



Computing

A Level

Computing

Exemplar Candidate Work

H447

Unit F454 Computing Project B

October 2015

Contents

Introduction	3
URS/Teacher's Comments	4
Candidate's Work	6

Introduction

This sample material serves as a general guide. It provides the following benefits to a teacher:

- Gives teachers an appreciation of the variety of work that can be produced for this unit
- Shows how the mark scheme has been applied by a senior assessor
- Provides examples of both good and weak application of different parts of the mark scheme
- Provides real examples of submitted for F454.

It is important to make the point that the teacher support materials play a secondary role to the Specification itself. The Specification is the document on which assessment is based and specifies what content and skills need to be covered in delivering the course. At all times, therefore, this teacher support should be read in conjunction with the Specification. If clarification on a particular point is sought then that clarification should be found in the Specification itself.

This project is an example which has not been scaled or adjusted during the moderation process; as such the marks awarded have been viewed as being 'within tolerance' when assessed by a moderator

URS/Teacher's Comments

Sample Material – Project 'B'

Centre marks/comments:

Definition, Investigation and Analysis

Definition

Max Mark: 3

There is a clear statement of what the end user's needs are and the system/job role that currently exists. An example of data capture is evident, although there is more that could have been collected. This becomes apparent later.

Centre Mark Awarded: 2

Investigation and Analysis

Max Mark: 11

There is extensive analysis completed and a range of live interviews have been conducted to develop a highly robust set of end user requirements. This is further supported through personal research into the system that he is going to develop, leading to more interaction with the end user about styles and ideas based around this research. This leads to a detailed list of EURs which are supported with some basic analysis of Hardware and Software requirements.

Centre Mark Awarded: 11

Total Section Mark : 13

Design

Nature of Solution

Max Mark: 6

The candidate further analyses the EURs and develops a modular approach. Each module is well described. A final review involving end user leads to definitions of variables and mention of validation. UML, Class Diagrams and design work all allow implementation by a 3rd party.

Centre Mark Awarded: 6

Algorithms

Max Mark: 5

Algorithms designed make a solution and the robustness of the algorithms are shown through the development of testing and checking with a range of test cases/traces.

Centre Mark Awarded: 4

Test Strategy

Max Mark: 6

The test strategy covers a range of test data to check the solution. Modular and Function testing is outlined to meet EURs, as well as Erroneous, Boundary and Normal testing. This is a detailed strategy.

Centre Mark Awarded: 5

Total Section Mark: 15

Software Development and Testing

Software Development

Max Mark: 16

There is excellent developmental evidence that has frequent reference to both Alpha testing and corrective action where needed. Code is included at the end and generally has sufficient annotation within to guide a reader as to the function. The develop solution clearly meets the design specification and is robustly developed.

Centre Mark Awarded: 15

Testing

Max Mark: 14

All Normal/Boundary and Erroneous testing is carried out with relevant screenshots/evidence and then corrective action is noted/developed as needed. The user tries to cover as many paths through the system as possible. The candidate provides space for the end user testing and this is signed off. There is evidence of some corrective action based on the feedback received.

Centre Mark Awarded: 13

Total Section Mark: 28

Documentation

Documentation

Max Mark: 10

The are plenty of pop up messages that show errors and how to fix these when running the program. The user guide shows a user normal use, including installation and backup. FAQ/Common errors are developed and incorporated.

Centre Mark Awarded: 9

Total Section Mark: 9

Evaluation

Discussion of the degree of success in meeting the original objectives

Max Mark: 4

The candidate reviews each fo the requirements and references these to successes earlier within the document via the use of page numbers. Some related issues were raised as part of the discussion.

Centre Mark Awarded: 3

Evaluate the user's response to the system

Max Mark: 3

There is acceptance from the user that the system meets needs and there are some developments identified for the future.

Centre Mark Awarded: 3

Desirable Extensions

Max Mark: 3

Sensible developments for the project are outlined, aimed at devleoping functionality of the system. Good and bad points are fully reviewed.

Centre Mark Awarded: 3

Total Section Mark: 9**Overall Project Mark: 74**

A1: Problem Definition

Description of client

My client is a teacher, working 9 to 5 every weekday. He currently has two classes, with a total of 11 students which he must monitor and keep track of. He sees both classes once a week for 1 hour 40 minutes where he is able to assess student's current progress. He would like a system which allows him to track each student's current grade, but in an easier way than his current system. My client currently wants this application focused on grading F454 projects. This information is then used to report to the head of department each month to monitor the classes that are taught.

Current problem and data collected

The problem is that my client is using a spreadsheet and paper based system which he often finds tricky to use and to set up. He would like an application which is easier to use and if possible, he would like the system to be accessible by more than one user at once. This could be another teacher or the head of department.

He wishes to have a system which allows him to be able to enter the results of pupil's marks for each section. He should be able to compare these grades if they are on target or not.

He would like the results to be shown as a graph / bar chart which displays the marks that a student has gained for the section and overall. The interface needs to be windows based and must be able to store names / scores / feedback in a file so that he can print out the data which will be eventually sent to parents and the exam board.

Current description of methods used or area in need of development

Method of use:

My client uses two systems, a publisher document and a spreadsheet. He needs to first open the publisher document to see what stage a student is at on their coursework. On this publisher document contains all the sections of the coursework, detailing what a student needs to pass, how far they are into the course, a bar representing their progress, their target and what they need to do to achieve a better grade. This provides paper evidence for the department meetings.

After reviewing a student's coursework, my client updates this publisher document so it's saved electronically. Then he updates the printed form of the publisher document which will be given to the head of department. Next, he must search through his documents and find his spreadsheet tracking system. He then updates what results were added on the publisher document to the spreadsheet which displays each student's names and also has boxes which allows my client to write down their current mark. At the end of the table is a UMS grade calculator which automatically detects what grade each student are at with a percentage of how much the student has progressed through the tasks.

Area in need of development:

The main area in need of development here is how my client has two systems for tracking a student's progress. Instead, one system can be made to do both tasks.

After receiving a paper based copy of both systems, it appears that both of the tracking systems give the same information, except the tracking system for students contains details about the specifications for the course. This can lead to mistakes such as duplication of data or loss of paper copies for students which can potentially cause mistakes. Loss of paper can also cause waste of time for both teacher and student as it means progress of work needs to be remarked and re-graded.

Forms of any relevant data

As stated above, my client is using a spreadsheet based system and paper based for students which is a duplicated version of data. He has two systems for the same information which needs to be looked at to see if there are ways to create one system which does both tasks.

Paper based:

The paper based document contains all the following information: Title, student name, centre and candidate number, each section of the Systems Life Cycle including the tasks they need to do for each section. At the end of each section contains a bar which states their final points.

Spreadsheet:

My client has given me a copy of the spreadsheet he uses.

		grades for each section																													
		A1			3	A2			11	B1			6	B2			5														
Tg		a	b	c	d	e	Sub	a	b	c	d	e	f	Sub	a	b	c	d	e	f	Sub	a	b	c	d	Sub	a	b	c	d	Sub
A		2	1	1	0	1	1																								
B																															
C		3	1	2	0	3	2	0	5	7	4	5	0	3																	
B		2	2	2	0	2	2	8	6	6	6	7	0	5																	
B																															
C		2	2	2	0	2		6	6	4	4	0	0	4																	
A																															
B		2	1	1	0	1	1	1	0	0	0	0	0	0	0																
B																															
D																															
B																															
A																															

The spreadsheet is colour coded, so depending on the marks that each student gets, it varies. For example, low grades are shown in red, average grades are displayed as orange, and high marks are shown as green.

The UMS points are not entered manually, but instead has a formula which calculates the points depending on a student's marks on each section.

What I intend to research

I intend on setting up a questionnaire to find out more about the users working environment, what type of equipment he takes to his work place and if he has easy access to a computer / laptop.

I also require knowing exactly what his current system does and so I would like to have a deep understanding of its flaws and how it can be difficult to use.

The reason I have chosen to do a questionnaire is because I do not have to directly meet the client as it may be inconvenient for him. The advantage of using the questionnaire is that my client is able to reply whenever he wishes to.

I plan to also give out a questionnaire to the other computing teachers and get feedback on their opinions of the current publisher tracker. I will do this to get more information about the tracker and to see if there needs to be any changes to the current system that the client may not have seen.

I plan on observing the client while he is using the system, this may not be possible but I will need to confirm this with my client. However, it is a big advantage to do so as I may be able to spot something the user does which he has not told me about. While observing, I am also able to ask any questions about the system, vice versa he is also able to ask me any questions if he is confused about something with the new system.

Next, I will have a face to face interview and talk about all the data that has been gathered through the questionnaire. I am able to then see if the user has changed their mind or has thought about something new. In this process, I am also able to reflect on the observation and prompt the client about anything I feel could have been done differently and any additions that may be implemented.

Different teachers may also require different data, for instance the Head of Department may require a summary of the data, where on the other hand the teacher needs more detailed marks etc.

Expanding onto this, I would like to have a second face-to-face interview with the client with any last questions or queries I may have. This will also allow my Client to ask me any questions about what could be implemented into the system. During the second interview, it will also allow me to go through the user requirements and can get his confirmation.

Finally, I would like to do my own research using the internet and gain as many pictures of suitable tracking systems as I can and create a mood board. I would like to display these mood boards to my client and get feedback which will allow me to see what direction my client is leaning towards for the interface.

A2: Investigation and Analysis

In this stage I plan on gathering as much information as I can about the current system and what the user wishes to have for his new developed program. This is to ensure that I know exactly what type of system the client wants and that communication is strong between the client and the programmer; thus also stopping mistakes and loss of time.

Questionnaire

The following questions were sent to my client, the screenshot does not fit and so I have transcribed it:

Timestamp	Are computers easily accessible in your workplace?	What do you find most difficult about your current system?	What are some good points about your system?	How are tasks normally carried out?	What details would you like to see on the interface?
08/10/2014 21:23:27	Yes. We have access to a computer almost always through the day	Updating three sheets can be long process. It also meant that you have to make sure that the data is correct on all of them. I want to produce graphs but it's awkward at the moment.	Colouring based on value	I log in. I then update my graphs. Feedback a	

Are computers easily accessible in your workplace?

Yes. We have access to a computer almost always through the day

What do you find most difficult about your current system?

Updating three sheets can be long process. It also meant that you have to make sure that the data is correct on all of them. I want to produce graphs but it's awkward at the moment.

What are some good points about your system?

Colouring based on values makes it easy to see who is doing well. The paper version works well for student.

How are the tasks carried out on your current system?

I log in. I then update my spreadsheet with the one I have marked. The marked work has my comments on that pupil's then need to reply to. I then update their manual sheet and give that back to the students as well as the work. The paper based sheet is also kept electronically for safety

What details would you like to be able to add for students?

Graphs. Feedback and reply boxes to log what they need to change and when they have changed it. Auto generate everything I need from one place.

What do you feel is the most important aspects of an application?

Easy to see grades. Ability to print marks for exam board, efficient tracking system

What would be your ideal application?

Electronic – not fussed how.

What do you expect from this new system?

Graphs, reports, print. For students and staff. Summary section for teachers. Tracking and feedback

Would you prefer a system which is entirely electronic?

Yes

What is something that is in the current system that you want for the new system?

I like the colour coding that is in my spreadsheet as this makes it easy to follow who is succeeding and where people are struggling.

Is there anything that you definitely do not want that your current system has?

I do not want to have to enter data more than once to get all of the information I need as this wastes too much time

What is the normal process of inputting a student's information onto your current system?

Normally I would mark the work. Then I would add the marks for each sub-section into the spreadsheet. This then needs to be copied over onto the student tracker (electronic version). After this I then update the paper based printed version of the tracker that the student has so that they currently know where they are at with regards to marking and success.

Additional Notes

SIGNATURE: _____

DATE: _____

Teacher feedback:

I sent a questionnaire to other Computer Science teachers regarding the current publisher tracking system and what they felt about it. I done this to gain feedback from a teacher who may suggest something that my client has not noticed.

Teacher feedback 1.

Looking at the interface of the tracker – How would you rate it as out of 10?

7

Can the tracker be improved? If so how?

Be more clear on when a student is on target with your target grade e.g. C, B, A. For example, if a student completes the first five sections, it will say I am at a C grade.

Additional notes

More colours

Signature:

Teacher feedback 2.

Looking at the interface of the tracker – How would you rate it as out of 10?

8

Can the tracker be improved? If so how?

Yes, if deadlines were written on the tracker.

Additional notes

If on a computer system, have a countdown until deadlines

Signature:

Teacher feedback 3.

Looking at the interface of the tracker – How would you rate it as out of 10?

7

Can the tracker be improved? If so how?

Yes, you can add popup messages to explain what you need to do get the marks to fulfil that criteria. This would be useful so we don't need to look at the syllabus when we mark coursework.

Signature:

Current forms

My client has printed out a form which displays all the inputs that would normally be filled in. The main problems with the forms are that if a mistake was to occur, my client would need to print off the paper again just to rectify it. On the second form, it was laid out well and the interface was easy to read with all notes beside it which stated what each section requires for completion of the stage. Therefore, this form may not have to be altered.

F454 Project Tracker

Computer Science Department
OCR RECOGNISING ACHIEVEMENT

Project Title: [REDACTED]	Student [REDACTED]	Centre Number: [REDACTED] Candidate Number: [REDACTED]						
Details of student and College ↗								
Grading points for each task ↘								
Section A1: Problem Definition Brief Description of the end user (e.g. Firm of Business involved, location, turnover, etc) Current description of problem Current description of methods used or area in need of development Clear statement of origins and form of any relevant data (e.g. paper based registration sheets that cover personal details) Discussion about how further research is needed to define problem and what you intend to do for research								
Section Mark	1	2						
3 Final:								
Grading points 'Green shows where they are upto' ↘								
Section A2: Investigation and Analysis I have included my end user fully in all discussions and have evidence of this I have detailed evidence of all interviews, research and observations I have fully researched all avenues and opportunities for the problem I have evidenced and discussed a full analysis of the problem to be computerised I have a detailed user requirements list that is justified and signed by the end user ↘ I have a detailed user requirements list that is justified and signed by the end user ↘								
Grading points for each section ↗ Student names ↘								
Tg	A1	3	A2	11	B1	5	B2	5
Tg	a b c d e Sub	a b c d e f Sub	a b c d e f Sub	a b c d e f Sub	a b c d e f Sub	a b c d e f Sub	a b c d e f Sub	a b c d e f Sub
A	2 1 1 0 1 1							
B								
C	3 1 2 0 3 2	0 5 7 4 5 0	3					
B	2 2 2 0 2 2	8 6 6 6 7 0	5					
B								
C	2 2 2 0 2	6 6 4 4 0 0	4					
A								
B	2 1 1 0 1 1	1 1 0 0 0 0 0	0					
B								
D								
B								
A								

As posted previously, this is the spreadsheet my client uses to see what stage each student is at:

Student names ↘

grades for each section ↗

Tg	A1	3	A2	11	B1	5	B2	5
Tg	a b c d e Sub	a b c d e f Sub						
A	2 1 1 0 1 1							
B								
C	3 1 2 0 3 2	0 5 7 4 5 0	3					
B	2 2 2 0 2 2	8 6 6 6 7 0	5					
B								
C	2 2 2 0 2	6 6 4 4 0 0	4					
A								
B	2 1 1 0 1 1	1 1 0 0 0 0 0	0					
B								
D								
B								
A								

Face-To-Face interview – Client

I have sat down with the client and have gathered details. I conducted a face-to-face interview in hopes that I am able to get as much information as possible.

When interviewing my client, he showed me his system and how he normally inputs data for each system. The main concern was that it was time consuming and frustrating at times to constantly having to refer to one system to another, and in some cases it was difficult to find the word document system as each student had their own file.

During the interview, my client also expressed some ideas that could be added to the system. He stated the main requirements, which were to have one system to do both tasks, but also wanted (if there was time) a system which allowed the head of department to log into the tracker and check progress online. This could be done with the school staff-shared network where upon clicking the document would display a screen which requested a username and password. My client has requested for a notes section where he is able to write down some information for the head of department to help the tracking – the head of department should be able to reply back. This is not required but something the client would like to have.

Another request my client had made was the system being a multi-user system so he can view and edit the data inside the system while another teacher may be looking at his progress.

There were other details also added of what type of system the client wanted to have, this has been added to the user requirements section.

SIGNATURE: _____ **DATE:** _____

Face-To-Face interview – Head of Department

After interviewing the Head of Department teacher, I have gathered extra data which are similar to my client's needs. He mainly wanted a feature where there could be feedback added to the system for evaluation and teacher support. Therefore, he is able to tell the teacher about any details of the coursework that may be missing. He also wishes to have a section of the tracker where the teacher is able to reply back to the feedback received. Another feature that the Head of Department brought up was having a simple interface. He liked the thought of being able to see every student's progress on one screen and being able to edit their current marks for each section. He would then like there to be an option where he can print out a certain pupil's progress for data meetings, but would like it if there was an option just to print out one or two sections, not the whole feedback as some parts of the course may not have been completed by the student.

Lastly, he requested for an option on the tracker where he is able to edit the exam specifications. This is because the exam board can sometimes change the way students get marks for sections. This feature should be added so that it is future proof and doesn't require for a whole new system to be built or another system developer to implement it. The Head of Department also brought up a suggestion for the new system. He stated he would like it to be added, but it would not be priority or necessary for it to be implemented. He would like it if he is able to put certain options for individual students who may be underachieving or may have a worrying background for his previous grades. Therefore, he is able to see what the individual student is progressing and can print off a feedback sheet for the student.

SIGNATURE: _____ **DATE:** _____

Face-To-Face – Second Interview

After interviewing the Head of Department, I arranged a second interview with the client.

When showing my client the mood board (shown in research section below), I had told him of the idea of having a profile system where he is able to see all students with an image of them. More details of the mood board has been listed below on page 10.

After speaking to my client regarding the layout of the system, I had told him about the idea that the Head of Department brought up. This was the ability for the client to select certain students and highlight their profile to allow the tutor to remember which pupils may be underachieving. My client liked the idea and stated that anything that the Head of Department or teachers have given feedback on will be taken on board and should be added to the new system if feasible.

Lastly, I spoke to the client regarding the user requirements. The full requirements have been posted in the user requirements section.

Observations

After talking with my client, I observed him using the spreadsheet and paper based system. There were evident things that needed to be changed as a lot of mistakes could occur during the process and could waste time.

How my client uses his system:

- My client opened a word document – This required going through folders.
 - He requested a system that centralised all the information so he doesn't have to go searching through folders.
- Search through folders to locate the class and the student.
- Open up a word document which displayed the student's current progress and what is required to get to the next stage.
 - This word document can be printed so it can be given to the head of department. My client has requested that each teacher could have their own profile which displayed all the information in the current system.
- My client then opened another application, this time it was a spreadsheet.
- This spreadsheet displayed each student's current marks for each section of F454.
- Each mark was coloured differently depending on how high or low it was. He would also need to change the colours manually. Therefore, this could be made easier by making the system change colour depending on (for example) if there grade was between 1 – 3, it indicates red, if it was 4 – 7, it indicates orange, and lastly from 8 – 10 displays the green colour.

My client needed to write down both the sub marks and the total marks for each section in the spreadsheet and had to total up the average manually – This could be made a lot easier by creating a system which calculates the grade based on the percentage of total marks available. Once the word document had been printed, he would also be able to write down data on the sheet. However, as he was noting down the information, he had made a mistake which resulted in the document being untidy and could have confused pupils. Therefore, anything that needed to be done physically on paper should have an option on the system which allows him to do that task.

Mood board (Internet research)

I have reviewed a large range of tracking systems and have selected a few which I feel may be appropriate for the client. (Posted in the next page)

Each mood board image has been labelled and reviewed below.

This image is a collage of nine screenshots from different educational and administrative software applications, each labeled with a large number (1-9) in a pink box.

- 1**: A screenshot of a seating chart titled "Period 2: Chemistry > Today's Attendance Status". It shows two rows of student names (Hill, Ernest; Roberts, Craig; Gonzalez, Janet; Phillips, Paula) with dropdown menus for attendance status (Present, Absent, Tardy).
- 2**: A screenshot of a course management system showing a grid of courses for the 8th grade (2013/2014). Courses include English, Math, and Science, with columns for Semester, Grade, Credits, and Action buttons.
- 3**: A screenshot of the "Student Tracking System" showing fields for First name, Surname, DOB, Age, Phone, and Course.
- 4**: A screenshot of a prospect details form for "Finch, Simon" with fields for address, phone numbers, email, and a list of inquiry details.
- 5**: A screenshot of a "SE Resistant Materials; Project Checklist" document with sections for Name and Teacher, and a table of 18 items related to project tasks.
- 6**: A screenshot of a "Review Grades: Course Summary" page for "RHA130: Red Hat Linux Core System Administration". It shows a grid of grades for various workbooks and a summary table.
- 7**: A screenshot of a "Class Attendance" sheet for September 2011, showing student names and attendance marks (T, F, S, Su, M, Tu, W, Th, F, Sa, Su, M, Tu, W, Th, F, Sa, Su, M, Tu, W, Th, F) for 30 students.
- 8**: A screenshot of a worksheet for "Short Beads" and "Chains" with sections for Counters & Cards, Spindles, Number Rod, Step Board, Tracing, Writing, Addition, Subtraction, Ten Board, Intro, and Long 2, 3, 4.
- 9**: A screenshot of a "Homework 2" page showing student names (Jack, Liz, Joe) and their scores for various assignments (100.0, 92.5, etc.).

- 1** – This type of interface is a windows-based profile system. This would allow my client to select a certain student and open up their profile. This could then be used to navigate through each of their grades and marks etc.

2 – This interface can be used to display each student on the left side. The user is then able to click on options to the side where he is able to change their grades etc.

3 – This interface is very simple, the user is able to note down the details of the student and also list their grades. Although, this may be too simple and may not have enough information that the client needs.

4 – This interface has many options with lots of drop down boxes to limit data. This ensures validation.

5 – This is similar to the current spreadsheet system. It's simple, has a list of all students and is easily editable.

6 – This is also similar to the spreadsheet as it allows the user to see all the students. This is however, a better interface look.

7 – This is a large spreadsheet which contains all the details needed. It contains extra information at the top which could be used to label which class and course it is.

8 – Similar to the current system, this spreadsheet contains colour coded symbols and various shapes to symbolise the progress of a student.

9 – This contains small drop down boxes and fields where the user can enter details on. It is easily editable, although the interface is still similar to the spreadsheet.

10 – After analysing this table, this isn't very user friendly as it doesn't contain much information to edit.

I have personally had a look at what could be possible and what would be appropriate for the new system. For the first layout, my Client had requested a menu based system. Therefore, it would be appropriate for the first layout to appear when the program starts up. This displays all the students in the course, but there should also be an option which displays all student's grades at once. Upon clicking the option to display all students, it would be appropriate for option 8 to be chosen. A very simple layout which will be easy for teachers to monitor.

Client's feedback on the mood board.

This feedback was given during the second interview stage.

My client liked the look of picture number 1 as it allowed him to see each student's progress. However, the Head of Department liked the current layout of being able to see all the students' progress at once. Therefore, there will be an option which allows the user to be able to see every student's marks. After the user clicks on one of the options, they will be greeted with a similar layout as number 8 on the mood board. This was decided as the current layout of the spreadsheet is very user friendly.

Current tracking system (Internet Research)

The benefit of researching the current tracking system is that it allows me to see what type of options currently exist and what features could be implemented. It also gives me ideas and suggestions for improving the current system which my client may not have thought of.

Microsoft Excel 2013

The main advantage with this application is that when inputting data into the system, a graph can be created automatically. This saves time as the user does not have to make the graph himself. However, one drawback of this feature is that it can be complicating to try and set up the graph as you need to select options to make sure it works. Expanding onto this, a macro can be created in Excel which can direct you to a certain section, this is one of Excel's great advantages.

Another feature of Excel is how easy it is to edit the information when a user inputs it into the spreadsheet. For instance, you are not required to go through options to change the input, but instead you can simply click on the inputted data and alter it. This also includes changing colours – You simply click on the spreadsheet tile and select the colour on the bar provided.

In conclusion, the change of colour feature is definitely something to be considered for the new system as Excel allows any colour to be inputted. The client may decide to change colours in the future.

After attempting to research for current systems, there were none that matched the description of the current system. Therefore, I have only reviewed Microsoft Excel as that is the current system my client is mainly using.

Analysis of Investigation

From all my investigations, I have gathered all the data required to understand what the problems of the current system are, and what type new system the client requires.

Summarising of Observation

My client currently has two systems, one system which allows him to see what needs to be done to complete a section of the coursework. This also contains the student's name, his current grade and what he needs to do to achieve the next stage. There is also a bar which indicates what grade the student is on.

On the other system, he uses a spreadsheet based software. Here, all his students are listed and he needs to identify and input their grades on each section. Then he needs to total up the scores and input an average. This average gets totalled up and decides how many marks the student currently has. The main problems with the current system is that my client has to use two systems for one job, which is to keep track of his students. Therefore, he requires an application which he is able to: Keep track of each student; system needs to display criteria for sections, total up an average without manually calculating and create a profile for each student where he is able to then print out.

Summarising Face-To-Face

Having Face-To-Face conversation with two teachers, it has given me a great insight to my Client's needs. All the worries and problems have been illustrated in the Face-To-Face section.

Summary of questionnaire:

The main concern from my client is that there are two applications that need to be open. This can be reduced to having one system which does both tasks, therefore it will be less time consuming and fewer mistakes can occur.

A problem that the teachers seem to have is that they are unaware of what grade students have currently reached as the tracking sheet is graded on their marks. Therefore, a system could be created where it notes down automatically what grade the student is currently at. On the other hand, the tracker could also display what each student needs to reach a certain grade.

Analysis of interview:

The client had made it clear what he found difficult when using the current system and displayed what could be made easier, such as having one system to do both tasks. Both current systems will be joined together and will be made on one application. This will make it easier for my client to track his system without needing to copy down the same information from one system to another. This also means that the new system will need to be formatted so that it is printable for A4 paper. Additionally, after the interview, it has made me aware of exactly what user requirements he wishes to have, which will be listed below. The variables for the marks have been made clear:

For target grades, there will be a different colour code as it will be judged on their target and not the full marks. If the user is on target, it will highlight green. If they are one grade below, colour will turn orange, if they are two grades or below, they will be given a red colour on the mark.

For overall grades:

- Green – 80% or higher grade.
- Red – Below 40%
- Amber – In-between %40 and %80.

This will have to be adjusted though as there is a section which has the maximum marks of 3. Furthermore, the final mark must shade against their target grade to display if the student is on target or not. Green, highlighting they are on

target, orange showing they are one grade below and red showing they are 2 or more grades below.

Detailed user requirements

During this phase, I have sectioned different requirements for different sets of research.

Client requirements

- Show criteria for each section of the F454 coursework.
- The ability for teachers to see what is missing in each section by comments.

Design requirements:

- Have a profile for each student which indicates their current and target grade for F454.
 - This has been requested by both Head of Department & Client in the interview section.
- List student's marks for each section of the F454 coursework.
- Show the criteria for each section of the F454 coursework
 - This is done so that the tutor can see how many marks a student gets.

Taken from the questionnaire which was sent to the client.

Input requirements:

- Be able to enter student's current marks for each section of F454.
- Add notes for students' progress.
- Easy to edit information such as current marks or criteria for coursework.
- Highlight certain students who teachers may be worrying about.

Input requirements taken from the interview stage.

Processing requirements:

- Calculate percentage of completion automatically.
- Automatically colour numbers once inputted depending on the grade.
- Save, so that the client can continue to keep track of his students (Database).

Output requirements:

- Be able to print of student information such as:
 - Name, section, marks, whether they are on target or not.
- Results should be shown in a graph which indicates the student's current grade.

General requirements

- Be windows based as the school operates on Windows 7.

My client would like, but not necessarily needs:

- Have teachers be able to track their classes through login.
 - After speaking to my client, this can be done through the school shared network which teachers have access to.
- Have teachers note down any information on the tracker for other teachers.
 - Username / Password login for
- Can be used by more than one user at one time.

SIGNATURE: _____

DATE: _____

Hardware and Software requirements

The following system requirements are based on handling Microsoft Windows 7 due to all school computer systems using the Operating System.

