2016-17

Explorations in Computer Science

Object Orientated Software Development C++

Keiran Brown 15000194
Callum Owen-Bridge
Karl Myers 15008411
The Engineers

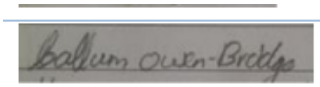
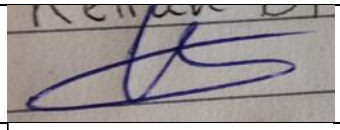
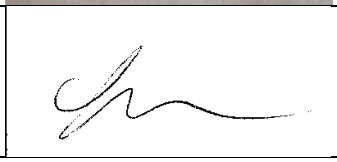
Plagiarism Statement

I confirm that this assignment is my own work, no part of it has copied from any other person's work (published or unpublished), and any text that has been paraphrased has been cited in the body of the document with the full details of the original work included in the reference list.

I also confirm that I have read the guidance provided at:

http://www.hope.ac.uk/gateway/supportandwellbeing/studentadministration/understandingyourdegree/

I also confirm that I understand the consequences of academic misconduct and that it may result in termination of my studies at Liverpool Hope University.

Name	Sign	Date
Callum Owen-Bridge		22/03/2017
Karl Myers		09/03/17
Keiran Brown		20/03/2017

Contents

Executive Summary 1

Requirements 1

Response 1

Benefits 1

Requirements 2

Functional Requirements 2

Non-functional Requirements 2

Installation 3

User Guide 6

Programme of Study 6

Courses 11

Adding Course 13

Assigning Assessments 14

Amending 14

Displaying 16

Deleting 16

Students 17

Displaying 19

Adding 19

Amending 21

Deleting 22

Assessment Marks 33

Security Considerations 38

Contingency Considerations 39

Reports 39

Executive Summary

This section outlines the solution requirements and the response to these requirements. The benefits of the chosen solution are also discussed here to validate why the solution has been chosen.

Requirements

What is required by Liverpool Hope University is a data structure able to store details of many students, courses, assessments and programmes. This data structure must also manage the relationships between these various entities. It must also store student enrolment into a course, course and student assessments, and finally, programme courses.
What is required on top of this is a useable interface. This interface is needed to allow a user to easily enter data about these various entities and their relationships.
This application and the data storage must be secured and have contingency options in place to ensure integrity of data storage.
What is of the highest priority, is the storage of the data correctly. Data must not be lost, and it must not become corrupt.

Response

A relational MySQL database will be created to allow storage of large amounts of data. This database should also take care of some of the security issues, as rules can be put into place on a per user basis.

To enable interfacing with this database by a non-experienced user, a console application will be developed using C++. This application will be able to send SQL commands to a server using a MySQL/C++ Connection framework.

The C++ application will use classes to represent various entities within the database, these will act as manipulators of those corresponding tables, these classes will share information to perform tasks for a user. Mediators will be used and the main program will have some functions that involve many tables within the database, such as outputting transcripts.

This application will be able to correct the user's inputs to try to reduce errors as much as possible, also searching through students can be made easier, as this application will implement a keyword search.

Benefits

Storing the data in a relational database will be beneficial to this solution over simpler methods such as a flat file database, which whilst applicable to this task, lacks the ability to easily set rules between tables and select and use keys for data searching.

Using an SQL database also means that users can be implemented, each of these users will have a password and their permissions can be set so that they may only view certain parts of the database.

Another benefit to this solution is the implementation of the keyword search, allowing users to misspell items and still find what they are looking for, as only most of the characters will need to match an item to be viable for the search. The advantage of using a console application as a solution to this is that this application will not require much computing power, this means it will be easier to run other applications alongside this one.

Requirements

Functional Requirements

There are some actions that must be performed by this project after completion. These actions are detailed here.

The system must be able to store details about:

- Students
- Assessments
- Courses
- Programmes
- The Relationships between these items.

The relationships will need to be stored in tables that use keys from all other tables to relate the items together. For instance, to relate a student to an assessment, a table "*student_assessment*" is needed, it would store keys from both student and assessment tables and would hold information such as the mark and grade of that student.

The system must account for user error. It must allow users to correct mistakes and verify data that users have entered. Whether by having the user check what they have entered, for instance a postcode is not always the same so only a user can check this. Although, a piece of data such as a date can be checked by the application, as these can be incorrect. Dates must also be checked for not only whether they are correct or not, but also whether they are correct in context to what they are describing. For example, a deadline date for an assessment, cannot be in the past. This means that the program must know what the current date is and have the ability to compare the input date by the user to the current date.

Another type of data that will need verification is items that have rules within only a university. One such item is weightings. These must equate to 100 from all courses inside a programme of study and across all assessments within a course. This means that all marks are accounted for. The application must check all inputs and verify that they add up to 100, if they don't then the application must act, and get the correct input from the user.

Non-functional Requirements

The location is important with this application, as it uses a database to function. Should the database be unavailable, this application will not work, as it will not know what to manipulate. Although, if the computer running the application is on the same network as the database and the address is known, the application will work. This location requirement can be made less stringent by having the database hosted and accessible through the internet, although, this means security becomes more of an issue.

A piece of required software is the MySQL database that is used by this application. It doesn't matter which form of application is being used to create and host the MySQL database, if the address can be found then this application will know how to interact with it.

Some support may be needed to maintain this application and the connected database. The application should never make any mistakes when entering into or amending the database, although, should some issue occur, somebody will possibly need to fix the issue using SQL commands. This requires a knowledge of SQL and requires some form of an administrator to perform these tasks. This person could also be responsible for making high level changes to the database. For instance, deleting a student can be done using the application, however, a tutor may not have permission to do so and so an administrator will have to perform this task.

Some training may be required to use this application along with the database. An important item is error messages. By informing users about the various error messages that occur within the database and application will enable users

to be more productive whilst using the application. Users would need to know what error is telling them they do not have permission to view or edit something, or when they have entered a duplicate key into the database and will need to enter a new one.

Installation

To install this application requires a few steps and resources. The following resources are required:

- .cpp and .h files
  - Database
  - Course
  - Date
  - Mark
  - Menu
  - Programme
  - Programme_course
  - Student
  - Student_course
  - Student_programme
  - Transcript
- Main_Program.cpp (This doesn't need a header file)
- Visual Studio 2015 Update 3
- Uniform Server (Or something similar able to run a MySQL Server)
- MySQL Script Detailing the design of the database

Step	Installation Contents
1	Database Setup
2	Application Setup
3	Visual Studio Version Verification
4	Building the Application – Creating a Project
5	Building the Application – Adding MySQL References
6	Building the Application – Adding the C++ Files

Database Setup

Uniform Server is an application that is recommended for testing this application. It is built to be used with any server as long as it is uses "MySQL", although, for testing, Uniform Server makes running the program quick and easy.

If Uniform Server is not already downloaded, it may be downloaded from the following link:
https://sourceforge.net/projects/miniserver/

Should that link be broken or out of date, the following link leads to the website for the application:
http://www.uniformserver.com/

To install Uniform Server, open the executable (.exe file). This will start the installer, make note of where the program installs to, otherwise it can be difficult to find. By default, it will install to where it is launched. After this get the SQL_Database.txt file containing the SQL code, then copy all of the text from within the file. Open Uniform

Server and a window will open with a few options available. If this is the first time running the server, it will ask for a password, type in a password that will be remembered, make note of this password.

Click on the option "*Start MySQL*" to start the server. Do not start the Apache server. Now click the button marked "*Server Console*", this will bring up a console. Type into the console "*mysql -u root -p*" then press enter. It will now ask for the password, type that in exactly as it was set earlier. If typed correctly, it will login successfully. Now, right click in the window to paste all of the SQL text from the file that was opened earlier, this will create the database. After this typing "EXIT" twice into the console will exit the server as a user, but the server will still be running as indicated by the green light in the main window.

Application Setup

Visual Studio Version Verification

Firstly, it is required that Visual Studio is installed with the C++ development support, it is also required that the user has Visual Studio 2015 and it is at "*Update 3*", if the user already knows that update 3 is installed, they may skip to the next section "*Building the Application*". This can be checked by clicking on "*Help*" at the top of visual studio after opening it and clicking "*About Microsoft Visual Studio*". This will bring up a window in which the Update version will be detailed.

Should the update version be incorrect, go to the following link to retrieve the correct version:
https://go.microsoft.com/fwlink/?LinkId=691129

That link is found at the following page on the Microsoft Website:
https://www.visualstudio.com/en-us/news/releasenotes/vs2015-update3-vs

Building the Application – Creating a Project

After the version has been verified, go to "*File*" → "*New*" → "*Project*" click that option and a window will open, allowing a project template to be chosen. Within this window, ensure that the pane on the left-hand side has selected "*Installed*" → "*Templates*" → "*Visual C++*" → "*Win 32*". Should the "*C++*" option be missing, please re-install Visual Studio but ensure that C++ options are chosen during the installation. After this click on "*Win 32 Console Application*". Put in the desired project name in the "*Name:*" box and click "*OK*".

When this step has been completed, a Wizard will open, click "*Next*". Ensure that the "*Console Application*" radio button is selected and under "*Additional Options:*" you will likely have to deselect "*Precompiled Header*" and "*Security Development Lifecycle (SDL) checks*" boxes, as these can interfere with the project compilation and aren't needed. Ensure that the "*Empty project*" box is ticked and click "*Finish*".

Building the Application – Adding MySQL References

An empty project will now be open inside Visual Studio. To begin, right click on the project inside the "*Solution Explorer*" on the right-hand side of the window, (It will have the same name as was given to the project) and click "*Properties*". On the left-hand side, expand "*Configuration Properties*" → "*General*". Under "*Project Defaults*" click on the option to the right of "*Common Language Runtime Support*" which will bring open a drop-down menu (It is likely set by default to "*No Common Language Runtime Support*"). Set this option to "*Common Language Runtime Support (/clr)*". This will allow the "*References*" section of the "*Solution Explorer*" to be shown.

Click "*Apply*" at the bottom of the properties window and then "*OK*". Next, in the Solution Explorer, right click on "*References*", and then click on "*Add Connected Service*". A new window will open, click on "*Find more services…*" at the bottom of the window. Another new window will open, ensure the menu on the left-hand side is expanded to "*Online*" → "*Visual Studio Gallery*". Now type into the search bar at the top right-hand side of the window "WCF Connected Service", Do Not Press The Enter Key this is an intuitive action, although it will close the search window! After typing in this search term an item named "*Visual Studio WCF Connected Service*" should appear, click on this

item and click "*Download*". This connector will then install, after this, click "*Close*" at the bottom of the extension window.

Now References may be added. Right Click on "*References*" in the "*Solution Explorer*" pane on the right-hand side of the window. Next, click "*Add Reference*". A new window will open with a search bar in the top right-hand corner. Search for "mysql" again Do Not Press The Enter Key. Some references will appear, tick "*MySql.Data*", if there are many, choose the uppermost item.

Next, search for "system", add the item called "*System*", again, if there are multiple, choose the uppermost item, do the same for "System.Data" which should have appeared with the search for "system" although it can be found by searching for "system.data". After all that, close the window using the button at the bottom marked "*Close*".

Building the Application – Adding the C++ Files

Now files can be added to this project. Get all of the "*.cpp*" and "*.h*" files and have them open in a file explorer window. Now, in Visual Studio right click again on the project (it will have the only the name of the project marked on it) in the Solution Explorer. Now click "*Open Folder in File Explorer*" Which will open the folder containing the project files. Move all the "*.cpp*" and "*.h*" files to this folder, adding them to this new project.

Right click on "*Header Files*" in the Solution Explorer. Next, click "*Add*" → "*Existing Item*". To add multiple items, hold "*Shift*" and click on each file wanted, in this case click on only the files appended with "*.h*", as they are clicked, they should appear in the bar at the bottom of the file selection window. After selecting all of these click "*Add*" at the bottom of the file adding window. After all of the files have been added, they should appear under the "*Header Files*" folder in the Solution Explorer. Repeat this process but instead right click on "*Source Files*" in the Solution Explorer and add only the "*.cpp*" files.

After completing all of these steps, press "*F5*" and the Visual Studio will begin to compile the application. A console window should appear, the application is now running, and on screen instructions can be followed. If the application cannot connect to the database, it will inform the user of this so they can fix the issue.

User Guide

Starting Up

When the application has started, the login screen appears (Figure 1a). This allows any user to login to the application. The privileges of that user then restrict what that user is able to do during their use of the application. This also stops unauthorised users from accessing the database and the functions of the application.

Figure 1a - Login Screen

Figure 1b shows the Main Menu. This is the first screen after logging onto the programme. This allows for navigation to any of the functions within the application. Each option has a number next to it, his is how navigation works within the programme. By entering the desired option via its associated number next to it, that option has been chosen, and the application will move into that option.

Figure 1b - Main Menu

Programme of Study

Displaying

To view a programme, the user must navigate to the programme menu as shown in figures 1 and 3. The user will then select one to view programme, here the user will be asked to enter a programme ID. If the database has the programme it will display the details otherwise the user will be returned to the second menu shown in figure 3. For convenience, the user can view another programme after previously viewing one.

The display function will show all the details of the programme as shown in figure 2

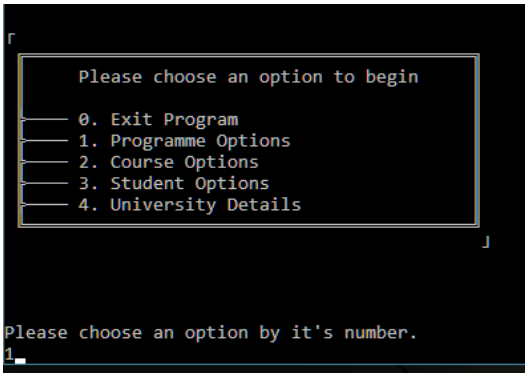


Figure 2 - Main Menu

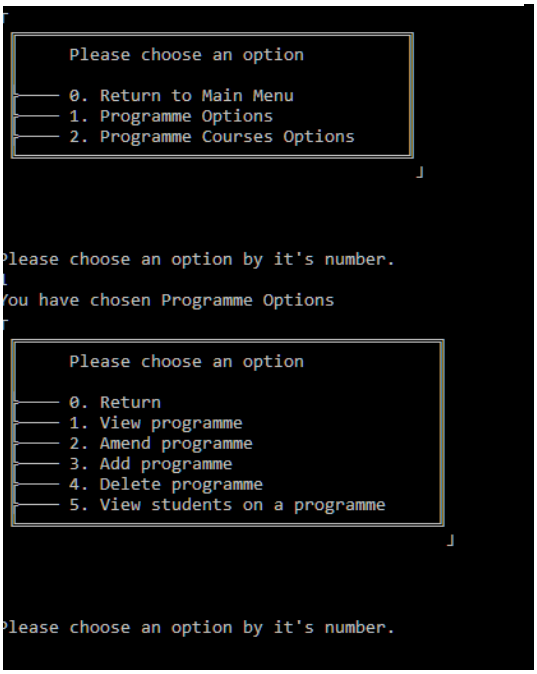


Figure 2 - Choosing a Programme Option

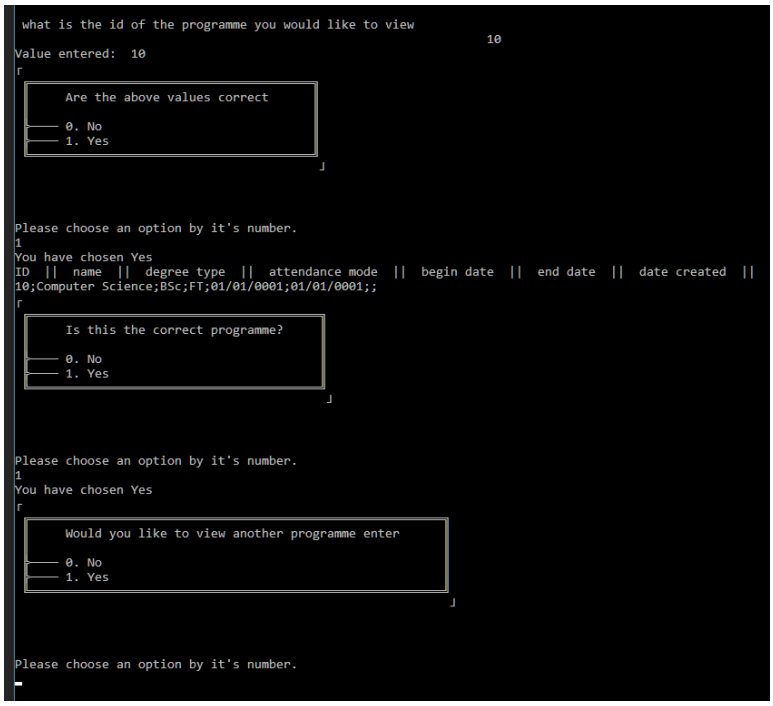


Figure 3 - Displaying a Programme

Adding a Programme

To add a programme, the user must navigate to the programme menu as shown in figures 1 and 2. The user will then select add programme. The user will then be asked to enter the programme name, research task name for level H, degree type, attendance mode, start date and end date. If any of these values are entered incorrectly the user can alter the value before entering the next value shown in figure 4. The user will be prompted if they enter an end date less than start date, preventing errors. The user will then be asked if the values are correct before entering the data into the database. If a value is incorrect the user can edit just that single value as shown in figure 5. An example of a programme being entered is shown in figure 6.

