it every time, as well as the database itself. But closer to the end of the project when it has already connected to the GUI, it became easier to understand and do everything with every query I have done there. Even hard queries like JOINs were quite easy for me to implement.

Arguably, the easiest and the most boring part was populating the tables. At the beginning, because of the reasons stated above, it was hard to understand what was wrong when errors during this process appeared, but later it began working properly, and it was just a matter of time before everything was populated.

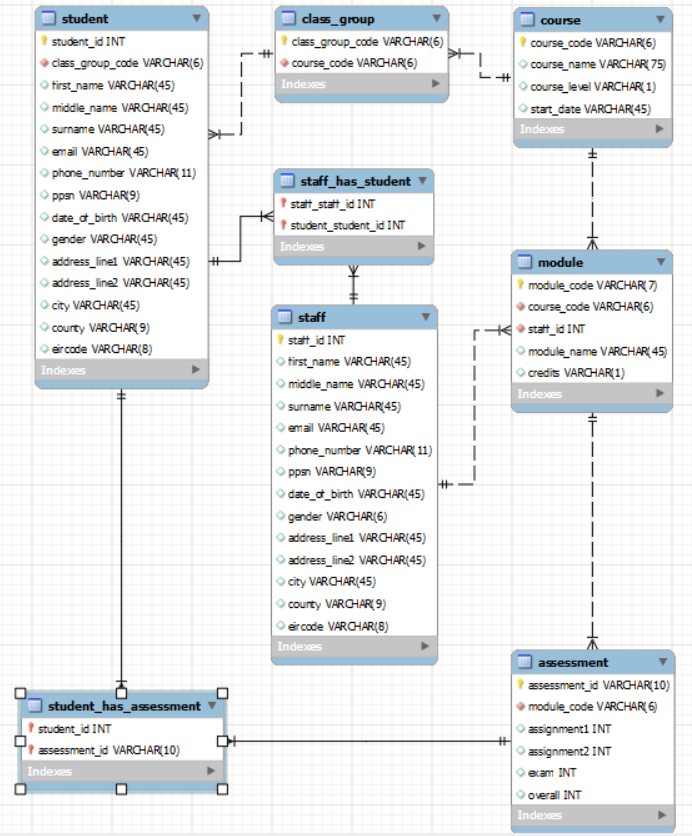
**Code**

The hardest part of the whole project was the coding part. Although I have the most experience in this particular part, it is always the hardest, because there are many tasks and problems, and each and every of them is different. Most of the issues in coding were due to lack of experience of solving them, and, sometimes, overcomplicating the code itself. At first, they happened the most in the phase of integration with database, but after I did it a couple of times, it became easy and pretty obvious, so I was able to find and fix mistakes clearly. It was never a problem of integration of data types, or even the Date or Time datatype issues; it was rather to create a query right and pass results to the right places. But even this type of errors and mistakes was quite limited, and it mostly came down to the problems inside the code itself.

Using Object Oriented Programming concepts proved to be very useful, and it limited the damage of not saving the data in the code properly. Still, at some point it became quite confusing going back and forward from the information in database and in code, so it was a challenge to connect the right information to the GUI and back. Overall, it was the most time-consuming part, but I am happy with the result that I achieved, as there were almost no bugs and no errors left by the end of the development.

## Table Relationships

**EER diagram:**



**Table relationships:**

Staff – Student (many-to-many)

Student – Assessment (many-to-many)

Class\_group – Student (one-to-many)

Course – Class\_group (one-to-many)

Course – Module (one-to-many)

Module – Assessment (one-to-many)

Staff – Module (one-to-many)

## Table Create and insert Statements

**Assessment Table:**

CREATE DATABASE IF NOT EXISTS `finalproject` /\*!40100 DEFAULT CHARACTER SET utf8mb3 \*/ /\*!80016 DEFAULT ENCRYPTION='N' \*/;

USE `finalproject`;

-- MySQL dump 10.13 Distrib 8.0.34, for Win64 (x86\_64)

--

-- Host: 127.0.0.1 Database: finalproject

-- ------------------------------------------------------

-- Server version 8.1.0

/\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/;

/\*!50503 SET NAMES utf8 \*/;

/\*!40103 SET @OLD\_TIME\_ZONE=@@TIME\_ZONE \*/;

/\*!40103 SET TIME\_ZONE='+00:00' \*/;

/\*!40014 SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0 \*/;

/\*!40014 SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0 \*/;

/\*!40101 SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='NO\_AUTO\_VALUE\_ON\_ZERO' \*/;

/\*!40111 SET @OLD\_SQL\_NOTES=@@SQL\_NOTES, SQL\_NOTES=0 \*/;

--

-- Table structure for table `assessment`

--

DROP TABLE IF EXISTS `assessment`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!50503 SET character\_set\_client = utf8mb4 \*/;

CREATE TABLE `assessment` (

`assessment\_id` varchar(10) NOT NULL,

`module\_code` varchar(7) NOT NULL,

`assignment1` int DEFAULT NULL,

`assignment2` int DEFAULT NULL,

`exam` int DEFAULT NULL,

`overall` int DEFAULT NULL,

PRIMARY KEY (`assessment\_id`),

KEY `fk\_assessment\_module1\_idx` (`module\_code`),

CONSTRAINT `fk\_assessment\_module1` FOREIGN KEY (`module\_code`) REFERENCES `module` (`module\_code`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `assessment`

--

LOCK TABLES `assessment` WRITE;

/\*!40000 ALTER TABLE `assessment` DISABLE KEYS \*/;