Hardware requirements:

- 1 gigahertz (GHz) or faster 32-bit (x86) or **64-bit (x64)** processor
 - This is required because it allows the end user to be able to use the system without any delay.
- 1 gigabyte (GB) RAM (32-bit) or 2 GB RAM (64-bit)
 - This is minimum RAM required to run Windows 7
- 16 GB available hard disk space (32-bit) or 20 GB (64-bit)
 - This is necessary as the system may be used over a few years and the database may continue to increase. Therefore, a larger disk space allows no errors involving memory management.
- DirectX 9 graphics device with WDDM 1.0 or higher driver
 - Needed to display outputs such as bar charts / graphs.

Software requirements:

- Windows 7 or later operating system.
- Email to allow the client to send copies of the student's progress.

Hardware requirements for input /output:

- Keyboard and mouse
 - Both to input data into the tracking system.
- Printer
 - Allows the ability to print out the tracking sheet for students.

B1: Design System Objectives

This stage is an expansion of the User Requirements and extra depth will be taken into consideration as the requirements will judge what type of interface the system will have. To illustrate the different interfaces the user can have, there will be various designs created which will follow the users input requirements.

Primary Objectives

The primary objectives are required features which are to be added to the system. This has been established in the user requirements list which has been agreed by the client.

User interface

- The screen size will be (1920x1080pixels) as it is the monitor size if the school PCs that the system will be run on.
- Based on target grades, the following will be the colour codes:
 - Blue is above target
 - Green indicates a student is on target
 - Yellow is a student with one grade below
 - Orange indicates the student is two grades below.
 - Red indicates student is not on target and is three grades or below.
- Background colour of the system will be plain white or very light blue.
- Profiles for each student will be grey, highlighted students will be slightly shaded.
- An exit button will be displayed on each page to exit to the main menu.

- Exit button colour will be red.

Input

- The client will have the ability to enter student's marks by selecting an in-built spreadsheet system. Expanding onto this, they can also select the student's profile and edit the details in there.
- Client will have the ability to add notes to students using the note box provided on the student profile.
- Client can easily edit data in two ways. They can open a student's profile and edit it there or can access the spreadsheet and edit the marks form there.
- Client will have the ability to highlight certain students who teachers may be worrying about. They will be able to do this by selecting the dropdown box.

Processing

- The total marks is 80, therefore the percentage of completion will be based on how many marks a student gets per section.
 - Formula: Marks / 80 * 100
- The colours are dependent on two factors: Target grade, total marks.
 - For total marks on each sub section, this will be:
 - Blue – Any grade above the target grade.
 - Green – On target
 - Amber – One grade below
 - Red – Two or more grades below
- A database will be created so each student's progress saves. I can use the feature which is already on Visual Studio.
- Every time a grade is inputted, the graph will increment or decrement a point (depending on the grade) on the graph so it joins together to the right grade.

Output

- My client will be able to print the tracker to see their current progress. This will include, as stated above:
 - Name, candidate number, name and project title will be on the top of the page.
 - Section of coursework, marks and percentage of completion will be the main page.
 - Whether the student is on target or not will be displayed under each section.
- An option will be made where my client will be able to view the results of students through a graph. On the side of the graph will display the following:
 - Highest grade – 80 marks or A grade.
 - Lowest grade – 0 marks or U grade.
- There will be a straight line going through the graph to display the students target grade. Therefore the teacher will be able to see which section a student may be struggling on.
- Results should be shown in a graph which indicates the student's current grade.

SIGNATURE: _____

DATE: _____

Secondary Objectives

These requirements are not necessary and instead are desired requirements. This has also been established in the user requirements list.

- As the school is based on a large network, there is a shared drive which allows teachers to have access to. There are set permissions on the drive which restrict access to students. Something that my client has brought up was the implementation of a login system for teachers which can be accessed in the shared drive. The advantage of this is that a separate website will not have to be created, but instead the shared drive can be used to an advantage as teachers can access it. The school shared drive is built in such a way that the administrators can allow permissions on certain folders. This will be helpful to differentiate between different staff.
 - The interface for this will be simple. A username and password field will be visible with a submit button at the bottom. This allows the teacher to access their tracking system.
 - This also means that a process needs to be created so that each teacher has a unique password and username.
 - Upon login, the user will be greeted with a menu which contains: Tracking and comments.
 - The notes section will open a new window, displaying the tutor's comments at the top and a textbox for a further comment.
- Can be used by more than one user at one time.
 - This means the application will need to adapt the multi-user, multi-tasking feature which will allow teachers to be able to edit it while another teacher is using it.
- Upon login for teachers, they will have an extra feature which will be labelled as email. This can be accessed by clicking on student options. The tutor can then email the student to let them know of deadlines etc.

SIGNATURE: _____

DATE: _____

Design Specification

This section will be generating design ideas based on the user requirements. Something that will be present on each module will be the font size: 12 and font family: Tahoma.

Here is a summary of each of the sections:

Module: 1 Main Menu

This page will contain two features. On the screen displays profiles for each student with their names written. This will allow the user to click on the student and access their details, he is then able to edit and add data such as their name, candidate number and of course he will be able to monitor each section of the coursework. This will also give an automatic average at the bottom of each section. The next feature that the main menu will have is a button which allows the user to access every student who he has. This was suggested by the Head of Department and confirmed by my client as they like to see the entire class progress at once.

There are no inputs in this stage, so no data needs to be validated. The only system that needs to be ensured is that all buttons are linked to the appropriate module (Module 2 and 3).

Module 2: Student profiles

As stated previously, this will be accessed through the main menu. The following features are required:

- Candidate name, candidate number, centre number, target grade and project title.
- Title of sections, this will be: Problem Definition, Investigation and Analysis, Design, Algorithms, Test Strategy, Software Development, Software Testing, Documentation, Evaluation – Object Success, Evaluation – User Response, Evaluation – Extensions.
- Description of each task under each section.
- Text field for each task which enables the user to input marks – This needs to be validated so the user cannot enter higher than the maximum grade.
- A read-only textbox which displays average score for each section. This will be totalled up and given an average accurate score for the student's final mark.
- Previous and Next page button to access the next stage of the tracking system as it will not fit on one page.

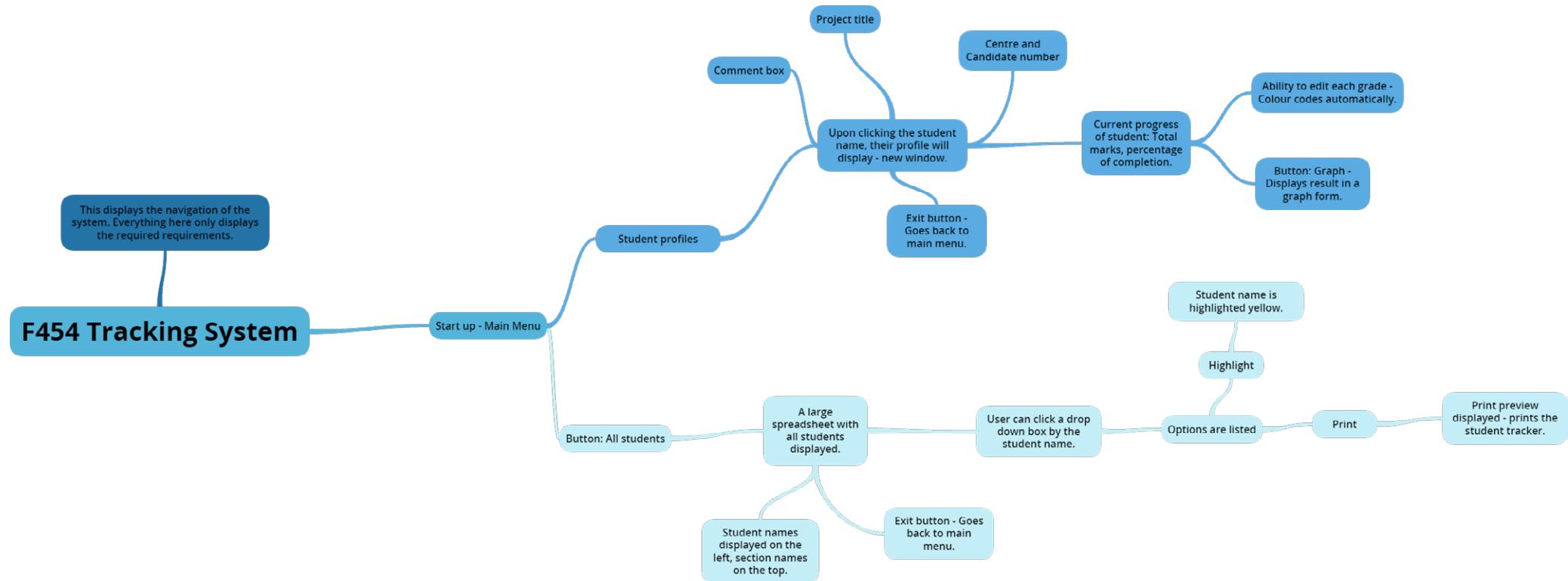
Module 3: Displaying all students

This button in the main menu allows the user to see every student's progress. It will be laid out similar to the current system – spreadsheet style. It will more than likely open Microsoft Excel but this will be investigated to see which is the most appropriate. This has been chosen as both client and Head of Department feel it is easy to use. On the page, it will contain:

- Textbox for:
 - Current marks for sections
 - Average marks
 - Completion percentage.
- Each student's name:
 - Can click on an option to do the following
 - Highlight (Done automatically upon clicking spreadsheet button).
 - Print (Microsoft Excel already provides this option)
 - Comment (Microsoft Excel already provides this option)

Walkthrough of the system (Design)

This displays what the user will see every time they click a button.



SIGNATURE: _____

DATE: _____

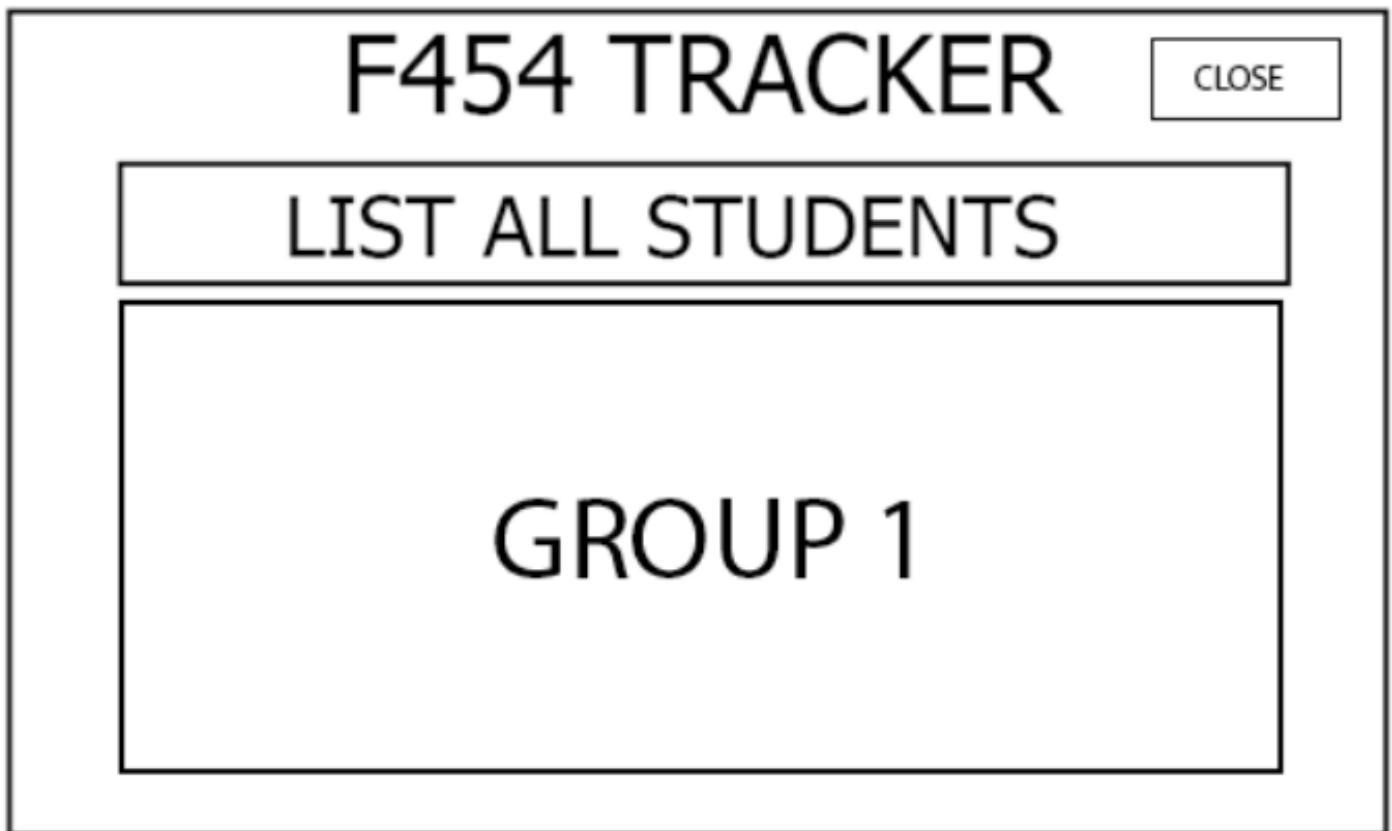
Interface Designs

During this process, I will be using the Top-Down modular design method. This method involves splitting up different sections of the system.

Main Menu Design

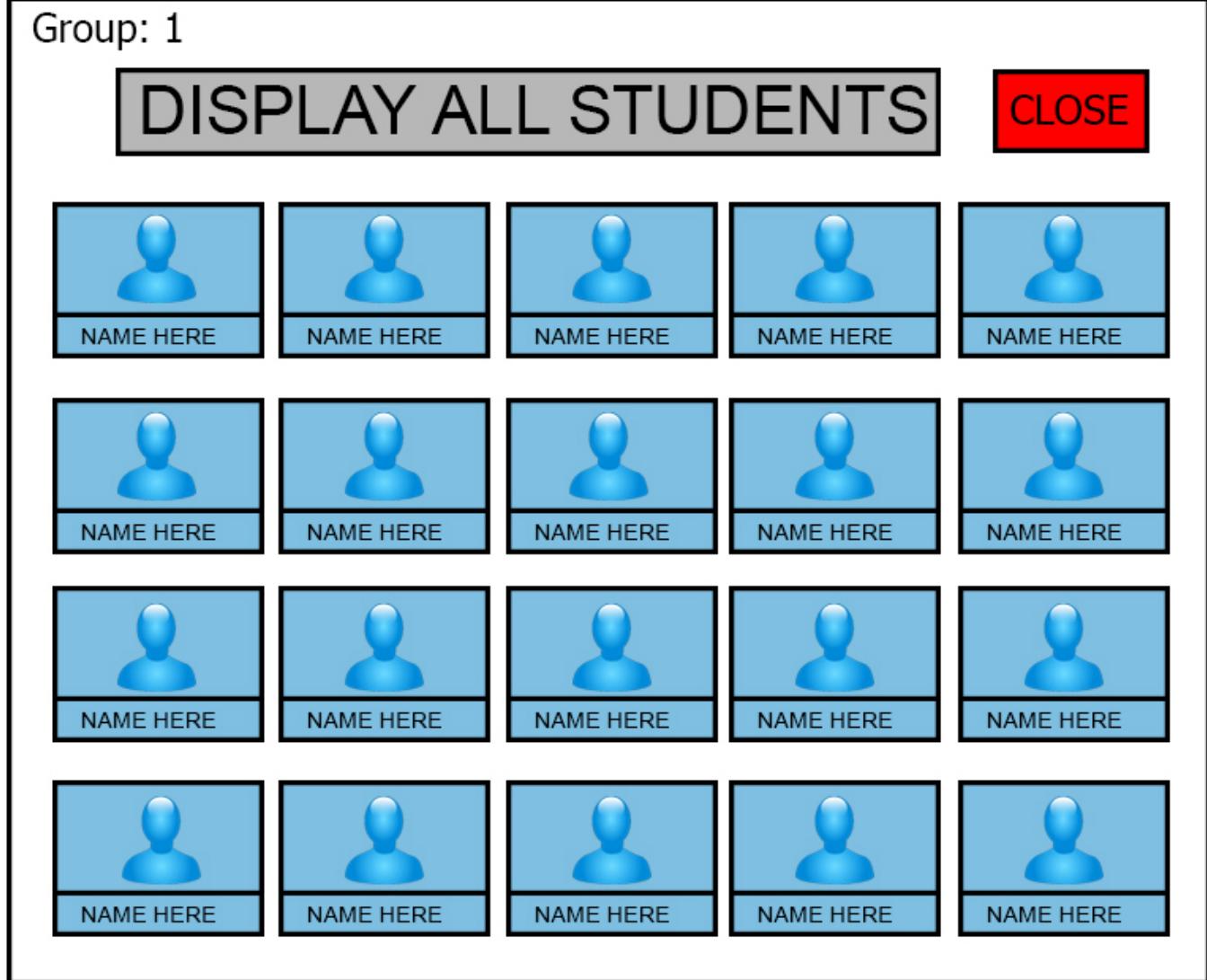
This is the first window displayed when the user runs the system. This allows the user to list all the students, which means accessing the spreadsheet. The Group 1 is for each of his students.

There are two designs that have been added below. One is a simple button click and the other is a dropdown menu box.



Personally, I prefer the Group 1 button as it is user-friendly.

Group: 1



As stated previously:

The main menu contains two options. One option is to search through profiles of student which enables the user to just focus on one pupil. The other option, the 'Display All Students' allows the user to search through the whole database and display all students on one screen. This will enable the user to see every student's progress.

Profile background colour will be light blue. 'Display all Students' button will be light grey. This has been chosen for simplicity sake and not to make it look overly done.

Validation

The data will be automatically validated as the user is limited to options and cannot enter their own piece of data. However, there will need to be a check for each button to ensure that it links to the right page.

Inputting student details

Before the client can use his application, he needs to set all the information of the candidates first.

STUDENT DETAILS

CLOSE

CANDIDATE NAME:

CANDIDATE NUMBER:

CENTRE NUMBER:

TARGET GRADE:

PROJECT TITLE:

SUBMIT

To add details to the profile section, my client must click on a profile. If it is empty, they will be greeted with this window. They are then able to enter the details and submit it. This will save and will replace the old profile.

They will be able to edit a student's information by going onto their profile and clicking the button located on the page.

Validation

- Candidate name can only be characters and no numbers. There will be a character limit of 30 and must be over 6 characters.
- Candidate number is limited to integers only and must be equal to 5 characters.
- Centre number is limited to integers only and must not exceed more than 4 characters.
- Target grade will be stored as a character variable, therefore it doesn't exceed over 1 character.
- Project Title will be a string variable, there will be no limit unless the characters exceed over 50.

Inputting student grades

CANDIDATE NUMBER:	<input type="text"/>	SHOW GRAPH
CENTRE NUMBER:	<input type="text"/>	EDIT
PROJECT TITLE:		
NAME:		Target Grade: #
Section A1: Problem Definition		
- CRITERIA HERE -	<input type="text"/>	
- CRITERIA HERE -	<input type="text"/>	
- CRITERIA HERE -	<input type="text"/>	
- CRITERIA HERE -	<input type="text"/>	
- CRITERIA HERE -	<input type="text"/>	
	Total: <input type="text"/>	
Section A2: Investigation and Analysis		
- CRITERIA HERE -	<input type="text"/>	
- CRITERIA HERE -	<input type="text"/>	
- CRITERIA HERE -	<input type="text"/>	
- CRITERIA HERE -	<input type="text"/>	
- CRITERIA HERE -	<input type="text"/>	
- CRITERIA HERE -	<input type="text"/>	
	Total: <input type="text"/>	
PREVIOUS		NEXT
Goes to previous stages		Goes to next stage

Upon selecting a profile for a student, the user will be greeted with this screen. This contains all the necessary details which had been outlined above.

However, to summarise once more:

The candidate number, centre number, project title, target grade and name are all variables and would be automatically carried over to the page. The tutor can input a number into the field which then totals up the average automatically in the read-only textbox provided, which has been labelled as 'Total'.

Bottom of the page contains two buttons, previous and next. As there are many sections, it is not possible to fit everything onto one screen unless the user requests the size of the system to be 1920x1080 dimensions. There will not be a 'Previous' button on the first section, however it has been shown here just as a preview.

Lastly, a graph button will be present which allows the user to see a student's progress in graph form. The interface for this has been posted below.

Validation

On this page, there will be validated data to ensure that no errors occur and it also avoids the system from crashing.

- Each section of the coursework will have a maximum amount of marks, this is stated below:

Section Name	Maximum amount of marks
Problem Definition	3
Investigation and Analysis	11
Design	6
Algorithms	5
Test Strategy	5
Software Development	16
Software Testing	14
Documentation	10
Evaluation – Objective Success	4
Evaluation – User Response	3
Evaluation – Extensions	3

- The minimum for each is 0, so anything below the set amount will be validated and will also result a prompt message if it goes below.
- Total section will be read only.
- Name and Target Grade will be updated automatically so the user will not be able to edit it on the page.
- Each of the marks will be integer value as you cannot get decimal numbers.

F454 TRACKER

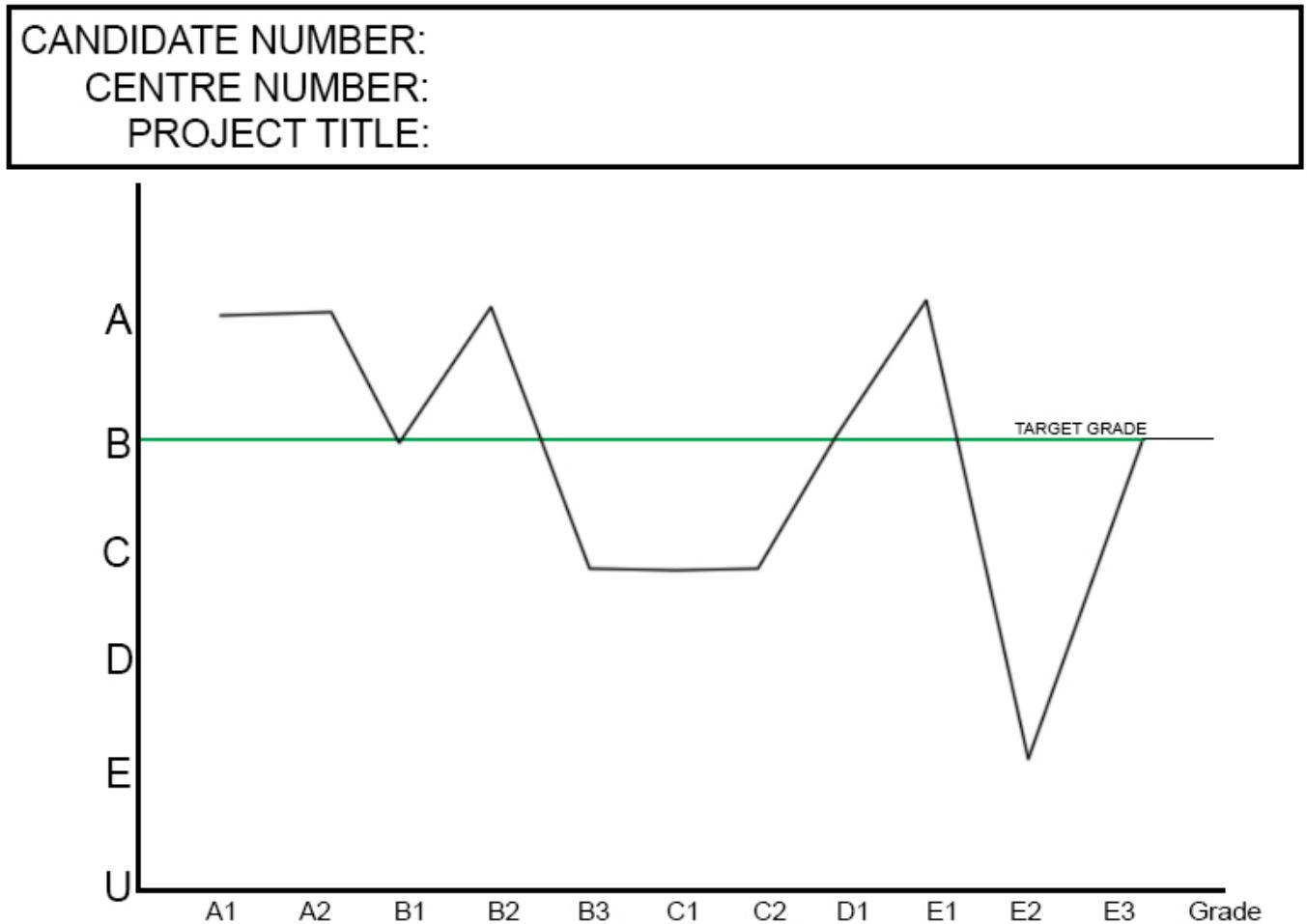
Spreadsheet tracking – All students

This is a rough design of how the tracker may look like. This window appears upon clicking the 'Display all students' button. T represents total for each section which the system will calculate automatically. The spreadsheet will continue to expand until the last section is displayed. At the end of all of the sections, there will be two extra fields: Percentage and grade.

Percentage shows how much the student has completed the coursework overall. The grade shows what the student's current progress is and will represent that by displaying a letter for the grade.

Validation on spreadsheet

Similar to the profile tracker, the spreadsheet will be validated with the amount of marks that can be added. The section for Names, %, Target and Grade will all be read-only as it will be calculated automatically.



This graph can be accessed through going on a student's profile. The graph above is an example of what a student's progress may look like if they have attempted every section of the coursework. If they have not, there will be gaps within the graph.

The line across B displays a student's target grade. This can change. This also means a variable needs to be created which judges the Y axis of the graph.

Validation

There will be no validation needed in this section as there are no inputs. The graph has been created through inputs from other sections of the system.

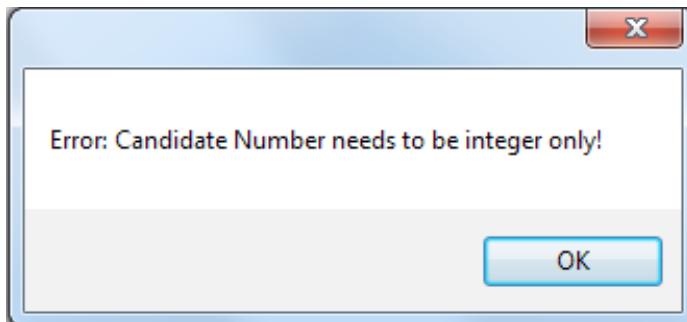
Error messages

If the data a user has inputted was incorrect, a message box will appear prompting the user that a mistake was made. To do this, I will be using the `MessageBox.Show();` syntax and will also be using try and catch method to ensure that the system does not crash if an input does not match the validated data.

Below will display each section of the error messages and what all of them will be used for:

Student Profile Input

There are several validated data in this section. The message box will look something like below:



Candidate Name: "Error: The characters you have entered are too long for candidate name!"

Candidate Number: "Error: Candidate number can only be integer!"

Centre Number: "Error: Centre Number can only be integer!"

Target Grade: "Please make sure the target grade is one letter!"

Project Title: "That Project title is too long!"

Inputting student grades

"Error: Please make sure that the grade you have entered is correct!"

Tracking spread sheet

"Error: The marks you have entered is incorrect!"

Interface Feedback (Interview)

An interview with the client in this section is essential as both parties (Programmer and Client) need to know exactly what is going to be built and must have an agreement.

Interview summary

After showing my client the designs that were created for his system, he agreed to most aspects of the application.

These are the features my client wanted to be added and edited into the system:

- Comment notes should be easier to see. My client requested a comment box in the profile section for students on the side of each section of the coursework.
- An end-of coursework cover sheet was also requested by my client. This will display each section of the coursework and an end mark. This will not have sub sections, but instead the total score for each main section such as Problem Definition or Investigation & Analysis.

On the sheet will also contain a small comment box under the sections.

- He was unsure whether or not he wanted a bar chart or a line graph to represent the results. Therefore, this is an undecided design and so different designs will be created which can be used in the near future for when the application is being developed.

SIGNATURE: _____

DATE: _____

Interview Response

After the interview, I took the feedback given and updated the sections which my client wanted.