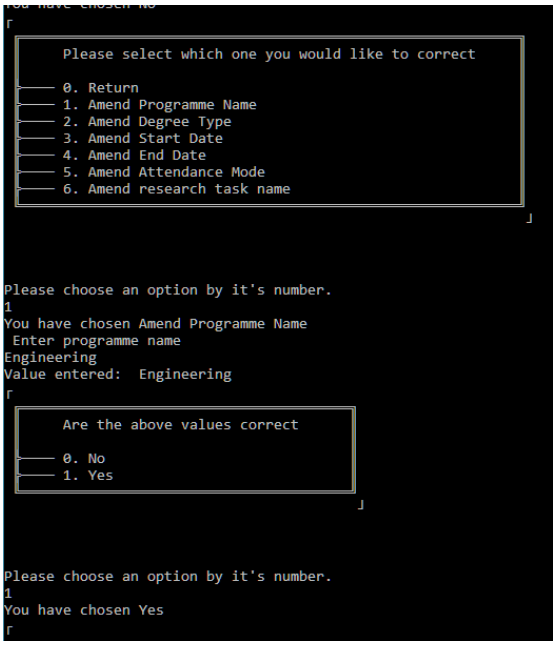


Figure 5 amending parts of input data menu

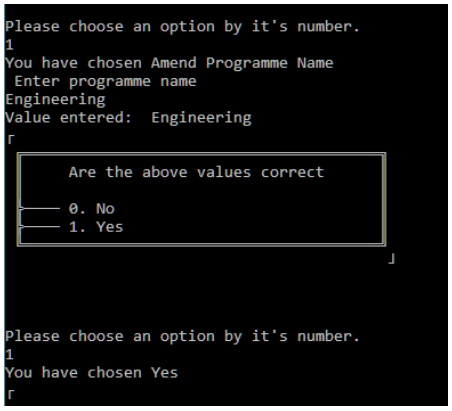


Figure 4 checking values are correct

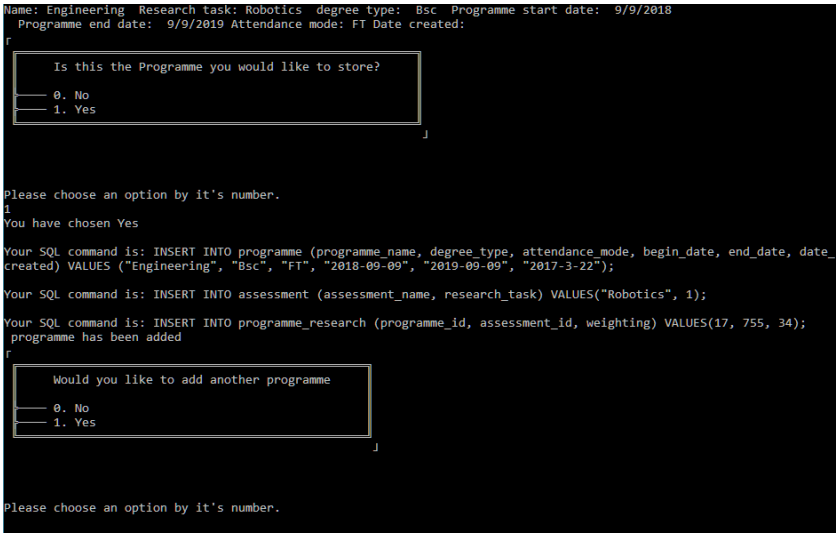


Figure 6 adding a programme

Amending

To amend a programme, the user must navigate to the programme menu as shown in figures 1 and 2. The user will be asked to enter the programme ID they would like to amend. If the programme exists its details will be displayed, otherwise the user will be returned to the programme menu. The user can edit the name, degree type, attendance mode, begin and start date. An example of a programme amendment is shown in figure 7

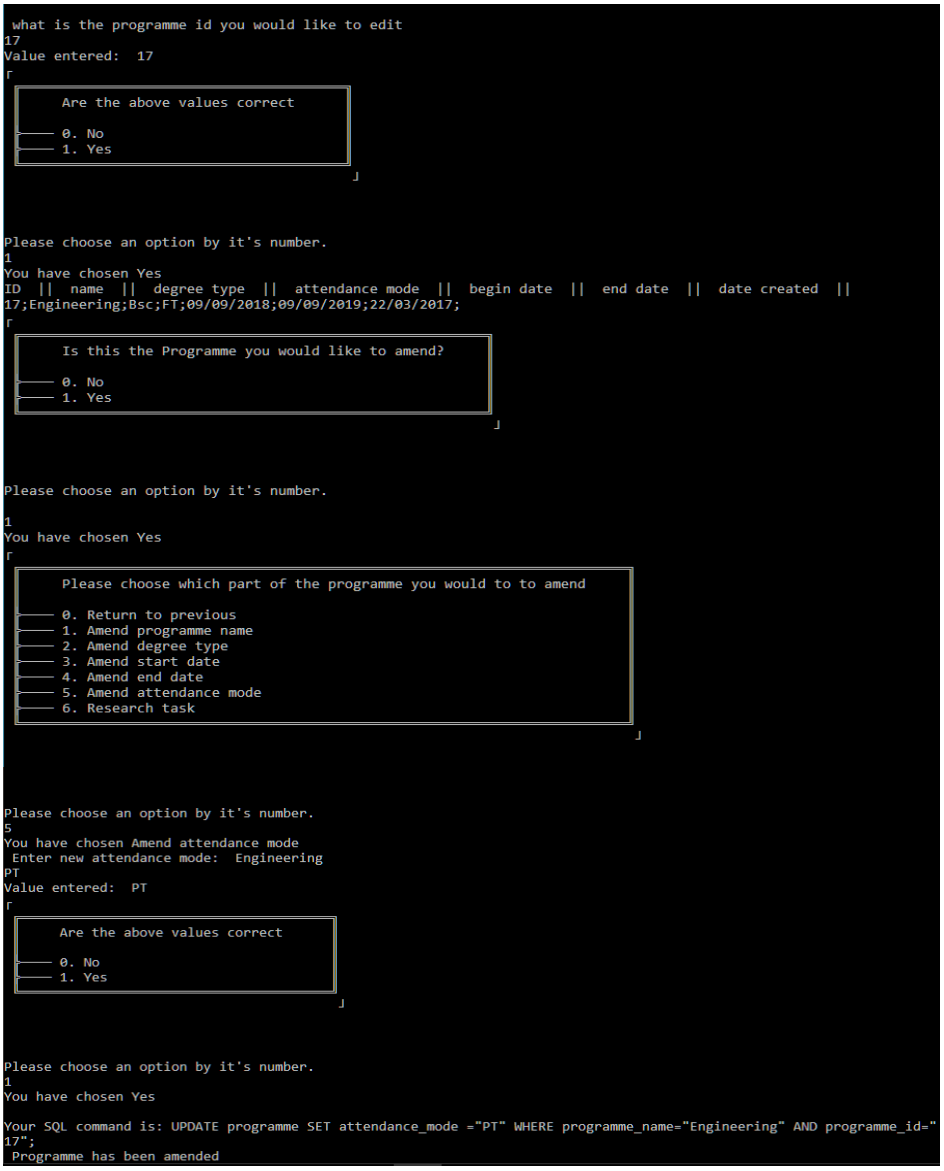


Figure 7 Amending a programme

Deleting

To delete a programme, the user must navigate to the programme menu as shown in figures 1 and 2. Once selected, the user will be asked for the programme ID they would like to delete. The details of the programme will be displayed, this allows the user to be sure it's the correct programme, otherwise the user can choose a different programme. An example of deleting a programme is shown in figure 8.

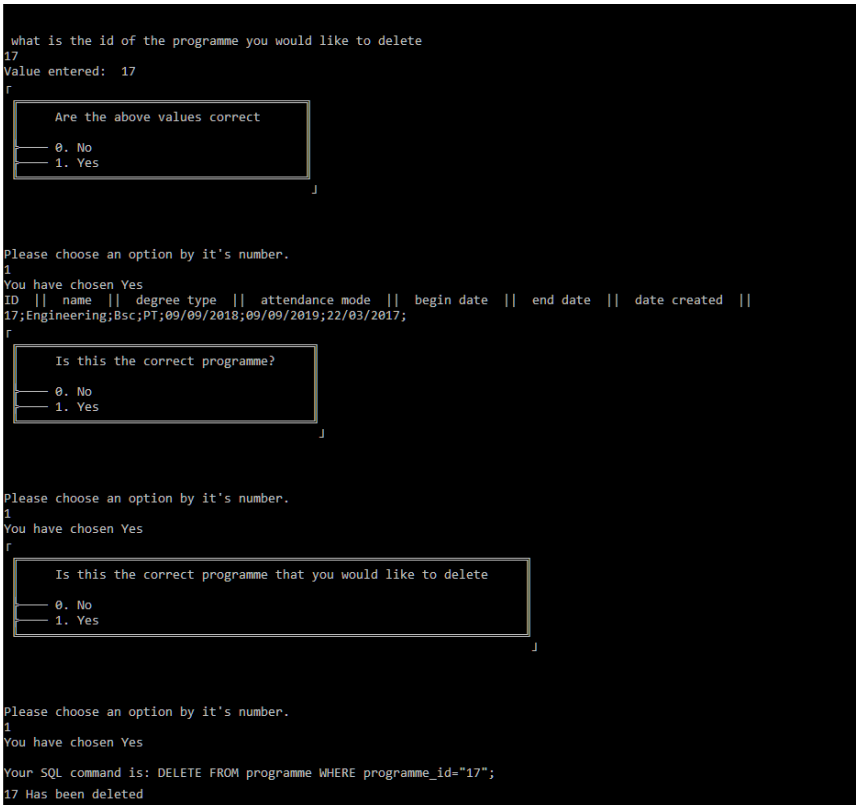


Figure 8 Deleting a programme

View Students on a Programme

To view students on a programme, the user must navigate to the programme menu as shown in figures 1 and 2. Once selected, the user will be asked for the programme ID and the details of the selected programme will be shown. If the programme is correct the programme will display the students enrolled on the programme; showing their first name last name and ID. If no students are present as shown in figure 9 a message will state that no students have been enrolled onto the selected programme.

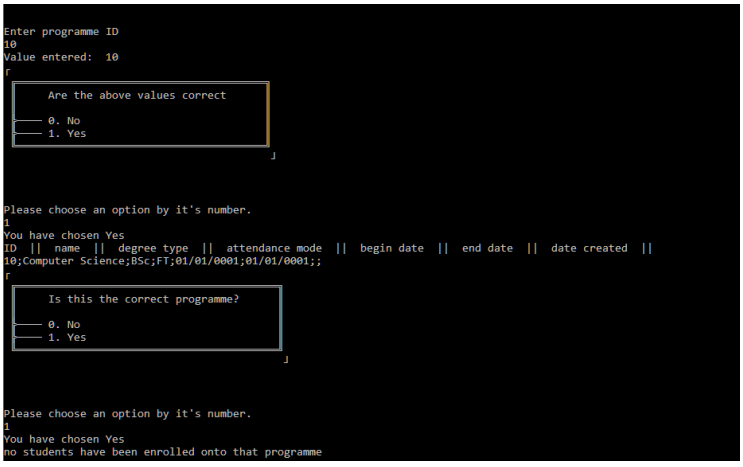


Figure 9 Viewing students on a programme

Programme Courses

Adding a course onto a programme

To add a course to a programme, select programme options (1) in the main menu, ten select programme courses options (2) in the next menu. In the programme courses options select Add course to programme (2) to add a course to a programme.

The programmes in the database will be displayed enter the programme ID as shown in figure 10. Then enter the number of courses on each level. Once entered, enter the course codes for each level of the programme and enter the begin and end years of the programme, shown in figures 10. A prompt will appear if the end date is less than the start date.

The entered values will be displayed to add them, choose option one otherwise option zero to change the values, as shown in figure 11. To do this function again select the yes option in add another programme course menu, as shown in figure 11.

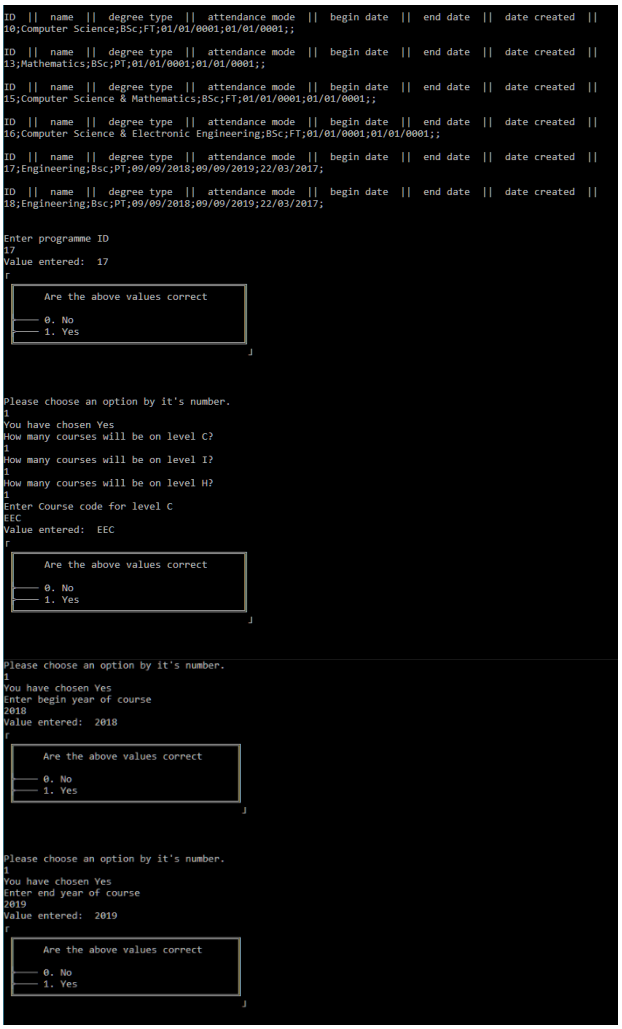


Figure 10 – showing programmes available, course, begin year and end year entry

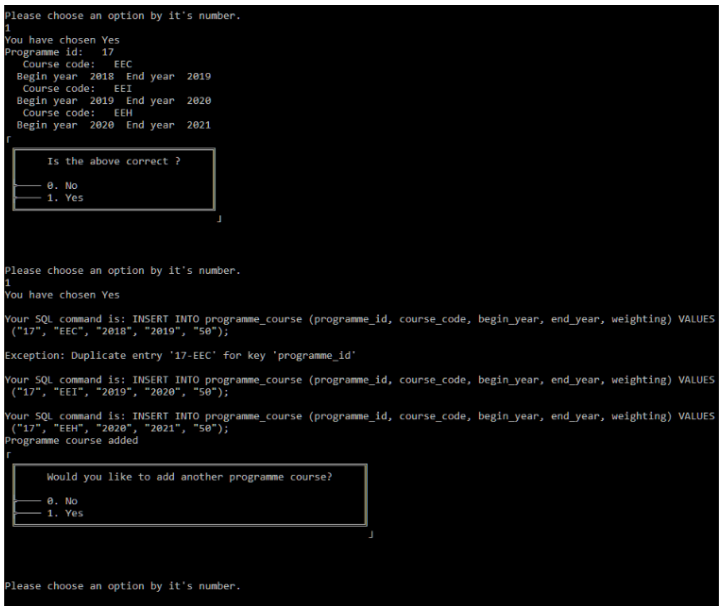


Figure 11 – Entering values to database

View courses on a programme

To view courses on a programme, select the programme options (1) in the main menu, then select programme courses options (2) in programme courses menu, then finally select view courses on programme (1). This is shown in figure 12.

Enter the programme ID to view the courses on the programme. The details of the programme will be displayed to ensure it is the correct programmed to be selected. Select yes if it is the correct programme. This is shown in figure 13.

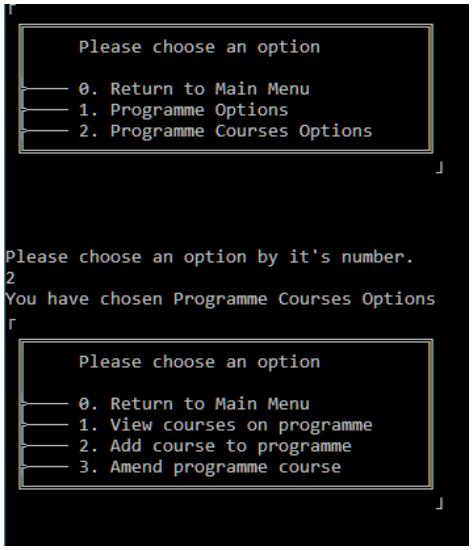


Figure 12 – Navigating menus to view courses on programme

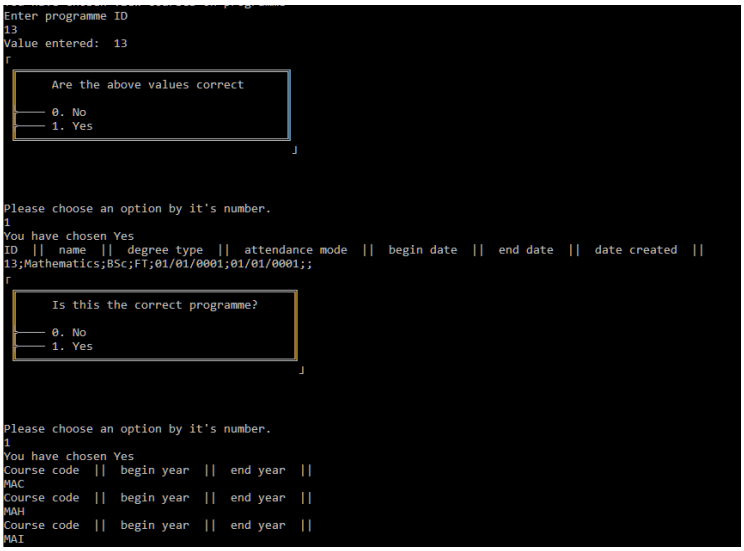


Figure 13 – viewing programme courses

If there are courses added to the programme the course codes will be displayed. As shown in figure 13.

Amend programme courses

To amend programme courses, for example the courses added, begin year or end year, select the programme options menu in main menu (1), then select programme courses options (2) in programme menu and select amend programme courses (3).

To edit courses added to the programme select option 1. To edit a of the programme containing courses select option 2. To edit end year of the programme containing courses select option 3. In figure 15 the begin year is edited. In figure 14 and 15 the course code is amended.

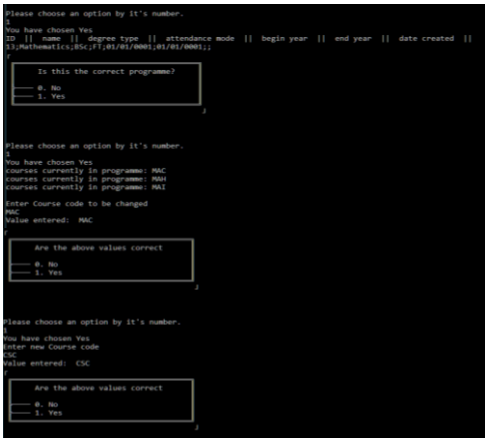


Figure 14 – amending courses on programme

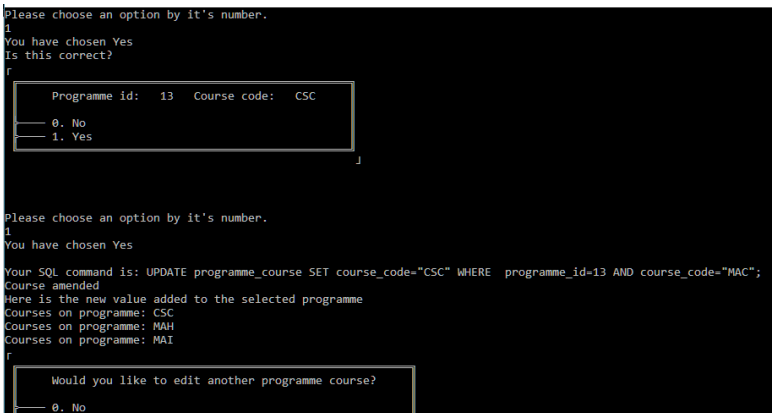


Figure 15 – Amending courses on programme

Courses

To add, display, amend, delete and assign assessments to a course the user must firstly, navigate to the course menu (fig 16) by choosing “course options” on the main menu.

```
============================
 Course Menu
 ------------
1. Add Course
2. Update Menu
3. Remove Course
4. Remove Assessement
5. Search Menu
0. exit menu
============================
Menu choice:
```

Figure 16 Course Menu

Adding Course

To add a course, the user must choose option 1 “Add Course” in the course menu. Firstly, the user is asked to enter the course code, then the course name. The user is then prompted to enter the level of the course, the level being either C, I, H or M.