INSERT INTO `assessment` VALUES ('13779COM','5N0690',0,0,0,0),('13779DM','5N0783',0,0,0,0),('13779MIT','5N18396',0,0,0,0),('13779MT','5N0580',25,35,20,80),('13779OOP','5N0541',0,0,0,0),('13779PDP','5N2927',0,0,0,0),('13779SA','5N2772',0,0,0,0),('13779TW','5N1367',0,0,0,0),('13779WE','5N1356',0,0,0,0),('16251GUI','6N0736',0,0,0,0),('16251LS','6N2191',0,0,0,0),('16251MH','6N3395',0,0,0,0),('16251MT','6N0734',30,25,10,75),('16251OOP','6N2108',0,0,0,0),('16251PM','6N4090',0,0,0,0),('16251RD','6N4785',0,0,0,0),('16251SA','6N1449',0,0,0,0),('16251WE','6N1946',0,0,0,0),('20416COM','5N0690',0,0,0,0),('20416DM','5N0783',0,0,0,0),('20416MIT','5N18396',0,0,0,0),('20416MT','5N0580',10,15,20,45),('20416OOP','5N0541',0,0,0,0),('20416PDP','5N2927',0,0,0,0),('20416SA','5N2772',0,0,0,0),('20416TW','5N1367',0,0,0,0),('20416WE','5N1356',0,0,0,0),('31156COM','5N0690',0,0,0,0),('31156DM','5N0783',0,0,0,0),('31156MIT','5N18396',0,0,0,0),('31156MT','5N0580',30,15,15,60),('31156OOP','5N0541',0,0,0,0),('31156PDP','5N2927',0,0,0,0),('31156SA','5N2772',0,0,0,0),('31156TW','5N1367',0,0,0,0),('31156WE','5N1356',0,0,0,0),('39850COM','5N0690',0,0,0,0),('39850DM','5N0783',0,0,0,0),('39850MIT','5N18396',0,0,0,0),('39850MT','5N0580',25,25,25,75),('39850OOP','5N0541',0,0,0,0),('39850PDP','5N2927',0,0,0,0),('39850SA','5N2772',0,0,0,0),('39850TW','5N1367',0,0,0,0),('39850WE','5N1356',0,0,0,0),('44808GUI','6N0736',0,0,0,0),('44808LS','6N2191',0,0,0,0),('44808MH','6N3395',0,0,0,0),('44808MT','6N0734',0,0,0,0),('44808OOP','6N2108',0,0,0,0),('44808PM','6N4090',0,0,0,0),('44808RD','6N4785',0,0,0,0),('44808SA','6N1449',0,0,0,0),('44808WE','6N1946',0,0,0,0),('63341GUI','6N0736',0,0,0,0),('63341LS','6N2191',0,0,0,0),('63341MH','6N3395',0,0,0,0),('63341MT','6N0734',0,0,0,0),('63341OOP','6N2108',0,0,0,0),('63341PM','6N4090',0,0,0,0),('63341RD','6N4785',0,0,0,0),('63341SA','6N1449',0,0,0,0),('63341WE','6N1946',0,0,0,0),('64430GUI','6N0736',0,0,0,0),('64430LS','6N2191',0,0,0,0),('64430MH','6N3395',0,0,0,0),('64430MT','6N0734',0,0,0,0),('64430OOP','6N2108',0,0,0,0),('64430PM','6N4090',0,0,0,0),('64430RD','6N4785',0,0,0,0),('64430SA','6N1449',0,0,0,0),('64430WE','6N1946',0,0,0,0),('69498COM','5N0690',0,0,0,0),('69498DM','5N0783',0,0,0,0),('69498MIT','5N18396',0,0,0,0),('69498MT','5N0580',0,0,0,0),('69498OOP','5N0541',0,0,0,0),('69498PDP','5N2927',0,0,0,0),('69498SA','5N2772',0,0,0,0),('69498TW','5N1367',0,0,0,0),('69498WE','5N1356',0,0,0,0),('88266COM','5N0690',0,0,0,0),('88266DM','5N0783',0,0,0,0),('88266MIT','5N18396',0,0,0,0),('88266MT','5N0580',0,0,0,0),('88266OOP','5N0541',0,0,0,0),('88266PDP','5N2927',0,0,0,0),('88266SA','5N2772',0,0,0,0),('88266TW','5N1367',0,0,0,0),('88266WE','5N1356',0,0,0,0);

/\*!40000 ALTER TABLE `assessment` ENABLE KEYS \*/;

UNLOCK TABLES;

/\*!40103 SET TIME\_ZONE=@OLD\_TIME\_ZONE \*/;

/\*!40101 SET SQL\_MODE=@OLD\_SQL\_MODE \*/;

/\*!40014 SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS \*/;

/\*!40014 SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS \*/;

/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/;

/\*!40111 SET SQL\_NOTES=@OLD\_SQL\_NOTES \*/;

-- Dump completed on 2024-04-29 22:57:19

**Class\_group Table:**

CREATE DATABASE IF NOT EXISTS `finalproject` /\*!40100 DEFAULT CHARACTER SET utf8mb3 \*/ /\*!80016 DEFAULT ENCRYPTION='N' \*/;

USE `finalproject`;

-- MySQL dump 10.13 Distrib 8.0.34, for Win64 (x86\_64)

--

-- Host: 127.0.0.1 Database: finalproject

-- ------------------------------------------------------

-- Server version 8.1.0

/\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/;

/\*!50503 SET NAMES utf8 \*/;

/\*!40103 SET @OLD\_TIME\_ZONE=@@TIME\_ZONE \*/;

/\*!40103 SET TIME\_ZONE='+00:00' \*/;

/\*!40014 SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0 \*/;

/\*!40014 SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0 \*/;

/\*!40101 SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='NO\_AUTO\_VALUE\_ON\_ZERO' \*/;

/\*!40111 SET @OLD\_SQL\_NOTES=@@SQL\_NOTES, SQL\_NOTES=0 \*/;

--

-- Table structure for table `class\_group`

--

DROP TABLE IF EXISTS `class\_group`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!50503 SET character\_set\_client = utf8mb4 \*/;

CREATE TABLE `class\_group` (

`class\_group\_code` varchar(6) NOT NULL,

`course\_code` varchar(6) NOT NULL,

PRIMARY KEY (`class\_group\_code`),

KEY `fk\_class\_group\_course1\_idx` (`course\_code`),

CONSTRAINT `fk\_class\_group\_course1` FOREIGN KEY (`course\_code`) REFERENCES `course` (`course\_code`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `class\_group`