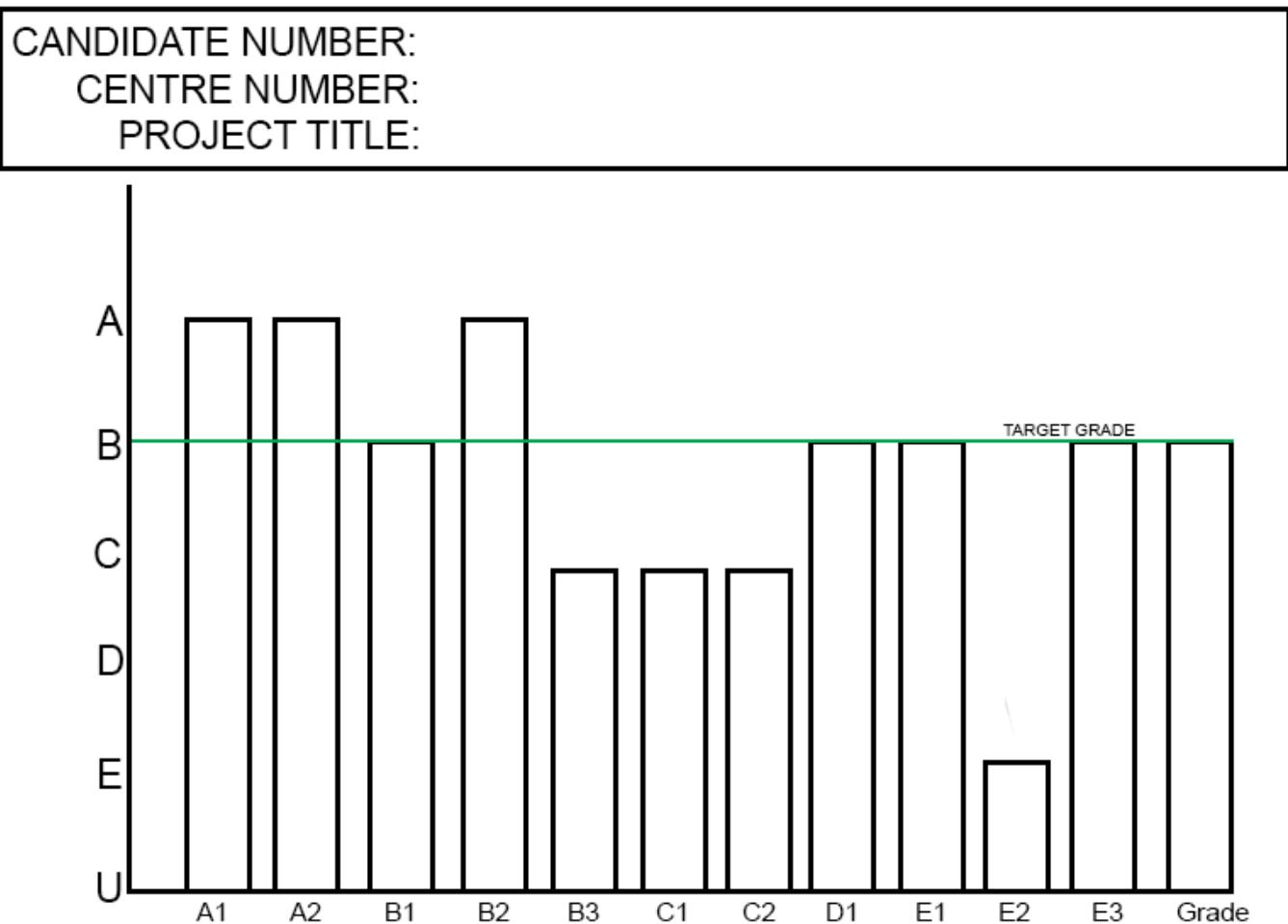
Inputting Student Grades

CANDIDATE NUMBER:	<input type="button" value="SHOW GRAPH"/>	
CENTRE NUMBER:	<input type="button" value="EDIT"/>	
PROJECT TITLE:	<input type="button" value="COVER SHEET"/>	
NAME: _____	Target Grade: #	
Section A1: Problem Definition		
- CRITERIA HERE -	<input type="text"/> <input type="button" value="COMMENTS"/>	
- CRITERIA HERE -	<input type="text"/> <input type="button" value="COMMENTS"/>	
- CRITERIA HERE -	<input type="text"/> <input type="button" value="COMMENTS"/>	
- CRITERIA HERE -	<input type="text"/> <input type="button" value="COMMENTS"/>	
- CRITERIA HERE -	<input type="text"/> <input type="button" value="COMMENTS"/>	
Total: <input type="text"/>		
Section A2: Investigation and Analysis		
- CRITERIA HERE -	<input type="text"/> <input type="button" value="COMMENTS"/>	
- CRITERIA HERE -	<input type="text"/> <input type="button" value="COMMENTS"/>	
- CRITERIA HERE -	<input type="text"/> <input type="button" value="COMMENTS"/>	
- CRITERIA HERE -	<input type="text"/> <input type="button" value="COMMENTS"/>	
- CRITERIA HERE -	<input type="text"/> <input type="button" value="COMMENTS"/>	
- CRITERIA HERE -	<input type="text"/> <input type="button" value="COMMENTS"/>	
Total: <input type="text"/>		
PREVIOUS		NEXT
Goes to previous stages		Goes to next stage

Inputting Student Grades

This is the new and improved feature that the client requested.

Firstly, at the top right an extra button has been created to access the coversheet. This allows the user to be able to input any details that are required. Next, there is a comments section to the side of the grades section which allows the user to be able to enter data which can be accessed by the student.

Tracking Graph – Bar Chart

This is simply a different version of the chart. Instead of a line graph, there is a bar chart as requested by the client.

F454 Coversheet
**Computing GCE H447 Unit F454
Coursework Cover Sheet**


Candidate Name: _____
Candidate Number: _____

Center Name: _____
Center Number: _____

(a) Definition, Investigation and Analysis

	MAX MARK	MARK
Definition - nature of the problem to be investigated // TEACHER COMMENTS CAN GO HERE		
Investigation and Analysis // TEACHER COMMENTS CAN GO HERE		
TOTAL		

(b) Design

	MAX MARK	MARK
(i) Nature of the solution // TEACHER COMMENTS CAN GO HERE		
(ii) Investigation and Analysis // TEACHER COMMENTS CAN GO HERE		
(iii) Test Strategy // TEACHER COMMENTS CAN GO HERE		
TOTAL		

(c) Software Development and Testing

	MAX MARK	MARK
(i) Software Development // TEACHER COMMENTS CAN GO HERE		
(ii) Testing // TEACHER COMMENTS CAN GO HERE		
TOTAL		

Data structures and variables

Data structures and variables need to be planned to ensure that as a programmer, I am naming variables sensibly and it also ensures that all names have been noted down to avoid confusion for future situations.

Variable Name	Data Type	Description	Validation
studentUniqueNumber	Global integer	Holds a student's number. Each student is incremented by one.	It is not an input and so there is no validation.
highlightButton	Global integer array	Holds which student has been highlighted	This is not an input and so there is no validation.
finalStudentTotals	Global integer array	Holds all the total marks for all students	Validation will have already happened. This is just addition.
studentGradesTotals	Global integer array	Holds all the total marks for each section (eg. A1, A2) for all students.	Validation will have already happened. This is just addition.
studentComments	Global string array	Holds the comments for all students	No more than 280 characters long.
allStudentGrades	Global int array	This holds each single mark of each student. This will eventually add up to give a total for the section.	Each variable has a maximum grade. This is validated in the textbox and not the array.
allStudentNames	Global string array	Holds the information on the students such as names.	Each array index will have its own limitation. All the data is validated in the textbox.

The above are all needed for most pages to be displayed on the top of the window.

Note: Comments will be written on each variable.

All the arrays contain the data for each sub mark. These sub marks will add up together and will be given an average number. This will be aTotal.

Formula:

`aOne[0]+aOne[1]+aone[2]+aOne[3] = aTotal;`

The try and catch syntax will be used. The code will try and see if it can divide the number, as some may result in a decimal number. If the number is a decimal, it will average to its closest full number.

The plan is that I will be using the C# database function. All the data for students will be stored in this XML database. After it has been saved in the database, data will be retrieved and will appear in all the sections of the system: Spreadsheet, profile and graph.

I will be using the list syntax that is available in Visual Studio. This will be used to store the data on a 3rd party file, which will more than likely be notepad. This will allow me to arrange the data in a sequential way, so that the data is stored in the order of: Name, Candidate number, centre number, project title and target grade. To differentiate between different students, we can use the candidate number as it is unique to every person.

PRINT OUT SHEET EXAMPLE

CANDIDATE NUMBER: [REDACTED]
 CENTRE NUMBER: [REDACTED]
 PROJECT TITLE: F454 Tracker

NAME: [REDACTED] **Target Grade:** A

Section A1: Problem Definition

- Brief Description of the end user-...
 Current description of problem-...
 Clear statements of origins-...
 Further research-...

3
3
3
3
3

Total: **3****Section A2: Investigation and Analysis**

- I have done-...
 I have detailed evidence-...
 I have fully researched-...
 I have a detailed user-...

3
3
5
5
4

Total: **4****Section B1: Design**

- I have created-...
 I a clear design-...
 I have clear evidence-...
 I have a detailed-...
 My design could-...

1
1
0
1
0
1

Total: **1****Section B2: Algorithms**

- A complete set-...
 I have described-...
 I have evidence-...
 I have shown-...

5
5
6
8

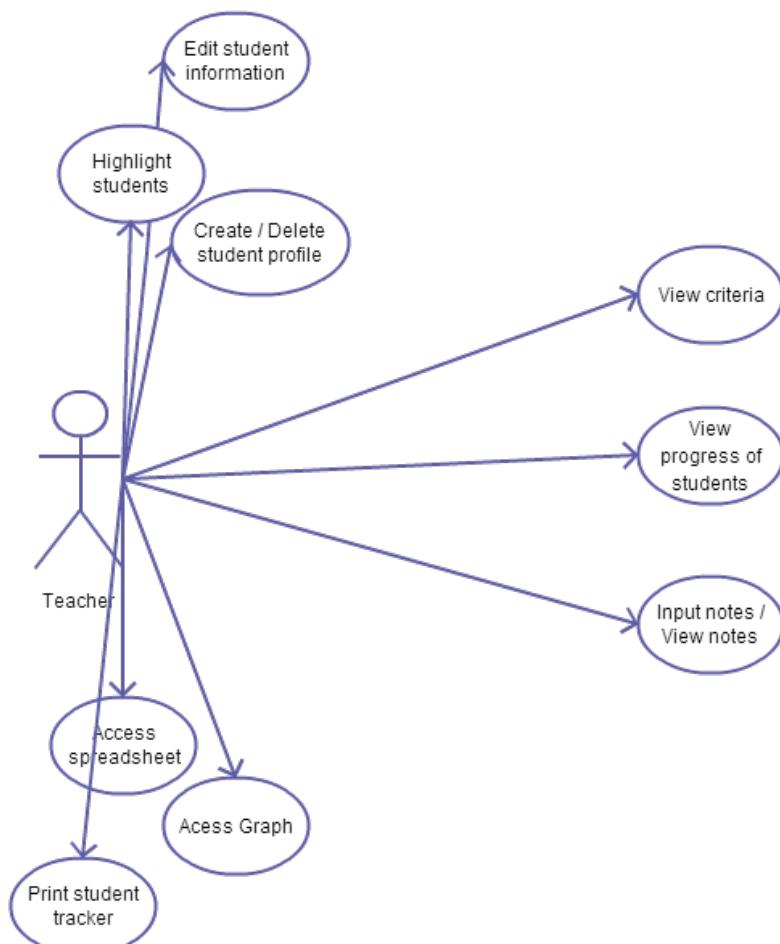
Total: **6**

UML Diagram (Use Case)

A Use Case diagram displays what each user is able to do on a system.

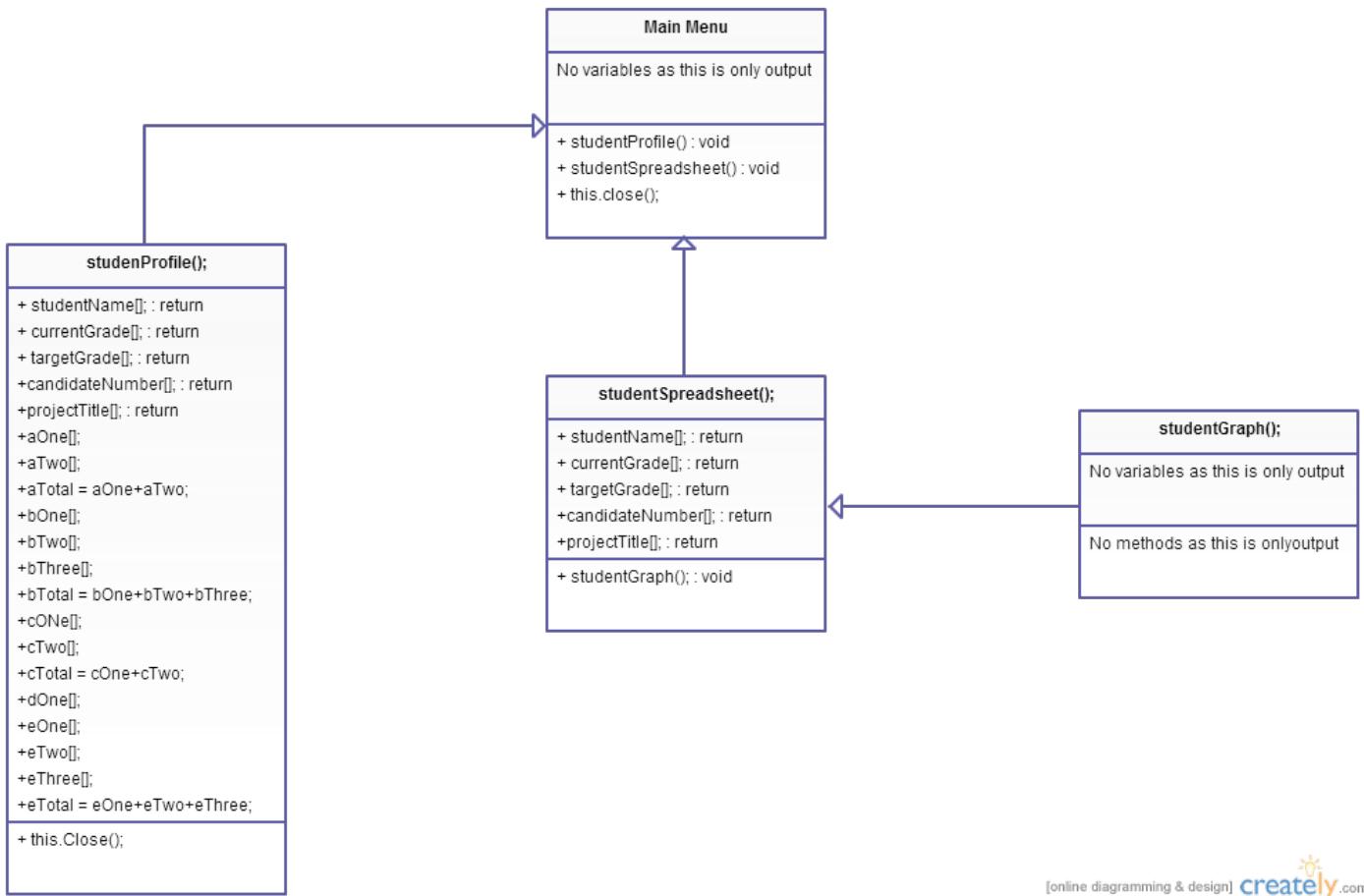
The teacher has access to everything, which means:

- Add grades / marks for each student
- Edit student information such as:
 - Name
 - Candidate number
 - Centre number
 - Target grade
- Create / delete a student profile
- View a graph version of a student's progress
- Have access to the spreadsheet
- Highlight students who the teacher may be worried about
- Print student tracker



UML Diagram (Class Diagram)

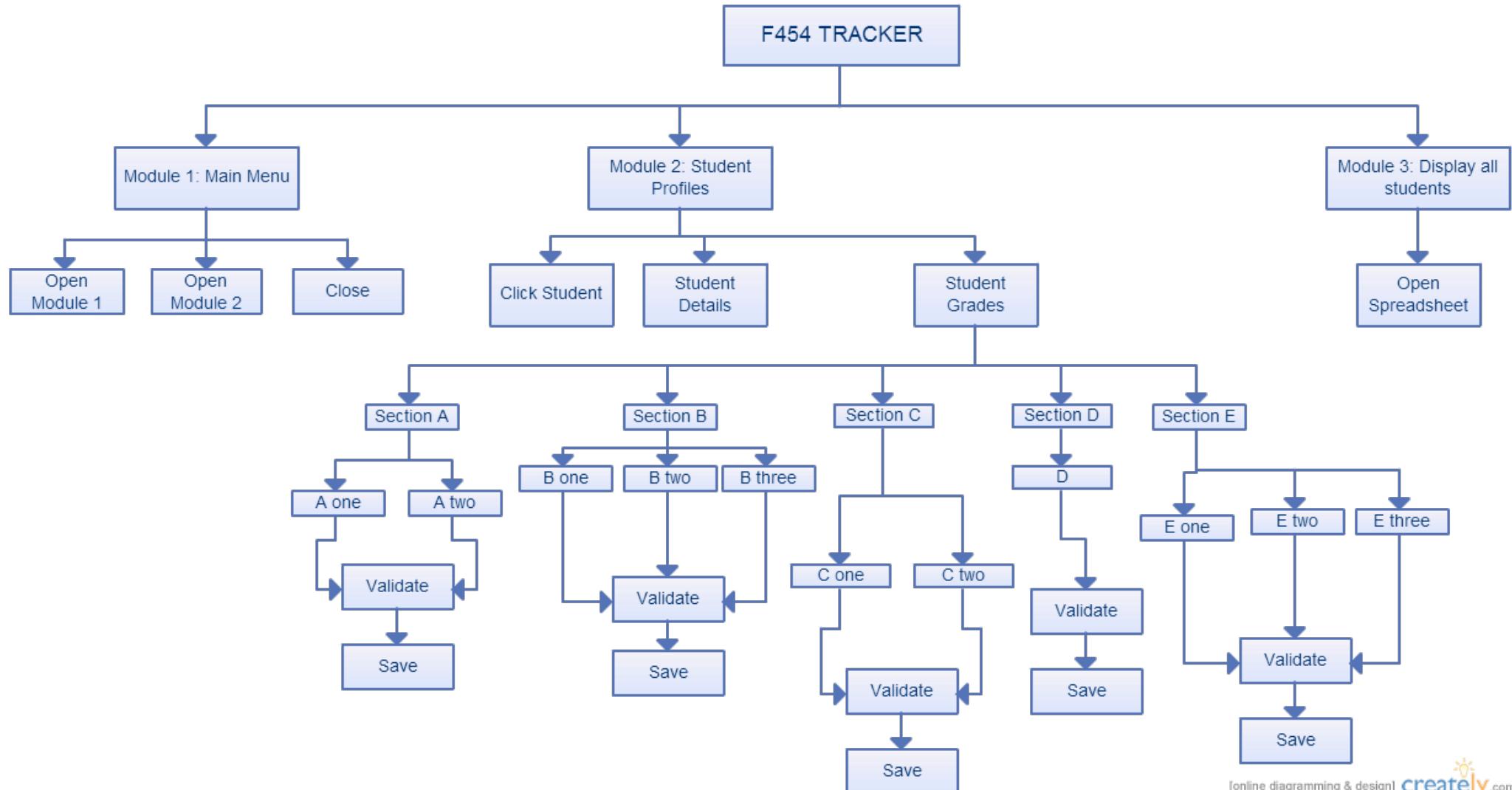
This UML diagram displays the methods and variables used in the new system.



[online diagramming & design]  [creately.com](#)

Modular Design

This graph shows all the sections of the system split into its own category.



B2: Algorithms

Before creating the program, algorithms need to be created so it is easier for the programmer to begin programming the system. It also allows to keep things in order. In the above Modular Design, I have labelled each module by number. Below I will describe each of the module.

Module 1: Main Menu

Main Menu contains two buttons. One to open up a spreadsheet (Module 2) and one to open up a classroom of students (Module 3)

Module 2: Student Profiles

Student Profiles button links to classrooms. The user can select a classroom to view the students inside it. The user can then select a student and fill in any details he wishes. The following data is validated: Project Title, Candidate Number, Candidate Name, Centre Number and Target Grade. This data will also link up to Module 3.

Module 3: Display all students

Module 3 opens up a spreadsheet. This spreadsheet will contain all the data that the user has inputted in Module 2. This will be done with the use of the Visual Studio database.

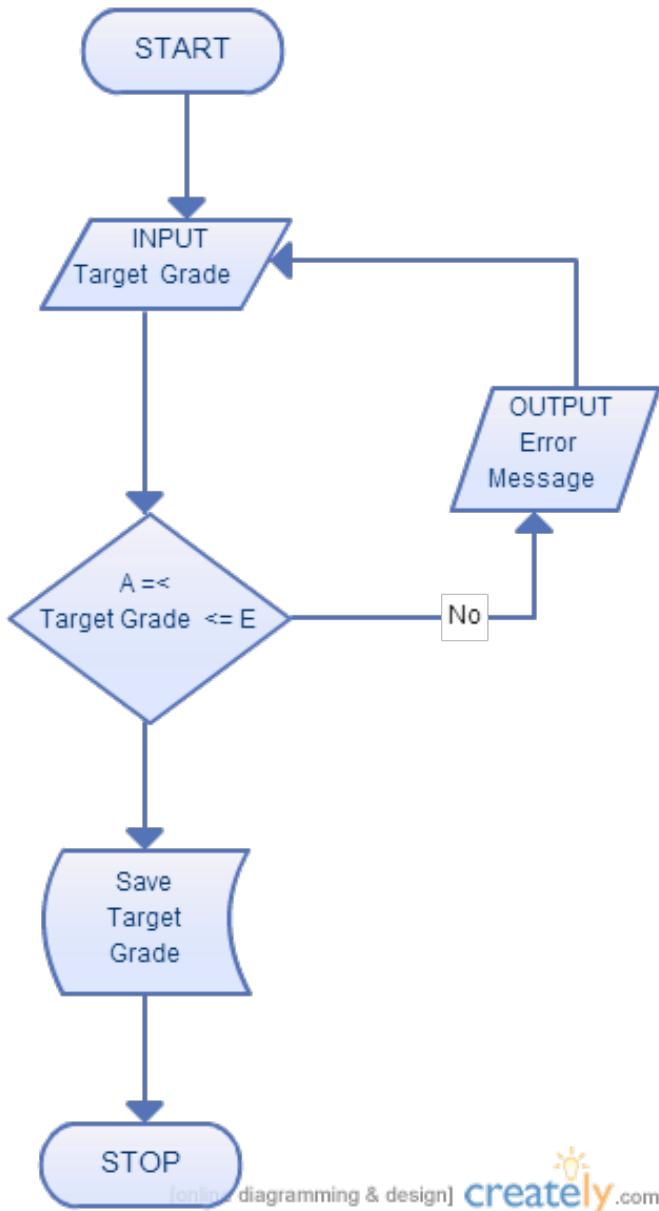
The following algorithms will be needed:

- Main Menu
- Student Profile / Group
- Spreadsheet

There will also need to be a plan of how validation will happen. This will be represented through Flowchart and Pseudo code.

Validation

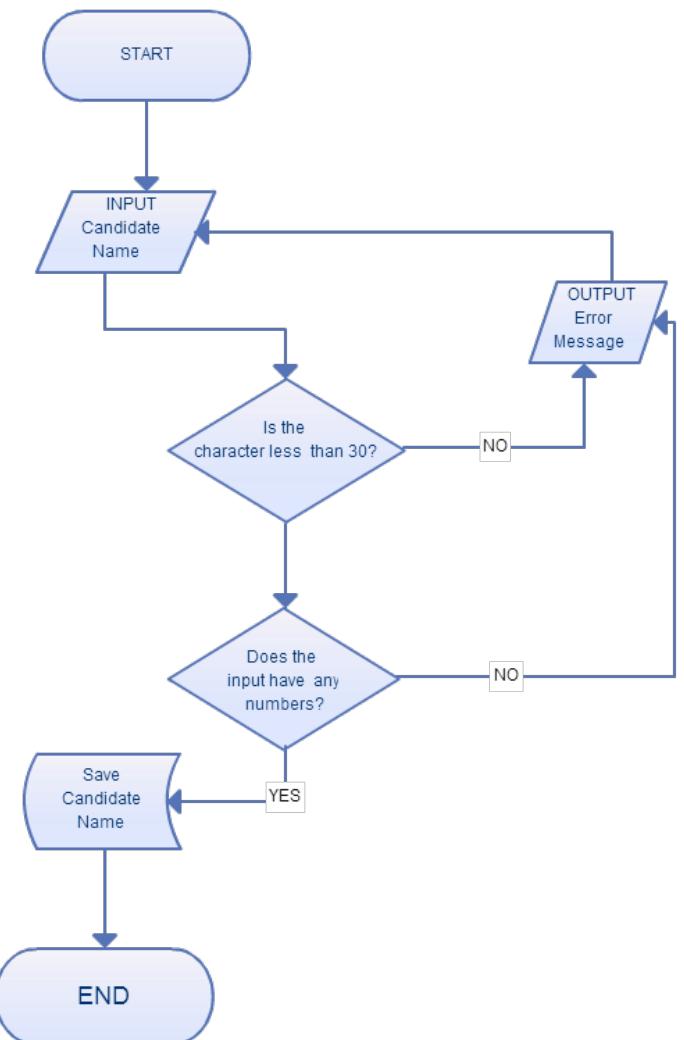
Target Grade Candidate Name



This is a simple method to ensure that the target grade is validated.

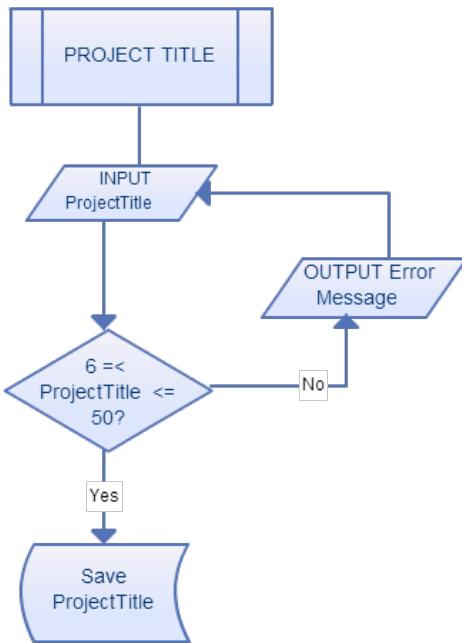
There will be a check which searches whether the input is between A to E. If it is not, then there will be an error message and the user will need to input the data again.

If the data is accepted, then it will be saved appropriately.



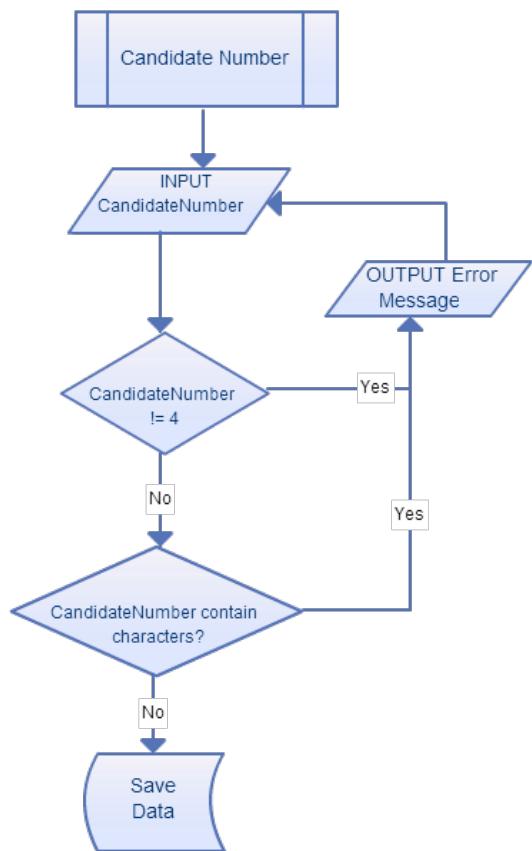
This is another simple way to ensure that candidate name is validated. This will check to see if both of the criteria is met, which is: Is the characters less than 30? Does the input have any numbers?