If the user enters an invalid year, for example, F the program gives an error message, provides examples of valid entry and prompts the user to re-enter a correct year (Fig 17).

```
Enter the level (C, I, H or M) of the course: f
Invalid level value. Allowed values are:
 C for Certificate Level (1st Year)
 I for Intermediate Level (2nd Year)
 H for Honours Level (3rd Year)
 M for Masters Level (Post Graduate)
 you entered: f
C
Enter the level (C, I, H or M) of the course:
```

Fig 17 Invalid level Error message

After a valid level has been entered, the program prompts the user to enter the credit value of the course, valid credit amounts for courses are 0 for no bearing, 15, 30, 45, 60, 90, and 120. If an invalid entry is made, the program will, again, produce an error message, give examples of valid entries and prompt the user to re-enter a valid credit amount (Fig 18).

```
Enter the credit value of the course: 10
Invalid credit value. Allowed values are:
 0 for no credit bearing
 15 for 15 credits
 30 for 30 credits
 45 for 45 credits
 60 for 60 credits
 90 for 90 credits
 120 for 120 credits
Enter the credit value of the course:
```

Fig 18 Invalid Credit Amount Error Message

After all the entries have been made, the program displays the course details for verification and asks the user if they wish to save the record (fig 19). If the user enters no, the program returns to the course menu. If yes is entered, the record is saved to the database, and the user is moved forward to assign assessments to the course.

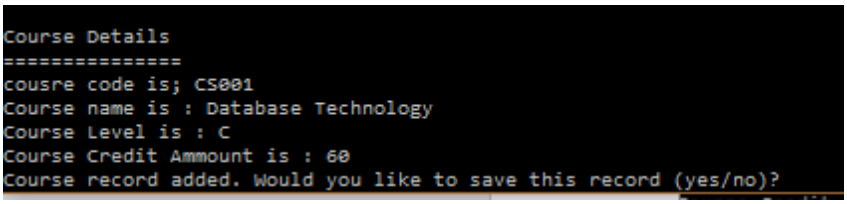


Fig 19 Course Details Verification

Assigning Assessments

After the course record has been saved to the database, the user the user is asked how many assessments are to be added, the user is then prompted to enter the assessment name, weighting and the deadline for the assessment. After the entries have been made for each assessment, the program displays the details of the assessment for verification and asks the user if they wish to save the record (fig 20). If the user enters no, they are returned to the beginning and asked to enter the assessment details again. If the user enters yes, the record is saved to the database. This is repeated for the number of times the user specified.

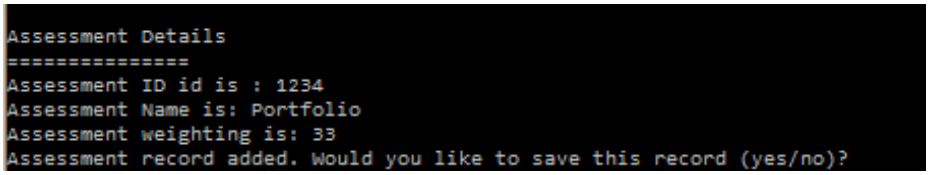


Fig 20 Assessment Details Verification

The total weighting for the assessments must equal 100. The program checks the total weighting of the assessments if it is found that the total weighting of the assessments is not equal to 100 the program produces an error message and the user is prompted to re-enter the weightings for the assessments (fig 21).

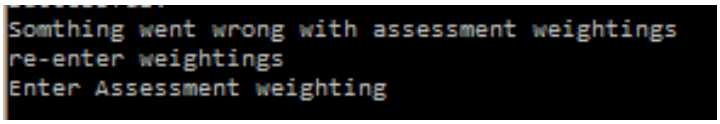


Fig 21 Assessment Weighting Error Message

Amending

To amend a course or its assessments the user must navigate from the course menu to the update menu, by choosing option 2 “Update Menu” (Fig 22).

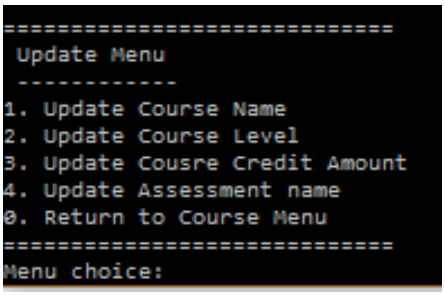


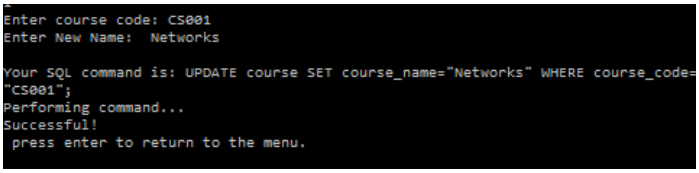
Fig 22 Update Menu

To update the name of a course, the user must choose option 1 “Update Course Name”. The user is then asked if they know the course code if the answer is yes the user is prompted to enter the course code and then the new name for the course.

If the answer is no, all the courses are displayed on the screen and the user is, again, prompted to enter the course code.

After the course name is changed the user is prompted to press enter to return to the Update menu (Fig 23)

Fig 23 Update Course Name



To update the level of a course, the user must choose option 2 “Update Course Level”. The user is then asked if they know the course code, if the answer is yes the user is asked to enter the course code and then the new level for the course. If the answer is no all of the courses saved in the database are displayed on the screen and the user is, again, prompted to enter the course code and the new level for the course.

If the user enters an invalid year the program gives an error message, provides examples of valid entry and prompts the user to re-enter a valid year

After the course level is changed the user is prompted to press enter to return to the Update menu (Fig 24)

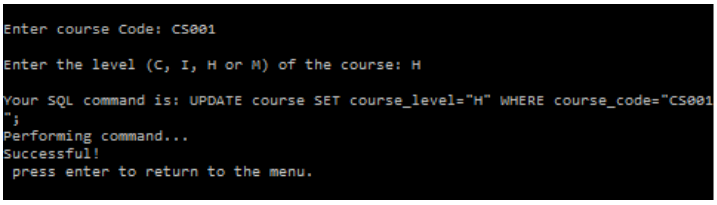


Fig 24 Update Course Level

To update the credit amount of a course, the user must choose option 3 “Update Course Credit Amount”. The user, as above, is asked if the course code is known. If yes, the user is asked to enter the course code and then the new credit amount for the course. If no, all the courses saved within the database are displayed and the user is then prompted to enter the course code and the new credit amount.

If the user enters an invalid credit amount the program gives an error message, provides examples of valid entry and prompts the user to re-enter a valid credit amount.

After the course credit amount is changed the user is prompted to press enter to return to the Update menu (Fig 25).

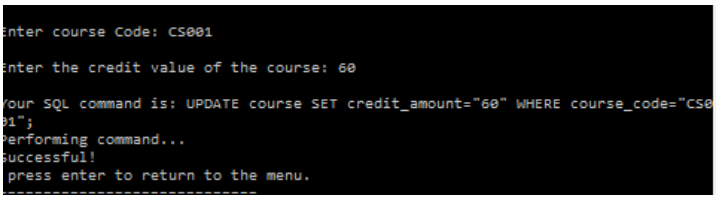


Fig 25 Update Course Credit Amount

Assessment

To update the name of an assessment, the user must choose option 4 “Update Assessment Name”. The user is then asked if the assessment ID is known, if yes the user is asked to enter the assessment ID and then the new name for the assessment.

If no, all the assessments saved within the database are displayed and the user is, again, prompted to enter the assessment ID and new assessment name.

After the assessment name is changed the user is prompted to press enter to return to the Update menu (Fig 26).

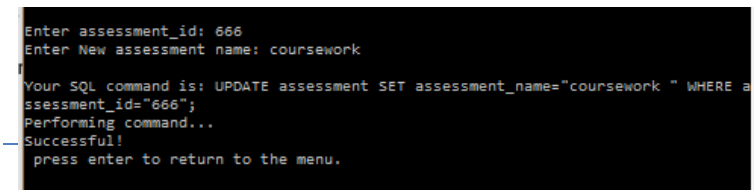


Fig 26 Assessment Name

Displaying

To display a course or its assessments the user must navigate form the course menu to the search menu, by choosing option 5 “Search Menu” (Fig 27).

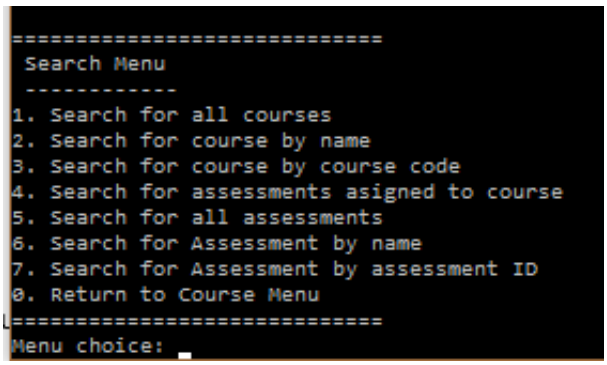


Fig 27 Search Menu

To display all courses, the user must navigate from the Search Menu by choosing option 1 “Search for all courses”. The program will then display all the courses saved within the database to the screen.

To display a single course, the user must navigate from the search menu by choosing, either, option 2 “Search for course by name” where the user will be prompted to enter the course name or by choosing option 3 “Search for course by course code” where the user will be prompted to enter the course code. The program will then display the course saved within the database to the screen.

To display the assessments assigned to a course, the user must navigate from the search menu by choosing option 4 “Search for assessments assigned to a course”. The user will be asked to enter the course code then all the assessments assigned to said course will be displayed on the screen.

To display all assessments, the user must navigate from the Search Menu by choosing option 5 “Search for all assessments”. The program will then display all the assessments saved within the database to the screen.

To display assessments by name or ID, the user must navigate from the search menu by choosing, either, option 6 “Search for assessment by name” where the user will be prompted to enter the assessment name or by choosing option 7 “Search for assessment by assessment ID” where the user will be prompted to enter the assessment ID. The program will then display the assessment, or assessments, saved within the database to the screen.

Deleting

Course

To remove a course, the user must navigate form the course menu by choosing option 3 “remove course”. The user will be asked if the course code is known, If the answer is yes the user will be asked for the course code (Fig 28) and the course will be removed from the database.

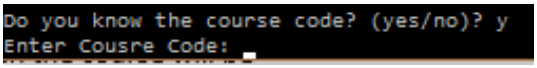


Fig 28 Removing a course by Course Code

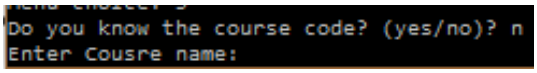


Figure 29 - Removing a Course by Course Name

If the answer is no the user will be asked to enter the course name and then the course will be deleted from the database.

Assessment

To remove an assessment, the user must navigate from the course menu by choosing option 4 “remove assessment”. The user will be asked the assessment ID, and the assessment will be removed from the database.

Students

To display, add, amend or delete a student the user must first navigate to the student editing menu. This is done by starting the application and choosing "*Student Options*" within the main menu, then the user must choose "*Edit and View Student Records*". It should be noted that this is different from student enrolment, which is done later. This section only adds students to the database so that they have records to be able to enrol them onto Programmes of Study later.

Student Search

Whenever a student needs to be found, the student search will likely be called. This allows the user to either specify a Student ID themselves, or to search using a parameter of the student that they know about.

The search begins as specified in figure 30.

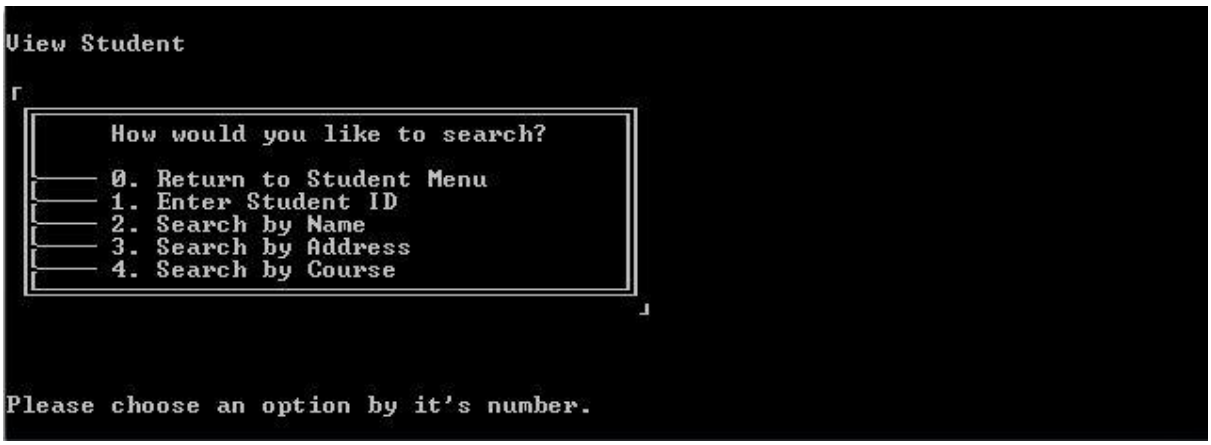


Figure 30 - Student Search Menu

After this, the user can choose which parameter they would like to search for (Figure 30), they all behave the same except for what parameter they search in, in this example, the student's name is searched for.

Next the user chooses what part of the parameter that would like to search for. For example, inside address, the following may be searched for:

- House Number (Specific, doesn't use a Keyword Search)
- Street Name (Uses Keyword Search)

- Town or City (Uses Keyword Search)
- Postcode (Uses Keyword Search)

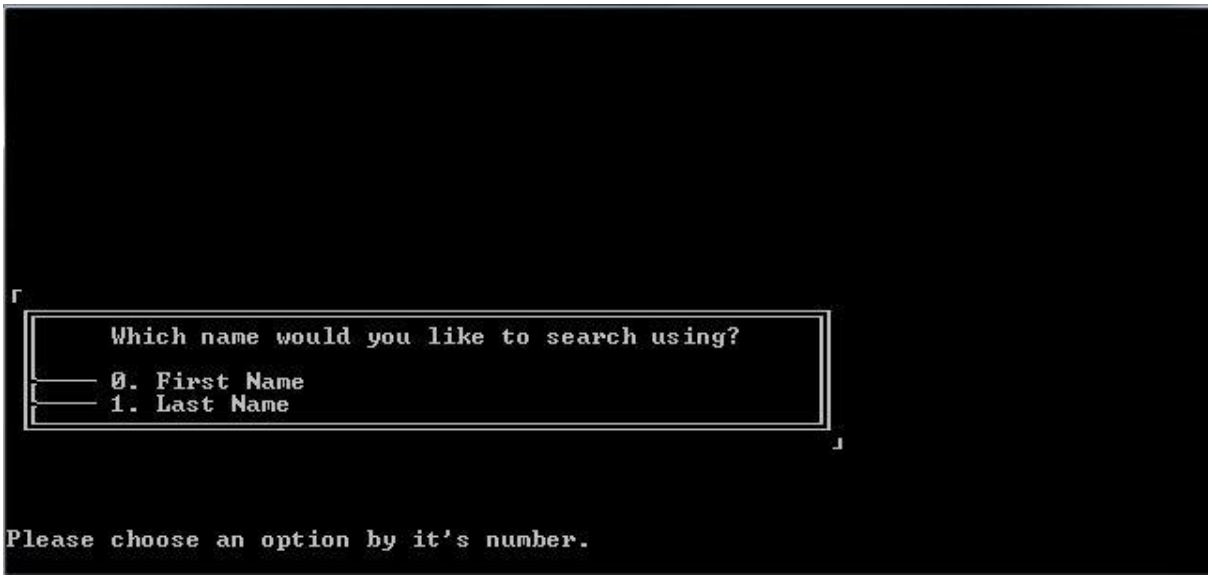


Figure 31 - Student Parameter Searching

After this, has been chosen (Figure 31), the keyword or specific term is specified by the user. Then a list of all search returns are shown (Figure 32), if applicable, these will be selectable from a menu for use in other functions (Figure 33).

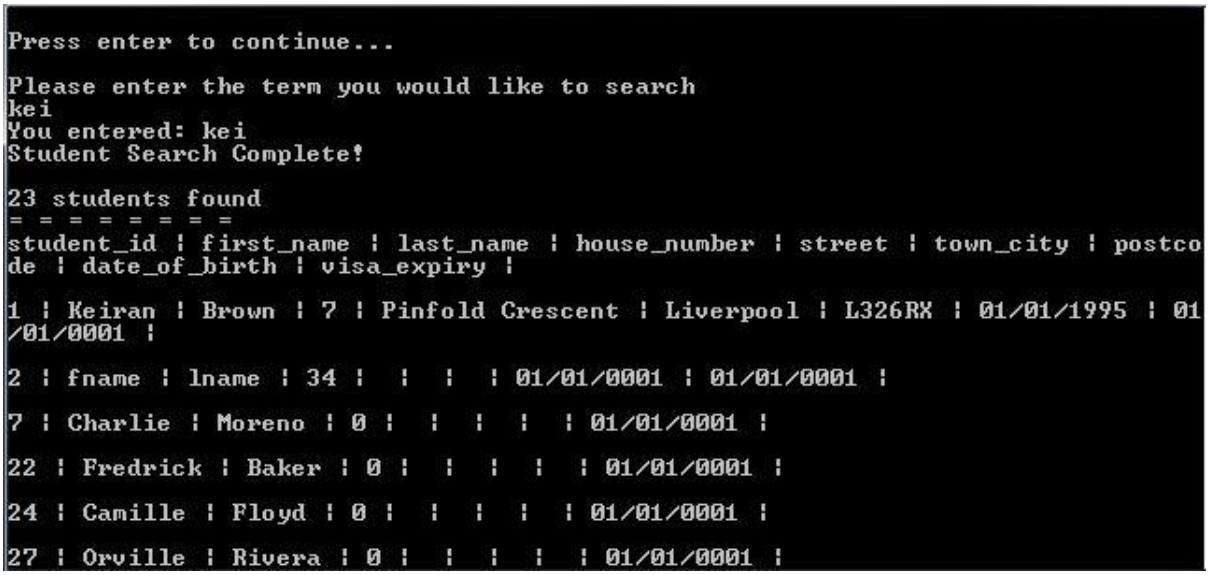


Figure 32 - Student Search Using Keyword

As can be seen, only strings that match at least half of the entered search term will match and be brought back(IDs 7 – 22 are missing, so did not match search term), this allows for flexibility in finding students and cuts down on shown values to the user.