--

LOCK TABLES `class\_group` WRITE;

/\*!40000 ALTER TABLE `class\_group` DISABLE KEYS \*/;

INSERT INTO `class\_group` VALUES ('PSD1','5M0529'),('PSD2','5M0529'),('PSD3','5M0529'),('ASD','6M0691');

/\*!40000 ALTER TABLE `class\_group` ENABLE KEYS \*/;

UNLOCK TABLES;

/\*!40103 SET TIME\_ZONE=@OLD\_TIME\_ZONE \*/;

/\*!40101 SET SQL\_MODE=@OLD\_SQL\_MODE \*/;

/\*!40014 SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS \*/;

/\*!40014 SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS \*/;

/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/;

/\*!40111 SET SQL\_NOTES=@OLD\_SQL\_NOTES \*/;

-- Dump completed on 2024-04-29 22:57:19

**Course Table:**

CREATE DATABASE IF NOT EXISTS `finalproject` /\*!40100 DEFAULT CHARACTER SET utf8mb3 \*/ /\*!80016 DEFAULT ENCRYPTION='N' \*/;

USE `finalproject`;

-- MySQL dump 10.13 Distrib 8.0.34, for Win64 (x86\_64)

--

-- Host: 127.0.0.1 Database: finalproject

-- ------------------------------------------------------

-- Server version 8.1.0

/\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/;

/\*!50503 SET NAMES utf8 \*/;

/\*!40103 SET @OLD\_TIME\_ZONE=@@TIME\_ZONE \*/;

/\*!40103 SET TIME\_ZONE='+00:00' \*/;

/\*!40014 SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0 \*/;

/\*!40014 SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0 \*/;

/\*!40101 SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='NO\_AUTO\_VALUE\_ON\_ZERO' \*/;

/\*!40111 SET @OLD\_SQL\_NOTES=@@SQL\_NOTES, SQL\_NOTES=0 \*/;

--

-- Table structure for table `course`

--

DROP TABLE IF EXISTS `course`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!50503 SET character\_set\_client = utf8mb4 \*/;

CREATE TABLE `course` (

`course\_code` varchar(6) NOT NULL,

`course\_name` varchar(75) DEFAULT NULL,

`course\_level` varchar(1) DEFAULT NULL,

`start\_date` varchar(45) DEFAULT NULL,

PRIMARY KEY (`course\_code`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `course`

--

LOCK TABLES `course` WRITE;

/\*!40000 ALTER TABLE `course` DISABLE KEYS \*/;

INSERT INTO `course` VALUES ('5M0529','Programming and Software Development with Cyber Security','5','29.08'),('6M0691','Advanced Certificate in Software Development','6','31.08');

/\*!40000 ALTER TABLE `course` ENABLE KEYS \*/;

UNLOCK TABLES;

/\*!40103 SET TIME\_ZONE=@OLD\_TIME\_ZONE \*/;

/\*!40101 SET SQL\_MODE=@OLD\_SQL\_MODE \*/;

/\*!40014 SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS \*/;

/\*!40014 SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS \*/;

/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/;

/\*!40111 SET SQL\_NOTES=@OLD\_SQL\_NOTES \*/;

-- Dump completed on 2024-04-29 22:57:19

**Module Table:**

CREATE DATABASE IF NOT EXISTS `finalproject` /\*!40100 DEFAULT CHARACTER SET utf8mb3 \*/ /\*!80016 DEFAULT ENCRYPTION='N' \*/;

USE `finalproject`;

-- MySQL dump 10.13 Distrib 8.0.34, for Win64 (x86\_64)

--

-- Host: 127.0.0.1 Database: finalproject

-- ------------------------------------------------------

-- Server version 8.1.0

/\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/;

/\*!50503 SET NAMES utf8 \*/;

/\*!40103 SET @OLD\_TIME\_ZONE=@@TIME\_ZONE \*/;

/\*!40103 SET TIME\_ZONE='+00:00' \*/;

/\*!40014 SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0 \*/;

/\*!40014 SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0 \*/;

/\*!40101 SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='NO\_AUTO\_VALUE\_ON\_ZERO' \*/;

/\*!40111 SET @OLD\_SQL\_NOTES=@@SQL\_NOTES, SQL\_NOTES=0 \*/;

--

-- Table structure for table `module`

--

DROP TABLE IF EXISTS `module`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!50503 SET character\_set\_client = utf8mb4 \*/;

CREATE TABLE `module` (

`module\_code` varchar(7) NOT NULL,

`course\_code` varchar(6) NOT NULL,

`staff\_id` int NOT NULL,

`module\_name` varchar(45) DEFAULT NULL,

`credits` varchar(1) DEFAULT NULL,

PRIMARY KEY (`module\_code`),

KEY `fk\_module\_course1\_idx` (`course\_code`),

KEY `fk\_module\_staff1\_idx` (`staff\_id`),

CONSTRAINT `fk\_module\_course1` FOREIGN KEY (`course\_code`) REFERENCES `course` (`course\_code`) ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT `fk\_module\_staff1` FOREIGN KEY (`staff\_id`) REFERENCES `staff` (`staff\_id`) ON DELETE CASCADE ON UPDATE CASCADE

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `module`

--

LOCK TABLES `module` WRITE;

/\*!40000 ALTER TABLE `module` DISABLE KEYS \*/;

INSERT INTO `module` VALUES ('5N0541','5M0529',76302,'Fundamentals of OOP','3'),('5N0580','5M0529',23010,'Mobile Technologies','2'),('5N0690','5M0529',65447,'Communications','2'),('5N0783','5M0529',58579,'Database Methods','2'),('5N1356','5M0529',65447,'Work Experience','1'),('5N1367','5M0529',58579,'Team working','2'),('5N18396','5M0529',72315,'Maths for Information Technology','3'),('5N2772','5M0529',58579,'Software Architecture','2'),('5N2927','5M0529',76302,'Programming & Design Principles','4'),('6N0734','6M0691',23010,'Mobile Technologies','3'),('6N0736','6M0691',76302,'Graphical User Interface Programming','3'),('6N1449','6M0691',23010,'Software Architecture','2'),('6N1946','6M0691',58579,'Work Experience','1'),('6N2108','6M0691',76302,'Object Oriented Programming','3'),('6N2191','6M0691',40102,'Leadership','2'),('6N3395','6M0691',72315,'Mathematics','3'),('6N4090','6M0691',76302,'Project Management','2'),('6N4785','6M0691',76302,'Relational Database','3');