1

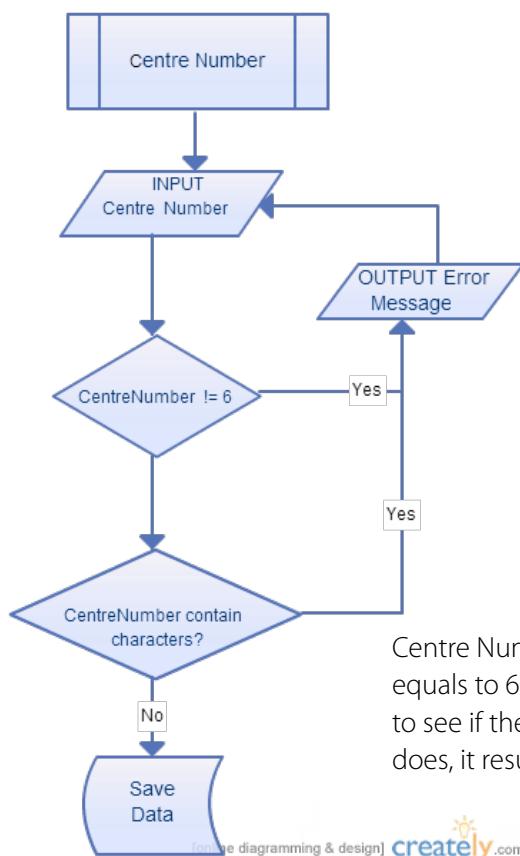


This is the validation for the Project Title. It checks to see whether the title is within the bounds of 6 to 50.

3



4



Centre Number checks to see if it equals to 6. If it does, then it checks to see if there are any alphabets, if it does, it results in an error.

This is the check for the Candidate Number, this checks whether or not the candidate number has an equal size of 4 characters. After that, it checks to see if there are any characters as Candidate Number must only have integers.

Candidate Number & Centre Number

INPUT CandidateNumber

Candidate Number

```

IF CandidateNumber.length != 4
THEN
    OUTPUT "Error: Your candidate number is 4 characters long!"
END THEN
ELSE IF Candidate.Number contains characters
THEN
    OUTPUT "Error: Your candidate number can only be numbers!"
END THEN
END IF
SAVE DATA

```

Centre Number

INPUT CentreNumber

```

IF CentreNumber.length != 6
THEN
    OUTPUT "Error: Your Centre Number must be less than 30 characters
long!"
END THEN
ELSE IF CEntreNumber contains characters
THEN
    OUTPUT "Error: Your centre can only be integers!"
END THEN
END IF

```

Project Title

```

01 INPUT ProjectTitle
02 IF ProjectTitle.length >= 50 characters
03 THEN
04     OUTPUT "Error: Your project title is too long!"
05 END THEN
06 ELSE
07     SAVE DATA
08 END ELSE
09 END IF

```

System

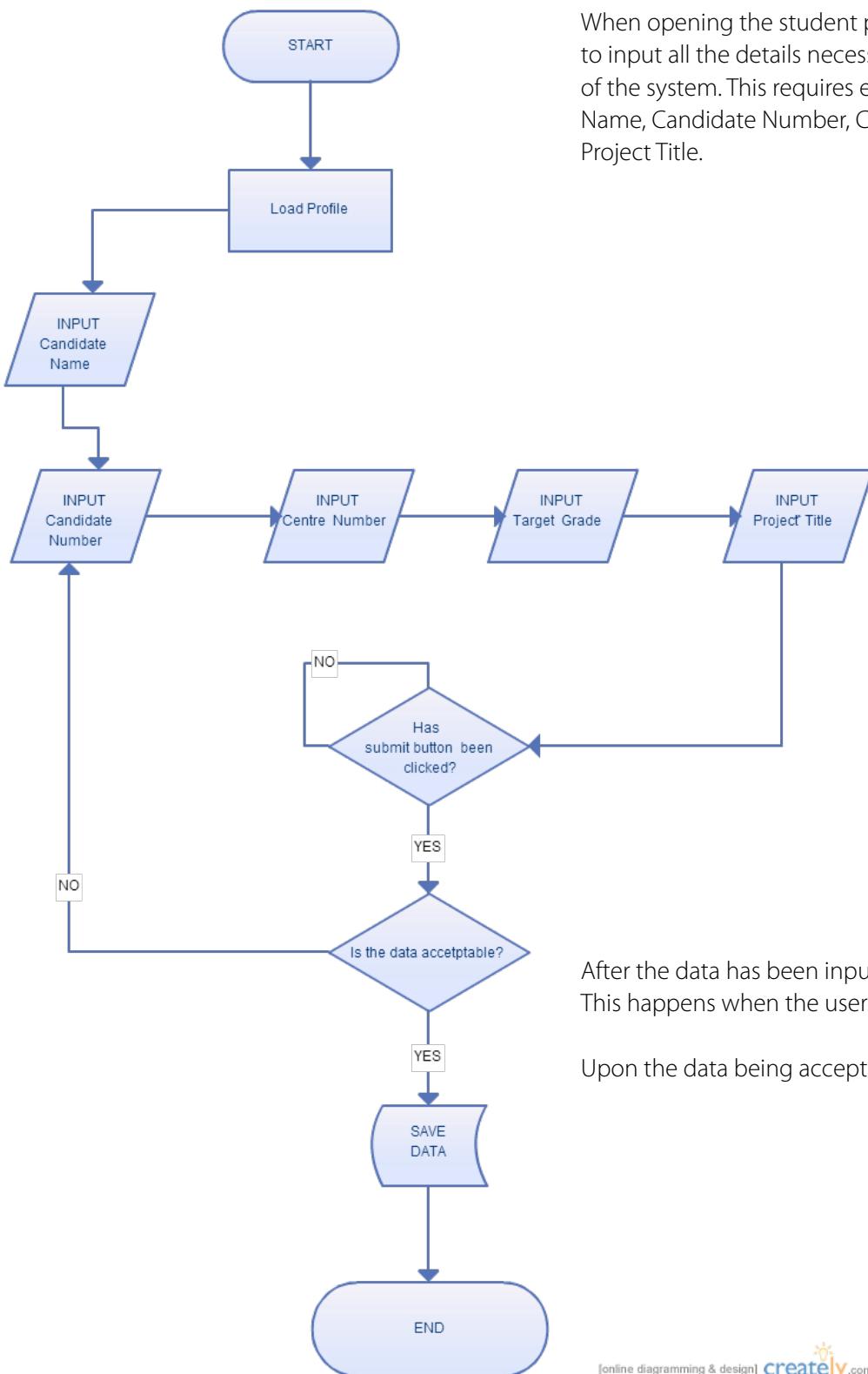
Main Menu

The main menu needs to be able to do the following:

- Open up different sections of the system –
 - Spreadsheet
 - Different classes / Groups

When the application first loads up, there will be two options. You can either load the group that the student is in, or you can load the spreadsheet which has all the information on every student. This will be two buttons which will open up a new window.

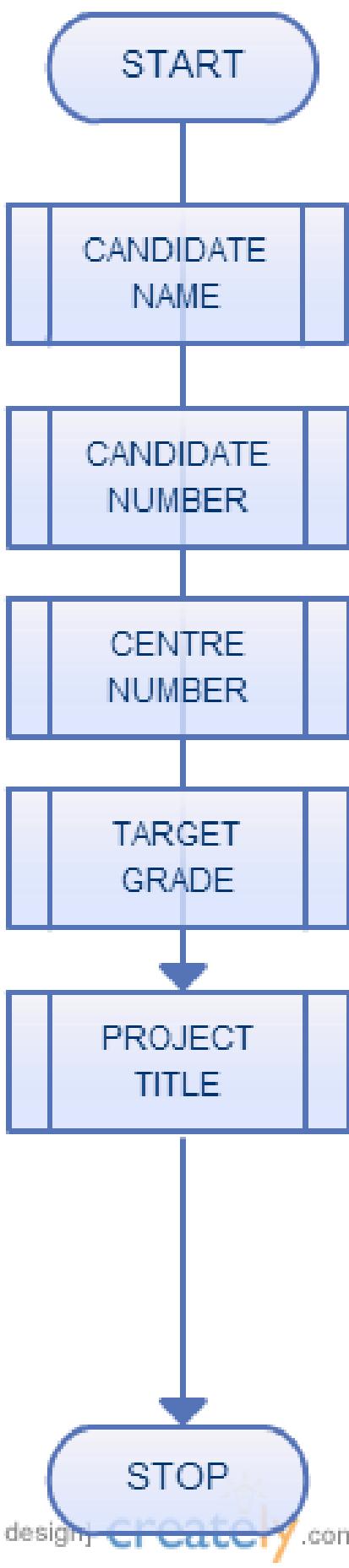
Group / Student Profile



When opening the student profile (Group), the user will need to input all the details necessary to progress to the next stage of the system. This requires entering the following: Candidate Name, Candidate Number, Centre Number, Target Grade and Project Title.

After the data has been inputted, validation needs to occur. This happens when the user hits the submit button.

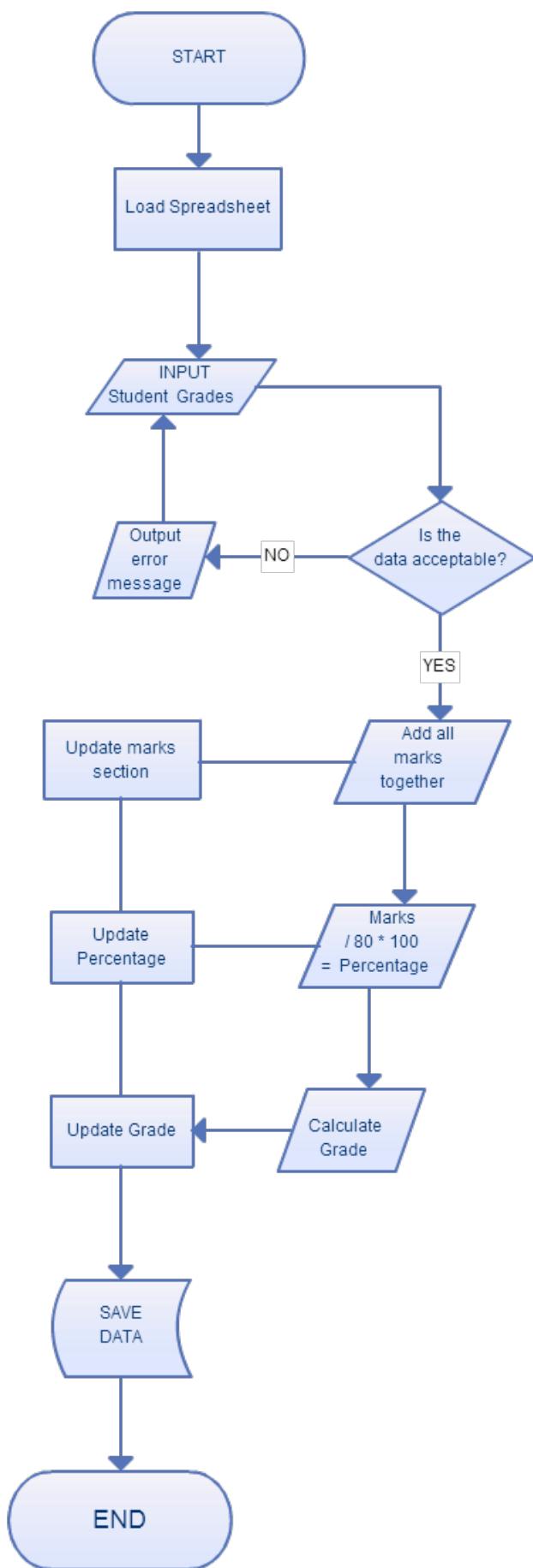
Upon the data being accepted, data will be saved.

Student

When opening a student profile, the following data need to be loaded. These are already stored in the database, and so it will be withdrawn from the database and printed onto the student profile. Similarly, this will happen on the spreadsheet.

If the data is not acceptable, then the user will be prompted a message. They will then need to edit the information and hit submit again to make sure all the data is correct. The error messages have been displayed in the Design section.

Spreadsheet



Upon the spreadsheet loading, the user has the ability to edit data.

This spreadsheet is not part of the system but instead opens the excel spreadsheet. Inside the spreadsheet there will need to be some calculations added.

Tracing Table

In this stage, I will be testing all the algorithms and pseudo code created to see its functionality. For each test, I will use different examples to show what would happen depending on various inputs.

Line	Input	Result	Comments
01	F454 Tracking System	N/A	This checks whether the input was empty or not
02		N/A	If there were no text, this would display an error message
03		N/A	If the title has over 30 characters, then it would display an error
04		N/A	Error message would appear
05		N/A	Checks whether the user has typed in anything on their keyboard
06		N/A	Stops any numbers from being inputted
07		candidateName save	Saves the location to the variable

To test algorithms, I will be using a table to display the input, the validation and whether data is accepted or not. I will also be using three different types of validation, which are: Extreme, Erroneous and Normal data. Extreme data is data that is on the border of acceptable data. Erroneous data is data that is not accepted and does not meet the criteria which it is bounded by. Lastly, Normal data is data which is acceptable.

Target Grade

Input TargetGrade	A = < TargetGrade <= E	Is data accepted?	Type of validation	Output
A	Yes	Yes	Extreme	None
E	Yes	Yes	Extreme	None
F	No	No	Erroneous	Error Message
1	No	No	Erroneous	Error Message
C	Yes	Yes	Normal	None
B	Yes	Yes	Normal	None

Candidate Name

Input CandidateName	6 >= CandidateName >= 30	Any numbers?	Is data accepted?	Type of validation	Output
Michael Smith	Yes	No	Yes	Normal	None
Samuel Jones	Yes	No	Yes	Normal	None
chukwuebuka abachachachachachacha	No	No	No	Erroneous	Error Message
Simon Jones 06	Yes	Yes	No	Erroneous	Error Message
Jr Aliaune Damala Badara Thiam	Yes	No	Yes	Extreme	None
Ann Li	Yes	No	Yes	Extreme	None

Candidate Number

In this stage and also centre number, extreme data cannot be tested as there can only be one type of input.

Input CandidateNumber	Character equal to 4?	Any characters?	Is data accepted?	Type of validation	Output
2957	Yes	No	Yes	Normal	None
4301	Yes	No	Yes	Normal	None
10984184	No	No	No	Erroneous	Error Message
2	No	No	No	Erroneous	Error Message
T3st	Yes	Yes	No	Erroneous	Error Message

Centre Number

Input CentreNumber	Characters equal to 5?	Any Characters?	Is data accepted?	Type of validation	Output
34336	Yes	No	Yes	Normal	None
34335	Yes	No	Yes	Normal	None
1928431	No	No	No	Erroneous	Error Message
123ab	Yes	Yes	No	Erroneous	Error Message

Project Title

The project title will need to be above 6 characters at least as users will have the word 'System' in their title.

Input ProjectTitle	6 >= ProjectTitle >= 50	Is data accepted?	Type of validation	Output
BMI Calculator	Yes	Yes	Normal	None
F454 Tracking System	Yes	Yes	Normal	None
System	Yes	Yes	Extreme	None
abcdefghijklmnoprstuvwxyzabcdefghijklmnopqrstuvwxyz	Yes	Yes	Extreme	None
ProjectTitleProjectTitleProjectTitleProjectTitleProjectTitle	No	No	Erroneous	Error Message
InputOutputInputOutputInputOutputInputOutputInputOutput	No	No	Erroneous	Error Message

Adding comments

Input comment	Is the comment 280 characters or less?	Is data accepted?	Type of validation	Output
Testing	Yes	Yes	Normal	None
12345	Yes	Yes	Normal	None
** A comment with 280 characters **	Yes	Yes	Extreme	None
** A comment over 280 characters **	No	No	Erroneous	Error Message

Pseudocode Testing

On the side of each Pseudocode I have labelled a process number. I will be using this in the table to make it clear which process is being executed.

```

01 INPUT CandidateNumber
02     IF CandidateNumber.length != 4
03         THEN
04             OUTPUT "Error: Your candidate number is 4 characters long!"
05         END THEN
06     ELSE IF Candidate.Number contains alphabet
07         THEN
08             OUTPUT "Error: Your candidate number can only be numbers!"
09         END THEN
10     END IF
11 SAVE DATA

```

Valid data

	Input	Output	Process
01	123456		User inputs their candidate number
02			Checks if the length is not 4. It is, so it skips to the next else statement.
06			Checks to see if the input has any alphabets. It doesn't, so it moves to the end of the IF statement.
11			Data is saved in the database. The variable holds the location in memory.

Invalid data

	Input	Output	Process
01	123		User inputs their candidate number
02			Checks if the length is not 4. It isn't and skips to the THEN statement.
04		Error: Your candidate number is 4 characters long!	Outputs error message.

Invalid data

	Input	Output	Process
01	abcdef		User inputs their candidate number
02			Checks if the length is not 4. It is, so it skips to the end of the IF statement.
06			Checks to see if the input contains any alphabets. It does, so it moves to the THEN statement.
08		Error: Your candidate number can only be numbers!	Outputs error message.

```

01 INPUT CandidateName
02     IF CandidateName.length >= 30
03         THEN
04             OUTPUT "Error: Your candidate name must be less than 30
                characters long!"
05         END THEN
06     ELSE IF CandidateName contains integers
07         THEN
08             OUTPUT "Error: Your candidate name must not have any integers!"
09         END THEN
10     END IF
11 SAVE DATA

```

Valid data

	Input	Output	Process
01	[REDACTED]		User inputs their candidate number
02			Checks if the length is above 30. It isn't, so it skips to the next IF statement.
06			Checks to see if the input has any integers. It doesn't, so it skips to the end of the IF statement.
11			Data is saved in the database. The variable holds the location in memory.

Invalid data

	Input	Output	Process
01	abcdefghijklmn opqrstuvwxyzabcdef		User inputs their candidate name
02			Checks if the length is above 30. It is so it moves to the next THEN statement.
04		"Error: Your candidate name must be less than 30 characters long!"	Outputs error message

Invalid data

	Input	Output	Process
01	1234		User inputs their candidate name
02			Checks if the length is above 30. It isn't so it skips to the next IF statement.
06			Checks to see if candidateName has integers. It does so it goes to the THEN statement.
08		Error: Your candidate name must not have any integers!"	Outputs error message.

Project Title

```

01 INPUT ProjectTitle
02 IF ProjectTitle.length >= 50 characters
03   THEN
04     OUTPUT "Error: Your project title is too long!"
05   END THEN
06   ELSE
07     SAVE DATA
08   END ELSE
09 END IF

```

Valid data

	Input	Output	Process
01	Project F454 Tracking Students Coursework Progress System		User inputs their Project Title
02			Checks if the length is over 50 characters.
03			It is so it moves to the THEN statement.
04		Error: Your project title is too long!	Message is displayed to alert the user.

Invalid data

	Input	Output	Process
01	F454 Tracking System		User inputs their Project Title
02			Checks if the length is over 50 characters.
06			It isn't, so it moves to the next else statement.
07			Data is saved

B3: Test Strategy

I will need to create a test plan to make sure that the system has no bugs or glitches. There will be several stages of testing.

Alpha, Beta, Acceptance Testing

The first test is Alpha, which is testing the basic functionality of the system. This is normally done before the system is ready but is there to ensure that the user requirements have been met. As part of **Alpha testing**, each of the modules will have been checked to ensure they all link up and have the right data validated. Within Alpha testing is White Box testing, during this phase I will need to test the code to make sure it is all correct. I will be doing this by:

- Main menu button links to the correct groups
- All marks are calculated correctly
- Data is transferred over appropriately from database to excel
- The right results for outputs displayed for bar chart
- Code is indented properly and commented well

During the Black Box testing, I will be looking mainly at the interface and how the inputs and outputs are done. I will be inputting various details to make sure that the correct results are outputted and that the correct restrictions are in place such as:

- Inputting correct and incorrect information in the student form
- Inputting correct and incorrect information for student marks
- Testing out each button to make sure they link to the right module
 - Next button links to the next page, cover sheet and graph will need to open an excel file.
- Bar chart represents the right data
- Comments are saved
- Candidate name and number, centre number, project title and target grade are properly displayed.

After Alpha testing has been conducted, I will refine the system and allow the end user to use the system to get some feedback; this is Beta testing. Every version of the system I give the user, I will be entering each of the feedback and criteria the user gives me. This will show evidence that the user has wanted specific features so in the future, there are no confusion. I have also created a set of 'profiles' that must be tested.

Profile 1

Candidate Name: John Smith

Candidate Number: 4910

Centre Number: 99999

Target Grade: B

Project Title: Web Design Database System

This profile should be accepted as all the data entered is correct.

PROFILE 1	ANY ERRORS?	COMMENTS
CANDIDATE NAME		
CANDIDATE NUMBER		
CENTRE NUMBER		
TARGET GRADE		
PROJECT TITLE		

Profile 2

Candidate Name: John Smith
 Candidate Number: 20191
 Centre Number: 3430
 Target Grade: C
 Project Title: Spreadsheet creator System

This profile should not be accepted as both Candidate Number and Centre Number are in the wrong range.

PROFILE 2	ANY ERRORS?	COMMENTS
CANDIDATE NAME		
CANDIDATE NUMBER		
CENTRE NUMBER		
TARGET GRADE		
PROJECT TITLE		

Profile 3

Candidate Name: 55smith.j
 Candidate Number: 1010
 Centre Number: 99999
 Target Grade: F
 Project Title: 2D Platform Game

This profile should not be accepted as both Candidate Name and Target Grade are in the wrong range.

PROFILE 3	ANY ERRORS?	COMMENTS
CANDIDATE NAME		
CANDIDATE NUMBER		
CENTRE NUMBER		
TARGET GRADE		
PROJECT TITLE		

Profile 4

This will be completely blank.

Candidate Name:
 Candidate Number:
 Centre Number:
 Target Grade:
 Project Title:

This profile should not be accepted as there has been no input.

Another way I can conduct Beta testing is allowing other users to test the system. A set of criteria can be given to this group of people which will allow a simulation of what real data could produce. The set of criteria I will be giving is the specification that has been listed above.

The last testing that will be conducted is acceptance testing which is when the final version has been created. It will be tested under the criteria of the system in all different types of scenarios. This also means that the end user will be able to see the testing's conducted and can say whether he agrees with the completion of the system or not. These are the following features that need to be tested and the set of criteria's that meet the user's requirement:

- Menu button links up to all appropriate modules.
 - Group will link up to a class full of students

- ‘LIST ALL STUDENTS’ will direct the user to an Excel file
- Class full of students will contain buttons with different students
 - Buttons with no data will transfer the user over to a form so they can add a new student
 - Button with data will transfer the user over to a form which allows the user to input grades
 - Close button will close the page; therefore the main menu will still be visible.
- Adding a new student will open a form, the following details will need to be filled in:
 - Candidate Name, Candidate Number, Centre Number, Target Grade, Project Title
 - All the details will need to be validated as shown in the Validation section.
 - Submit button will need to save all the data that has been input, unless there are missing / unacceptable data.
- For profiles that have already been created, the user will be able to input grades.
 - Grades will need to be validated – this has been shown in the validation section.
 - Comments are saved upon input, if this does not work, a save button will be created.
 - Student profiles contain two buttons at the bottom which will allow the user to view the next / previous page.
 - There will also be a graph button which allows the user to view the excel file but the data will be represented in a graph / bar chart form.
 - An edit button will be there to allow the user to edit the information of the student.
- ‘LIST ALL STUDENTS’ will direct user to an Excel file and data that the user has input will be transferred over to the spreadsheet with the use of a Database.
- Printing will happen in the Excel file and the student input page.

Each of the tests have been split into the modules:

Module 1

- 1a.** Click ‘GROUP 1’ button to display the list of students
- 1b.** Click ‘LIST ALL STUDENTS’ to display the spreadsheet full of students

Module 2

- 2a.** Create a new student
- 2b.** Edit the students information
- 2c.** Input valid grades into the students details
- 2d.** Input invalid grades into the student details, this should result in an error
- 2e.** Click ‘save all’ and ensure that all the totals are correct
- 2f.** Enter a comment and save
- 2g.** Highlight a student
- 2h.** Un-Highlight a student
- 2i.** Click on graph button
- 2j.** Check to see if the graph is accurate from the grades that you have inputted

Module 3

- 3a.** Check the colour scheme to see whether they have been correctly displayed
- 3b.** Check to see if the correct students are displayed
- 3c.** Input some grades into the spreadsheet
- 3d.** Check to see if the totals are correct
- 3e.** Check to see if the percentage is correct
- 3f.** Check to see if grade is correct
- 3g.** Print

This is the form the user will input information in:

Module	Test Number	Target met?	Comments
1	1a		
	1b		
2	2a		
	2b		
	2c		
	2d		
	2e		
	2f		
	2g		
	2h		
	2i		
	2j		
3	3a		
	3b		
	3c		
	3d		
	3e		
	3f		
	3g		

All of the above are each of the sections of the system which will need to be tested. Some of the stated features are not compulsory by the user but it something that can be added if there is time.

Input, Processing, Output

When testing the application, all forms of input, output and processing need to be tested. The following listed below will be all these forms.

Input

- Candidate Name: Ensuring that up to 30 characters are acceptable.
- Candidate Number: Ensuring that only 4 integers are acceptable.
- Centre Number: Ensuring that only 5 integers are acceptable.
- Target Grade: Ensuring only 1 character is acceptable, between A to E.
- Project Title: Ensuring that only 50 characters.
- Grades for each section: Limited per section, these limits are in the validation section.

Processing

- Adding up all marks for total
- Creating a percentage
- Updating marks will change the way bar charts look

Output

- Bar chart displays the correct data.
- All saved data should be correct.

- Spreadsheet should display correct data.

Normal, Extreme and Erroneous Data

There are three different types of validation, normal, extreme and erroneous. This will allow me to ensure that all the validated data and inputs are correctly programmed. Normal validation is data which is acceptable and is within the bounds that it has been limited to.

Target Grade is limited to 1 character alphabet

Test Case: Candidate name	Steps	Expected Results	Status (Failed / Pass)
1)	Enter alphabet A	Data accepted	
2)	Enter alphabet E	Data accepted	
3)	Enter alphabet Z	Error message	
4)	Enter alphabet AB	Error message	
5)	Enter one digit	Error message	

Candidate name will be restricted to 30 characters and no numbers will be accepted. The failed / pass box will be filled in once the system is complete.

Test Case: Candidate name	Steps	Expected Results	Status (Failed / Pass)
1)	John Smith	Data accepted	
2)	JJ	Error message	
3)	55smith.j	Error message	

Candidate number is restricted to 4 integers.

Test Case: Candidate name	Steps	Expected Results	Status (Failed / Pass)
1)	4321	Data accepted	
2)	41	Error message	
3)	abcd	Error message	
4)	abc	Error message	
5)	1.021	Error message	
6)	2.0	Error message	

Project Title is limited to 50 characters.

Test Case: Project Title	Steps	Expected Results	Status (Failed / Pass)
1)	F454 Tracking System	Data accepted	
2)	Abcdef421klmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzdefghijklmnopq12uvwxyz	Error message	
3)	abcdefghijklmnoprstuvwxyzabcdefghijklmnopqrstuvwxyzdefghijklmnopqrstuvwxyz	Data accepted	

Comment box is limited to 280 characters.

Test Case: Comment box	Steps	Expected Results	Status (Failed / Pass)
1)	Testing	Data accepted	
2)	Enter a comment with 280 characters	Data accepted	
3)	Enter a character with over 280 characters	Error message	

Inputting Marks will have different settings per section. In this example, we will look at Problem definition which as a maximum amount of marks of 3.

Test Case: Inputting Marks	Steps	Expected Results	Status (Failed / Pass)
1)	2	Data accepted	
2)	3	Data accepted	
3)	4	Error message	
4)	a	Error message	
5)	-4	Error message	

All marks will be added up in this section.