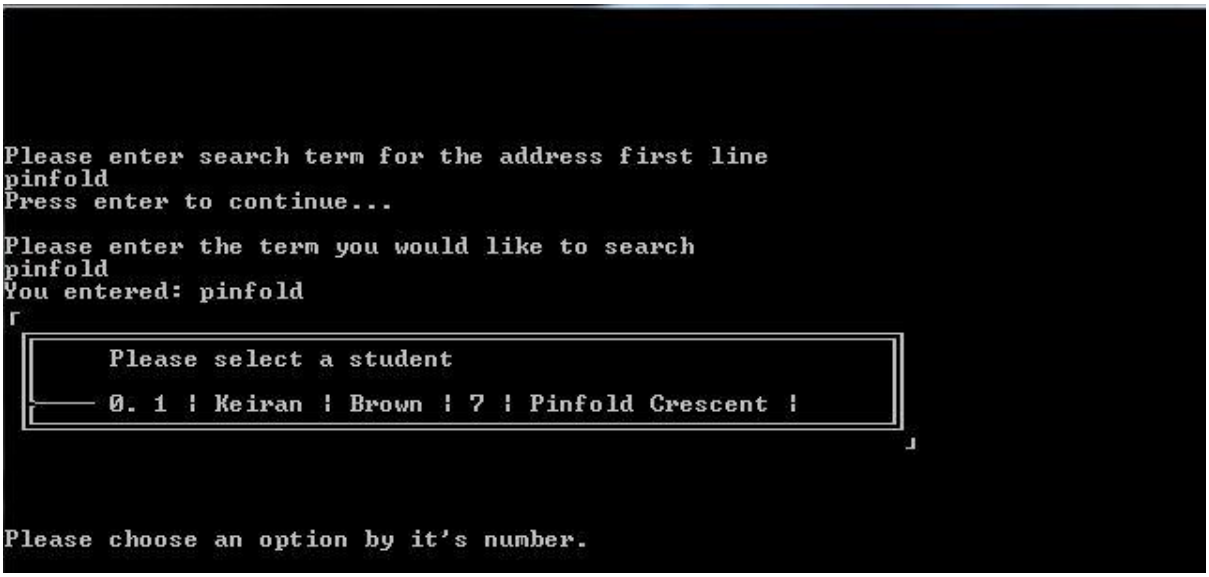


Figure 33 - Student Search with Use for Search Afterwards

Displaying

To display students, the user firstly chooses the option "view student" from the student menu. They are then asked if they would like to view all students or search for a particular parameter related to the student. Should they choose to search, they are taken to the student search function specified earlier.

After the search term is input, a set of students will appear matching the search term, this includes student IDs which make searching simpler if known, and a student count displaying the number of records matching that search.

Should the search term not match any students in the database, the application will report to the console that no students were found by printing "No Students Found".

Adding

To add a student, the user must choose "create new student" from the student menu. This will then ask the user if they would like to add a new student record, or if they would like to add a selection of many student records from a file.

Adding by File

Should the user choose to add student records by file, they will be asked to specify where the file is located, what delimiter if used in the file, newlines are also appropriate, and can be mixed with another delimiter. For Example:

```
12;Mark;16;Long Lane
56;Mary;7;Water Drive
```

This set of values is appropriately formatted for input by the program into the database. The delimiter could also be used in place of the newline.

After specifying a file path and a delimiter, the user must now specify what information is contained within the file. This is done using a menu displaying all the information the database can hold. The user must select the fields in the order they are in the file. This process is shown in figure 34.

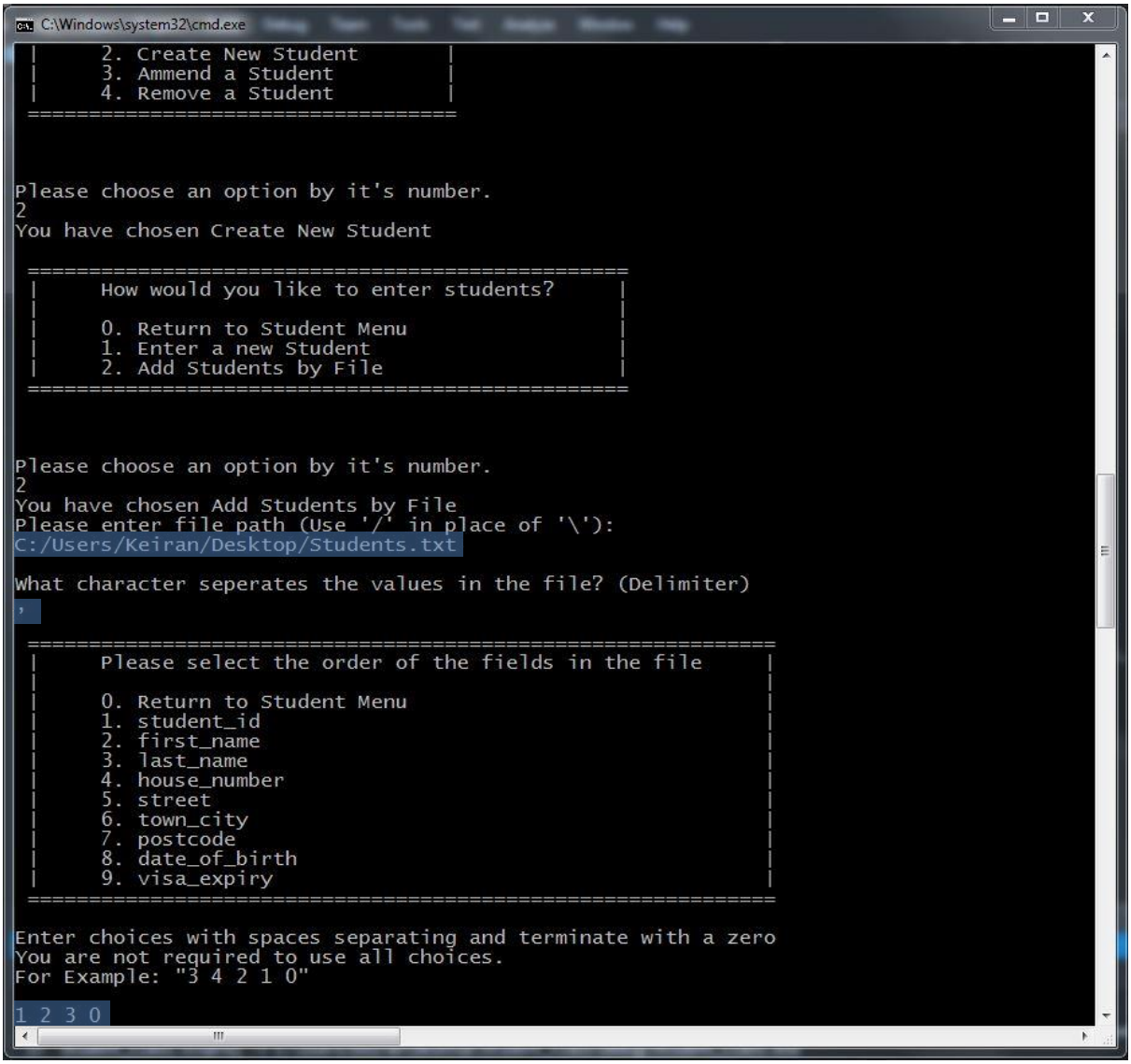


Figure 34 - The process of inserting student values from a file.

After this process the file is automatically read and dissected using the information provided. This information is then input into the database. The user it told what information is entered using the values gathered. If, however, for some reason, any of the specified student IDs within the file are duplicates inside the database, then the application will automatically amend the ID until there are no duplicates, the user is informed of the IDs that failed and the students are still entered with different IDs.

Figure 20 shows this process happening. The user has specified the location of the file, how the data is separated, and then the order of the data in the file. Note that Student ID can be specified here, this allows the user to be able to find the students afterwards, but they should be wary of students within the database already using these IDs. Should this occur, data will not be successfully entered, and the user will be notified of this.

Adding a Single Record

Should the user want to enter a single record, they should choose to enter a new student from the menu. After this they are taken through the steps of adding a new student. This involves entering each value in turn (Figure 35), but not the student ID, as this is done automatically.

```
└─── 2. Add Students by File

Please choose an option by it's number.
1
You have chosen Enter a new Student
Press return(Enter) for an empty value
Please enter first_name
fname

Please enter last_name
lname

Press return(Enter) for an empty value
Please enter house_number
1

Please enter street
street
```

Figure 35 - Entering a New Student's Details

If the user would like to leave a field blank or null in a record, for instance; Visa Expiry Date, they can just press enter to send an empty string to that field. This is not recommended for certain values as it can make searching more difficult, although, these can be filled in later using amending features in the application, this is also how any mistakes can be rectified.

Amending is not always necessary for mistakes however. After entering all values for the student, the values entered are displayed and the user has the option of re-entering a certain value by choosing the field from the menu. The values are displayed alongside the fields so the user can verify the data they have entered (Figure 36).

```
Choose an entry to amend it
┌
    Please verify that all of your entries are correct

└─── 0. All Inputs are correct
└─── 1. first_name - fname
└─── 2. last_name - lname
└─── 3. house_number -
└─── 4. street - street
└─── 5. town_city -
└─── 6. postcode -
└─── 7. date_of_birth -
└─── 8. visa_expiry -

Please choose an option by it's number.
```

Figure 36 - Verifying Student Record Values

After all of this is complete, the student is entered into the database and the data has been saved.

Amending

To amend a student, the user must first choose the menu option “amend a student” from the student menu. This then takes the user through the standard student search.

After finding a specific student, the user then chooses which field they would like to edit for that record. They are then prompted to enter new data for that field. After going through verification of that data the record is then amended in the database with the new data.
This process is shown in Figure 37.

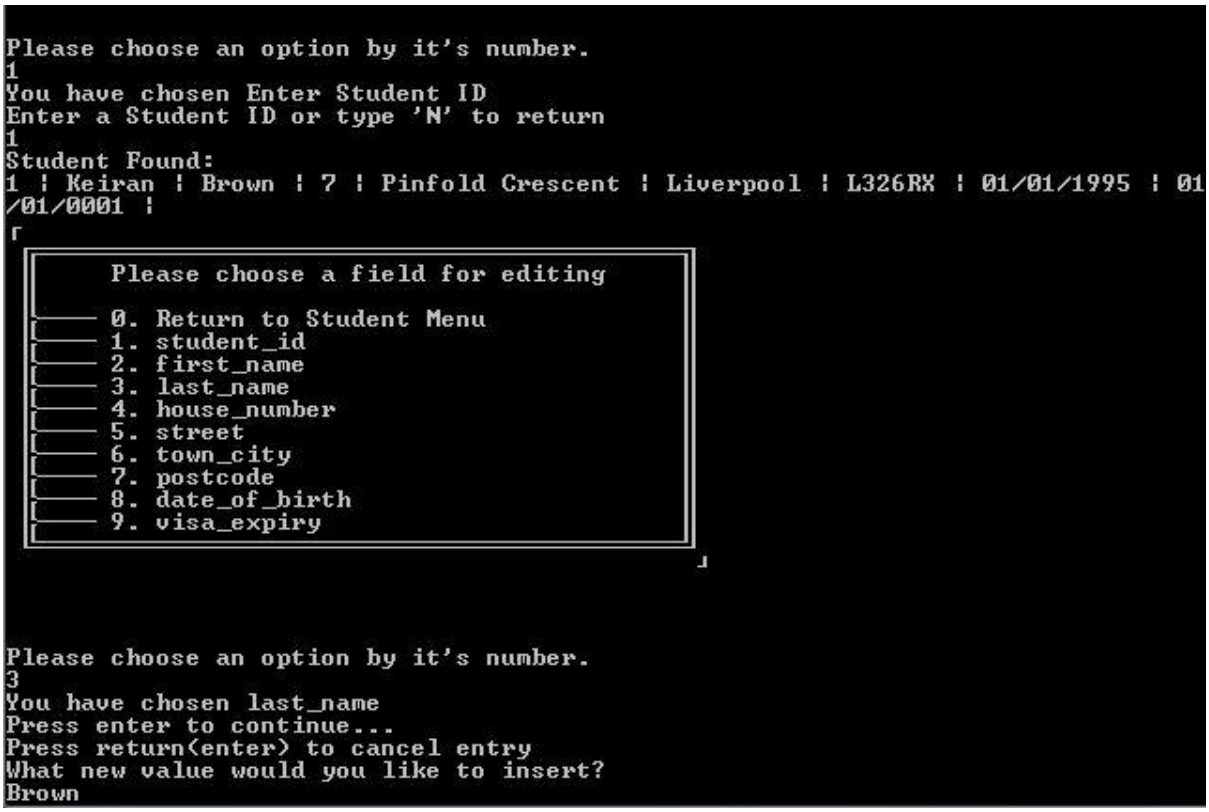


Figure 37 - Amending A Student Record

Deleting

To delete a student, the option "Remove Student" is first chosen. This then goes through the same student search as specified earlier. When a record is found, the record is displayed and the user must re-enter the student's ID to ensure verification of the deletion operation. This is a more complex verification than a simple yes or no answer, this makes sure the user must be paying full attention and will be sure of deletion as it cannot be undone.

The deletion process is shown in Figure 38.

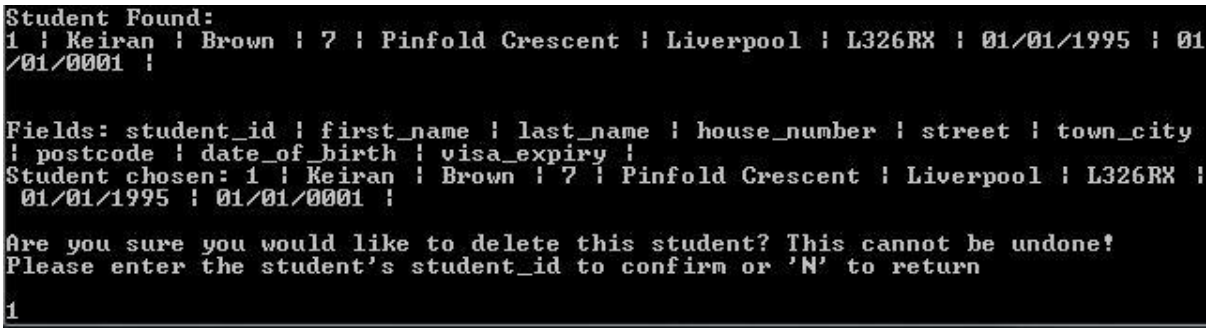
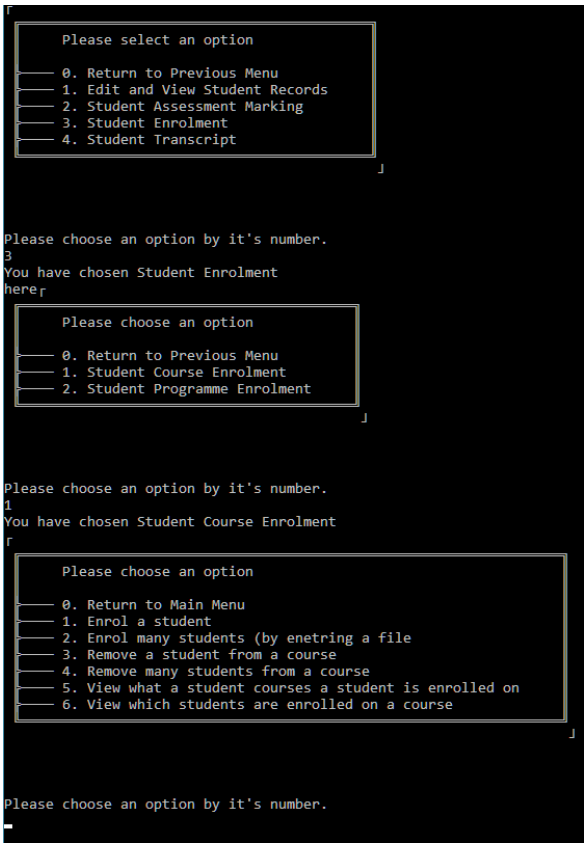


Figure 38 - Student Deletion

Enrolling Students on Courses

To enrol a student onto a course, the user must select student options (3), then student enrolment (3) and select student course enrolment (1). As shown in figure 40.

To enrol a batch of students, select option 2 in the menu. The program will ask the user to enter the file path of the students to be enrolled, the delimiter used to separate the students, for example a semi-colon. The user will be prompted if the file failed to open, the application will return to the student course enrolment menu if the file failed to open; as shown in figure 39. The user will be asked to enter course code for the students to be enrolled onto. The program will then enrol the students onto the course code entered.



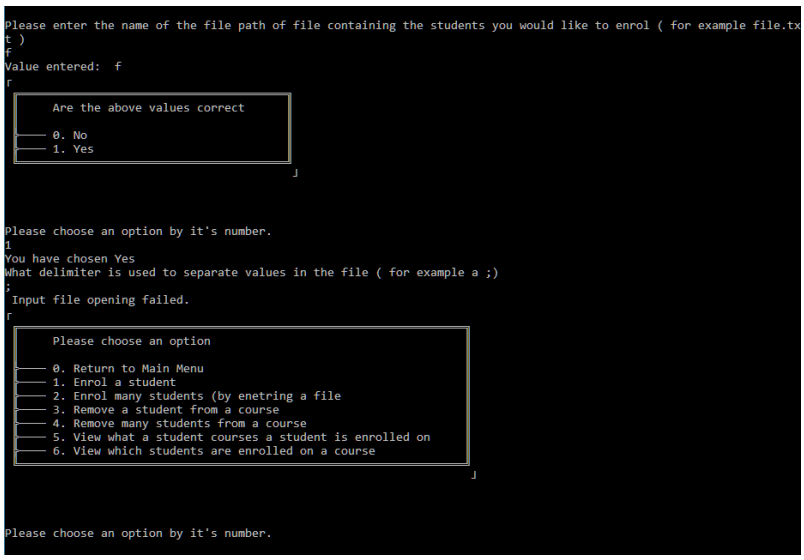
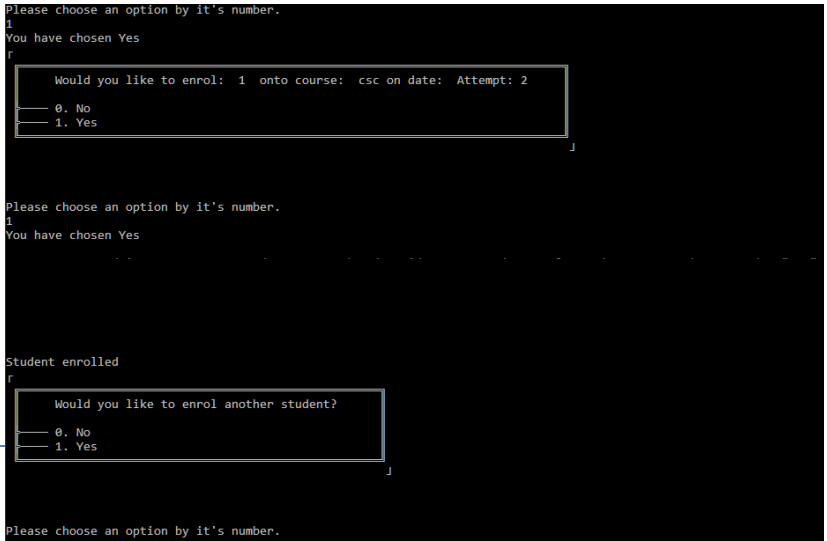


Figure 39 – file read failed

Figure 40 – navigating to enrol student on course

To enrol a single student onto a course code, select option 1 in the student course enrolment menu shown in figure 40. Enter the



course code for the student to be enrolled onto. Its details will be displayed, so the user can be assured they selected the correct course. Then, enter the student's ID number, this can also be found if the ID is unknown by selecting option 2 in the find a student menu shown in figure 41. If, the ID number is entered the details of the student will be shown. The user will then be asked if it is the first attempt at the course by the student selected; if yes then 1 will be entered in attempt if user selects no, they can enter a value.

