|  |
| --- |
|  |

|  |
| --- |
| Java Group Project – Student Management System |
| Unit Leader: Dr. Mehmet Aydin Authors: Cevdet Basaran  Paul Smith  Cheslav Starinskis  Alex Drabek  Chad Glover |
| The aim of the project is to design a model of a student management system using a client-server architecture that accesses a database with the procedure of using SQL, similar to the one that is currently implemented by the university for students, staff and administrators. This involves storing all relevant student information, comprising personal information, course related information and financial information. The database is deployed on the server, and the users are given access to the database via the client’s interface.  The report demonstrates the recorded efforts of the group with the implementation of the system and analysis of the project encompassing meticulous details and evaluations. Functional testing was improvised and recorded to examine and approve the specifications of the system to produce the desired outcomes and to add a superfluous feature to the phase of testing which is an integral part of the success of the project. |



# Java Group Project – Student Management System

Contents

[Java Group Project – Student Management System 1](#_Toc356468488)

[Introduction 2](#_Toc356468489)

[The Tasks 3](#_Toc356468490)

[Required Features for Solution of Case Study 4](#_Toc356468491)

[Class Diagram 5](#_Toc356468492)

[The Role of the Database 5](#_Toc356468493)

[Coding Design 6](#_Toc356468494)

[Client/Server Architecture 8](#_Toc356468495)

[Problems encountered during the development process 9](#_Toc356468496)

[Conclusion 9](#_Toc356468497)

[Summary 10](#_Toc356468498)

[Appendices 11](#_Toc356468499)

[UML Diagrams 11](#_Toc356468500)

# Introduction

The project is designed in accordance to the cooperation of the group members participating to envision the processes of software development to design an imperative elementally developed system. The report outlines the procedures used in developing the system for the student management system. An in-depth analysis of the implementation of the project and an explanation of how the group coerced an effective communication method for the decision making which epitomises the professional approach of its members in general.

The group was able to assign tasks to each of its members according to the strongest attributes of that person whether it was coding ability, networking and organising the group’s arrangements and writing the documentation for this report. Each person was backed up with a helper for whatever task they were performing and the programming or coding was done in pairs when the group was not apart. Participating in meetings was organised and segregation of tasks, dividing the workload was implemented only to a certain degree so that the integration of all the parts of the system would be operational. A realisation that the developments had to be coherently working so that the parts of the system could be integrated into a whole curtailed too much individual work on tasks without communication and conferring. The group also planned to work together as much as possible on a specific task but a realisation of how slow progress could be was the disadvantage of that although any problems that occurred could be much more easily worked out by the group. The optimal approach was planned and attempted with success and only minimal mishaps which were amended by a rectifying enthusiasm from the participants. Any discrepancies between agreements were mitigated by consulting the other members who decided on the best solution for the circumstances.

# The Tasks

The design of the system depended on the completion of the following tasks:

1. Considerations about the case study were ascertained as a group. The group agreed on the features that were required to be part of the solution to the problem described, and on how they might be able to deliver these features. This included determining what data may be required to support functionality in each case. A collection and description of these assumptions is subsequently provided in the latter part of this report.

2. An appropriate and comprehensive Class diagram was developed based on the Use Case diagram provided in the brief. All the features that the group agreed on are included in the Class diagram. A draft was used initially to assist the programmers in the group finalising the prototype of the Class diagram from eventual feedback from programs used and conclusive influence of the definitive approved solution.

3. The group developed an appropriate database to store all student information and retrieve it once needed for operations and dissemination (which can be defined as a sender sending information, the receiver collecting the information processing it and sending information back).

4. The group coded and tested the critical features of the Student Management System analysed using the programming facilities of the Java language such as I/O file operations, networking classes, JDBC etc.

5. The group developed graphical user interfaces or GUIs for each of the forms that are used within the system as a user interface.