Test Case: Spreadsheet Total	Steps	Expected Results	Status (Failed / Pass)
1)	Enter a set of marks & check the total	Produce the correct total	
2)	Enter marks under 0	Error message	

Spreadsheet percentage is calculated by using the formula: Marks / 80 * 100

Test Case: Spreadsheet Percentage	Steps	Expected Results	Status (Failed / Pass)
1)	Enter 40	50%	
2)	Enter 80 marks	100%	
3)	Enter 0 marks	0%	
4)	Enter over 80 marks	Error message	

Will vary depending on the grades inputted, this means that each bar must be at the right height level.

Test Case: Bar chart results	Steps	Expected Results	Status (Failed / Pass)
1)	Enter 40 marks	Bar chart should be half-way.	
2)	Enter full 80 marks	All bars are at maximum level	
3)	Enter 0 marks	Bars do not appear / rise to 0 marks	

Centre Number is limited to 5 integers at all times.

Test Case: Centre Number	Steps	Expected Results	Status (Failed / Pass)
1)	Enter 5 digits	Data accepted	
2)	Enter digits other than 5	Error message	
3)	Enter 5 alphabets and numbers	Error message	
4)	Enter alphabets and numbers that's not 5	Error message	

C1: Software Development

During this phase of the Systems Life Cycle, the system will be built. Below will show the progression throughout the system development phase. The first part of Systems Development will show how I have progressed through the creation of the interface. Each step of this will be done through modules.

When I first began to code the system, I set up several global variables so that all of the sections of my system can access the data. This is shown below.

```
// public static variable type name = "";
public static int studentUniqueNumber = 1;

// Holds all student names
public static string[,] allStudentNames = new string[30, 5];

// Contains all student section totals
public static int[,] studentGradesTotals = new int[30, 12];

// Holds the total for each student
public static int[,] studentAddedTotals = new int[30, 0];

// Holds student comments
public static string[,] studentComments = new string[31, 5];

// This will store every students grades
public static int[,] allStudentGrades = new int[31, 49];
```

After this, I started to develop the first module. During the creation, I performed Alpha testing based on the requirements I posted previously which are:

Alpha Testing (White Box)

- Main menu button links to the correct groups
- All marks are calculated correctly
- Data is transferred over appropriately from database to excel
- The right results for outputs displayed for bar chart

Alpha Testing (Black Box)

- Inputting correct and incorrect information in the student form
- Inputting correct and incorrect information for student marks
- Testing out each button to make sure they link to the right module
- Next button links to the next page, cover sheet and graph will need to open an excel file.
- Comments are saved

- Candidate name and number, centre number, project title and target grade are properly saved and displayed.
- Code is indented properly and commented well

The test above is all linked to my previous test plan which can be found on page 47.

Software Development – Module 1

I first started to create the main menu of the system. The main menu will not be 1920x1080 as it will contain very minimal options for the user. Using the design I had created in the Design section, I replicated it in Visual Studio. In the main menu, there are several buttons. The close button closes the system. The ‘LIST ALL STUDENTS’ button opens up a spreadsheet with all the details of each student and their grades. The ‘GROUP’ buttons open classrooms which will have several buttons for each student. When the program is first opened, it will read through all the data that has been inputted. This has been done by using the StreamReader tool provided by Visual Studio. This is the code:

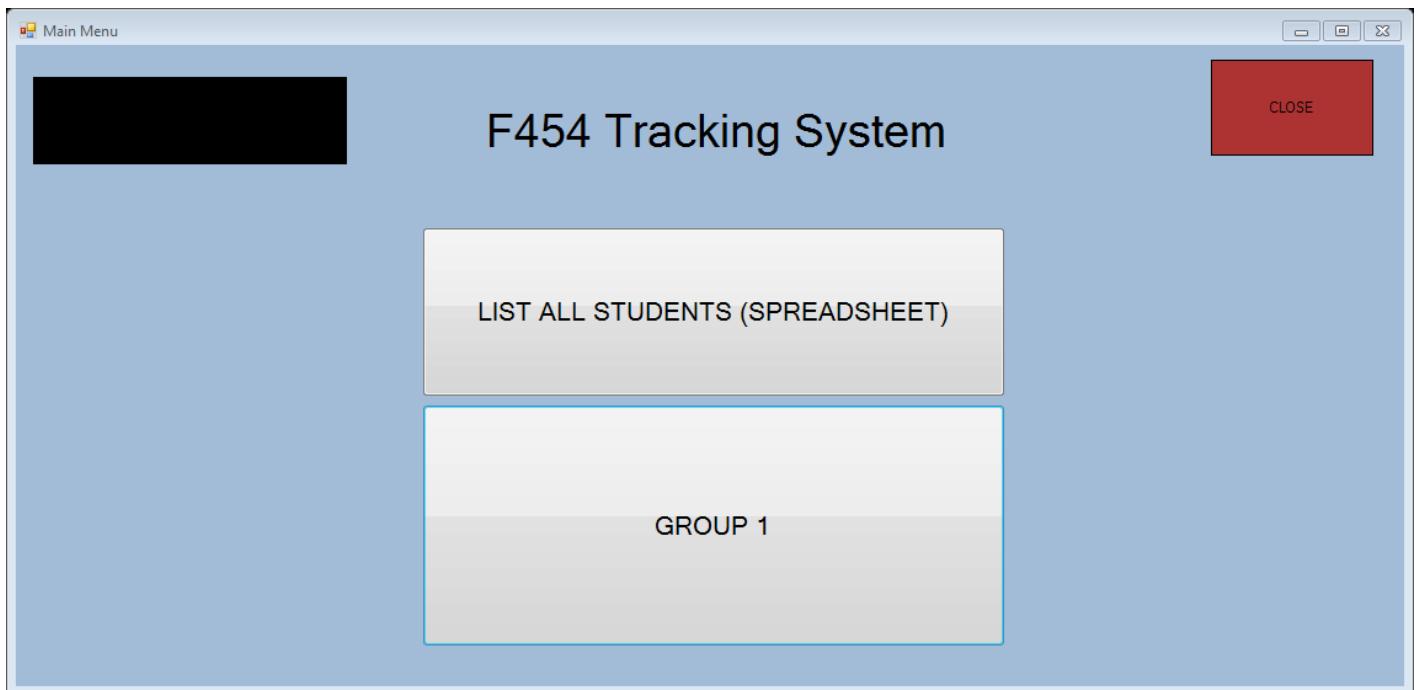
```
StreamReader readMe = new StreamReader(@"S:\studentdata.txt", false);
StreamReader gradesRead = new StreamReader(@"S:\studentgrades.txt", false);

for (int a = 0; a < 30; a++)
{
    for (int b = 0; b < 5; b++)
    {
        Variables.allStudentNames[a, b] = readMe.ReadLine();
    }
}
readMe.Close();

for (int x = 0; x < 30; x++)
{
    for (int y = 1; y < 47; y++)
    {
        Variables.allStudentGrades[x, y] = Convert.ToInt32(gradesRead.ReadLine());
    }
}
gradesRead.Close();
```

A for loop runs through the variables and sets them as saved in the text file.

This is the appearance of the main menu. The way this was created with the use of an image box to display The Studio School’s logo. A label for the title ‘F454 Tracking System’ and three buttons. One which is laid out in a flat style.



Testing Button Click

After programming the main menu, I decided to test each of the buttons to make sure they all link up properly. Here are the screenshots:

List All Students test

When clicking on the button, it loads the code that had been entered which opens the excel file.

This code not only opens Microsoft Excel but also declares 'ws' as a key term which can write to the excel file.

```
// This opens the Excel file
Microsoft.Office.Interop.Excel.Application xla = new Microsoft.Office.Interop.Excel.Application();
Workbook wb = xla.Workbooks.Add(XlSheetType.xlWorksheet);
Worksheet ws = (Worksheet)xla.ActiveSheet;
xla.Visible = true;
```

To load the actual data that has been inputted, this was the code that was used:

```
// each section of the coursework will be listed below.
ws.Cells[1, 1] = "Candidate Name";
ws.Cells[1, 2] = "A1";
ws.Cells[1, 3] = "A2";
ws.Cells[1, 4] = "B1";
ws.Cells[1, 5] = "B2";
ws.Cells[1, 6] = "B3";
ws.Cells[1, 7] = "C1";
ws.Cells[1, 8] = "C2";
ws.Cells[1, 9] = "D1";
ws.Cells[1, 10] = "E1";
ws.Cells[1, 11] = "E2";
ws.Cells[1, 12] = "E3";
ws.Cells[1, 13] = "TOTAL";
ws.Cells[1, 14] = "%";
ws.Cells[1, 15] = "GRADE";
ws.Cells[1, 16] = "TARGET";
```

The brackets represent the row and column.

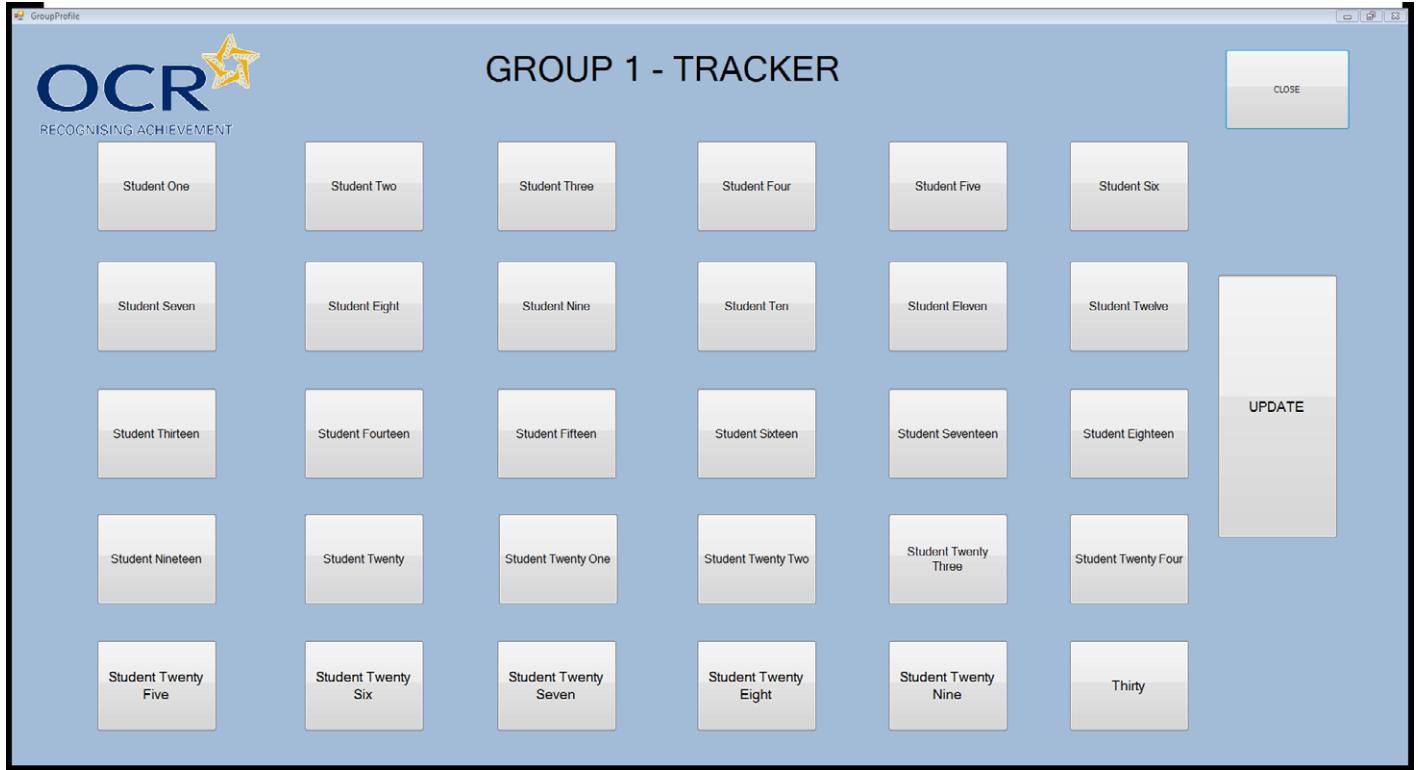
Upon loading the system, it was displayed correct. There were no data shown as there were no inputted information.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Candidate Name	A1	A2	B3	B2	B3	C1	C2	D1	E1	E2	E3	TOTAL	%	GRADE	TARGET
2																
3																
4																
5																
6																
7																

The code shown above will be repeated for each row of students and data will be taken from the array as it is loaded automatically upon opening the application.

Group 1 Button Test

Upon clicking the group button, the Group 1 Tracker loaded with the list of all the students. They have been named as 'Student One, Two, Three etc.' because no data have been entered. The way I created this page was placing a set of buttons representing different students around the page in a 6x5 format. Upon clicking a button, a set of code will load. This code will be shown below. All the features of this page will be listed in the 2nd module section.

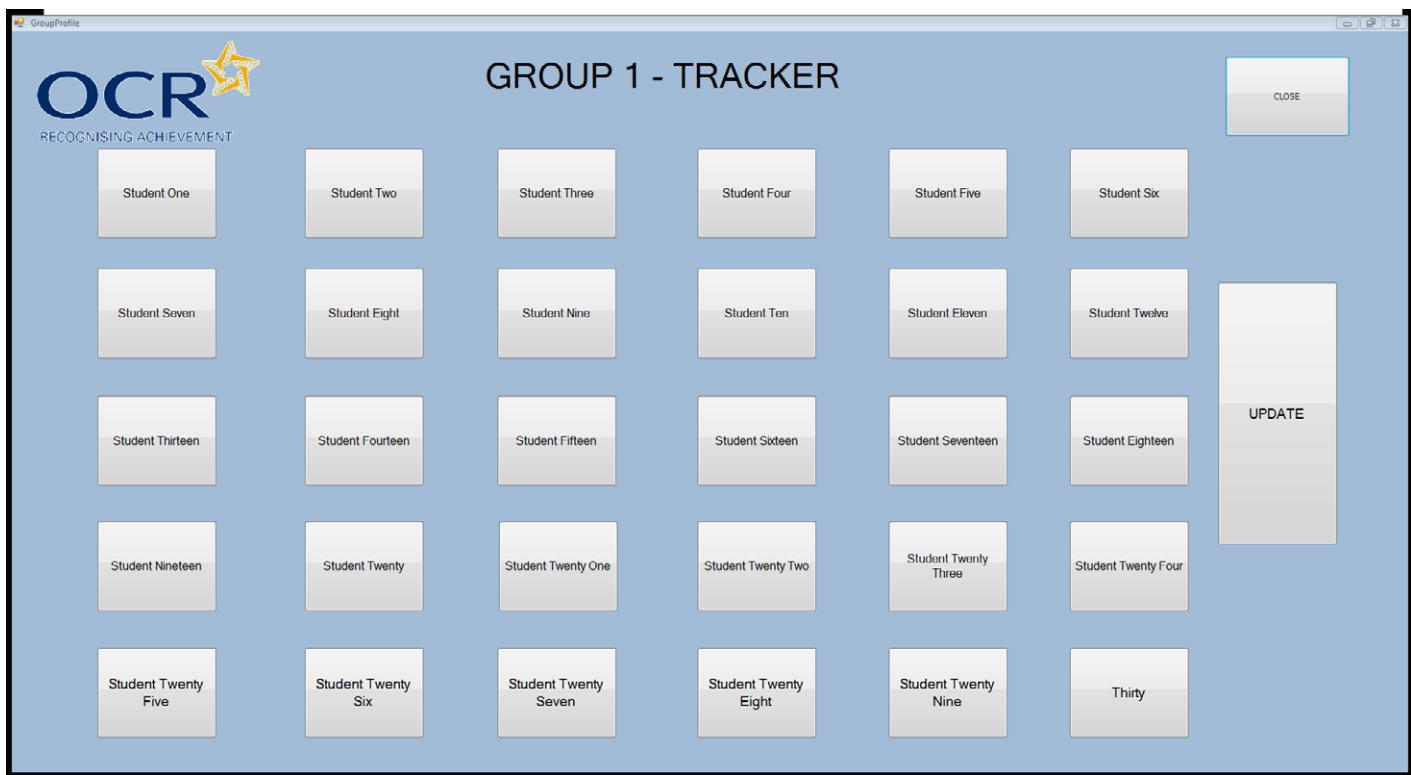


Software Development – Module 2

Module 2 is the group page which displays all the students in the class. The size of this page is 1920x1080 as requested by the client. The user can individually click on a student to input all their grades.

As shown above, this is the page that the user is first greeted with. Each button represents a student and upon clicking one, it will either load a screen which contains a set of textboxes to be filled in, or a page which contains the mark scheme to input the student's grade. The 'UPDATE' button is used to update the buttons. There are two actions this will do, the first is changing the name of the student and the second the highlighting certain students.

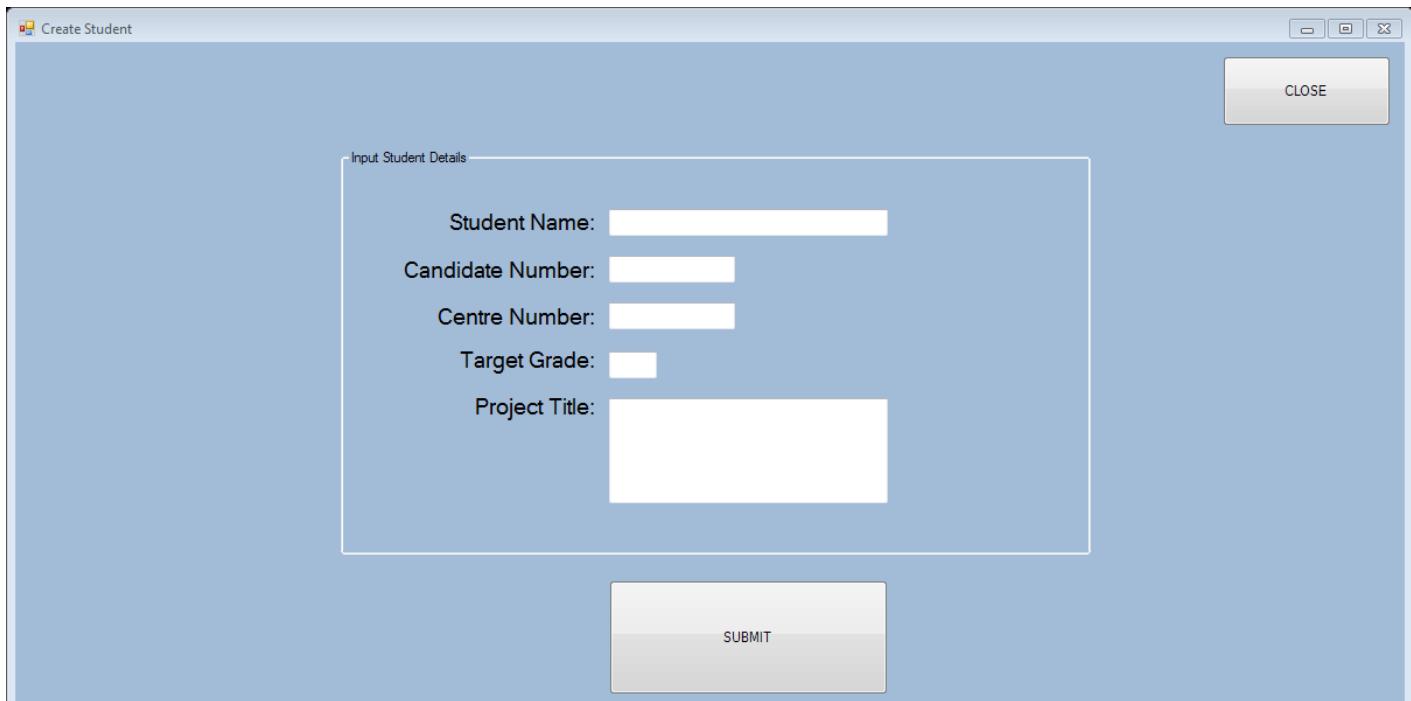
The way this page was created was with the use of several buttons. Each button automatically has its own label. The OCR image was with the use of an image box provided by Visual Studio. The 'GROUP 1 – TRACKER' is a label with increased font size.



Student One Test

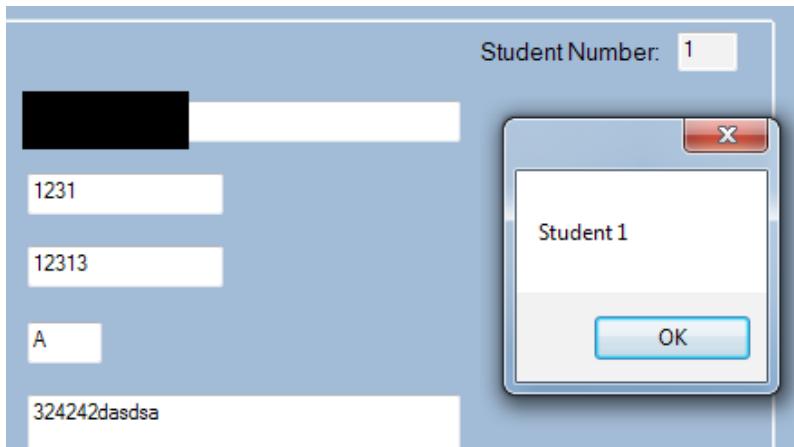
Upon clicking Student One, the student details page opened which displayed all the input needed for the student. Each textbox have been validated so that no errors will appear while the system is running. All the tests will be shown in C2.

This is the Create Student page. This was created with the use of several textboxes and labels. There is a 'Groupbox' feature added to the design which was simply put for a better interface look.

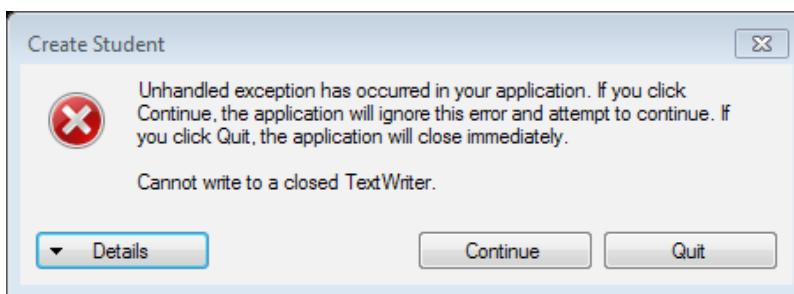


During the process of creating the page to add a new student, I realised that there would be a problem due to having so many students and having to separate each student. I created a new variable named 'studentUniqueNumber'. This means that every time the user clicks on a student, the student will have a unique number which will be saved as

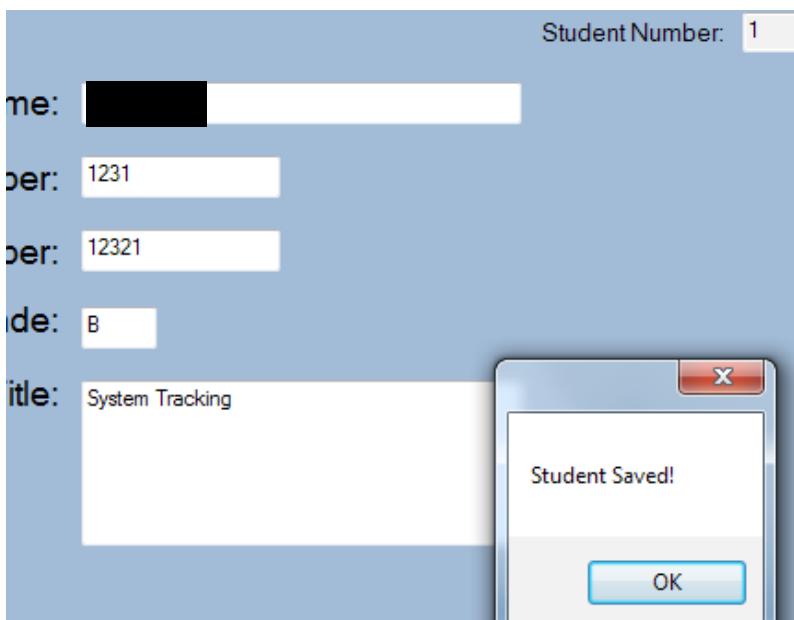
student unique number. This can then be used to differentiate which student is which and also allows to load different data. Originally, I was planning on differentiating students with the use of candidate numbers, however since that can vary a lot, I chose to do a unique student number as it would be a lot easier, but still work the way it is supposed to.



After the data was submitted, the input is directly supposed to be held on a 2D array. This would then be easier to write to a text file. However, when submitting the data, an error appeared which is shown below:



The problem was as stated on the error message, I had closed the `WriteFile()` statement which meant that no input could be saved in the text file. I deleted the statement and pasted it at the end of the for-loop. This saved properly as shown below.



The code:

```
for (int y = 0; y < 30; y++)
{
    for (int z = 0; z < 5; z++)
    {
        studentInformation.WriteLine(Variables.allStudentNames[y, z]);
    }
}
studentInformation.Close();
```

Saving & Beta Testing

During this part of the production of the system, I carried out my Beta testing. Below are the Beta testing's which I had planned originally. Under each profile will have the test shown and the results.

Profile 1

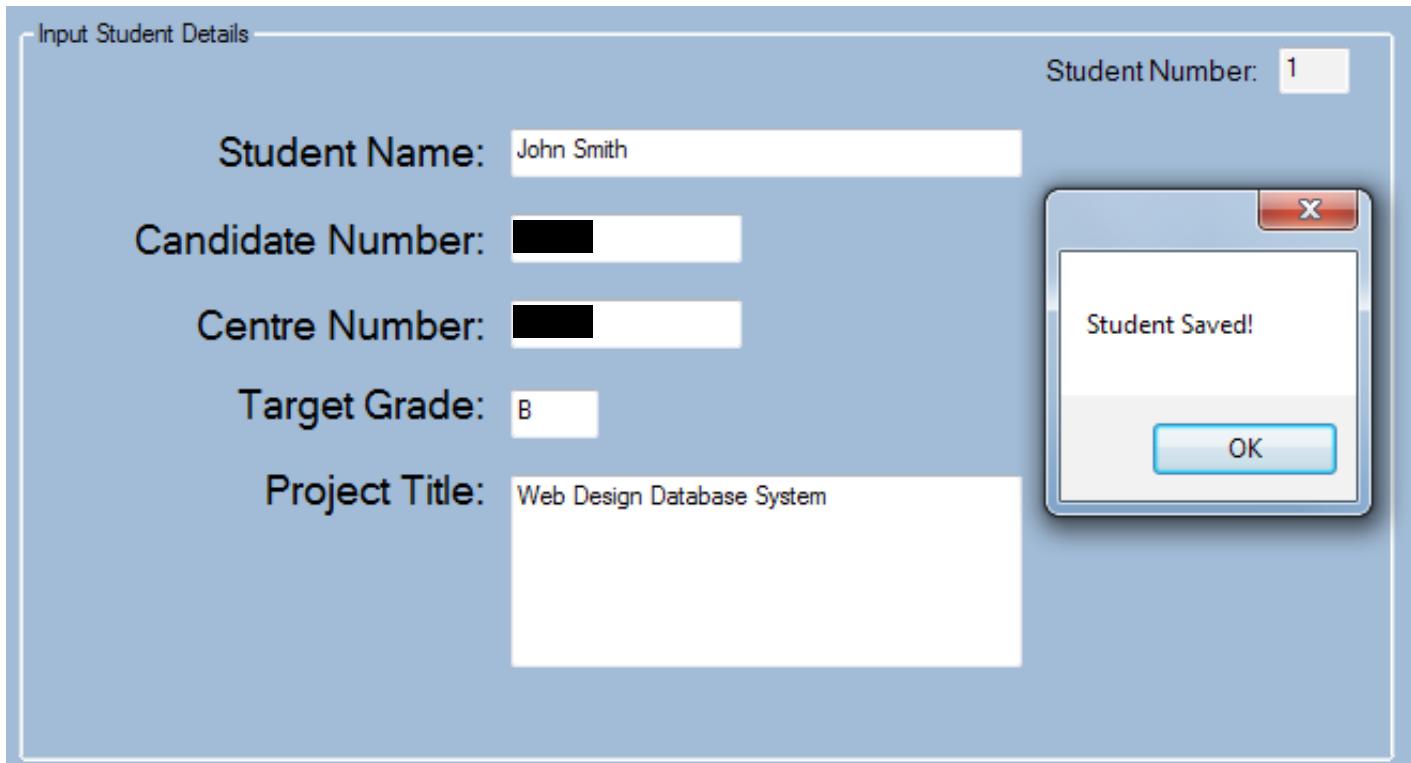
Candidate Name: John Smith

Candidate Number: [REDACTED]

Centre Number: [REDACTED]

Target Grade: B

Project Title: Web Design Database System



This profile should be accepted as all the data entered is correct.