/\*!40000 ALTER TABLE `module` ENABLE KEYS \*/;

UNLOCK TABLES;

/\*!40103 SET TIME\_ZONE=@OLD\_TIME\_ZONE \*/;

/\*!40101 SET SQL\_MODE=@OLD\_SQL\_MODE \*/;

/\*!40014 SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS \*/;

/\*!40014 SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS \*/;

/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/;

/\*!40111 SET SQL\_NOTES=@OLD\_SQL\_NOTES \*/;

-- Dump completed on 2024-04-29 22:57:19

**Staff Table:**

CREATE DATABASE IF NOT EXISTS `finalproject` /\*!40100 DEFAULT CHARACTER SET utf8mb3 \*/ /\*!80016 DEFAULT ENCRYPTION='N' \*/;

USE `finalproject`;

-- MySQL dump 10.13 Distrib 8.0.34, for Win64 (x86\_64)

--

-- Host: 127.0.0.1 Database: finalproject

-- ------------------------------------------------------

-- Server version 8.1.0

/\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/;

/\*!50503 SET NAMES utf8 \*/;

/\*!40103 SET @OLD\_TIME\_ZONE=@@TIME\_ZONE \*/;

/\*!40103 SET TIME\_ZONE='+00:00' \*/;

/\*!40014 SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0 \*/;

/\*!40014 SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0 \*/;

/\*!40101 SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='NO\_AUTO\_VALUE\_ON\_ZERO' \*/;

/\*!40111 SET @OLD\_SQL\_NOTES=@@SQL\_NOTES, SQL\_NOTES=0 \*/;

--

-- Table structure for table `staff`

--

DROP TABLE IF EXISTS `staff`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!50503 SET character\_set\_client = utf8mb4 \*/;

CREATE TABLE `staff` (

`staff\_id` int NOT NULL,

`first\_name` varchar(45) DEFAULT NULL,

`middle\_name` varchar(45) DEFAULT NULL,

`surname` varchar(45) DEFAULT NULL,

`email` varchar(45) DEFAULT NULL,

`phone\_number` varchar(11) DEFAULT NULL,

`ppsn` varchar(9) DEFAULT NULL,

`date\_of\_birth` date DEFAULT NULL,

`gender` varchar(6) DEFAULT NULL,

`address\_line1` varchar(45) DEFAULT NULL,

`address\_line2` varchar(45) DEFAULT NULL,

`city` varchar(45) DEFAULT NULL,

`county` varchar(9) DEFAULT NULL,

`eircode` varchar(8) DEFAULT NULL,

PRIMARY KEY (`staff\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `staff`

--

LOCK TABLES `staff` WRITE;

/\*!40000 ALTER TABLE `staff` DISABLE KEYS \*/;

INSERT INTO `staff` VALUES (23010,'Mark',NULL,'Shaw','mark.shaw@gretb.ie','085 3311577','5484880UA','1976-12-19','Male','83 Maoilín','Knocknacarra','Galway','Galway','H91 Y38E'),(33137,'Mike','Hovard','Hainley','mike.hainley@gretb.ie','083 8292257','1234567PH','1990-02-14','Male','Wild Geese GAA','The Orchard','Fingal','Dublin','A91 HR98'),(40102,'Fiona',NULL,'Holden','fiona.holden@gretb.ie','083 6595831','7814326FH','1975-11-22','Female','15 Ardmore','Salthill','Galway','Galway','H91 N9WK'),(58579,'Declan',NULL,'Doherty','declan.doherty@gretb.ie','087 3651105','7492680UH','1982-04-04','Male','House R474','','Kilmaley','Clare','V95 CF30'),(65447,'Suzanne',NULL,'Kenny','suzanne.kenny@gretb.ie','086 7645756','1059277ZH','1973-06-15','Female','Mayo Footgolf','Lakeview','Deerpark','Mayo','F28 FW74'),(72315,'John',NULL,'Feeney','john.feeney@gretb.ie','083 7003272','4559216DA','1961-08-19','Male','Ballyclare Ct','','Ballyclare','Roskommon','N39 TH77'),(76302,'Una',NULL,'Keary','una.keary@gretb.ie','086 8232761','1764260BA','1980-09-30','Female','3 Pine Grove\r Treanrevagh\r','Treanrevagh','Mountbellew','Galway','H53 WE00');

/\*!40000 ALTER TABLE `staff` ENABLE KEYS \*/;

UNLOCK TABLES;

/\*!40103 SET TIME\_ZONE=@OLD\_TIME\_ZONE \*/;

/\*!40101 SET SQL\_MODE=@OLD\_SQL\_MODE \*/;

/\*!40014 SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS \*/;

/\*!40014 SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS \*/;

/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/;

/\*!40111 SET SQL\_NOTES=@OLD\_SQL\_NOTES \*/;

-- Dump completed on 2024-04-29 22:57:19

**Staff\_has\_student Table:**

CREATE DATABASE IF NOT EXISTS `finalproject` /\*!40100 DEFAULT CHARACTER SET utf8mb3 \*/ /\*!80016 DEFAULT ENCRYPTION='N' \*/;

USE `finalproject`;

-- MySQL dump 10.13 Distrib 8.0.34, for Win64 (x86\_64)

--

-- Host: 127.0.0.1 Database: finalproject

-- ------------------------------------------------------

-- Server version 8.1.0

/\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/;

/\*!50503 SET NAMES utf8 \*/;

/\*!40103 SET @OLD\_TIME\_ZONE=@@TIME\_ZONE \*/;

/\*!40103 SET TIME\_ZONE='+00:00' \*/;