6. Client/Server architecture was used to devise the whole system.

7. The Database was deployed on the server side, while the users access via a GUI-based client program.

8. The database was carefully designed using knowledge of SQL and improvising in skills to check the design which was expertly converted into SQL using SQL converter software to speed up the processes. This demonstrated the ingenious behaviour of the group relative to aspired computing skills and critical thinking ability.

# Required Features for Solution of Case Study

The solution to part of the problem was devised by the group within two meetings at the start of the project. The group discussed the features needed for implementation for the functionality of the program. The group devised the following for the solution; MyPHPAdmin for database purposes, and GUI’s needed for purposes such as login, registration, admin, student, and staff. Furthermore the group needed to create a server and client for communication purposes utilising MyPHPAdmin, plus a database for information to be read from and written to. Unified Modelling Language diagrams were created so we could see the functionality of the program at different levels.

The features as previously discussed needed to be detailed as to what each feature is used for, and inclusion of the data involved within the features. The task of creating the UML diagram was quite time consuming, because the group had to carefully think about the class diagram by thinking how many classes were going to be needed and what was required in them.

MyPHPAdmin is a MySQL RDBMS (relational database management system) tool that can handle administration of databases, by either updating or deleting information within a particular database. The group designed the database table requirements after the UML diagrams were devised. Furthermore, the group was able to identify relationships between tables and establish what were primary keys, foreign keys and unique constraints. For example, we established what was going to be identified as a unique constraint, because a person’s bank account details cannot be the same as another so therefore it is unique.

The GUI’s that were created all have special features such as their sophisticated layouts and finely honed which was an attractive presentation for the human eye. The data behind all of them had different layouts such as; Group-bag layout, grid-layout, card-layout and border-layout. Using these layouts made the GUI’s more users friendly with box down menu text fields, and various other attributes which can be observed much more straightforwardly directly in the demonstration of the GUI’s.

The server and client was devised by the group with some careful planning in mind, because the client is merged with all the GUI’s, and this is achieved by merging the GUI classes in a package. So when a specific request is handled exemplified as follows, for example: when a user is requesting certain information the request is sent to the server from which the data in the background of the server is handled by a switch statement which contains specific SQL statement commands, which then in return is processed and the information is sent back through the server which the client/GUI’s receives and delivers an output with the correct required presentation of it, if it is a vital requisite.

# Class Diagram

The class diagram was prepared in the analysis stage. However, during the implementation stage the presented solution required modification in the class diagrams in order to improve overall functionality and user experience. This clearly shows that the Waterfall methodology is not suitable for this project and therefore the group considered end-user needs equally with the functionality. On the late implementation stage as a group we agreed to switch to Iterative and Incremental development methodology to be used in this project. This decision had huge impact on ability to correct the logical errors made in the design stage as well in the overall outcome of the project. The outcome of each iteration was presented internally to the group members.

On the illustration it can be seen that the class diagram presents and describes two packages of classes with relations between them. All of the classes are needed in order to provide whole functionality to the client.

# The Role of the Database

The role of the database is a very crucial role, because certain aspects of the database needed to be contemplated to ascertain the appropriate functionality. The way this was devised by the group was by conferring altogether about the draft UML class diagram we created originally so that we could split actions between each user. There are three types of users that we took into consideration, which were; the administrator, staff which are not the type of administrator staff but the other type, and students.

The group’s thought’s on what should be included for a student within a database is what they would be able to view with the GUI’s for student when using the program. This was characterised as follows: a student can view their grades, units and course details. So in effect these requirements were inserted into the database. Furthermore, the staff’s information in the database was talked about in great detail as to what functionality would be needed for the staff. The requirements are as follows: view student grades, student courses, units, and updating grades, units or student courses.

This database was created firstly within Microsoft Access and after the correct relationships were established we used a software program to convert the database to MySQL which has saved us some time. However after conversion the relationships were broken and INNER JOINS needed to be re-established for the full functionality of the database. The way the INNER JOINS are established is within each command executed between clients.

The administrator serves a very important role within the system, because they have control over details between staff and students. The administrators control can be anything from updating to deleting details whether it is concerning staff or student. The administrator also has privileges to create a new database or delete an existing one.