PROFILE 1	ANY ERRORS?	COMMENTS
CANDIDATE NAME	NONE	
CANDIDATE NUMBER	NONE	
CENTRE NUMBER	NONE	
TARGET GRADE	NONE	
PROJECT TITLE	NONE	

Profile 2

Candidate Name: John Smith

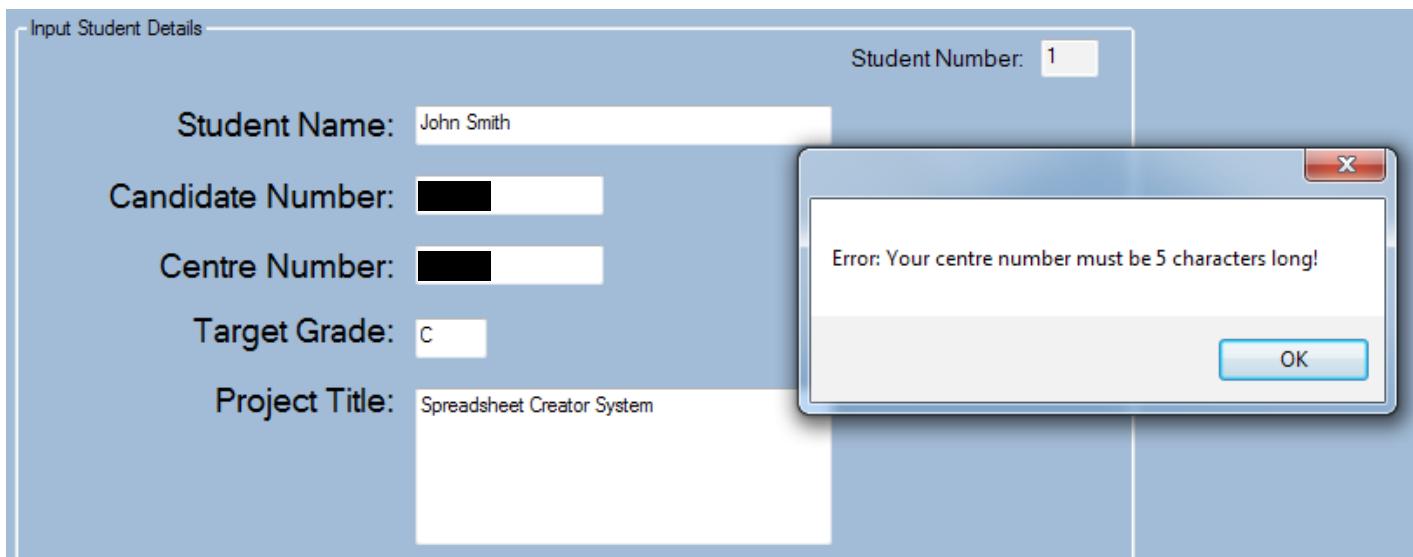
Candidate Number: 99999

Centre Number: 9999

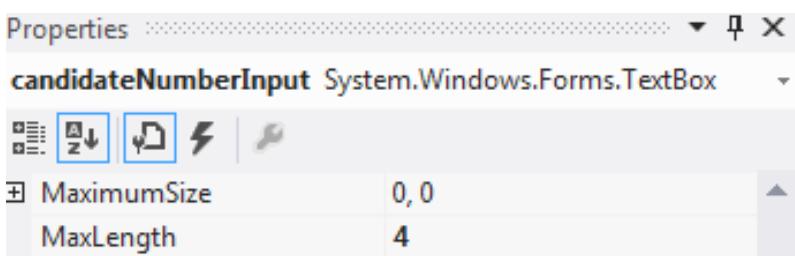
Target Grade: C

Project Title: Spreadsheet creator System

This profile should not be accepted as both Candidate Number and Centre Number are in the wrong range.



The reason I did not enter my candidate number as '20191' was because I limited the textbox max size to 4 to stop the user from typing over the character limit. This is shown below:



PROFILE 2	ANY ERRORS?	COMMENTS
CANDIDATE NAME	NONE	
CANDIDATE NUMBER	NONE	
CENTRE NUMBER	YES	PRODUCED AN ERROR MESSAGE
TARGET GRADE	NONE	
PROJECT TITLE	NONE	

Profile 3

Candidate Name: 55smith.j

Candidate Number: 1010

Centre Number: 36300

Target Grade: F

Project Title: 2D Platform Game

This profile should not be accepted as both Candidate Name and Target Grade are in the wrong range.

The screenshot shows a user interface for inputting student details. The main window is titled "Input Student Details". It has fields for "Student Name" (containing a redacted value), "Candidate Number" (1010), "Centre Number" (36300), "Target Grade" (F), and "Project Title" (2D Platform Game). A modal dialog box is displayed, stating "Your Candidate Name must be in alphabets only!" with an "OK" button. The "Candidate Number" field is highlighted in yellow, indicating it is the source of the error.

Target Grade did not produce an error message because the code stops as soon as it hits its first error.

PROFILE 3	ANY ERRORS?	COMMENTS
CANDIDATE NAME	YES	PRODUCED AN ERROR MESSAGE
CANDIDATE NUMBER	NO	
CENTRE NUMBER	NO	
TARGET GRADE	NO	
PROJECT TITLE	NO	

Profile 4

This will be completely blank.

Candidate Name:

Candidate Number:

Centre Number:

Target Grade:

Project Title:

Student Name:	<input type="text"/>
Candidate Number:	<input type="text"/>
Centre Number:	<input type="text"/>
Target Grade:	<input type="text"/>
Project Title:	<input type="text"/>

Please ensure that all fields have been inputted!

OK

PROFILE 4	ANY ERRORS?	COMMENTS
CANDIDATE NAME	YES	PRODUCED AN ERROR MESSAGE
CANDIDATE NUMBER	YES	PRODUCED AN ERROR MESSAGE
CENTRE NUMBER	YES	PRODUCED AN ERROR MESSAGE
TARGET GRADE	YES	PRODUCED AN ERROR MESSAGE
PROJECT TITLE	YES	PRODUCED AN ERROR MESSAGE

Upon checking the section, only the student unique number appeared. Therefore I had to go back into the code to see what was going wrong.

Student Number:	<input type="text" value="1"/>
Candidate Name:	<input type="text"/>
Candidate Number:	<input type="text"/>
Centre Number:	<input type="text"/>
Target Grade:	<input type="text"/>
Project Title:	<input type="text"/>

I went through the code and realised that I had put the declaration of code the wrong way round. I had essentially told the code to write from the text file to the array, when instead it should have been writing to the text file from the array. I switched the code appropriately and ran the program once again.

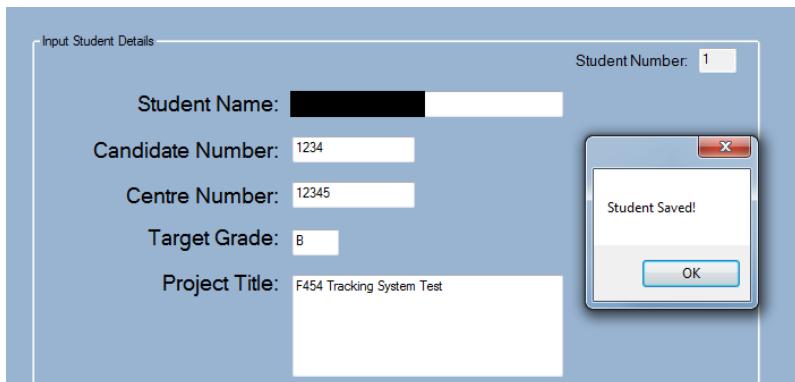
This was the code I used to display the student details on each section, it was written in a case statement:

```

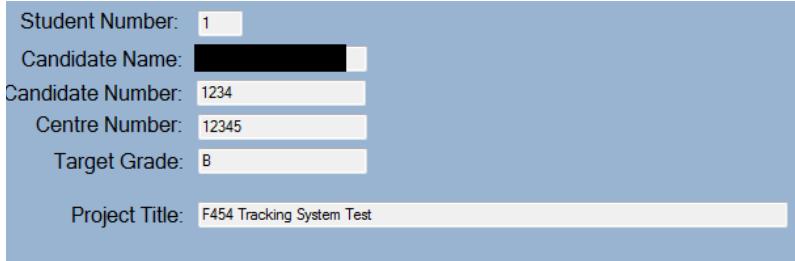
// Each case represents a student - setting the textbox as the data inputted by the user.
case 1:
{
    courseSectionCandidateName.Text = Variables.allStudentNames[0, 0];
    courseSectionCandidateNumber.Text = Variables.allStudentNames[0, 1];
    courseSectionCentreNumber.Text = Variables.allStudentNames[0, 2];
    courseSectionTargetGrade.Text = Variables.allStudentNames[0, 3];
    courseSectionProjectTitle.Text = Variables.allStudentNames[0, 4];
    // Calling the array to lower the amount of code.
    arrayToTextBox();
    break;
}

```

Below is the data I entered which should now display the information on the read-only textboxes.



After the code was fixed, the correct information was displayed. This code was then copied onto the other student boxes.



Testing button name change

I created a student and set the name as '[redacted]'. I updated the system and it showed the name as shown below. There were no errors for this.



This was the code I used to display the name (wrote in a method):

```
private void buttonNames()
{
    if (Variables.allStudentNames[0, 0].Length != 0)
    {
        GroupOneStudentOne.Text = Variables.allStudentNames[0, 0];
    }
}
```

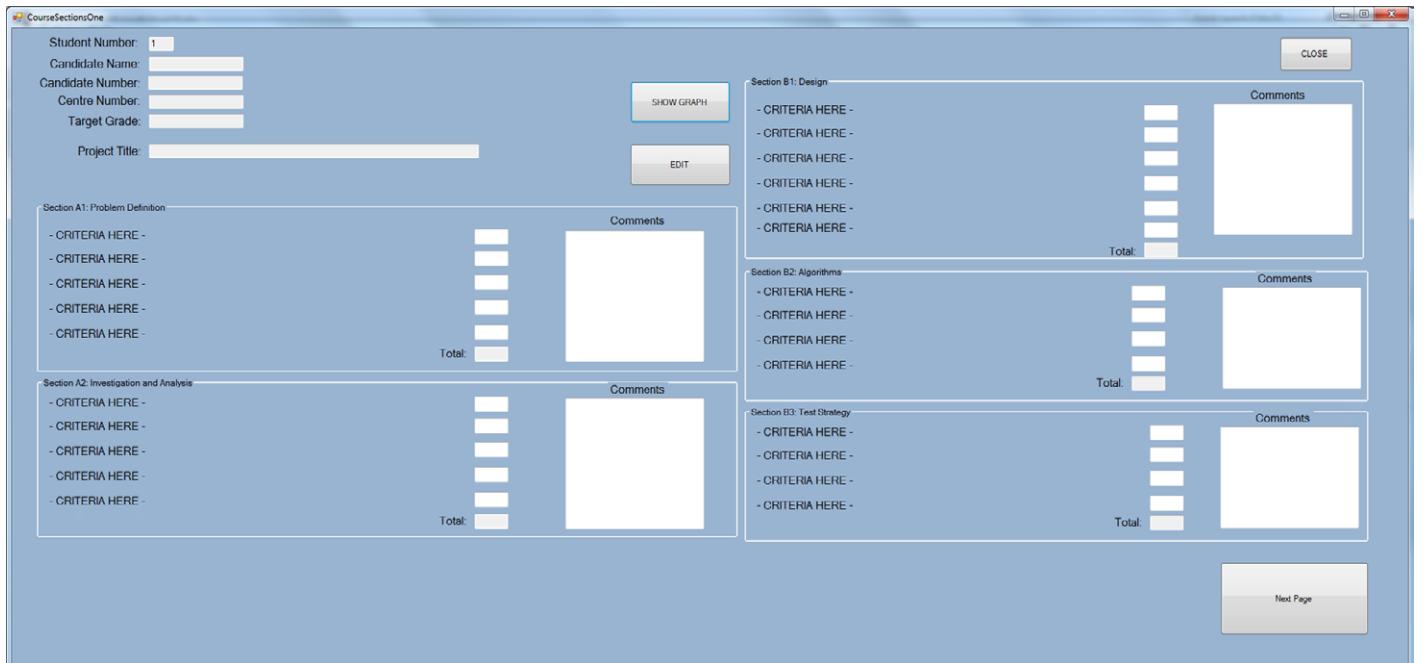
The code checks to see if the length isn't 0. If it isn't, it changes the text of the button to what is inside the array.

Button click test

When clicking on a button that already contains inputted data, it should automatically take the user to section one. Here is my first test:

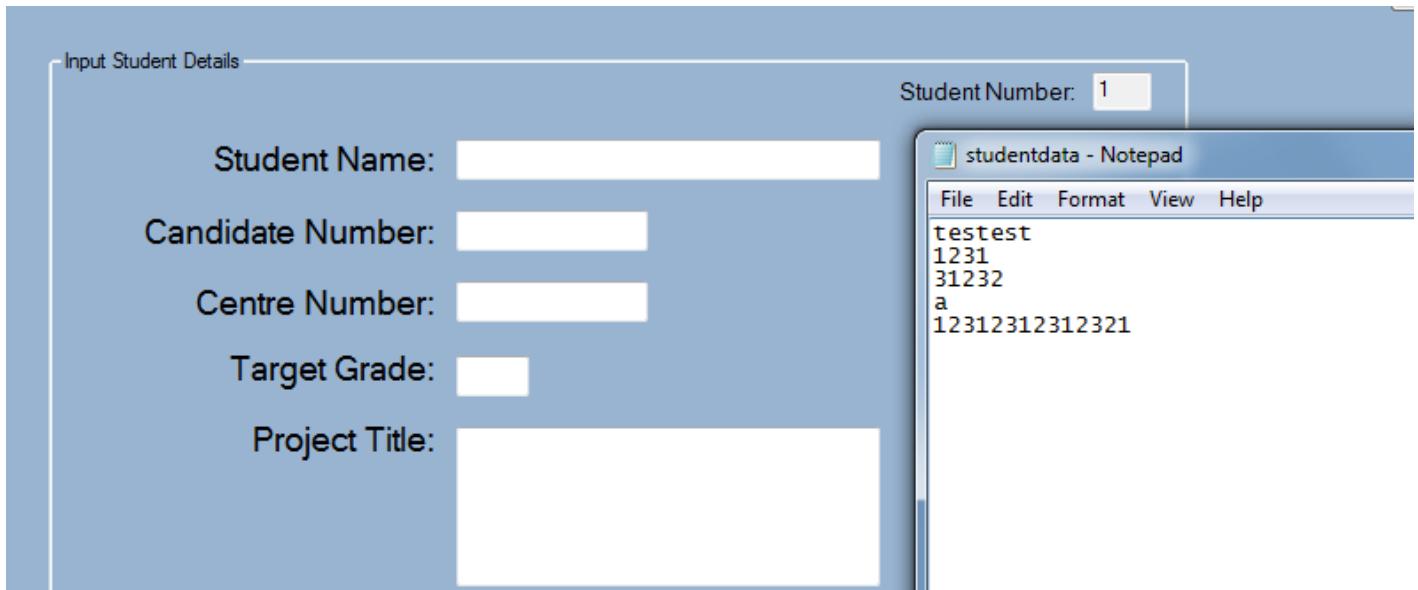
I clicked on the button and even though it had no data, it still took me to section one. You can see that there were no input on the top left of the page as no student details are shown.

The page was created with the use of read-only textboxes, normal textboxes, buttons and labels. Read-only textboxes were used as some of the data did not need to be inputted.

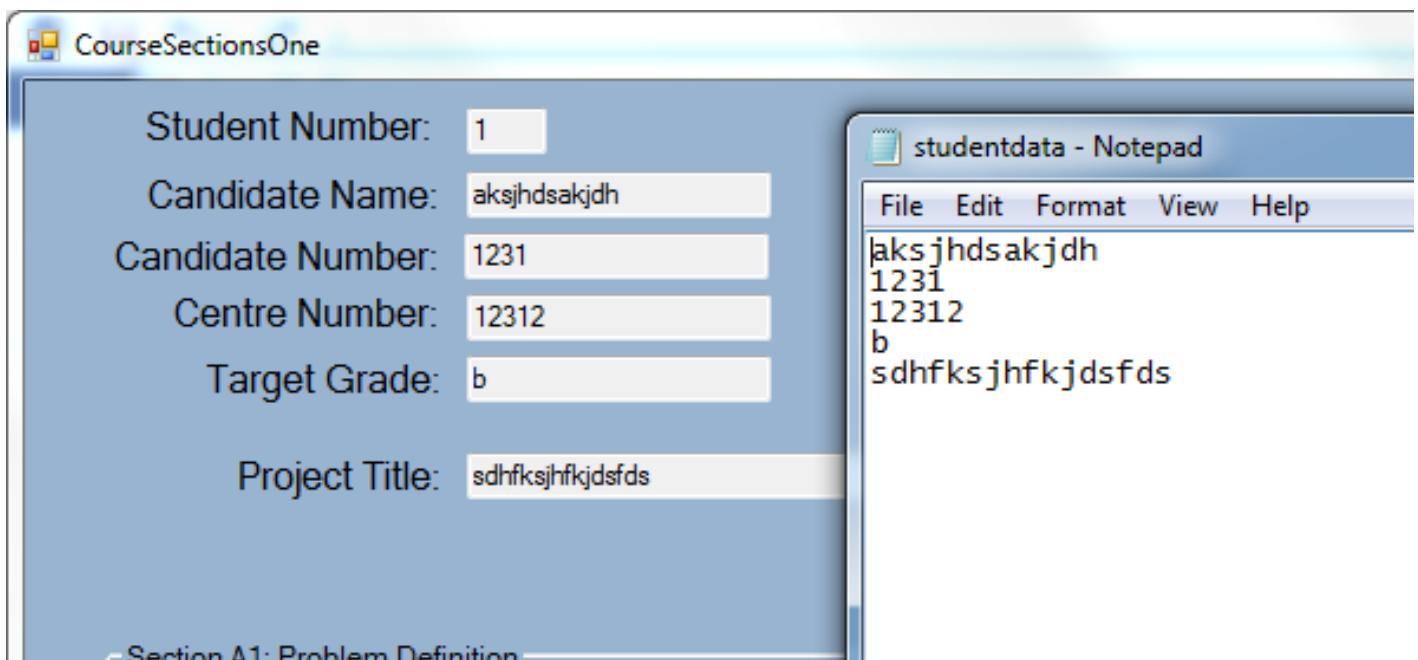


I went back over my code and edited the if statement so if the array didn't have any data written inside of it, then it would make the 'create student' appear first.

However, after inputting the details and it saving, the buttons still did not link correctly as it kept telling me to input the details.



I then decided to create an IF statement which would check to see if the array equalled to null. This worked and the code saved correctly and opened appropriately as shown below.

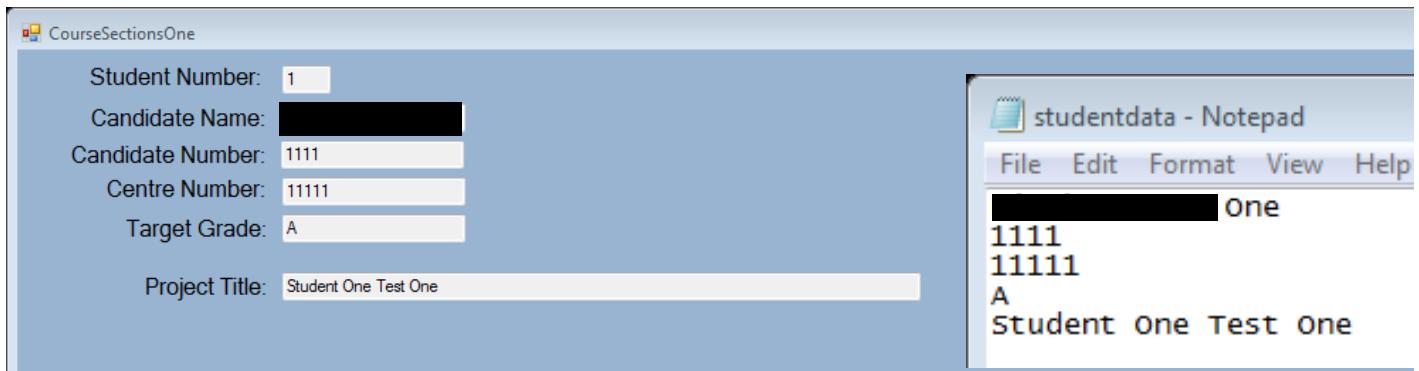


This was the code I used:

```
private void groupOneOne_Click(object sender, EventArgs e)
{
    Variables.studentUniqueNumber = 1;
    CourseSectionsOne sectionOne = new CourseSectionsOne();
    CreateStudent createNewStudent = new CreateStudent();

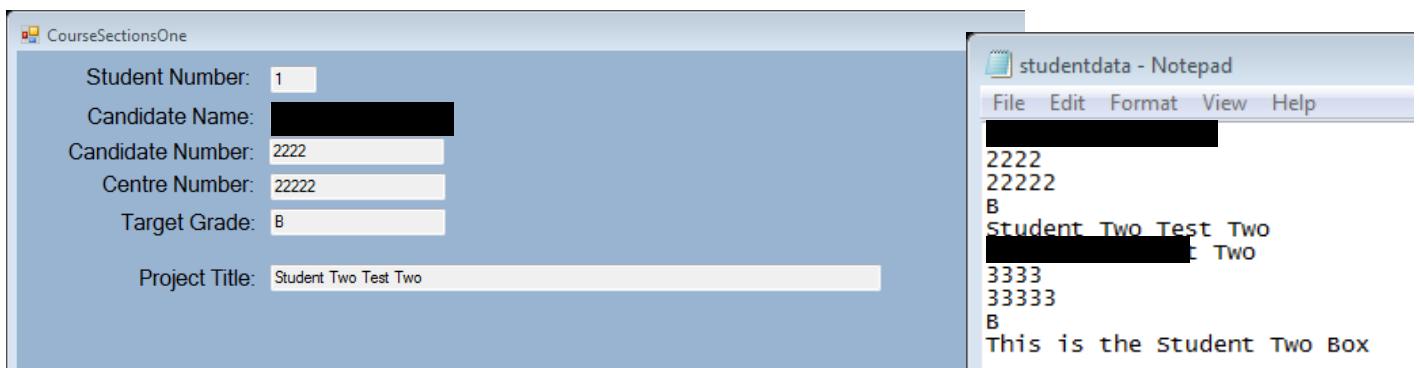
    if (Variables.allStudentNames[0, 0] == null)
    {
        createNewStudent.Show();
    }
    else
    {
        sectionOne.Show();
    }
}
```

Next, I decided to check to see if the 'edit' button worked for each of the students. This would mean that the current data already inputted for the student would need to be overwritten.



Above shows how the data saved correctly.

Below is a screenshot of the data I inputted after clicking edit. After this, I also created another student in the second button to make sure it adds both of the data.



After this point, I programmed the main section. Firstly, I will be testing the validation for each box.

Here, one of the input is over 3, when it must be 3 or below. I have limited the textbox to one integer only. This will also ensure that no minus numbers can be inputted.



This was the code I used to validate the data:

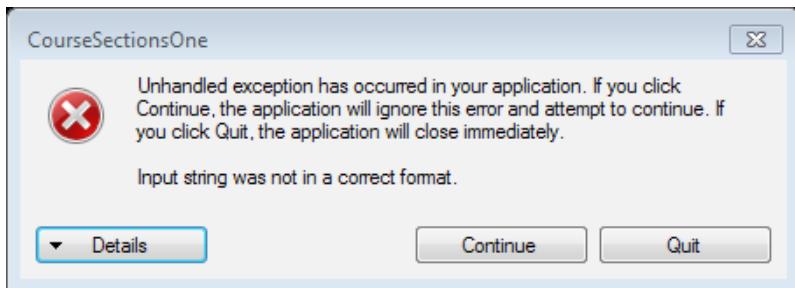
```

if (sectionOneAOnePOne.TextLength > 0 || sectionOneAOnePTwo.TextLength > 0 ||
    sectionOneAOnePThree.TextLength > 0 || sectionOneAOnePFour.TextLength > 0 ||
    sectionOneAOnePFive.TextLength > 0)
{
    if (Variables.allStudentGrades[Variables.studentUniqueNumber, 1] > 3 ||
        Variables.allStudentGrades[Variables.studentUniqueNumber, 2] > 3 ||
        Variables.allStudentGrades[Variables.studentUniqueNumber, 3] > 3 ||
        Variables.allStudentGrades[Variables.studentUniqueNumber, 4] > 3 ||
        Variables.allStudentGrades[Variables.studentUniqueNumber, 5] > 3)
    {
        MessageBox.Show("Error, section A One must be between 0 and 3");
    }

    else if (Variables.allStudentGrades[Variables.studentUniqueNumber, 1] < 0 ||
              Variables.allStudentGrades[Variables.studentUniqueNumber, 2] < 0 ||
              Variables.allStudentGrades[Variables.studentUniqueNumber, 3] < 0 ||
              Variables.allStudentGrades[Variables.studentUniqueNumber, 4] < 0 ||
              Variables.allStudentGrades[Variables.studentUniqueNumber, 5] < 0)
    {
        MessageBox.Show("Error, section A One must be between 0 and 3");
    }
}

```

I then inputted some valid data, but this message appeared when I wanted the total average.



Following this, I changed the arrays as I then realised that I was using the wrong naming conventions. After altering the code appropriately, there was no longer an error message. However, another problem had risen.

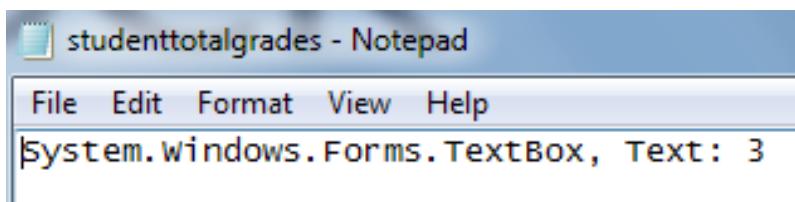
Although the program did not crash, the average was still not being calculated properly. The code was rounding itself down instead of rounding to the nearest integer as shown below:

Section A1: Problem Definition	
Brief Description of end user (e.g Firm of Business involved, location, turnover etc)	3
Current description of problem	3
Current description of methods used or area in need of development	3
Clear statement of origins and form of any relevant data (e.g paper based registration)	3
Describe how further research is needed to define a problem and its intent	1
Total:	1

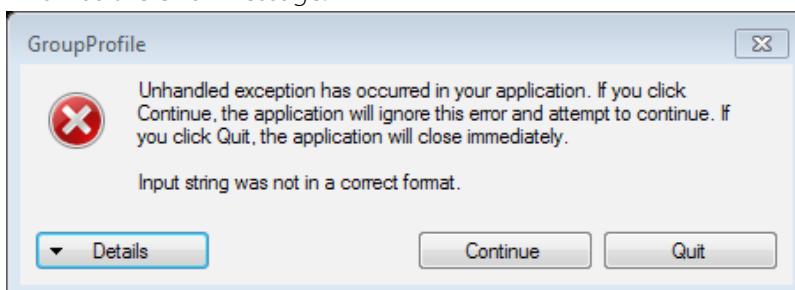
I couldn't get my head around the code, so I decided to create a temporary float. This would then allow me to round up the number appropriately as shown below:

Section A1: Problem Definition	
Brief Description of end user (e.g Firm of Business involved, location, turnover etc)	3
Current description of problem	3
Current description of methods used or area in need of development	3
Clear statement of origins and form of any relevant data (e.g paper based registration)	3
Describe how further research is needed to define a problem and its intent	1
Total:	3

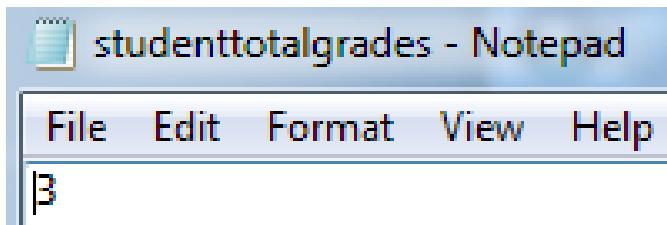
After checking the notepad, the file did save as shown below. However, this resulted in yet another error message.