/\*!40014 SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0 \*/;

/\*!40014 SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0 \*/;

/\*!40101 SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='NO\_AUTO\_VALUE\_ON\_ZERO' \*/;

/\*!40111 SET @OLD\_SQL\_NOTES=@@SQL\_NOTES, SQL\_NOTES=0 \*/;

--

-- Table structure for table `staff\_has\_student`

--

DROP TABLE IF EXISTS `staff\_has\_student`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!50503 SET character\_set\_client = utf8mb4 \*/;

CREATE TABLE `staff\_has\_student` (

`staff\_id` int NOT NULL,

`student\_id` int NOT NULL,

PRIMARY KEY (`staff\_id`,`student\_id`),

KEY `fk\_staff\_has\_student\_student1\_idx` (`student\_id`),

KEY `fk\_staff\_has\_student\_staff1\_idx` (`staff\_id`),

CONSTRAINT `fk\_staff\_has\_student\_staff1` FOREIGN KEY (`staff\_id`) REFERENCES `staff` (`staff\_id`) ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT `fk\_staff\_has\_student\_student1` FOREIGN KEY (`student\_id`) REFERENCES `student` (`student\_id`) ON DELETE CASCADE ON UPDATE CASCADE

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `staff\_has\_student`

--

LOCK TABLES `staff\_has\_student` WRITE;

/\*!40000 ALTER TABLE `staff\_has\_student` DISABLE KEYS \*/;

INSERT INTO `staff\_has\_student` VALUES (23010,13779),(58579,13779),(65447,13779),(72315,13779),(76302,13779),(23010,16251),(40102,16251),(58579,16251),(72315,16251),(76302,16251),(23010,20416),(58579,20416),(65447,20416),(23010,31156),(58579,31156),(65447,31156),(23010,39850),(65447,39850),(72315,39850),(76302,39850),(23010,44808),(40102,44808),(58579,44808),(72315,44808),(76302,44808),(23010,63341),(40102,63341),(58579,63341),(72315,63341),(76302,63341),(23010,64430),(40102,64430),(58579,64430),(72315,64430),(76302,64430),(23010,69498),(58579,69498),(65447,69498),(72315,69498),(76302,69498),(23010,88266),(65447,88266),(72315,88266),(76302,88266);

/\*!40000 ALTER TABLE `staff\_has\_student` ENABLE KEYS \*/;

UNLOCK TABLES;

/\*!40103 SET TIME\_ZONE=@OLD\_TIME\_ZONE \*/;

/\*!40101 SET SQL\_MODE=@OLD\_SQL\_MODE \*/;

/\*!40014 SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS \*/;

/\*!40014 SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS \*/;

/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/;

/\*!40111 SET SQL\_NOTES=@OLD\_SQL\_NOTES \*/;

-- Dump completed on 2024-04-29 22:57:20

**Student Table:**

CREATE DATABASE IF NOT EXISTS `finalproject` /\*!40100 DEFAULT CHARACTER SET utf8mb3 \*/ /\*!80016 DEFAULT ENCRYPTION='N' \*/;

USE `finalproject`;

-- MySQL dump 10.13 Distrib 8.0.34, for Win64 (x86\_64)

--

-- Host: 127.0.0.1 Database: finalproject

-- ------------------------------------------------------

-- Server version 8.1.0

/\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/;

/\*!50503 SET NAMES utf8 \*/;

/\*!40103 SET @OLD\_TIME\_ZONE=@@TIME\_ZONE \*/;

/\*!40103 SET TIME\_ZONE='+00:00' \*/;

/\*!40014 SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0 \*/;

/\*!40014 SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0 \*/;

/\*!40101 SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='NO\_AUTO\_VALUE\_ON\_ZERO' \*/;

/\*!40111 SET @OLD\_SQL\_NOTES=@@SQL\_NOTES, SQL\_NOTES=0 \*/;

--

-- Table structure for table `student`

--

DROP TABLE IF EXISTS `student`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!50503 SET character\_set\_client = utf8mb4 \*/;

CREATE TABLE `student` (

`student\_id` int NOT NULL,

`class\_group\_code` varchar(6) NOT NULL,

`first\_name` varchar(45) DEFAULT NULL,

`middle\_name` varchar(45) DEFAULT NULL,

`surname` varchar(45) DEFAULT NULL,

`email` varchar(45) DEFAULT NULL,

`phone\_number` varchar(11) DEFAULT NULL,

`ppsn` varchar(9) DEFAULT NULL,

`date\_of\_birth` date DEFAULT NULL,

`gender` varchar(45) DEFAULT NULL,

`address\_line1` varchar(45) DEFAULT NULL,

`address\_line2` varchar(45) DEFAULT NULL,

`city` varchar(45) DEFAULT NULL,

`county` varchar(9) DEFAULT NULL,

`eircode` varchar(8) DEFAULT NULL,

PRIMARY KEY (`student\_id`),

KEY `fk\_student\_class\_group\_idx` (`class\_group\_code`),

CONSTRAINT `fk\_student\_class\_group` FOREIGN KEY (`class\_group\_code`) REFERENCES `class\_group` (`class\_group\_code`) ON DELETE CASCADE ON UPDATE CASCADE

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `student`

--

LOCK TABLES `student` WRITE;

/\*!40000 ALTER TABLE `student` DISABLE KEYS \*/;