With the database created there are five tables inserted into the database. There are four tables’ which can be identified as unique, because as discussed before bank details cannot be repeated and require a unique constraint. Furthermore students, staff and courses also have a unique constraint because their identification is in fact different from one another and no student ID, staff ID or courses ID can repeat itself. All of the tables except results have a primary key and the purpose of this is so there is no repetition of data throughout the database. It also is unique identifier for the rows of the database which contains individual data for each attribute within.

There is a password field in staff and students tables which when the program is run, and a staff or student user tries to log in there is a verification method used with an SQL statement that checks the username and current password before proceeding to the required GUI for that user.

# Coding Design

The functionality of the coding achieves everything asked for by the brief of the group project assignment. Firstly when the program is executed the main login GUI run’s and this presents three logins in one. The three logins which can be used is from a choice of staff, student or admin. The administrator cannot be considered to be general staff but of the sort that has special privileges (for administrative purposes) so therefore has been identified as unique and is distinctly exclusive, because the administrator as previously mentioned in the report has full control of entries and deletions within the database.

When the student logs in to the system they are presented with three buttons on their GUI, which are view, update and logout. The view option presents a drop down menu which ‘behind the scenes’ in the data is formed from card layout. The options presented to the student are: view their current grades, furthermore they can look at what units they are currently taking, and view their course details. This provides great functionality for the student to see what their current status is within the current year in terms of grades information, and consequently what units they have left to take. The update option is for updating information such as email address, phone number and change password option. As for all GUI’s there is a logout button which is clearly conveyed as to be self-explanatory.

If a member of staff is using the system they are presented with the following options: view options such as student grades, students on the course and students on a particular unit. Once again the data ‘behind the scene’ to see these options is articulated to be card layout which presents a drop down menu. The options as previously mentioned are self-explanatory and do not need elaborating on. The update option will entail updating student’s grades only. Furthermore there is a logout function for staff when they have finished completing the required task, which as mentioned before is self-explanatory.

The GUI for administrating purposes serves a very important role, because of database administration editing. The GUI for admin can update, delete, whether that is staff or student information. The options on the GUI are as follows: there are separate tabs for student and staff, and within those tabs are options to view, update, clear and delete information. Additionally there are two extra buttons which include going straight to MyPHPAdmin or closing the actual GUI itself. The data used behind the scenes to create for the GUI is JTabbedPane layout, which presents a clean and attractive layout for the buttons and tabs. There is a box in the middle of the GUI which is for the database representation of the required table when the admin selects a particular field or tuple for administration purposes.

The registration GUI is for staff and for students only and this does not include administrator registration, because this would in effect create a very unsecure system so that anyone can register as an administrator which could create havoc for the system. The group decided on a staff registration page to bring extra functionality to the registration system. Originally there was an option for registration for an admin, however this was not carefully thought through at that time of production within the group project.

Within the registration system there are all the features that can be expected such as: title, name, last name, DOB, gender, address, postcode, telephone number, email, bank details, University ID, and password. The data used ‘behind the scene’ for the GUI was achieved by creating a grid-bag layout and grid/card layout. Choosing this method created a very appealing and attractive visual GUI for the user, and this happened to be easily read and understandable, due to the intuition of the group’s creativity and decisions.

The problems were tackled well within the group where coding ranged. Everyone in the group are strong programmers, however there were two people that stood out within the group that was consistent at delivering GUI’s and server/client architecture at a functional and visual level. All of the group had and shared opinions whether things could be improved at different stages, i.e. the design and layout of the GUI’s and SQL prepared statements.

# Client/Server Architecture

Inspired by the presented solution for the server and a client structure during the lecture, the client server architecture was implemented and extended by another server in this case My SQL server. The proposed solution has got two layers. Firstly, Client is communicating through the network with the other Java application called a middle server which will maintain the communication with the My SQL server and the client, furthermore it is acting like a middle man in the communication.