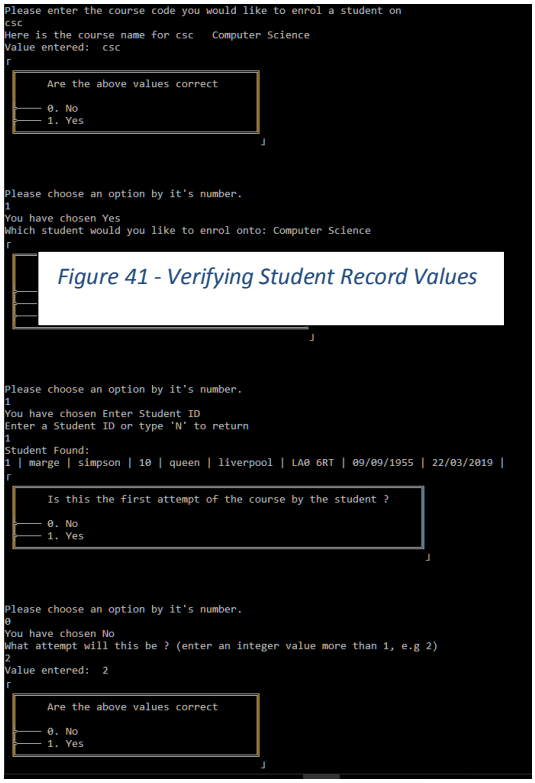


Figure 41 - Verifying Student Record Values

The user will be prompted if the values entered are correct, select option 1 to enter values into the database. Once, enrolled the student will automatically be enrolled onto the assessments associated with that course code. Figures 41 and 42 show an example of enrolling a student onto a c

Figure 42 - Verifying Student Record Values

Removing Students from Courses

To remove a student from a course, first select student options in the main menu, then select student enrolment, then select remove a student from a course and select the option to not use a file.

Once selected, enter the course code that the student will be removed from. The course name will be displayed and the program will ask if it the correct course, select yes by entering 1 and pressing enter, as shown in figure 43. Now enter student ID or search for student. In figure 44, the student ID was known and entered. The student's details will be displayed, the program will then ask if this is the correct student and course values for the student to be removed from the selected course, if so enter 1 for yes, as shown in figure 45. Otherwise, the function will restart.

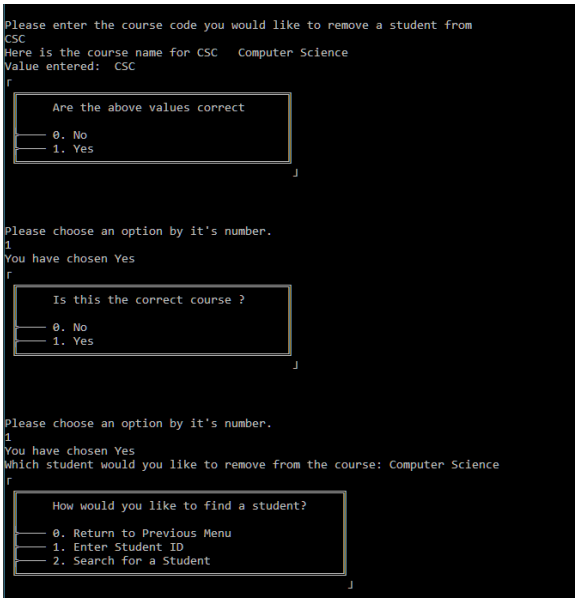


Figure 43 – entering course code and student ID

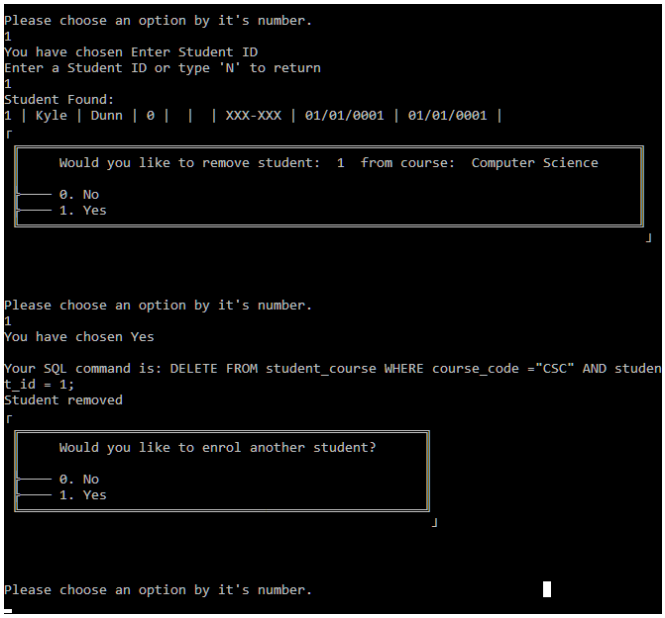


Figure 44 – removing student from course code

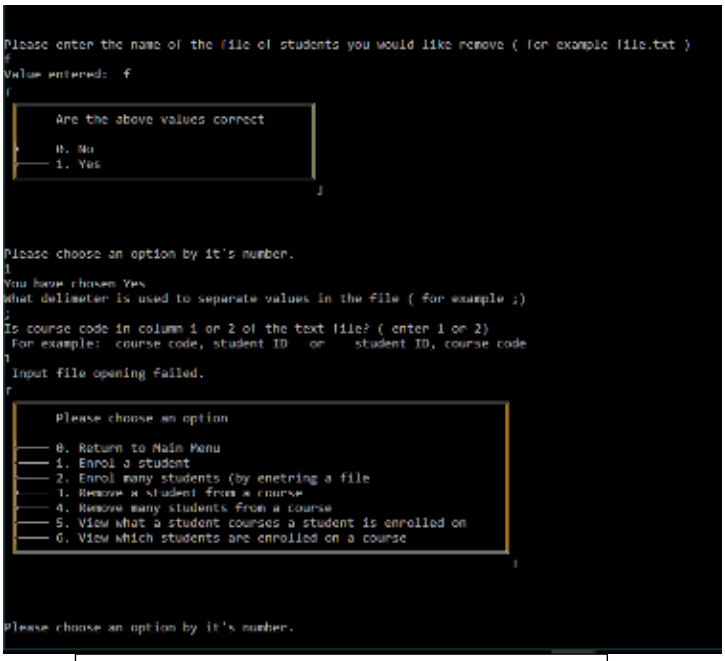


Figure 45 – batch removing students from course

View student courses

To view a student's list of courses, first select student options in the main menu, then select student enrolment, then select option 5.

Once selected, the program will ask for the student ID this can either be searched or entered in figure 46, the ID is known and so was entered. The student details will be displayed select yes by entering 1 if the student is correct, otherwise the function will ask for the ID again. The student ID and the courses the student has been enrolled onto will be displayed, as shown in figure 47. Otherwise, a message will state that the student is not enrolled onto any courses as shown in figure 46.

The course code can then be selected to view the student's current results. To do this enter the course code from what is displayed, as shown in figure 48. If it is the correct course code entered and the student has assessment marks, the current course result will be displayed as shown in figure 48.

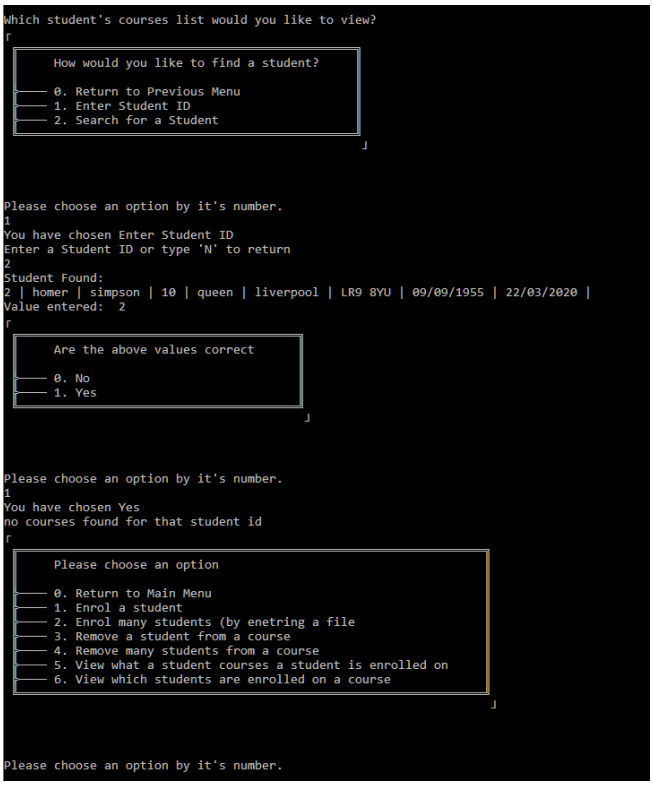


Figure 46 – student not enrolled on courses

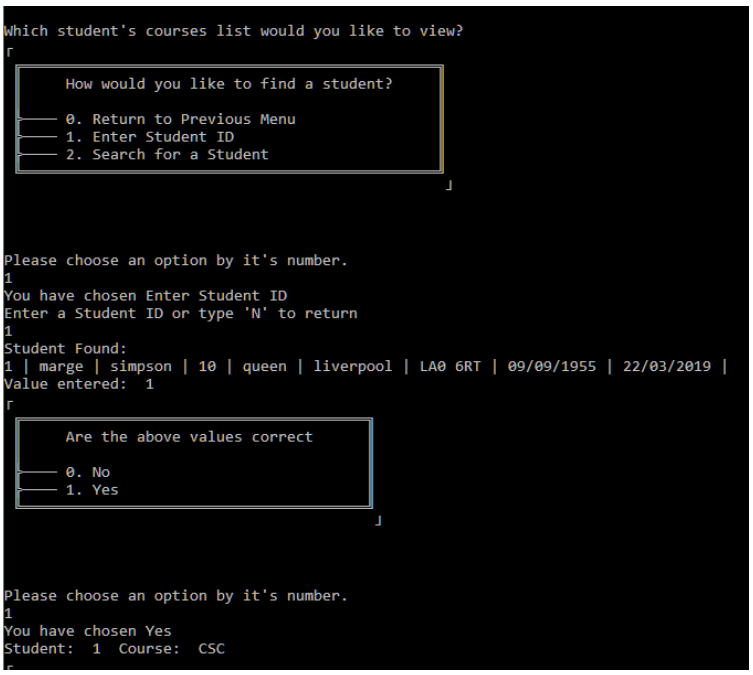


Figure 47 – Student enrolled on courses

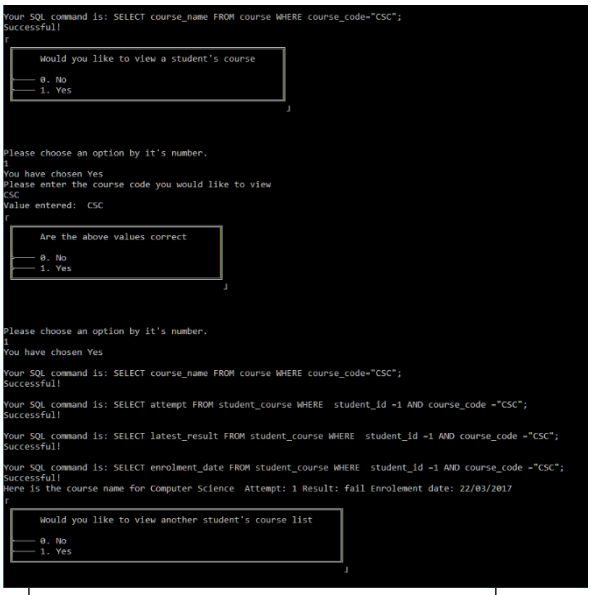


Figure 48 – Student course result

View students on course

To view the students on a course, select student options in the main menu, then select student enrolment, then select view students on course; option 6.

Once selected, enter the course code. Its name will then be displayed and select yes by entering 1 if the course code is correct. All the student ids enrolled onto that course code will then be displayed, as shown in figure 49.

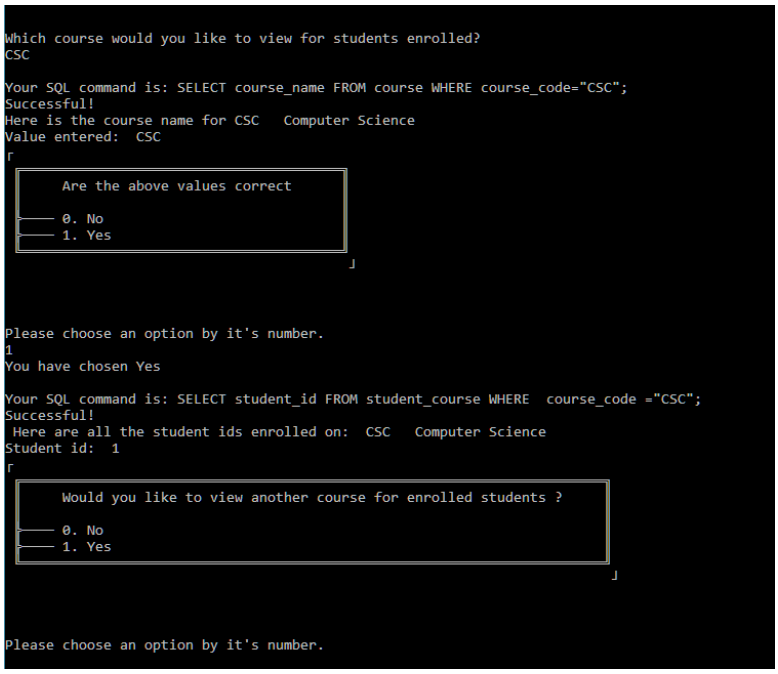


Figure 49 – Student enrolled on course

Student Programme of Study Enrolment

Batch enrolment

To enrol a batch of students onto a programme, select student options in main menu, then select student programme enrolment. To enrol a batch of students, select yes by entering 1.

The program will ask for the file path of the file to be entered. Once entered, enter the delimiter that separates the fields in each record for example a semi-colon. If the file opens successfully the program will read the file and enrol each student onto the program selected. It will then enrol them onto the courses related to that program. If the program fails top open the file it will return to the previous menu, and prompt that the file failed to open, as shown in figure 50.

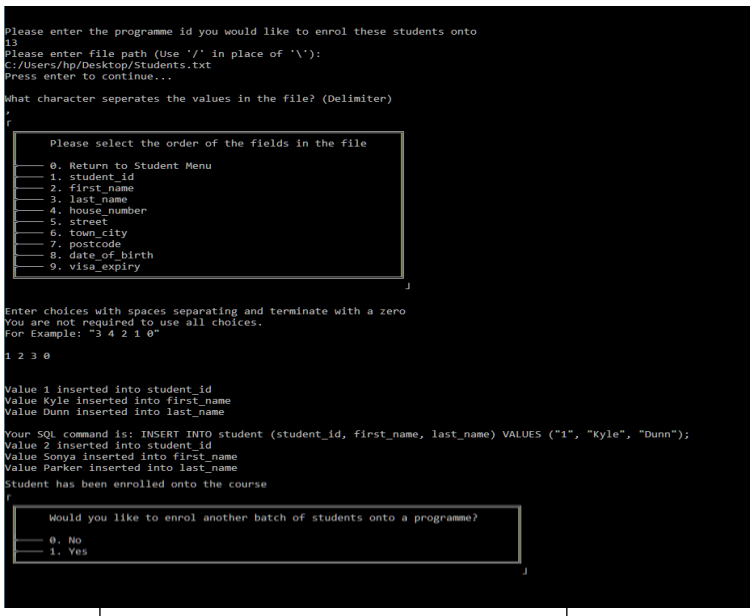


Figure 50b – Student enrolled on course

Single enrolment

To enrol a student individually, select no by entering 0, when asked to enrol by file. A list of programmes will be displayed, as shown in figure 51. Enter the student ID or search for it, as shown in figure 51. Then enter the programme ID to enrol the student on and the level they will be enrolled onto, as shown in figures 51 and 52.

The student will be enrolled onto the courses associated with the course, automatically. Now enter the attempts the student has made for the programme. Once complete, the student will be enrolled onto the programme and the courses associated with it and the assessments associated with the course. If a program has a research task the student will be enrolled onto it also.

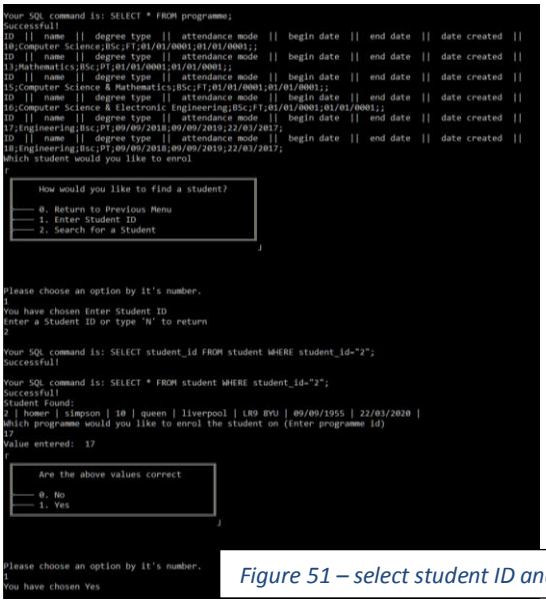


Figure 51 – select student ID and programme

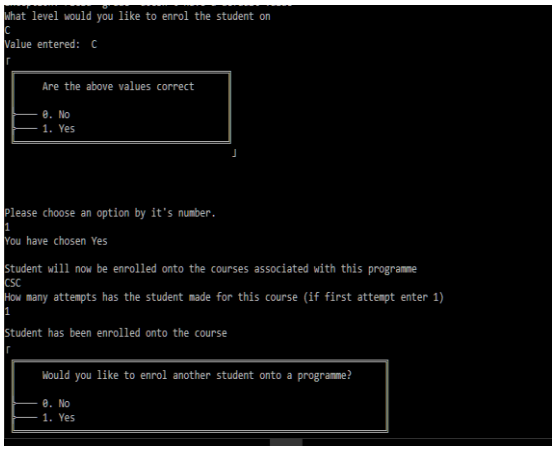


Figure 52 – Student enrolled on course

Withdrawing a student from programme

To remove a student from a programme, select option 3 in student enrolment menu.