INSERT INTO `student` VALUES (13779,'PSD1','Macie',NULL,'Shelton','macie.shelton@outlook.com','085 4124177','7700225VH','2002-06-17','Female','25 Riverrun','Lower Road','Riverrun','Cavan','H14 FX69'),(16251,'ASD','Jeffrey','Aaron','Baron','jeffrey.baron@gmail.com','085 2625464','5452407DH','2003-10-05','Male','41 Riverview','Chambersland','New Ross','Wexford','Y34 C821'),(20416,'PSD3','Evelyn','Hanna','Cameron','evelyn.cameron@yahoo.com','087 3098114','1241125LH','2003-02-16','Female','132 Mountain View Park','Whitehall','Dundrum','Dublin','D14 KH93'),(31156,'PSD3','Idris',NULL,'Saunders','idris.sauders@gmail.com','087 4394241','6993459TH','1999-01-01','Male','25 Carraig Ban','Terryland','Galway','Galway','H91 ET6N'),(39850,'PSD2','Tomas','Hovard','Russell','tomas.russell@gmail.com','083 7985141','4115381JA','2004-09-08','Male','16 Botanic Park','Drumcondra','Dublin','Dublin','D09 PT78'),(44808,'ASD','Kobe',NULL,'McLeod','kobe.mcleod@yahoo.com','086 8935317','8268048MA','2001-05-25','Male','32 The Greens','','Duncormick','Wexford','Y35 V402'),(63341,'ASD','Macie','Alissa','Connor','macie.connor@gmail.com','085 8077283','4835734QA','2002-12-30','Female','84 Merrion Square South','','Dublin','Dublin','D02 KF79'),(64430,'ASD','Joanne',NULL,'Blair','joanne.blair@gmail.com','085 7627896','9857460SH','2001-10-10','Female','Kilferagh House','Nore Valley Walk','Bennettsbridge','Kilkenny','R95 CK71'),(69498,'PSD1','Amina',NULL,'Shelton','macie.shelton@outlook.com','085 4124177','7141506RA','2004-12-25','Female','25 Riverrun','Lower Road','Riverrun','Cavan','H14 FX69'),(88266,'PSD2','Ross',NULL,'Johnston','ross.johnston@yahoo.com','086 9772921','0691084DH','2005-06-07','Male','68 Donaghmede Road','Kilbarrack Upper','Dublin','Dublin','D05 H2K2');

/\*!40000 ALTER TABLE `student` ENABLE KEYS \*/;

UNLOCK TABLES;

/\*!40103 SET TIME\_ZONE=@OLD\_TIME\_ZONE \*/;

/\*!40101 SET SQL\_MODE=@OLD\_SQL\_MODE \*/;

/\*!40014 SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS \*/;

/\*!40014 SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS \*/;

/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/;

/\*!40111 SET SQL\_NOTES=@OLD\_SQL\_NOTES \*/;

-- Dump completed on 2024-04-29 22:57:19

**Student\_has\_assessment Table:**

CREATE DATABASE IF NOT EXISTS `finalproject` /\*!40100 DEFAULT CHARACTER SET utf8mb3 \*/ /\*!80016 DEFAULT ENCRYPTION='N' \*/;

USE `finalproject`;

-- MySQL dump 10.13 Distrib 8.0.34, for Win64 (x86\_64)

--

-- Host: 127.0.0.1 Database: finalproject

-- ------------------------------------------------------

-- Server version 8.1.0

/\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/;

/\*!50503 SET NAMES utf8 \*/;

/\*!40103 SET @OLD\_TIME\_ZONE=@@TIME\_ZONE \*/;

/\*!40103 SET TIME\_ZONE='+00:00' \*/;

/\*!40014 SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0 \*/;

/\*!40014 SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0 \*/;

/\*!40101 SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='NO\_AUTO\_VALUE\_ON\_ZERO' \*/;

/\*!40111 SET @OLD\_SQL\_NOTES=@@SQL\_NOTES, SQL\_NOTES=0 \*/;

--

-- Table structure for table `student\_has\_assessment`

--

DROP TABLE IF EXISTS `student\_has\_assessment`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!50503 SET character\_set\_client = utf8mb4 \*/;

CREATE TABLE `student\_has\_assessment` (

`student\_id` int NOT NULL,

`assessment\_id` varchar(10) NOT NULL,

PRIMARY KEY (`student\_id`,`assessment\_id`),

KEY `fk\_student\_has\_assessment\_assessment1\_idx` (`assessment\_id`),

KEY `fk\_student\_has\_assessment\_student1\_idx` (`student\_id`),

CONSTRAINT `fk\_student\_has\_assessment\_assessment1` FOREIGN KEY (`assessment\_id`) REFERENCES `assessment` (`assessment\_id`) ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT `fk\_student\_has\_assessment\_student1` FOREIGN KEY (`student\_id`) REFERENCES `student` (`student\_id`) ON DELETE CASCADE ON UPDATE CASCADE

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `student\_has\_assessment`

--

LOCK TABLES `student\_has\_assessment` WRITE;

/\*!40000 ALTER TABLE `student\_has\_assessment` DISABLE KEYS \*/;

INSERT INTO `student\_has\_assessment` VALUES (13779,'13779COM'),(13779,'13779DM'),(13779,'13779MIT'),(13779,'13779MT'),(13779,'13779OOP'),(13779,'13779PDP'),(13779,'13779SA'),(13779,'13779TW'),(13779,'13779WE'),(16251,'16251GUI'),(16251,'16251LS'),(16251,'16251MH'),(16251,'16251MT'),(16251,'16251OOP'),(16251,'16251PM'),(16251,'16251RD'),(16251,'16251SA'),(16251,'16251WE'),(20416,'20416COM'),(20416,'20416DM'),(20416,'20416MIT'),(20416,'20416MT'),(20416,'20416OOP'),(20416,'20416PDP'),(20416,'20416SA'),(20416,'20416TW'),(20416,'20416WE'),(31156,'31156COM'),(31156,'31156DM'),(31156,'31156MIT'),(31156,'31156MT'),(31156,'31156OOP'),(31156,'31156PDP'),(31156,'31156SA'),(31156,'31156TW'),(31156,'31156WE'),(39850,'39850COM'),(39850,'39850DM'),(39850,'39850MIT'),(39850,'39850MT'),(39850,'39850OOP'),(39850,'39850PDP'),(39850,'39850SA'),(39850,'39850TW'),(39850,'39850WE'),(44808,'44808GUI'),(44808,'44808LS'),(44808,'44808MH'),(44808,'44808MT'),(44808,'44808OOP'),(44808,'44808PM'),(44808,'44808RD'),(44808,'44808SA'),(44808,'44808WE'),(63341,'63341GUI'),(63341,'63341LS'),(63341,'63341MH'),(63341,'63341MT'),(63341,'63341OOP'),(63341,'63341PM'),(63341,'63341RD'),(63341,'63341SA'),(63341,'63341WE'),(64430,'64430GUI'),(64430,'64430LS'),(64430,'64430MH'),(64430,'64430MT'),(64430,'64430OOP'),(64430,'64430PM'),(64430,'64430RD'),(64430,'64430SA'),(64430,'64430WE'),(69498,'69498COM'),(69498,'69498DM'),(69498,'69498MIT'),(69498,'69498MT'),(69498,'69498OOP'),(69498,'69498PDP'),(69498,'69498SA'),(69498,'69498TW'),(69498,'69498WE'),(88266,'88266COM'),(88266,'88266DM'),(88266,'88266MIT'),(88266,'88266MT'),(88266,'88266OOP'),(88266,'88266PDP'),(88266,'88266SA'),(88266,'88266TW'),(88266,'88266WE');