This was the error message:



I removed the text from the notepad and the error message no longer appeared. Therefore, I will need to find a way to stop the writeline command in entering the 'System.Windows.Forms.Textbox' as it is a string value and not an integer. I went back through the code and realised that I had only put the name of the textbox and did not enter .text after it. This then fixed the problem and it saved appropriately as shown below:



This was the code I used:

```
sectionOneAOnePOne.Text = Variables.allStudentGrades[Variables.
studentUniqueNumber, 1].ToString();
```

After this, I moved onto section A2. The first problem that occurred was that the average was not being totalled up as shown below:

Section A2: Investigation and Analysis		Comments
- CRITERIA HERE -	3	
- CRITERIA HERE -	2	
- CRITERIA HERE -	2	
- CRITERIA HERE -	10	
- CRITERIA HERE -	2	
Total:		

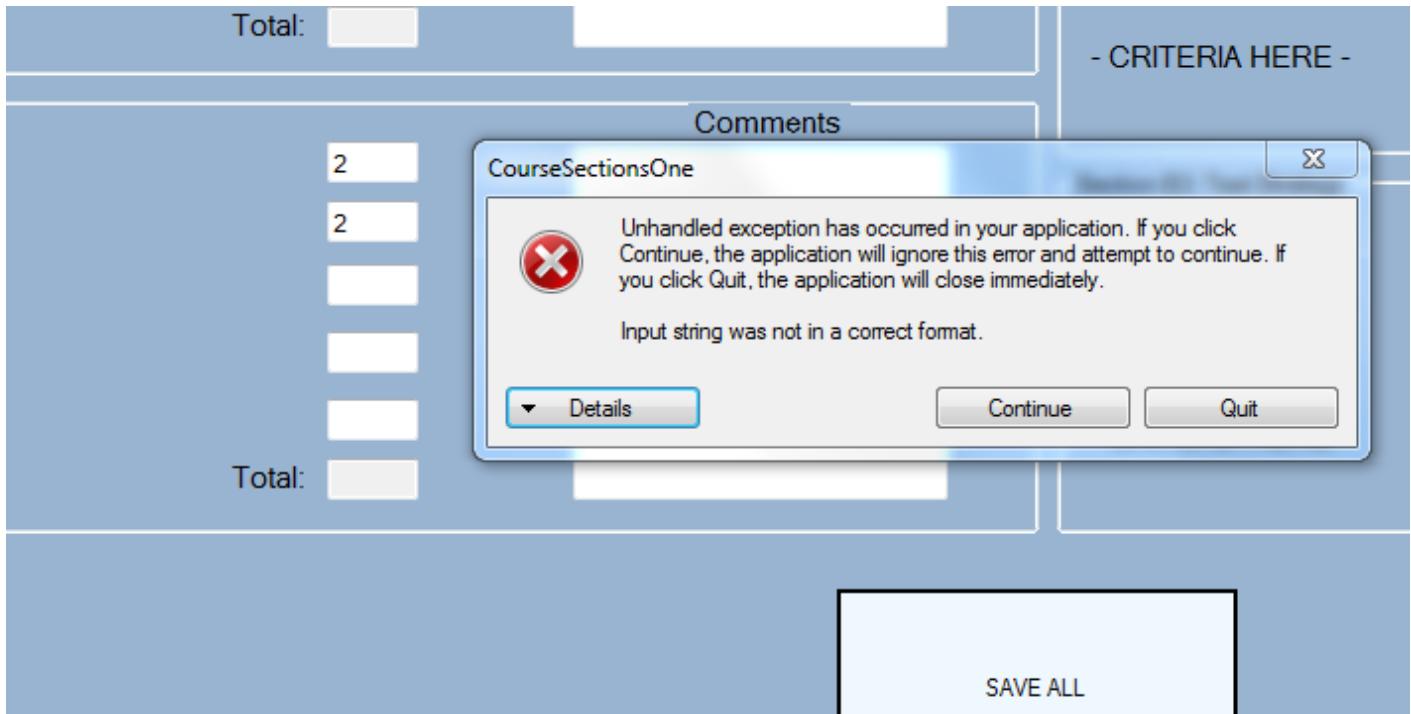
I went through the code and needed to convert the variables to an integer. I done this by using the following statement

```
Variables.studentGradesTotals[Variables.studentUniqueNumber, 2] =
(int)tempRoundATwo;
sectionOneATwoTotal.Text =
Variables.studentGradesTotals[Variables.studentUniqueNumber, 2].ToString();
```

As you can see, by converting the variable, it made the correct data appear (shown on next page).

Section A2: Investigation and Analysis		Comments
- CRITERIA HERE -	5	
- CRITERIA HERE -	4	
- CRITERIA HERE -	3	
- CRITERIA HERE -	2	
- CRITERIA HERE -	1	
Total:	3	

I then tested to see whether or not it would work if I didn't input anything. This created an error:



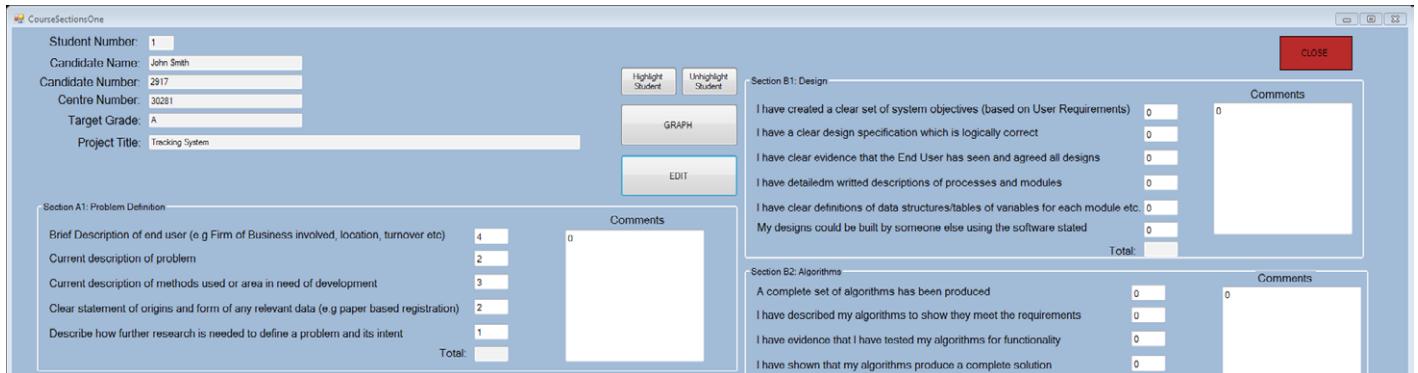
From this error, I decided that each textbox will automatically have 0 inputted to stop this error as shown below. This was created automatically when the system read through all the data as it found missing inputs and so it filled it with 0s.

This is the code I used:

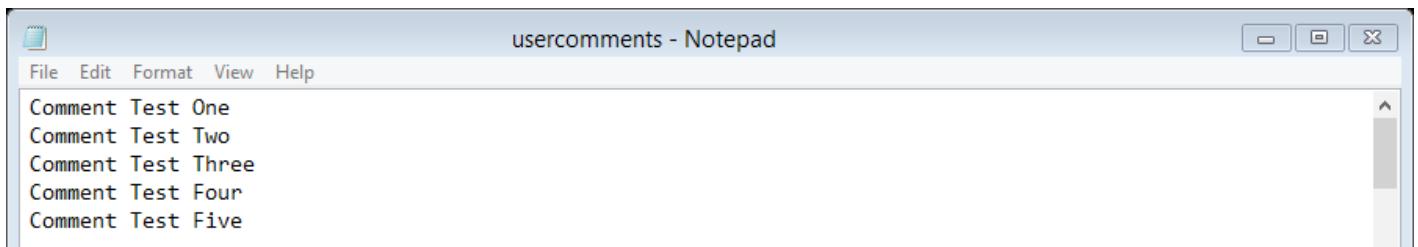
```
// Contains all individual grades - Reads the grades - Arrays are set so it automatically
// sets the textboxes to what the arrays equal to.
for (int x = 1; x < 31; x++)
{
    for (int y = 1; y < 49; y++)
    {
        Variables.allStudentGrades[x, y] = Convert.ToInt32(gradesRead.ReadLine());
    }
}
```

Comments

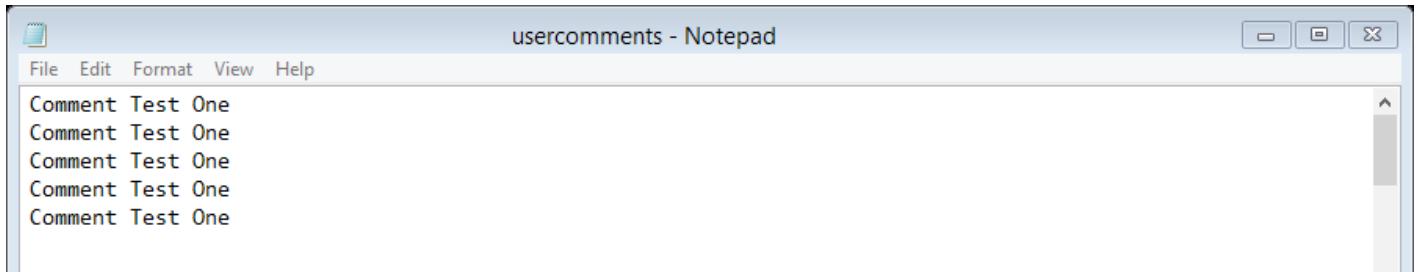
In section one, I inputted comments on each box as 'Comment Test One', 'Comment Test Two', 'Comment Test Three' etc. This meant that it should have saved as that in the file.



When looking at the text file, it did not save as inputted. Instead, it just went through a loop of the first comment.



I edited the code. The problem was that I had parsed the wrong array with the textbox and so it was only writing the first comment box. After testing it again, it worked.



This is the code I used to gather the comments:

```
Variables.studentComments[Variables.studentUniqueNumber, 0] = aOneComment.Text;
Variables.studentComments[Variables.studentUniqueNumber, 1] = aTwoComment.Text;
Variables.studentComments[Variables.studentUniqueNumber, 2] = bOneComment.Text;
Variables.studentComments[Variables.studentUniqueNumber, 3] = bTwoComment.Text;
Variables.studentComments[Variables.studentUniqueNumber, 4] = bThreeComment.Text;
```

After this happens, a for loop will save all the comments.

```
for (int x = 1; x < 30; x++)
{
    for (int y = 0; y < 11; y++)
    {
        writeComment.WriteLine(Variables.studentComments[x, y]);
    }
}
writeComment.Close();
```

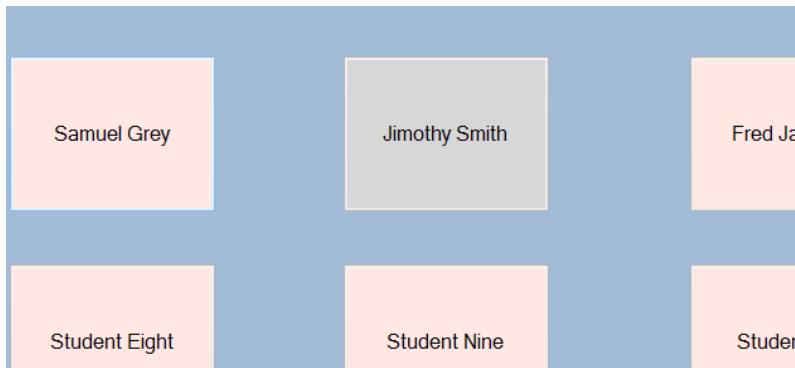
Highlighting students

As part of my client's requirements, he wanted the ability to highlight certain students who they may have worries about. I created two buttons for this, one to highlight and one to un-highlight to make it simple for both the programmer and for the user.

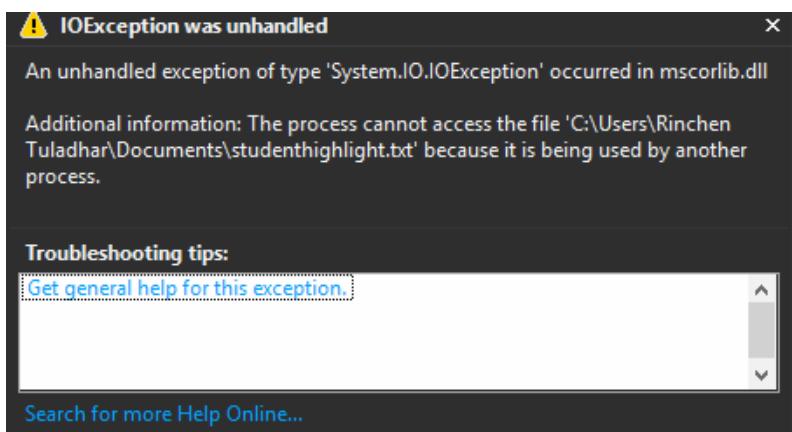
Student Number: 3
Candidate Name: Timothy Smith
Candidate Number: 4214
Centre Number: 12312
Target Grade: C
Project Title: Power on system

Highlight Student Unhighlight Student
SHOW GRAPH
EDIT

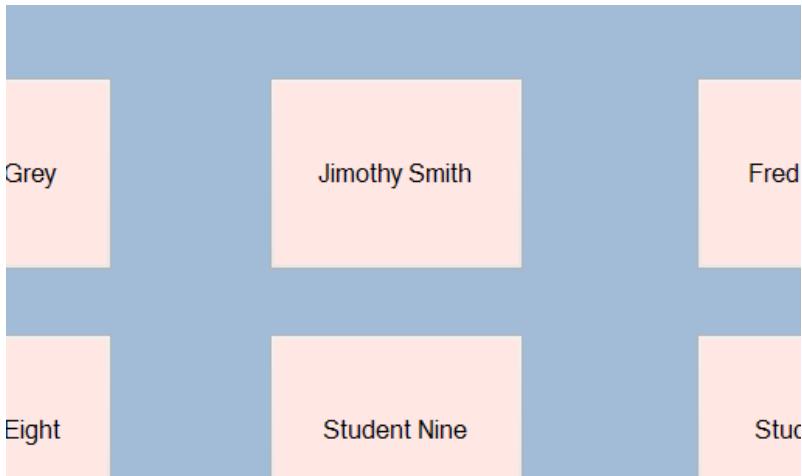
It worked, as shown below. A small tint of grey displayed which was done on purpose so it doesn't look out of place. I then attempted to un-highlight the student but it didn't work and stayed on the same tinted colour.



Some of the textboxes were not named properly. This was the issue, but again another error appeared:



This was a simple case of forgetting to close a file when writing to it. I found it and closed the writeline statement. After that, the test was performed one more time and the student highlight was removed as shown below:



This is the code I used to (un)highlight the students:

Firstly, clicking the highlight button will set the array to 1.

```
case 1:
{
    Variables.highlightButton[Variables.studentUniqueNumber, 0] = 1;
    break;
}
```

Clicking the un-highlight button will set the array to 0.

```
case 1:
{
    Variables.highlightButton[Variables.studentUniqueNumber, 0] = 0;
    break;
}
```

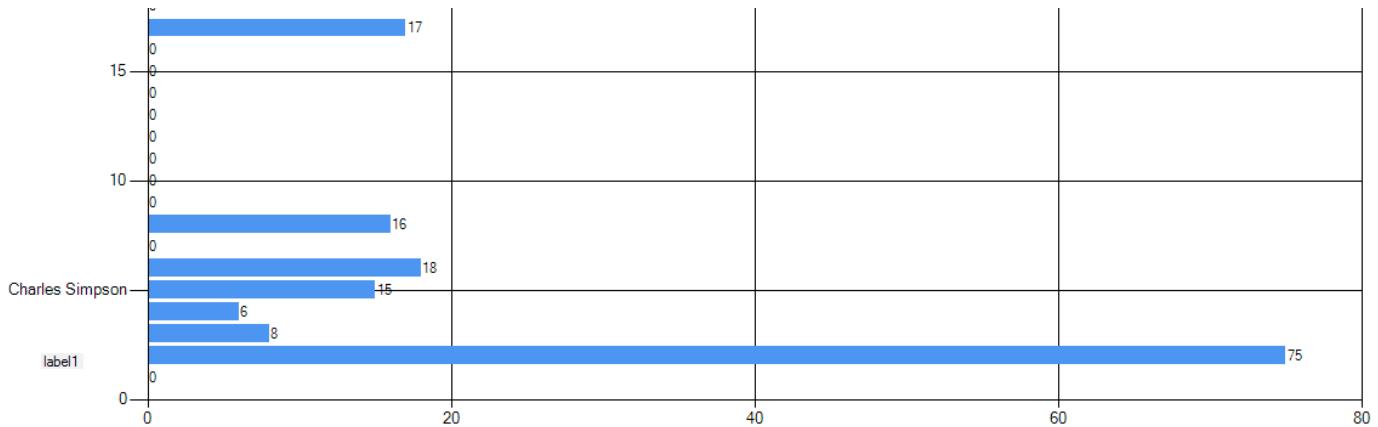
This is then checked using an IF statement which loops around all the Variable.highlightButton array and checks whether they are 0 or 1.

```
private void studentHighlight()
{
    if (Variables.highlightButton[1, 0] == 1)
    {
        GroupOneStudentOne.BackColor = Color.LightGray;
    }

    if (Variables.highlightButton[1, 0] == 0)
    {
        GroupOneStudentOne.BackColor = Color.MistyRose;
    }
}
```

Graph

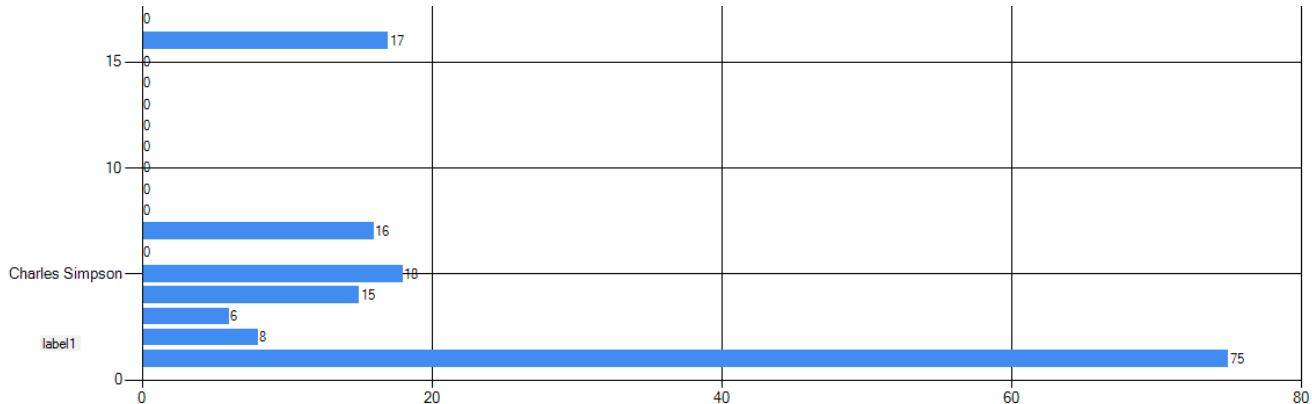
The last part of the system was the graph, this would display all the students' current total grades. Originally, my client requested a line graph, but he had some doubts and also wanted to see a version of the bar chart. The issue with the graph was that the student names only appeared every 5 students. It didn't show the students individually which was the issue.



This was created with the use of the graph tool provided by Visual Studio.

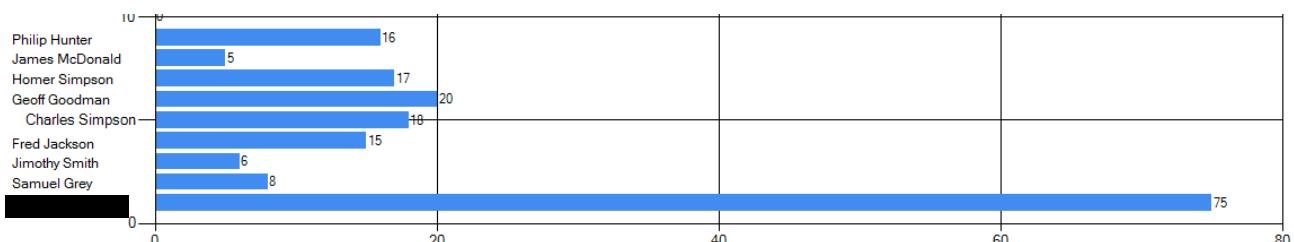
After comparing the data that was represented on the graph to the actual totals, I realised that another problem was that the student 'Charles Simpson' did not have 15 marks, but instead had 18.

I went back over the code and realised that the wrong array was being declared and instead I had to decrement each array by 1 to display the correct student as shown below.



After searching through for ways to make it so all students appear, I could not figure out specific way of automatically setting every students name appear.

Therefore, I decided to create several labels to represent each student. This has been shown below:



The labels have been programmed so they will automatically adapt depending on what the user sets the students name to.

The code for the graph is shown below:

```
this.gradesChart.Series["Student Grades"].Points.AddXY(Variables.allStudentNames[0, 0], Variables.finalStudentTotals[1, 0]);
this.gradesChart.Series["Student Grades"].Points.AddXY(Variables.allStudentNames[1, 0], Variables.finalStudentTotals[2, 0]);
this.gradesChart.Series["Student Grades"].Points.AddXY(Variables.allStudentNames[2, 0], Variables.finalStudentTotals[3, 0]);
this.gradesChart.Series["Student Grades"].Points.AddXY(Variables.allStudentNames[3, 0], Variables.finalStudentTotals[4, 0]);
this.gradesChart.Series["Student Grades"].Points.AddXY(Variables.allStudentNames[4, 0], Variables.finalStudentTotals[5, 0]);
this.gradesChart.Series["Student Grades"].Points.AddXY(Variables.allStudentNames[5, 0], Variables.finalStudentTotals[6, 0]);
this.gradesChart.Series["Student Grades"].Points.AddXY(Variables.allStudentNames[6, 0], Variables.finalStudentTotals[7, 0]);
```

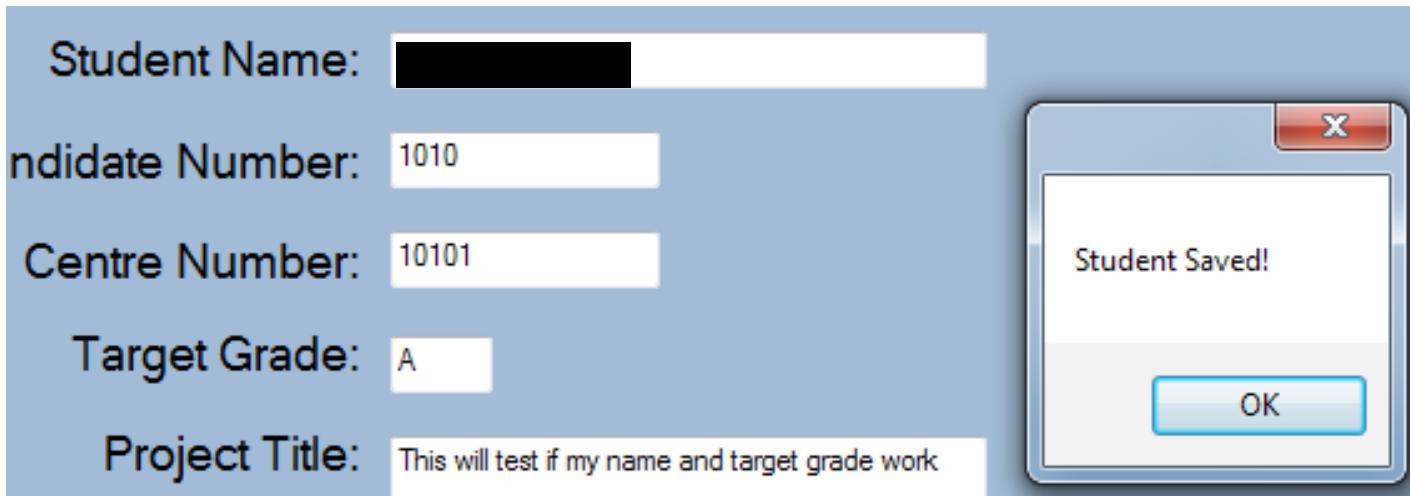
This is the code which automatically changes the name of the labels depending on the users input:

```
graphFirstStudent.Text = Variables.allStudentNames[0, 0].ToString();
graphSecondStudent.Text = Variables.allStudentNames[1, 0].ToString();
graphThirdStudent.Text = Variables.allStudentNames[2, 0].ToString();
graphFourthStudent.Text = Variables.allStudentNames[3, 0].ToString();
```

Software Development – Module 3

Module 3 is the 'LIST ALL STUDENTS' button. This will load a large amount of code which imports all the saved data into Microsoft Excel.

Firstly, I need to test whether the candidate name and target grade work as they have already been programmed.



After going back to the main menu, the results did save and were presented in the spreadsheet.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Candidate Name	A1	A2	B3	B2	B3	C1	C2	D1	E1	E2	E3	TOTAL	%	GRADE	TARGET
2	[Redacted]															A

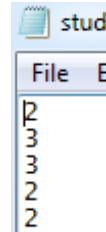
After this, I selected the spreadsheet button and the data appeared.

	A	B	C	D	E	F	G	H	I	J	K	L
1	Candidate	A1	A2	B1	B2	B3	C1	C2	D1	E1	E2	E3
2	[Redacted]	1	2	3	4	5	0	0	0	0	0	0

As part of the user requirements, the total of each section should be displayed. Here is my first test: I first inputted the following.

Section B1: Design		Section B2: Algorithms		Section B3: Test Strategy	
- CRITERIA HERE -	4	- CRITERIA HERE -	4	- CRITERIA HERE -	2
- CRITERIA HERE -	4	- CRITERIA HERE -	0	- CRITERIA HERE -	2
- CRITERIA HERE -	4	- CRITERIA HERE -	0	- CRITERIA HERE -	4
- CRITERIA HERE -	0	- CRITERIA HERE -	4	- CRITERIA HERE -	4
- CRITERIA HERE -	0	- CRITERIA HERE -	0	- CRITERIA HERE -	2
Total:	3	Total:	2	Total:	2

I then clicked update and it saved to the notepad:



I clicked on 'Show Graph' and this was the result:

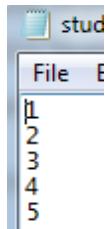
	A	B	C	D	E	F	G	H	I	J	K
1	Candidate A1	A2	B1	B2	B3	C1	C2	D1	E1	E2	
2	JKHSAKJH	2	3	3	3	2	0	0	0	0	0

For some reason, 'B2' had the result 3 instead of 2.

After testing the database numerous times, it continued to save the previous instead of the data inputted. I went back into the code and realised that some textboxes were not being compiled, this did result in the correct output:

Section B1: Design		Section B2: Algorithms		Section B3: Test Strategy	
- CRITERIA HERE -	3	- CRITERIA HERE -	3	- CRITERIA HERE -	5
- CRITERIA HERE -	3	- CRITERIA HERE -	4	- CRITERIA HERE -	5
Total:	3	Total:	4	Total:	5

This result was the same in the text file.



After I had done this, I began to program it so that each of the total grades would be highlighted in their own colour, which depend on whether or not they are close to the maximum marks.

I tested the spreadsheet to see if I could get it to function the first section of 'A1' for all students. It only worked for the first student and didn't go through the other grades.

Sheet1 - Microsoft Excel (Product Activ																		
		Normal																
		Bad																
A1	f _x	Candidate Name	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Candidate A1	A2	B1	B2	B3	C1	C2	D1	E1	E2	E3	TOTAL	%	GRADE	TARGET			
2	[REDACTED]	3	0	0	0	0	0	0	0	0	0	0	3	3.75	F	A		
3	Samuel Gr	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	F	
4	Jimothy Si	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	F	
5	Fred Jacks	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	D	
6	Dawa Kals	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	B	
7		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	F	
8		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	F	
9		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	F	
10		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	F	
11		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	F	
12		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	F	

I had a nesting for loop which was placed the wrong way round. I edited this and added it for section A2 as well. It worked this time as shown below.