Once selected, enter the programme ID, as shown in figure 53. Then enter the student ID or search for it as shown in figure 54, the ID was entered. The student details will be displayed to ensure it is the correct student, as shown in figure 54. If both the programme and student ids are correct select yes, the student will then be withdrawn from the programme, as shown in figure 55.

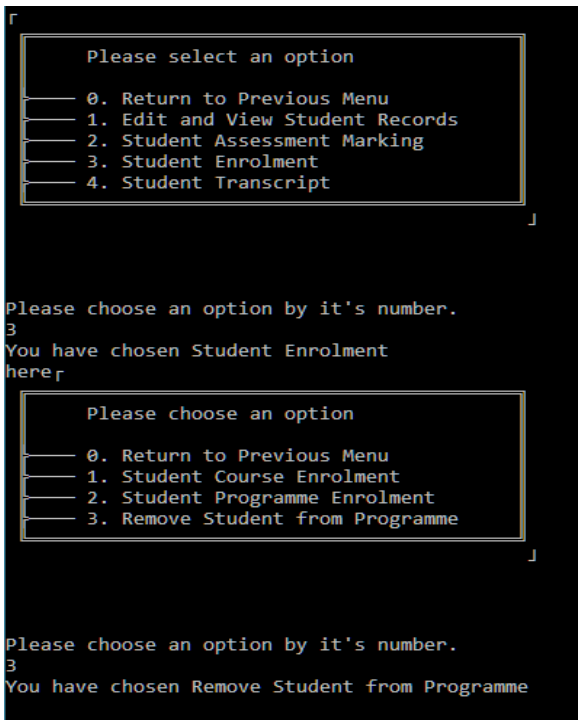


Figure 53 – navigating menu to remove student from programme

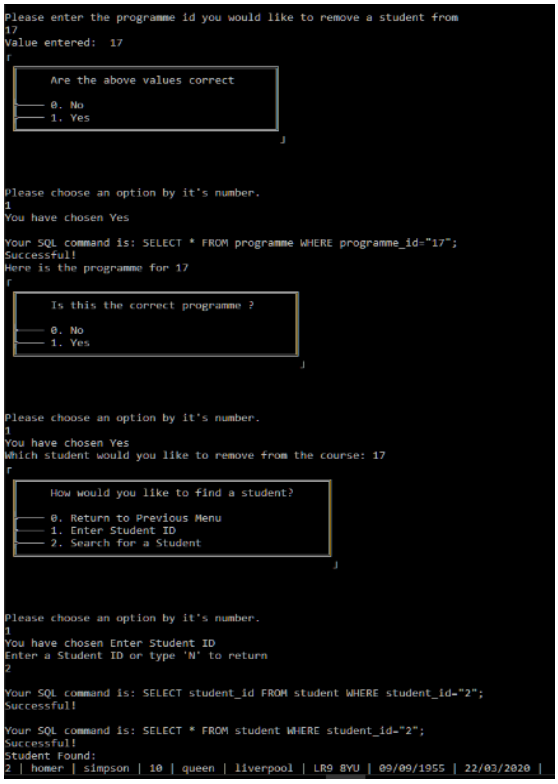


Figure 54 – entering student and programme ID

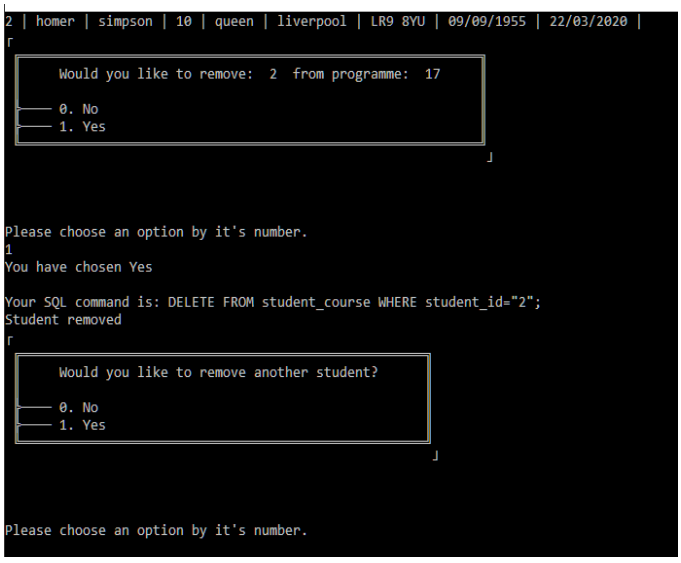


Figure 55 – Removing student from programme

Assessments

This section details how to manipulate Individual Assessments. This could be to fix an error in an Assessment for a particular Course, or to enrol a Student on only a particular Assessment. To begin, the user must first navigate to the Assessment menu, this is done after logging in and choosing the option "*Assessment Options*" on number "*3*" (Figure

57). This brings up the Assessment menu (Figure 56)

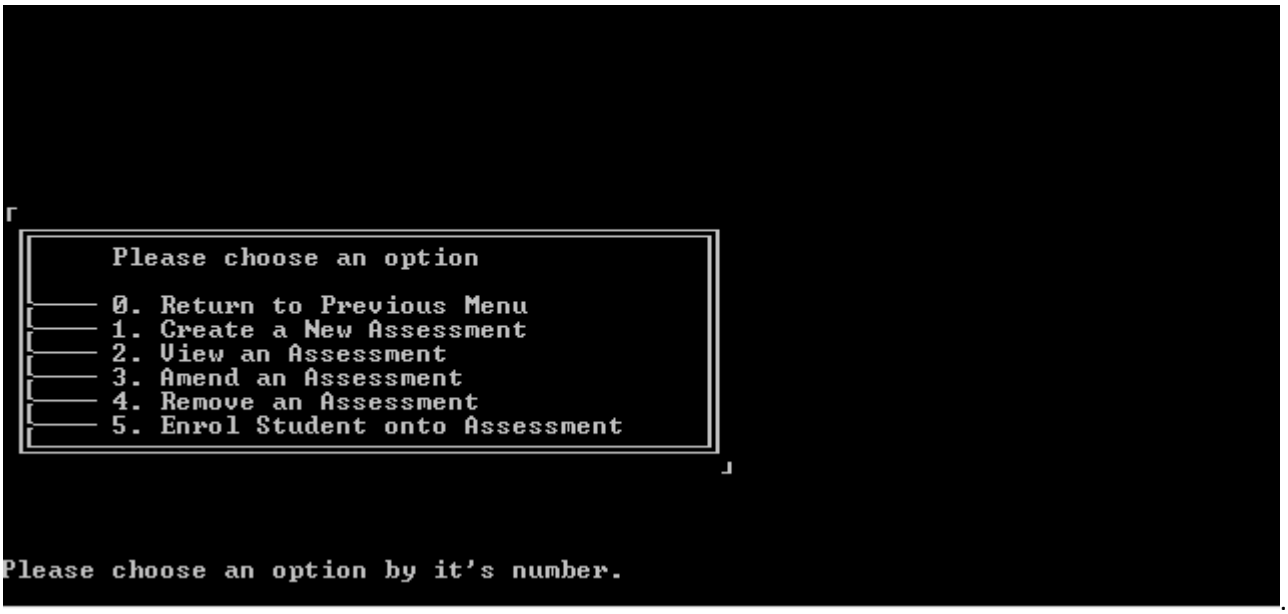


Figure 56 - Main Menu

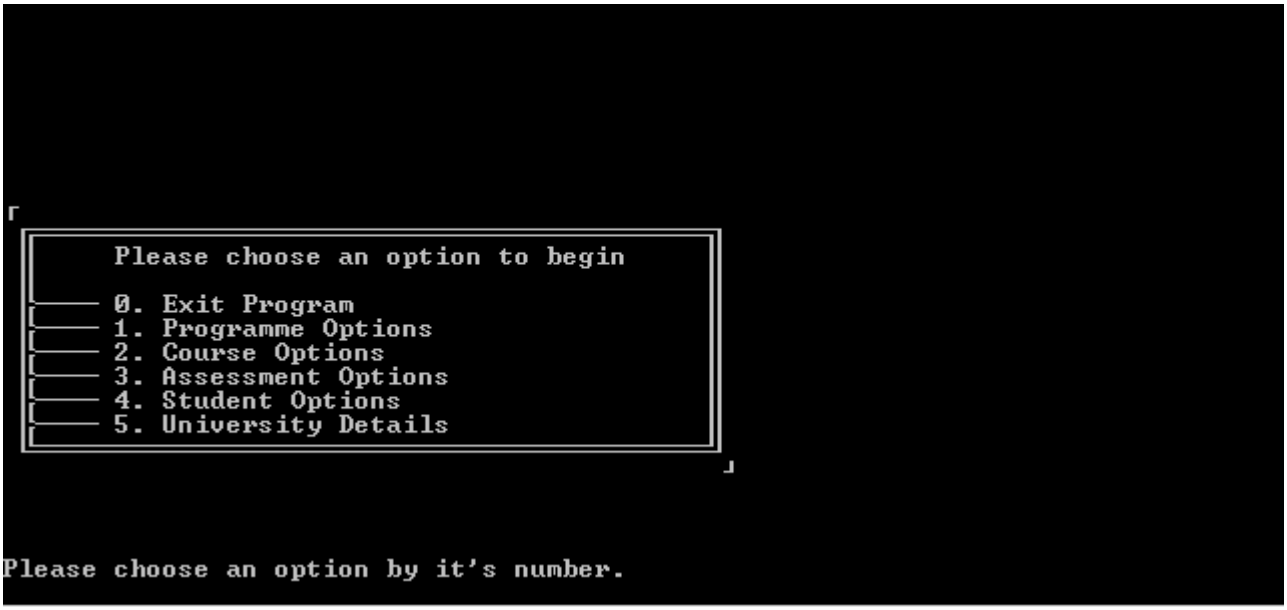


Figure 57 - Assessment Menu

Searching Assessments

Assessments can be searched in two ways. Either by Assessment ID or by Assessment name. Whenever this search is invoked, a menu will appear asking if they can specify the ID of the Assessment to find. If they can, they will be prompted to enter an ID, this is then checked for validity (if the entered value is a number).

Should the user be unable to specify an ID, then they may search for an Assessment using its name (Figure 58). This uses the keyword search specified under "Student". After completing this search, the application may continue with the function it is doing using the ID that the user has specified.

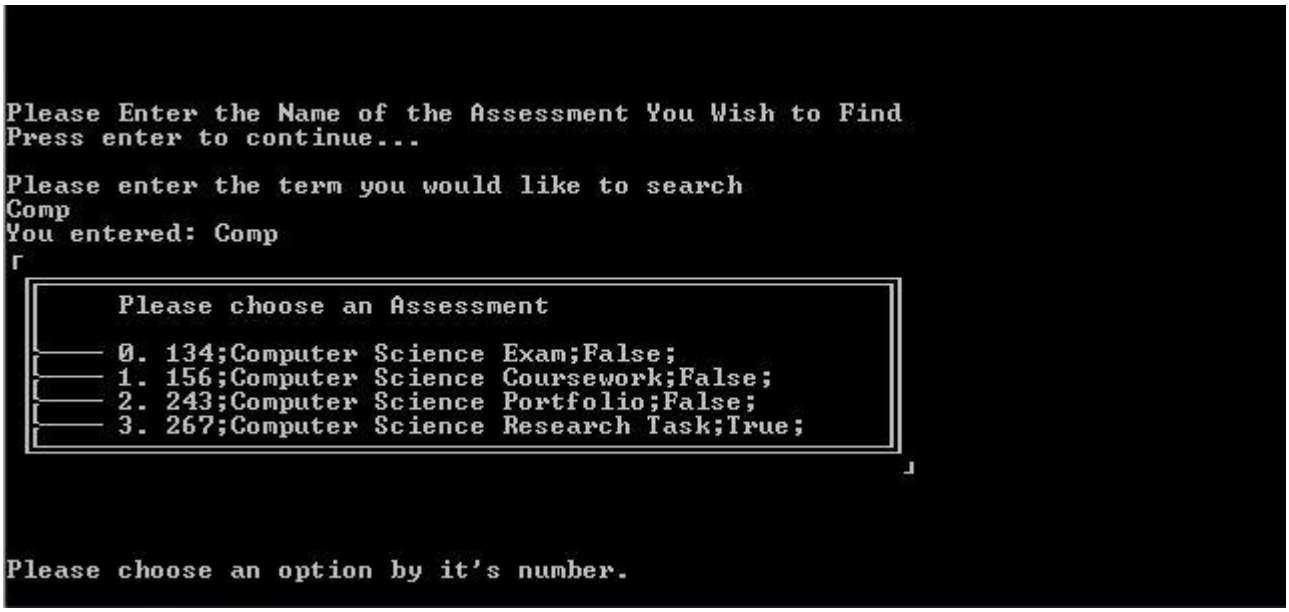


Figure 58 - Assessment Searching

Viewing Individual Assessments

To view an Assessment, the user must choose the option "*View an Assessment*". This takes them to the viewing process, which allows them to search for an Assessment by entering its ID number, or by searching for its name.

After this, the Assessment they have chosen is displayed and they will be prompted to "*Press Enter to Continue*" when they have finished reading the Assessment details. The viewing of an Assessment is shown in Figure 59.

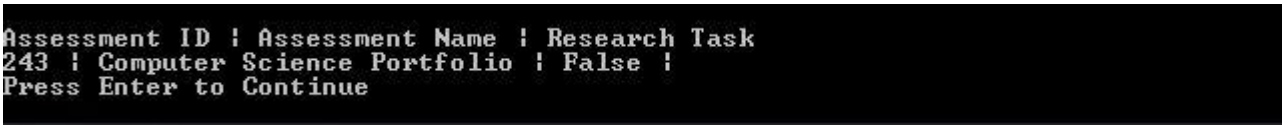


Figure 59 – Viewing An Assessment

Creating Individual Assessments

To create an individual Assessment, the user must choose the option "*Create an Assessment*" which takes them to the Assessment input process. This allows them to enter the Assessment Name and then asks them if the Assessment is a Research Task or not (Figure 60).

The user is then shown the details they have entered and they are asked if they are sure if they would like to enter these details, if they choose "*No*", then the function will exit with no changes, if they choose "*Yes*" then the new record is entered into the database and the user is informed.

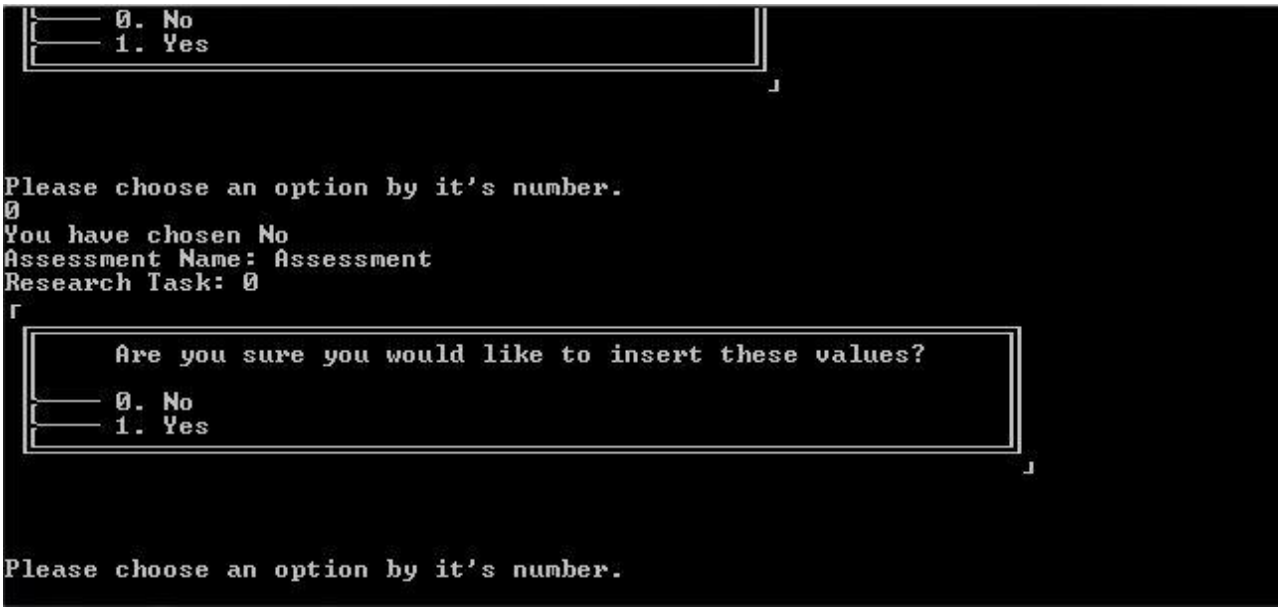


Figure 60 – Creating a New Assessment

Amending Individual Assessments

To amend an Assessment, the user must choose the "*Amend an Assessment*" option in the Assessment menu. This brings up the amending Assessment process, the user is then asked which field they would like to edit, "*Assessment Name*" or "*Research Task*", if they choose the name, they are then asked to enter the new name, otherwise, the user is asked whether they would like to make this Assessment a Research Task or not (Figure 61).