/\*!40000 ALTER TABLE `student\_has\_assessment` ENABLE KEYS \*/;

UNLOCK TABLES;

/\*!40103 SET TIME\_ZONE=@OLD\_TIME\_ZONE \*/;

/\*!40101 SET SQL\_MODE=@OLD\_SQL\_MODE \*/;

/\*!40014 SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS \*/;

/\*!40014 SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS \*/;

/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/;

/\*!40111 SET SQL\_NOTES=@OLD\_SQL\_NOTES \*/;

-- Dump completed on 2024-04-29 22:57:20

## Queries

**Select teacher ID, name, and surname based on a teacher ID**

SELECT staff\_id, first\_name, surname FROM staff WHERE staff\_id=?

**Select all teacher information based on a surname**

SELECT \* FROM staff WHERE surname=?

**Select all teacher information based on a teacher ID**

SELECT \* FROM staff WHERE staff\_id=?

**Select module code and name based on a teacher ID**

SELECT module\_code, module\_name FROM module WHERE staff\_id=?

**Update teacher email, phone number, and address based on a surname**

UPDATE staff SET email=?, phone\_number=?, address\_line1=?, address\_line2=?, city=?, county=?, eircode=? WHERE surname=?

**Delete all teacher information based on a surname**

DELETE FROM staff WHERE surname = ?

**Select all student information based on a surname with wildcard**

SELECT \* FROM student WHERE surname=? OR surname LIKE 'R\_\_\_\_\_\_'

**Select all student information based on a student ID**

SELECT \* FROM student WHERE student\_id=?

**Update student email, phone number, and address based on a surname**

UPDATE student SET email=?, phone\_number=?, address\_line1=?, address\_line2=?, city=?, county=?, eircode=? WHERE surname=?

**Delete all student information based on a surname**

DELETE FROM student WHERE student\_id=?

**Add all teacher information**

INSERT INTO staff (staff\_id, first\_name, middle\_name, surname, email, phone\_number, ppsn, date\_of\_birth, gender, address\_line1, address\_line2, city, county, eircode) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)

**Add all student information**

INSERT INTO student (student\_id, class\_group\_code, first\_name, middle\_name, surname, email, phone\_number, ppsn, date\_of\_birth, gender, address\_line1, address\_line2, city, county, eircode) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)

**Select student id, first name, and surname based on teacher id**

SELECT student.student\_id, student.first\_name, student.surname FROM staff\_has\_student INNER JOIN student ON staff\_has\_student.student\_id = student.student\_id WHERE staff\_id=?;

**Select unique class group code based on teacher id**

SELECT DISTINCT student.class\_group\_code FROM staff\_has\_student INNER JOIN student ON staff\_has\_student.student\_id = student.student\_id WHERE staff\_id=?;

**Select class group code based on student id**

SELECT class\_group\_code FROM student WHERE student\_id=?;

**Select module code and name based on teacher id and class group code**

SELECT module\_code, module\_name FROM module INNER JOIN class\_group ON module.course\_code = class\_group.course\_code WHERE staff\_id=? AND class\_group\_code=?

**Select student id, first name and surname based on class group code**

SELECT student\_id, first\_name, surname FROM student WHERE class\_group\_code=?;

**Select assessment id, assignment1, assignment2, exam, and overall results based on student id and module code**

SELECT assessment.assessment\_id, assignment1, assignment2, exam, overall FROM assessment INNER JOIN student\_has\_assessment ON assessment.assessment\_id = student\_has\_assessment.assessment\_id WHERE student\_id=? AND module\_code=?;

**Update assignment1, assignment2, exam, and overall results based on assessment id**

UPDATE assessment SET assignment1=?, assignment2=?, exam=?, overall=? WHERE assessment\_id=?

**Select assessment id and results based on student id**

SELECT assessment.assessment\_id, assignment1, assignment2, exam, overall FROM assessment INNER JOIN student\_has\_assessment ON student\_has\_assessment.assessment\_id = assessment.assessment\_id WHERE student\_id=?;

**Select student id, assessment id and results based on class group and module codes**

SELECT student\_has\_assessment.student\_id, assessment.assessment\_id, assignment1, assignment2, exam, overall FROM assessment INNER JOIN student\_has\_assessment ON student\_has\_assessment.assessment\_id = assessment.assessment\_id INNER JOIN student ON student.student\_id = student\_has\_assessment.student\_id WHERE class\_group\_code=? AND module\_code=?;

## Database screenshots

**Staff Table:** stores teacher information.

**Зображення, що містить текст, програмне забезпечення, знімок екрана

Автоматично згенерований опис**

**Student Table:** stores student information.

Зображення, що містить текст, знімок екрана, число, програмне забезпечення

Автоматично згенерований опис

**Staff\_has\_student Table:** matches the teacher's ID with the student's ID and vice versa.

Зображення, що містить текст, знімок екрана, програмне забезпечення, Комп’ютерна піктограма

Автоматично згенерований опис

**Module Table:** stores module information.

Зображення, що містить текст, знімок екрана, програмне забезпечення, Веб-сторінка