This provides a large field of possibilities in terms of adding more functionality on top of this structure such as a statistics service and other services provided by a middle server apart from communication to the database. The proposed structure makes a huge improvement on the security by separating the database server from client and extra validation on the middle server.

Moreover, this is an easy way to maintain large numbers of clients while keeping a possibility to improve and change the database without disrupting the client. Once change is required only the middle server needs to be changed and there will be no need to update each client.

However, the proposed complex solution when a whole amount of traffic goes through the middle server cause delays and response time could be improved in some cases.

GUI’s use general class/packages to invoke communication methods, and then the functionality of intensions of communications that induce the following; while button is clicked it sends a chosen message to a server. A server listens to a new message and processes the message. The processing within a server has got a couple of stages, firstly, the server is determining which option the client is requesting whether it is view information, deleting or updating. Secondly it determines the type of user which requests information and chooses/adjusts appropriate SQL statements for a specific type of user. However, another stage, which is not the last, is that the server is querying the SQL database and listening for a response. The latest stage is processing and forming a response to the client with a complete set of results.

# Problems encountered during the development process

The group at first seemed very strong in different aspects through the start of the project. However some problems started to arise during the Easter break, because one of the group members encountered family problems but proceeded to work until things became apparent that this person needed a break. When the group came back the other group members needed to take over that person’s role. However the group’s communication through the Easter period left much to be desired. After the Easter break the group re-devised a second action plan because it was clear the first one, for various reasons including those previous mentioned required this. The coders approach for the solution to the problem was very complex and some of the code was new to them and had not been taught by the University, so in effect they stepped out of their comfort zone. However this solution even though it was complex was very good because it gave the user fantastic functionality and it was easy to understand when being used. The coders achieved this by studying examples on oracle, implementing these examples and expanding them until they were satisfied they would use that specific layout for that GUI. Another problem that stood out was the use of JTables within the GUI’s as this was a new concept to the group and very advanced. Advice from the internet and other sources was needed to achieve full functionality across all GUI’s except the login GUI because that was a basic layout used in the design process. The group came to the conclusion of having created a very successful interesting project, doing what was necessary in order to do so.

# Conclusion

The reflection of the group about the project is one of optimism about the experiences gained and lessons learnt. The system is satisfactory touching on all the points required by the brief. It was acknowledged that designing a perfect system with all components included and coherently working integrated together can be quite difficult to achieve. The logic used to obtain all the functionalities were not lacking in the group. More thought could have been put into requirements gathering and working together which may have suffered marginally due to the time restraints and priority related circumstances that bear importance in the organisation of a less professional environment and due to the problems that the group encountered. The group worked well together and communicated extremely well most of the time but realised that the tasks had to be somewhat divided to obtain results, especially more towards the end. At times people in the team required more help but it would not be available until structured meetings were organised which were sometimes hard to ensure or arrange for whatever the reason. Overall the project was very fine and completely successful. The system incorporated features that were specified in the criteria and more in terms of coding design which is well presented. This is quite an achievement considering the difficulties and time restraints. The group concluded that the project could have been improved with the following which they wish to implement in future projects: extra security measures using authentication for passwords and users which was done to some extent anyway; but would be an interesting domain to investigate i.e. such as the securing of systems implementing Java such as Java web applications and their platforms, etc., which is an immense topic for discussion.

# Summary

The group project aids to envision the software development processes required in delivering a complicated system. It showed the group how to innovate designs that can be complementary in that the thought processes can be used to develop intricate, elaborate and sophisticated systems used in Software Engineering. It can also be a stepping stone for innovation within this field of Computer Science and Information Technology, thus the nature of system simulation being mainly specialised for the purposes of new developments. Technology is a facet of modern everyday life, people being surrounded by software and systems everywhere. This is a very interesting phenomenon that can now be monitored by the onlooker to see how this relatively new discipline will advance; and by the oncoming generations into the indefinite time period after the present, which will inevitably be pursued by mankind for voluminous forthcoming improvements in technology.

# Appendices

## UML DiagramsD:\Users\Administrator\Desktop\Class\(default package).jpg