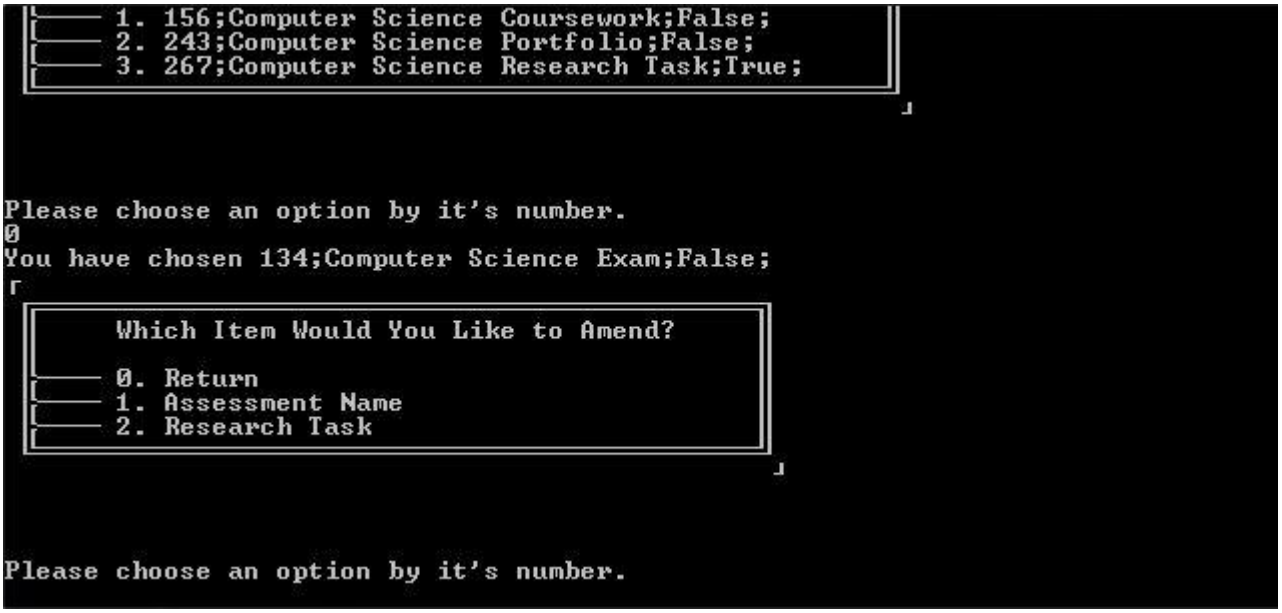


Figure 61 – Amending An Assessment

After entry, the value is updated in the database and the user is informed of the change.

Removing Individual Assessments

To remove an individual Assessment, the user must first choose "*Remove an Assessment*" from the Assessment menu, after this, they are asked to search for the Assessment in the same way as is specified under "Viewing Individual Assessments". When the Assessment is found, the user is prompted to enter the Assessment's ID to ensure that they would like to delete this Assessment (Figure 62).

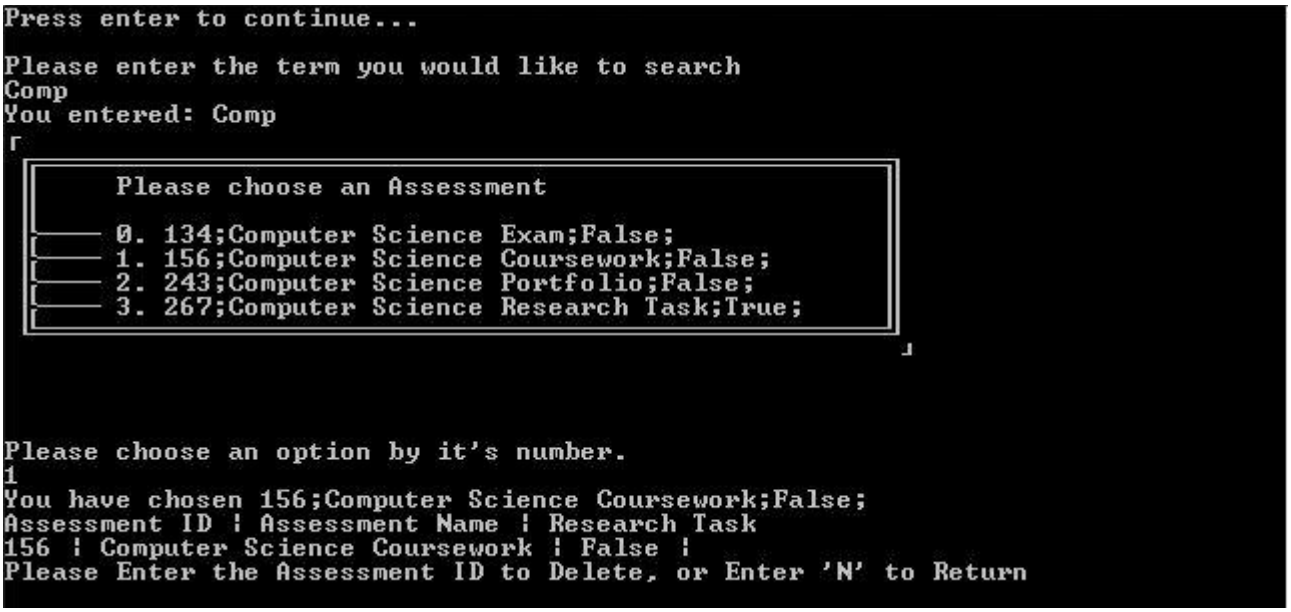


Figure 62 – Deleting An Assessment

Assessment Marks

Displaying Assessment Mark

To view a student's assessment mark, select student options in the main menu. Then, select student assessment marking. Here the assessment marks can be displayed, amended and added.

Enter the student id or search for student. If the student details are correct select yes, as shown in figure 62a. Enter the assessment id if correct select yes, as shown in figure 62a. If the student details are correct select yes. The student's assessment grade, result, mark and attempt will then be displayed. As shown in figure 62b.

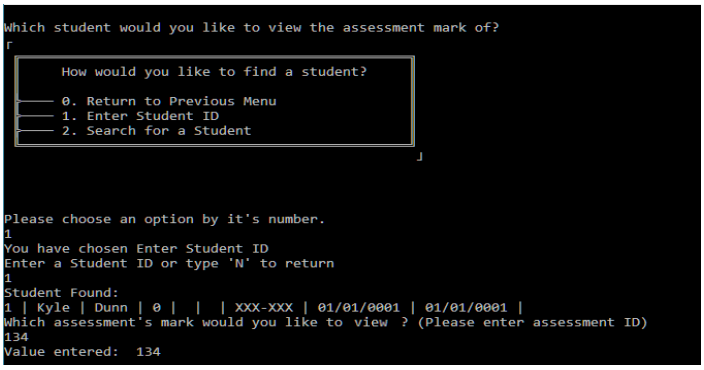


Figure 62a – searching for student

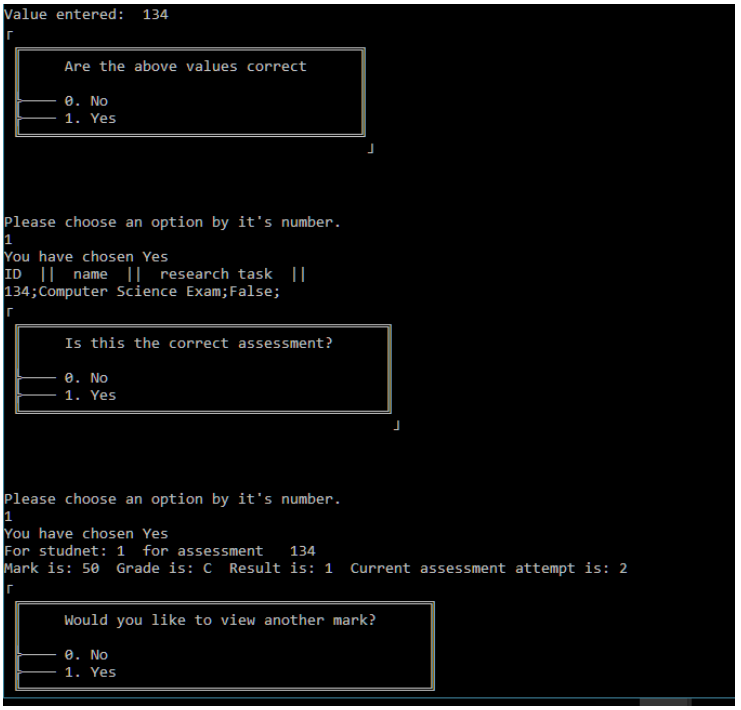


Figure 62b – assessment mark

Adding assessment mark

To add a student's assessment mark, select student options in the main menu. Then, select student assessment marking. Here the assessment marks can be displayed, amended and added.

Once selected, enter the student ID if known otherwise search for the student. Then enter the assessment ID if correct select yes. The details of the assessment will be shown, as shown in figure 63. Then enter the mark the student received along with any concessional codes, as shown in figure 63. The grade, attempt and result will be automatically calculated. The mark, grade, attempt and result will then be displayed as shown in figure 64. If correct select yes and the data will be stored into the database.

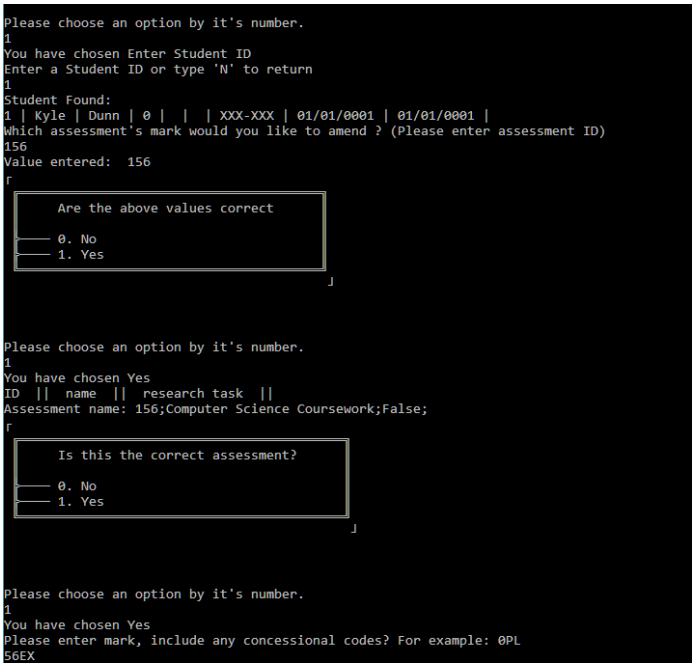


Figure 63 – assessment mark adding

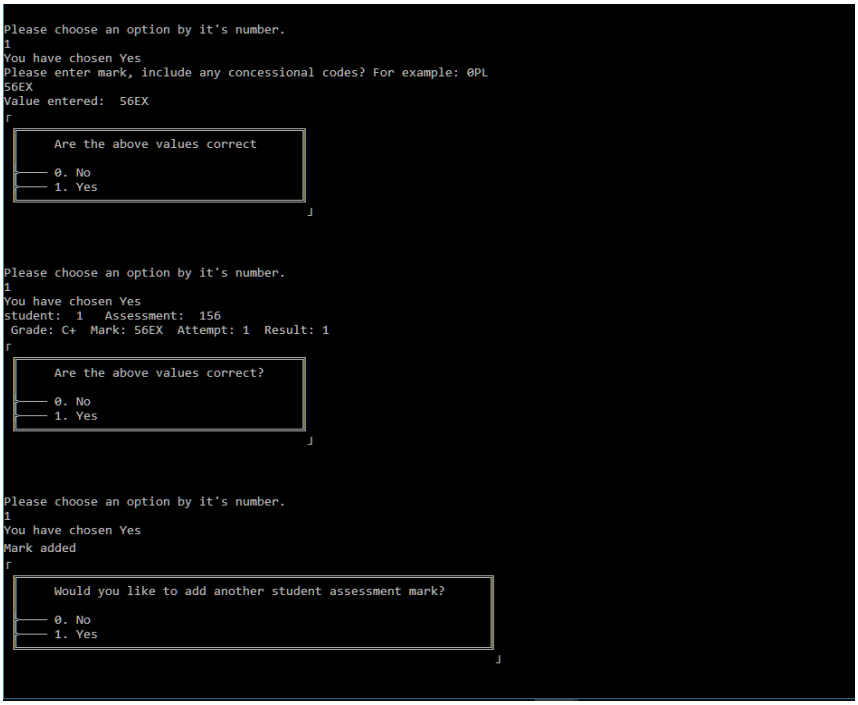


Figure 64 – viewing mark, grade result and attempt of assessment mark

Amending

To amend a student's assessment mark, select student options in the main menu. Then, select student assessment marking. Here the assessment marks can be displayed, amended and added.

Once selected, enter the assessment ID and its details will be shown if correct select yes. Enter the student ID. Then enter the new assessment mark and any concessional code to go with it. The grade, result and attempt is calculated. If student reaches maximum attempts and fails the second attempt, the course result will show a fail. If the mark is correct select yes and the mark will be amended, as shown in figure 65.

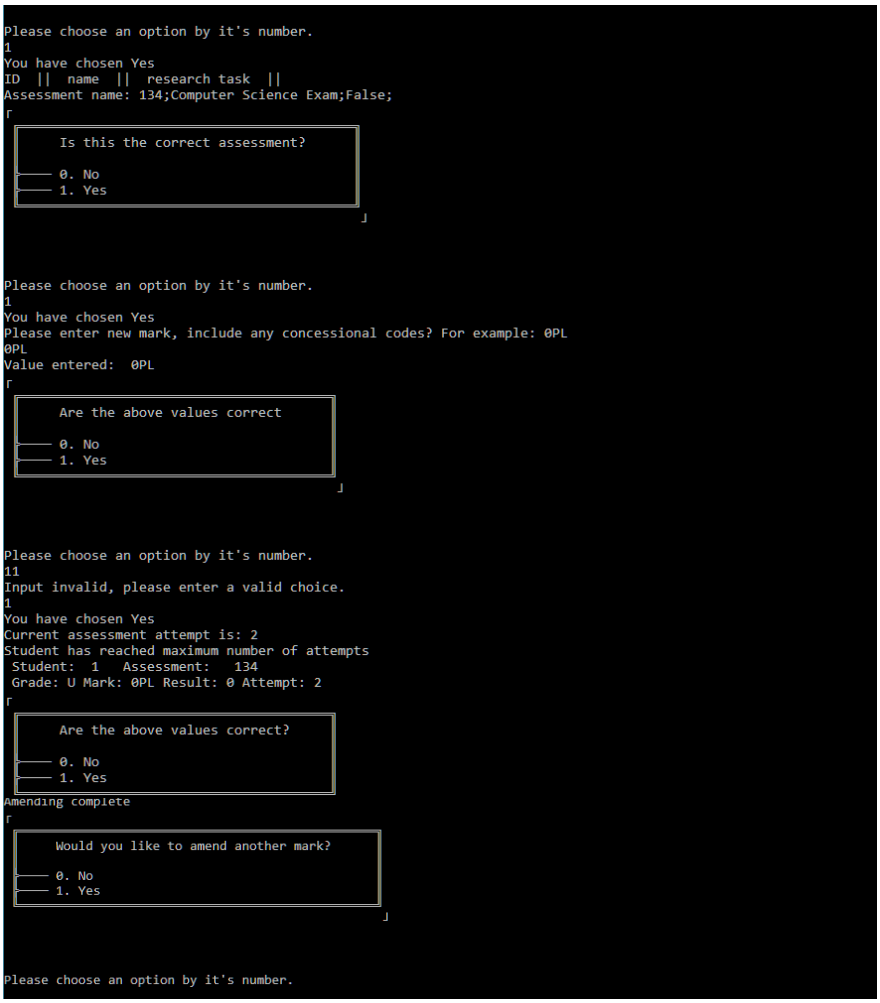


Figure 65 – amending assessment mark

View course mark

To view a student's course mark, select student options in the main menu. Then, select student assessment marking, then select view course mark.

Once selected, enter the student's Id or search for the student. Enter the course code to view the student's mark. If correct select yes. I the course is correct select "Yes" .the student's course more for the selected course code will then be displayed as shown in figure 67.

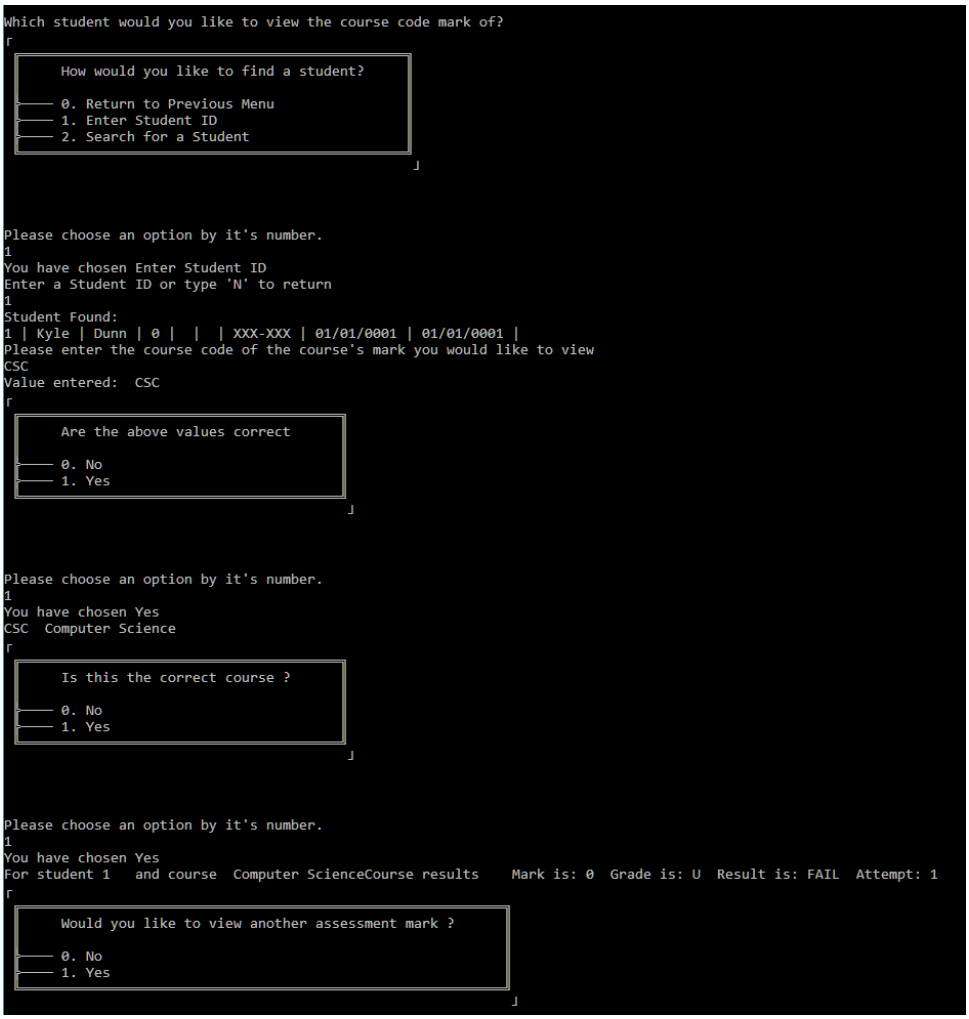


Figure 67 – View course mark

View overall level mark

To view a student's course mark, select student options in the main menu. Then, select student assessment marking, then select view level mark.

Once selected, enter the student's ID or search for the student, the student's details will then be displayed. Select which level to view in figure 68 level C was chosen. If correct select yes. The mark, grade and resuly will then be displayed. To view another level mark select yes.