Автоматично згенерований опис

**Course Table:** stores course information.

Зображення, що містить текст, знімок екрана, програмне забезпечення, Комп’ютерна піктограма

Автоматично згенерований опис

**Class\_group Table:** stores class group information.

Зображення, що містить текст, програмне забезпечення, Комп’ютерна піктограма, Веб-сторінка

Автоматично згенерований опис

**Assessment Table:** stores assessment information.

Зображення, що містить текст, знімок екрана, програмне забезпечення, Веб-сторінка

Автоматично згенерований опис

**Student\_has\_assessment Table:** matches the student’s ID with the assessment’s ID and vice versa.

Зображення, що містить текст, знімок екрана, програмне забезпечення, Комп’ютерна піктограма

Автоматично згенерований опис

## Security Considerations

I have added a password to access some actions like add, update, and delete information through the graphical user interface. Also, to enter assessment results and generate a report, teacher must enter his id. This ensures upmost security and protection of personal student and tutor records.

* Password: dmn
* Teacher ID: see database records

# Test Cases

## Test Case 1

**Test Case 1:** Add student information

**Test Description:** In the Add Student frame, enter all information and click “Add” button

**Expected Result:** Entered information should be stored in the database in Student table and cleared from the frame

**Actual Result:** Information is successfully saved in the database in the right place, information cleared from the frame and ready for the next information addition

Зображення, що містить текст, знімок екрана, програмне забезпечення, Комп’ютерна піктограма

Автоматично згенерований опис **Зображення, що містить знімок екрана, текст, програмне забезпечення

Автоматично згенерований опис**  
**Status:** Pass

## Test Case 2

**Test Case 2:** Search for and update teacher information

**Test Description:** In the Teacher Search frame, search for a teacher using ID option with ID 77761, and update his email, phone number, and address in correct format

**Expected Result:** Teacher information found and fully displayed, information in database in the Student table updated successfully

**Actual Result:** Teacher information found and fully displayed, information in database in the Student table updated successfully

**Зображення, що містить текст, знімок екрана, програмне забезпечення, монітор

Автоматично згенерований опис**

**Зображення, що містить текст, програмне забезпечення, знімок екрана, число

Автоматично згенерований опис**

**Status:** Pass

## Test Case 3

**Test Case 3:** Search for and delete teacher information

**Test Description:** In the Teacher Search frame, search for a teacher using Surname option with surname Haskie, and delete his information from the system

**Expected Result:** Teacher information found and deleted from both frame and database

**Actual Result:** Teacher information found and deleted from both frame and database

Зображення, що містить текст, знімок екрана, програмне забезпечення, монітор

Автоматично згенерований опис **Зображення, що містить текст, знімок екрана, програмне забезпечення, Комп’ютерна піктограма

Автоматично згенерований опис**

**Зображення, що містить текст, програмне забезпечення, число, Шрифт

Автоматично згенерований опис**

**Status:** Pass

## Test Case 4

**Test Case 4:** Search and update the assessment results

**Test Description:** In the Assessment frame, enter teacher ID 23010, choose Student, Jeffrey Baron, Mobile Technologies module, and update the marks to get Distinction

**Expected Result:** Frame opened; student, class group, module, and assessment with results found, results updated to Distinction and in the database in Assessment table

**Actual Result:** Frame opened; student, class group, module, and assessment with results found, results updated to Distinction and in the database in Assessment table

**Зображення, що містить текст, електроніка, знімок екрана, монітор

Автоматично згенерований опис Зображення, що містить текст, знімок екрана, монітор, програмне забезпечення

Автоматично згенерований опис**

**Зображення, що містить текст, знімок екрана, програмне забезпечення, Комп’ютерна піктограма

Автоматично згенерований опис**

**Status:** Pass

## Test Case 5

**Test Case 5:** Generate a report

**Test Description:** In the Report frame, enter teacher ID 23010, choose a list of students for a module based on grade type of report, for module 6N0734 Mobile Technologies, with Distinction grade, and click Generate button

**Expected Result:** Frame opened; appropriate type of report generated with right student results

**Actual Result:** Frame opened; appropriate type of report generated with right student results

**Зображення, що містить текст, електроніка, знімок екрана, монітор

Автоматично згенерований опис**

**Status:** Pass

# Project Development Plan and Design Spec Review

In my opinion, the completed version of the project meets the original requirements and design specifications. Everything that was asked for in the brief is done, and application is functioning well, without errors or issues. It is able to add, search, update, and delete the information about both teacher and student, doing so only if you have password (except for search); enter and update assessment results for both student and class group, allowing to do so only for modules of specific teacher, using and asking for teacher ID before allowing access to the frame, so that no other teacher or unauthorised person will be able to change them; the same goes for generating different types of reports, specifically 5 that were asked in the brief: full learner record, results for a particular module for a class group or across all class groups, a list of students for a module based on a grade (Distinction, Merit, Pass, Fail), and a full results report for a class group. The system is user-friendly with the minimum amount of input required (selection rather than typing), navigation is simple, and all the appropriate information is displayed everywhere where it is needed.

The main critical decision that I made is when I decided to stick to my own design and create different frames for different actions, rather than keep everything in one. It proved to be more efficient, as navigation became easier and more intuitive, as well as all the buttons for the actions were located in appropriate places and with minimum additional actions required from the system. I also made a change to the specification when I decided to not only do the report of distinction students for a module, but also for other grades. I took responsibility for such a decision based on my own initiative and resourcefulness.

Kanban Agile project management methodology proved to be quite effective, as it allowed more space for decision-making thanks to visualisation of the tasks. Trello has sent me notifications about the tasks that should be completed in the near future and helped me to keep them up to date. GanntChart showed the detailed progress of the project overall, and in its parts, which encouraged me to keep going and see if I fall behind in some parts. All of this contributed to the success of the project.

I think my design is effective in terms of functionality, as it allows easy access (if you have the password or id) to the main actions. All of the main buttons are highlighted by bright colours, and most controls have tooltips, so that user would not be confused much. I am confident that it suits the requirements and can work properly when used.