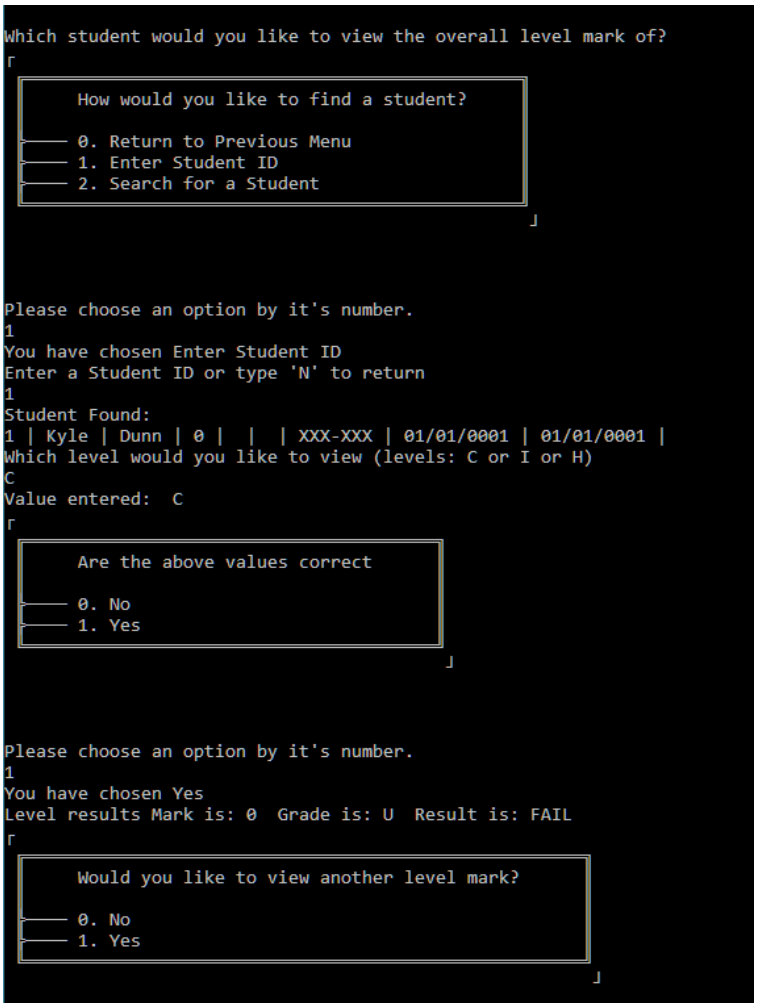


Figure 68 – View level mark

Confirming Marks

The marks are confirmed by the user before they are entered, for example in figure 64 the program asks the user if the mark entered is correct. If the mark entered is incorrect the user can select no and can re-enter the mark.

Security Considerations

Access is restricted to specific tables based on which type of user is accessing them. For instance, administrators have complete access as they need to be able to make changes to data at the most specific level, in case errors occur or there is a problem with the database.

A tutor can view everything but cannot make all changes to the database, as some changes require approval.

Students only have access to view items, and some tables such as assessment are not viewable by students as this could lead to sensitive information about assessments, the only information they need to see is in their own student assessment table rows.

Authentication is done using MySQL users. Users are setup with a username and a password. This password is encrypted and stored inside the database. When wanting to login to the database, a user must enter both of these correctly. The username is used to identify that specific user and the password is used to authenticate that user, so only that user should know their own password.

Contingency Considerations

There are various solutions for backing up a database. This solution uses MySQL as the solution for storage of data ad relationships. MySQL has commands for backing up a database, however, there are various applications that are readily available for backing up a database (Oracle Corporation, 2017) (Pranas.NET, 2017).

Some applications are free and will do scheduled backups automatically (Pranas.NET, 2017). As for the frequency, this may be up to the institution as needs change. It should be noted beforehand, that backing up a database does not necessarily mean backing up the entire database at once, in fact, as they become very large, this becomes impractical. Instead, only parts that change need to be backed up, this is called a "Differential Backup". This is generally how it is done with most databases. Sometimes, a backup can combine the two techniques and perform mostly Differential Backups alongside fewer Complete Backups (Yu Ping, Hu Hong-Wei and Zhou Nan, 2014).

So, if traffic is low between the database and its users, then more regular full daily backups could suffice with more frequent differential backups throughout the day.

Uniserver Zero implements a backup solution by giving the user an option to create "*.sql*" backup files that are then read in by the application to restore the database in its current state. This however is only a manual solution. A better solution is to use a provided plugin for Uniserver Zero that uses a batch file that can be launched regularly by a task scheduler regularly to backup the database.

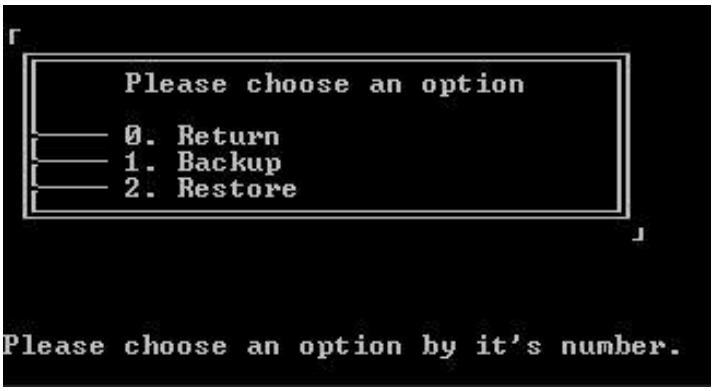


Figure 69 – Contingency Main Menu

The issue with this backup solution is that it isn't accessible through the application. Also, this solution won't work if a new MySQL Database Management application is used. Instead, a backing up feature has been implemented into the application itself. This is on the Main Menu on option "*7*". This allows for file backups of all Students, Courses and Programmes in the database. All values are written to a text file and can be retrieved through the application also.

Figure 69 shows the Contingency menu, this allows access to the backup and restoration menus which are shown in Figures 70 and 71 respectively.

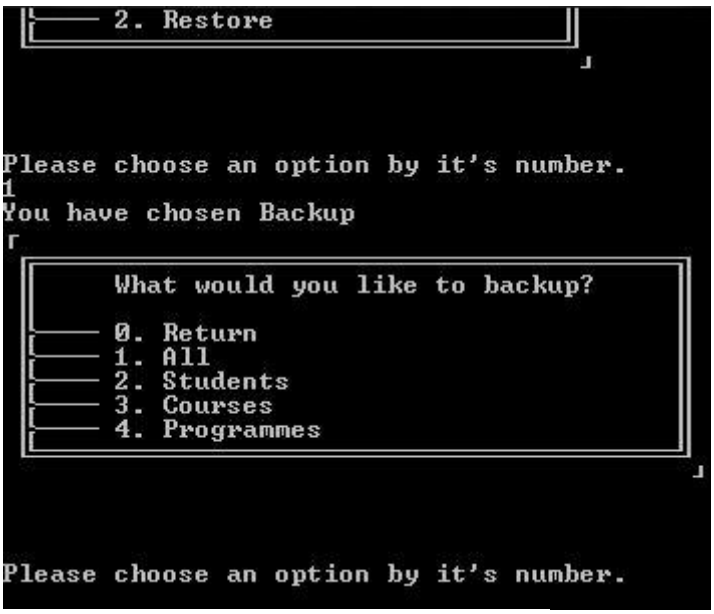


Figure 70 – Backup Menu

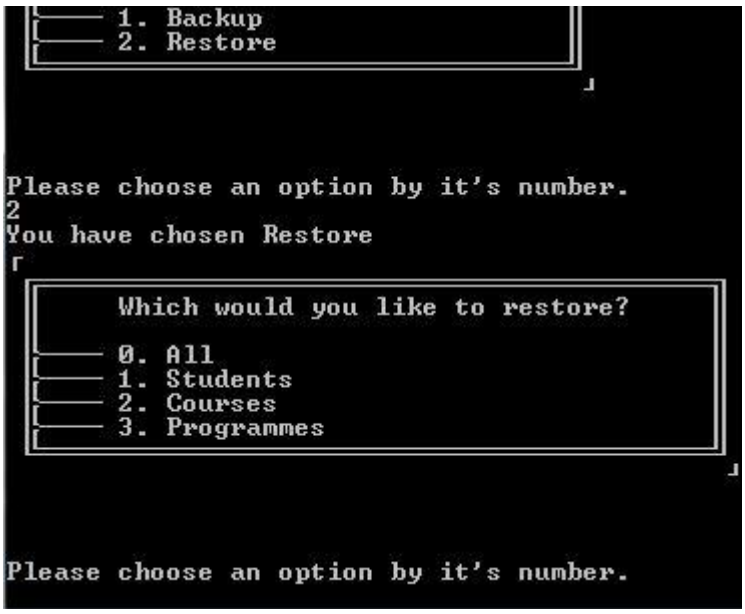


Figure 71 - Restoration Menu

Reports

Display a Student's Transcript

To view a Student's transcript, the user must navigate to the Student menu as is stated under the "Student" section. Then, the user can choose "Student Transcripts", this opens the Transcript menu which allows the user to choose between a single or a batch of Transcripts. When selecting a single Transcript, the user is asked to search for a Student using the Student Search (Detailed the "Student" section under "Student Search"). After this the application will display the appropriate transcript Figure 72, if the specified Student is not enrolled on any Programme of Study, then the Transcript will display "*Student not Enrolled on any Programme of Study*".

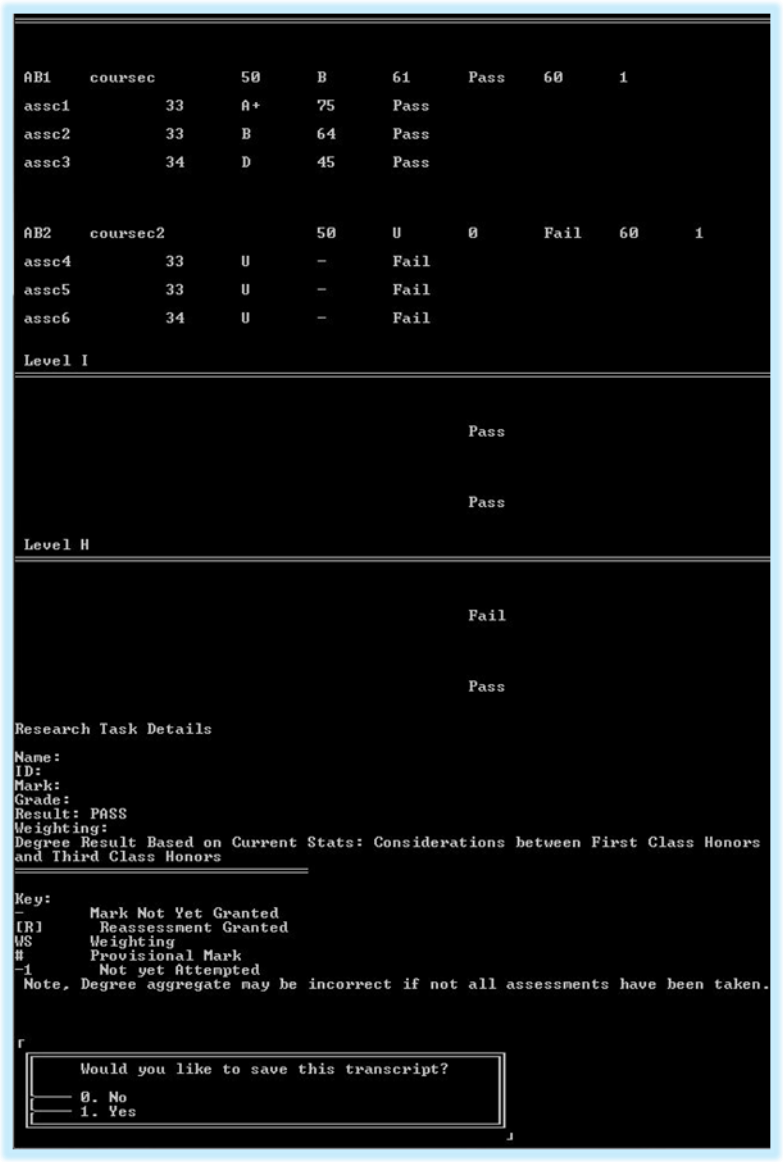


Figure 72 - Single Student Transcript

A single Transcript will give aggregate grades and marks for each Course worked out from all Assessments within each Course. The Aggregates for each Course are used to work out the Aggregate for each level in turn. Afterwards, the Overall Predicted Degree to be obtained is worked out from all of these aggregates. The overall predicted degree can be very inaccurate with little information for a Student, although, it will try to predict from what is available. As more information is added, this prediction becomes more accurate. After the Transcript, the user is asked whether they want to save this Transcript, which would print to a file with today's date in the filename if so.

Display or Print a Batch of students

Multiple students can be viewed using this application. The option to view all students is within the application, although, this search might not be specific enough. So, searches exist to view all students on a programme of study and to view all students on a course.

A transcript can be printed out to a text file using this application. A batch of as many transcripts as the user wants can also be printed. When using a batch, the user can view each one in turn and can navigate to the next or previous transcript. They may also save as many as they want to a text file. If using just one transcript, then the option to save is there when viewing it. If using batches of transcripts, then the user can print the currently viewed transcript, or print all of them.

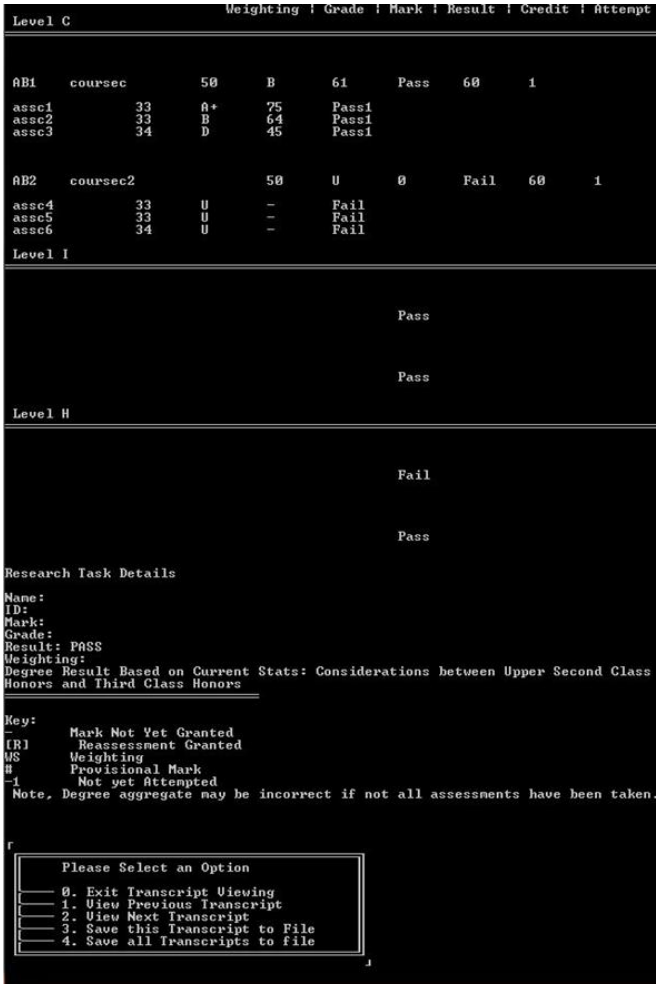


Figure 73 - Batch of Transcripts - Transcript View

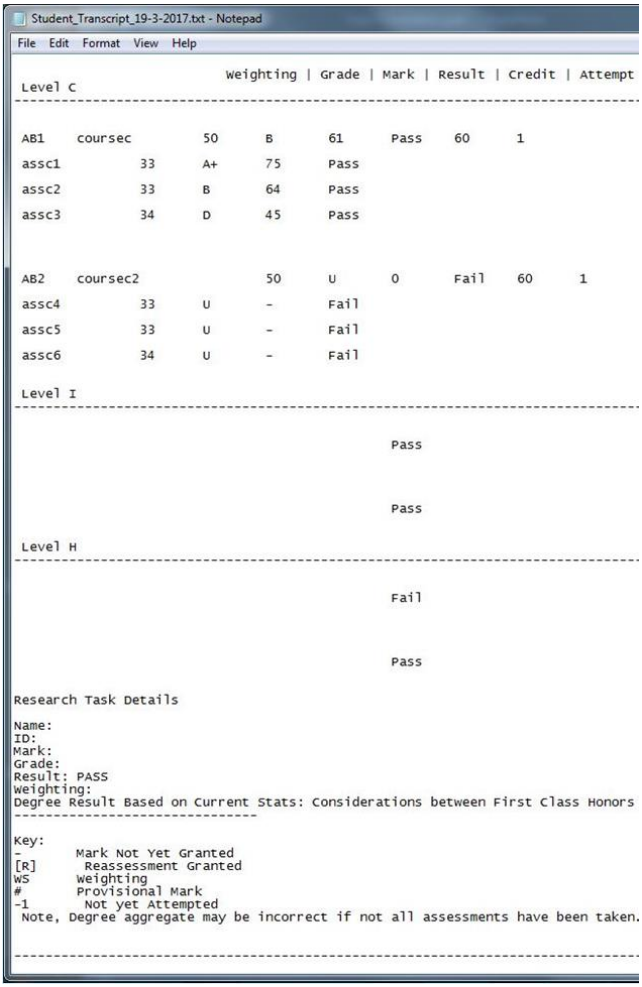


Figure 74 - Transcript File Output

The user can navigate through each transcript with the menu shown at the bottom of Figure 73, the user may also Save any single Transcript or all Transcripts in the batch Figure 74.

Contributions

This section details the contributions of each team member for this application. This details the classes and features of the application.

Karl Myers

- Course Class
- Course Search
- Course Assessment
- Course Assessment Search

Keiran Brown

- Student Class
- Student Search
- Backup Class
- Assessment Class
- Admin Class
- Date Class
- Menu Class
- Database Class
- Creation of Database
- Transcript

Callum Owen-Bridge

- Mark Class
- Programme Class
- Programme Course Class
- Student Programme Class
- Student Course Class

Citations

Yu Ping, Hu Hong-Wei and Zhou Nan, "Design and implementation of a MySQL database backup and recovery system," Proceeding of the 11th World Congress on Intelligent Control and Automation, Shenyang, 2014, pp. 5410-5415.
doi: 10.1109/WCICA.2014.7053638

Oracle Corporation, (2017). MySQL :: MySQL Enterprise Backup. [online] Mysql.com. Available at: https://www.mysql.com/products/enterprise/backup.html [Accessed 18 Mar. 2017].

Pranas.NET, (2017). MySql Backup FTP. [online] Mysqlbackupftp.com. Available at: https://mysqlbackupftp.com/ [Accessed 18 Mar. 2017].