Sheet1																	
		Normal															
		Bad															
A17	f _x	Candidate Name	A1	A2	B1	B2	B3	C1	C2	D1	E1	E2	E3	TOTAL	%	GRADE	TARGET
1	Candidate Name	A1	2	7	0	0	0	0	0	0	0	0	0	10	12.5	F	A
2	[REDACTED]	3	0	0	0	0	0	0	0	0	0	0	0	13	16.25	F	A
3	Samuel Grey	2	1	1	0	0	0	0	0	0	0	0	0	1	1.25	F	C
4	Jimothy Smith	1	0	0	0	0	0	0	0	0	0	0	0	1	1.25	F	C
5	Fred Jackson	3	0	0	0	0	0	0	0	0	0	0	0	3	3.75	F	D
6	Dawa Kalsang	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	B
7		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	F
8		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	F
9		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	F
10		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	F
11		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	F
12		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	F
13		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	F
14		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	F
15		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	F
16	James Bond	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	E
17		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	F
18		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	F
19		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	F
20		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	F
21		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	F
22		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	F
23		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	F
24		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	F
25		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	F
26		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	F

Following this, I began to program all of the total grades using the same code shown below:

```

for (int i = 0; i < 30; i++)
{
    if (Variables.studentGradesTotals[i, 1] == 3)
    {
        ws.Cells[i+2, 2].Interior.Color =
System.Drawing.ColorTranslator.ToOle(System.Drawing.Color.Green);
    }

    else if (Variables.studentGradesTotals[i, 1] == 2)
    {
        ws.Cells[i + 2, 2].Interior.Color =
System.Drawing.ColorTranslator.ToOle(System.Drawing.Color.Orange);
    }

    else if (Variables.studentGradesTotals[i, 1] < 2)
    {
        ws.Cells[i + 2, 2].Interior.Color =
System.Drawing.ColorTranslator.ToOle(System.Drawing.Color.Red);
    }
}

```

This was the end result –

	A	B	C	D	E	F	G	H	I	J	K	L
1	Candidate A1	A2	B1	B2	B3	C1	C2	D1	E1	E2	E3	
2		3	11	4	4	4	16	14	9	4	3	3
3	Samuel G	2	2	2	1	1	0	0	0	0	0	0
4	Jimothy S	3	0	0	2	1	0	0	0	0	0	0
5	Fred Jacks	3	3	3	3	3	0	0	0	0	0	0
6	New Stud	0	0	0	0	0	0	0	0	0	0	0
7		0	0	0	0	0	0	0	0	0	0	0
8	RINCHEN T	3	6	3	4	4	0	0	0	0	0	0
9		0	0	0	0	0	0	0	0	0	0	0
10		0	0	0	0	0	0	0	0	0	0	0
11		0	0	0	0	0	0	0	0	0	0	0
12		0	0	0	0	0	0	0	0	0	0	0
13		0	0	0	0	0	0	0	0	0	0	0
14		0	0	0	0	0	0	0	0	0	0	0
15		0	0	0	0	0	0	0	0	0	0	0
16		0	0	0	0	0	0	0	0	0	0	0
17	Rinchen T	2	3	3	4	5	0	0	0	0	0	0
18		0	0	0	0	0	0	0	0	0	0	0
19		0	0	0	0	0	0	0	0	0	0	0
20		0	0	0	0	0	0	0	0	0	0	0
21		0	0	0	0	0	0	0	0	0	0	0
22		0	0	0	0	0	0	0	0	0	0	0
23		0	0	0	0	0	0	0	0	0	0	0

The current grade for students were not programmed in the actual system but instead was calculated on Excel as there is an IF statement provided. At first, I was confused but found a website which gave a description of how to use IF statements.¹

Below shows the code that was used to calculate the current grade of the student.

```
ws.Cells[2, 15] =
"=IF(M2>=67, \"A\", IF(M2>=58, \"B\", IF(M2>=49, \"C\", IF(M2>=41, \"D\", IF(M2>=33, \"E\",
IF(M2<=32, \"F\"))))));"
```

I chose these grade boundaries as they were the official F454 grade boundaries²:

- A = 67=+
- B = 58=+
- C = 49=+
- D = 41=+
- E = 33=+
- F = 32=-

Here is a screenshot of the result:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Candidate A1	A2	B1	B2	B3	C1	C2	D1	E1	E2	E3		TOTAL	%	GRADE	TARGET
2		3	11	4	4	4	16	14	9	4	3	3	75	93.75	A	A
3	Samuel G	2	2	2	1	1	0	0	0	0	0	0	8	10	F	b
4	Timothy S	3	0	0	2	1	0	0	0	0	0	0	6	7.5	F	b
5	Fred Jacks	3	3	3	3	3	0	0	0	0	0	0	15	18.75	F	b
6	New Stud	0	0	0	0	0	0	0	0	0	0	0	0	0	0	F
7		0	0	0	0	0	0	0	0	0	0	0	0	0	0	F
8	RINCHEN	3	6	3	4	4	0	0	0	0	0	0	20	25	F	B
9		0	0	0	0	0	0	0	0	0	0	0	0	0	0	F
10		0	0	0	0	0	0	0	0	0	0	0	0	0	0	F
11		0	0	0	0	0	0	0	0	0	0	0	0	0	0	F
12		0	0	0	0	0	0	0	0	0	0	0	0	0	0	F
13		0	0	0	0	0	0	0	0	0	0	0	0	0	0	F
14		0	0	0	0	0	0	0	0	0	0	0	0	0	0	F
15		0	0	0	0	0	0	0	0	0	0	0	0	0	0	F
16		0	0	0	0	0	0	0	0	0	0	0	0	0	0	F
17		0	0	0	0	0	0	0	0	0	0	0	0	0	0	F
18		0	0	0	0	0	0	0	0	0	0	0	0	0	0	F
19		0	0	0	0	0	0	0	0	0	0	0	0	0	0	F
20		0	0	0	0	0	0	0	0	0	0	0	0	0	0	F
21		0	0	0	0	0	0	0	0	0	0	0	0	0	0	F
22		0	0	0	0	0	0	0	0	0	0	0	0	0	0	F
23		0	0	0	0	0	0	0	0	0	0	0	0	0	0	F
24		0	0	0	0	0	0	0	0	0	0	0	0	0	0	F
25		0	0	0	0	0	0	0	0	0	0	0	0	0	0	F
26		0	0	0	0	0	0	0	0	0	0	0	0	0	0	F

¹ <http://fiveminutelessons.com/learn-microsoft-excel/using-multiple-if-statements-excel>

² <http://www.ocr.org.uk/Images/142071-unit-level-raw-mark-grade-boundaries-june-2013.pdf>

C2: Software Testing

Normal, Extreme and Erroneous Data – Page 51

TESTING CANDIDATE NAME

All these tests have been created on page 51.

During the process of testing, I realised that it was not possible to prove some data as there were no messages. To combat this, I have added a message box every time the user has successfully added a student.

Test Case: Candidate name	Steps	Expected Results	Status (Failed / Pass)
1)	John Smith	Data accepted	Pass
2)	JJ	Error message	Pass
3)	08smith.j	Error message	Pass

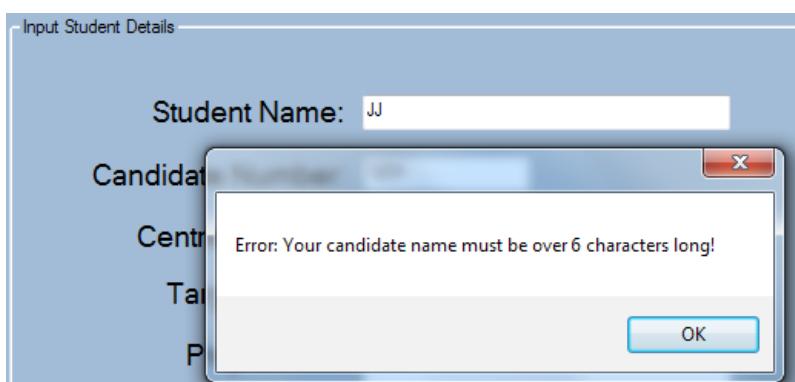
1st Test

The message box shows that the student data has been saved.



2nd Test

When entering the name 'JJ', the data still went through and it allowed to enter the next page. After looking back at my code, it appears that I had forgotten to add validation for the amount of characters for names. I have amended this and tested it out again and it passed, as shown below:

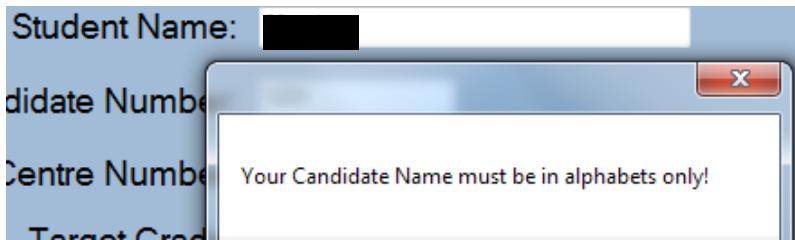


I then realised that even though there was an error, the page still took me to the next module which it shouldn't have. I went back to my code and realised that I had forgotten to add 'return;' after the statement as shown below:

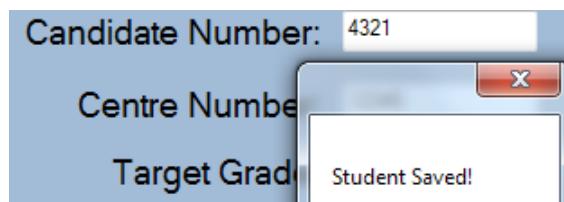
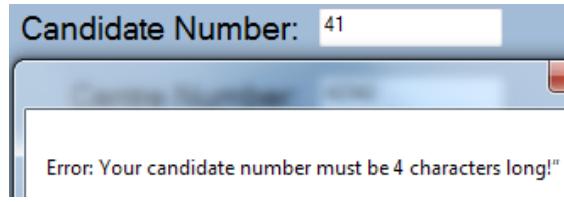
```
else if (studentNameInput.Text.Length < 6)
{
    MessageBox.Show("Error: Your candidate name must be over 6 characters long!");
    return;
}
```

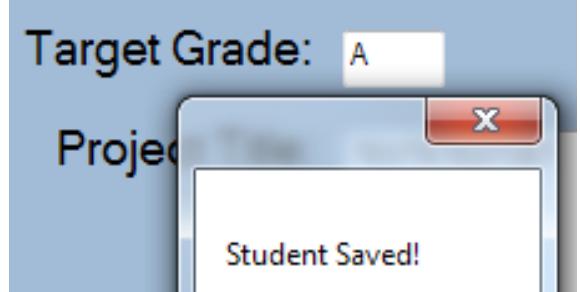
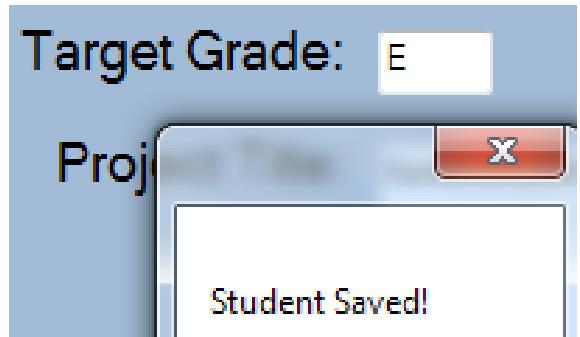
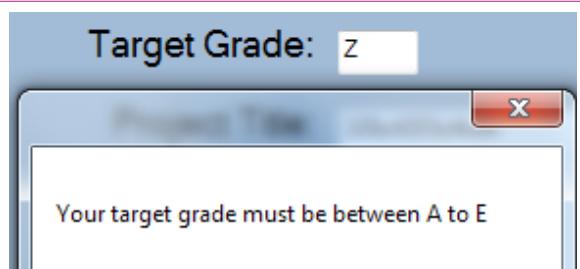
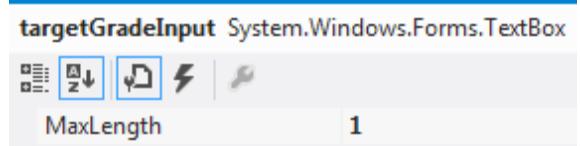
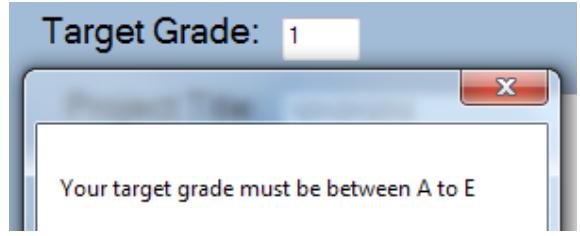
3rd Test

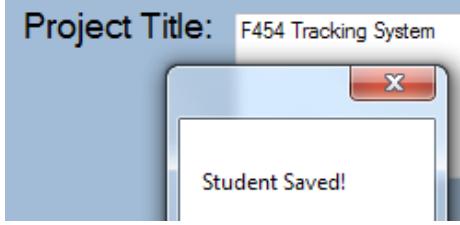
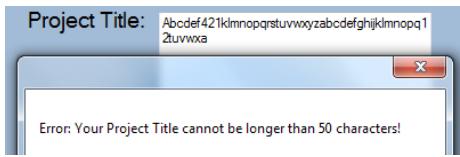
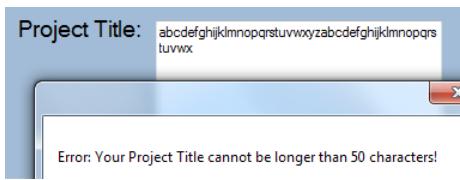
Inputting [REDACTED] resulted in an error as stated.

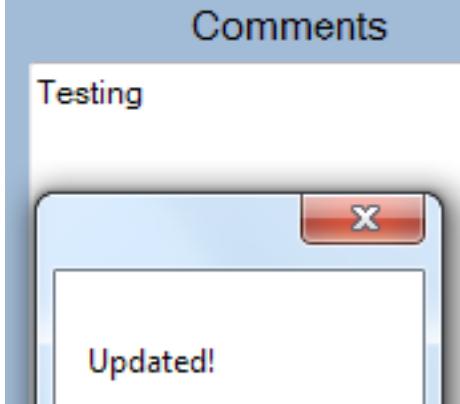
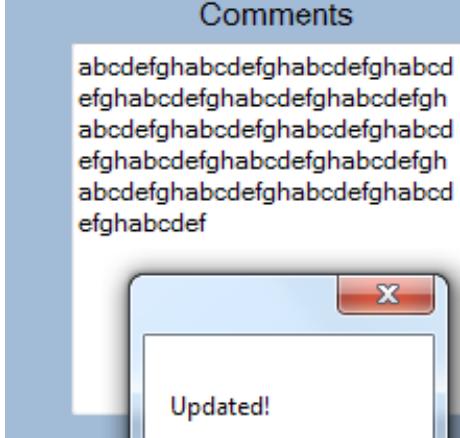
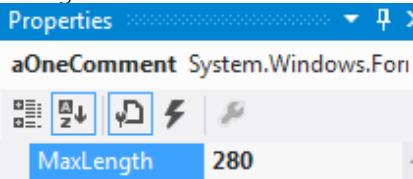
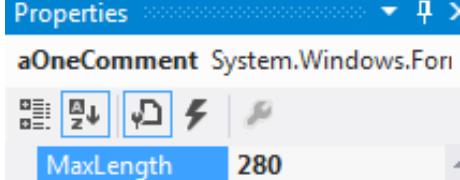


TESTING CANDIDATE NUMBER

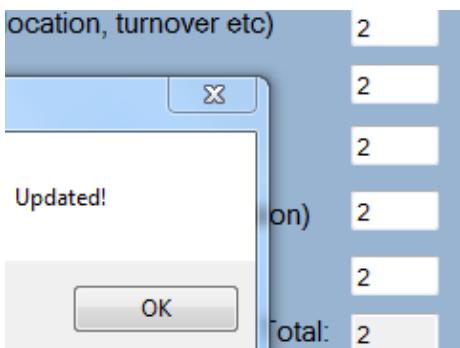
Candidate number	Steps	Expected Results	Evidence	Changes / Notes
1)	4321	Data accepted		
2)	41	Error message		
3)	abcd	Error message	// Checks to see if the candidate name only has alphabets if (!System.Text.RegularExpressions.Regex.IsMatch(studentNameInput.Text, "^[a-zA-Z]")) { MessageBox.Show("Your Candidate Name must be in alphabets only!"); studentNameInput.Text.Remove(studentNameInput.Text.Length - 1); return; }	This code restricts the textbox for alphabets only
4)	abc	Error message	// Checks to see if the candidate name only has alphabets if (!System.Text.RegularExpressions.Regex.IsMatch(studentNameInput.Text, "^[a-zA-Z]")) { MessageBox.Show("Your Candidate Name must be in alphabets only!"); studentNameInput.Text.Remove(studentNameInput.Text.Length - 1); return; }	This code restricts the textbox for alphabets only
5)	1.02	Error message	// Checks to see if the candidate name only has alphabets if (!System.Text.RegularExpressions.Regex.IsMatch(studentNameInput.Text, "^[a-zA-Z]")) { MessageBox.Show("Your Candidate Name must be in alphabets only!"); studentNameInput.Text.Remove(studentNameInput.Text.Length - 1); return; }	This code restricts the textbox for alphabets only
6)	2.0	Error message	// Checks to see if the candidate name only has alphabets if (!System.Text.RegularExpressions.Regex.IsMatch(studentNameInput.Text, "^[a-zA-Z]")) { MessageBox.Show("Your Candidate Name must be in alphabets only!"); studentNameInput.Text.Remove(studentNameInput.Text.Length - 1); return; }	This code restricts the textbox for alphabets only so the decimal does not appear.

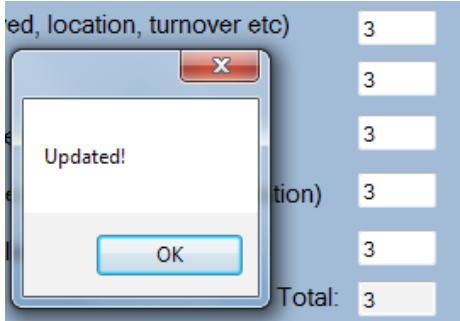
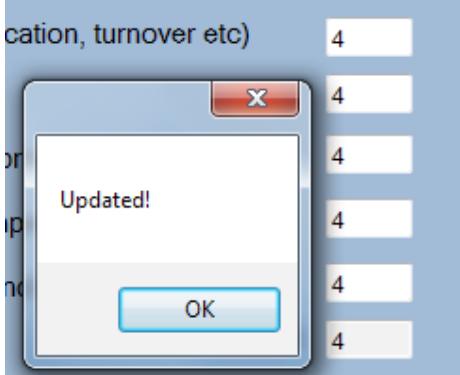
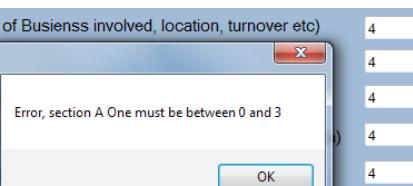
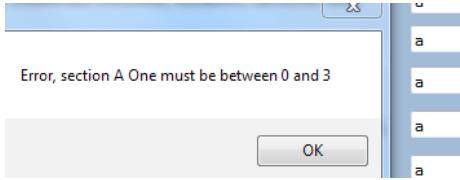
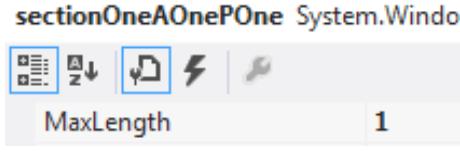
Target Grade	Steps	Expected Results	Evidence	Changes / Notes
7)	Enter alphabet A	Data accepted		
8)	Enter alphabet E	Data accepted		
9)	Enter alphabet Z	Error message		
10)	Enter alphabet AB	Error message		Visual Studio provides an option to limit the textbox to 1
11)	Enter one digit	Error message		

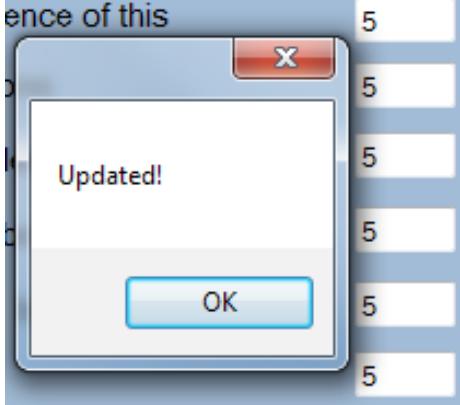
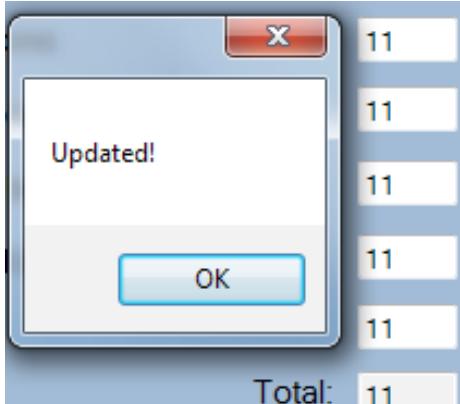
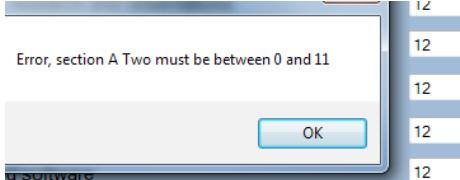
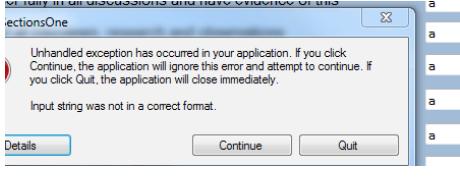
Target Grade	Steps	Expected Results	Evidence	Changes / Notes
12)	F454 Tracking System	Data accepted		
13)	Abcdef421k lmnopqrstuvwxyz vwxyzabcd efghijklmno pq12uvwxyzxa	Error message		
14)	abcdefghijkl mnopqrstuvwxyz wxyzabcdef ghijklmnop qrstuvwxyzx	Data accepted		<p>Resulted in an error message. I went back to the code, I had wrote the operators wrong and so I changed it so the error message only appeared when it reached over 50 and not equal to or over to 50.</p> <pre>else if (ProjectTitleInput.Text.Length > 50) { MessageBox.Show("Error: Your Project Title cannot be longer than 50 characters!"); return; }</pre>

Comment box	Steps	Expected Results	Evidence	Changes / Notes
15)	Testing	Data accepted		
16)	Enter a comment with 280 characters	Data accepted		I was not able to enter 280 characters. I realised I had limited the textbox to 150 instead of 280. I changed it here: 
17)	Enter a character with over 280 characters	Error message		I made it so you cannot enter above 280, shown here:

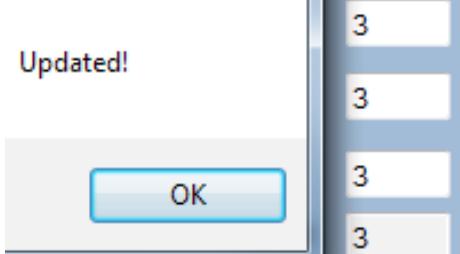
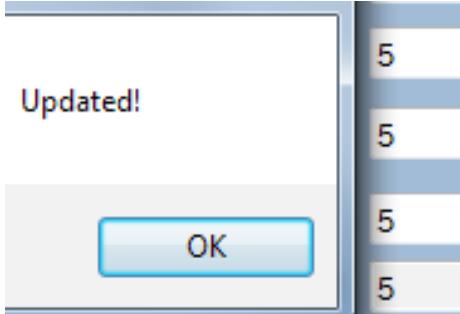
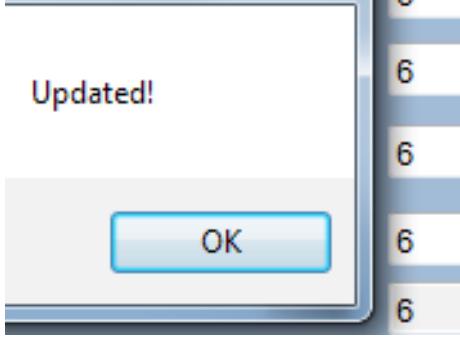
TESTING MARKS

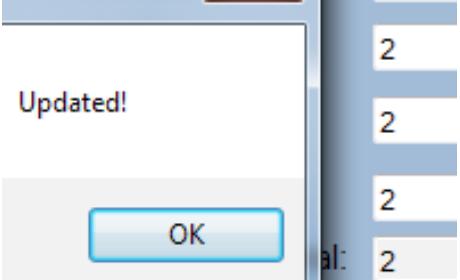
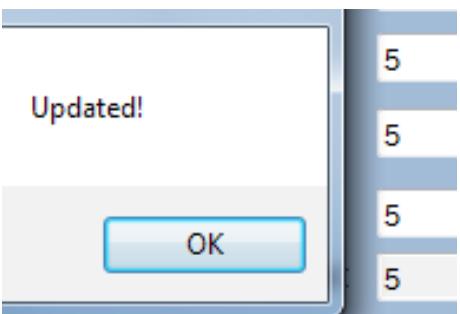
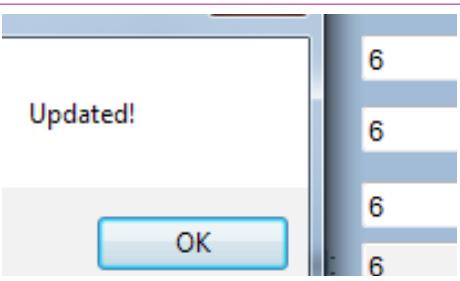
Inputting Marks A1	Steps	Expected Results	Evidence	Changes / Notes
18)	2	Data accepted		

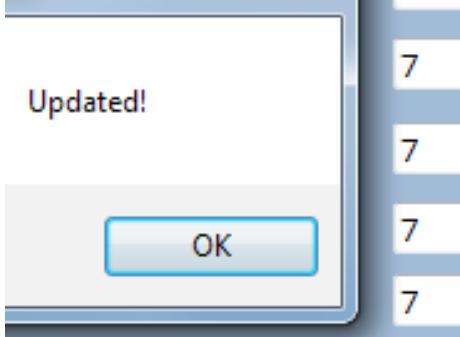
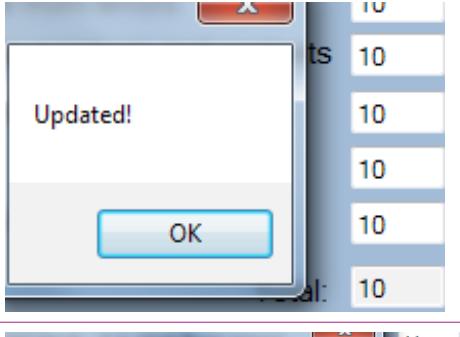
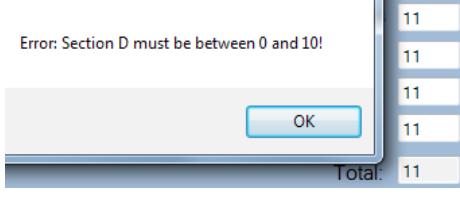
Inputting Marks A1	Steps	Expected Results	Evidence	Changes / Notes
19)	3	Data accepted		
20)	4	Error message		<p>The error did not show. I went back to the code and realised I had minimised it originally. I maximised it and the error message appeared.</p> <pre>private void sectionOneSaveAll_Click(object sender, EventArgs e){}</pre> <p>Error message shown below:</p> 
21)	a	Error message		
22)	-4	Error message		Textbox is limited to one so minus cannot be entered.

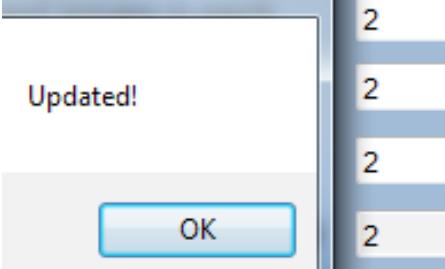
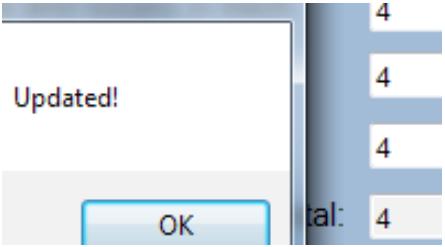
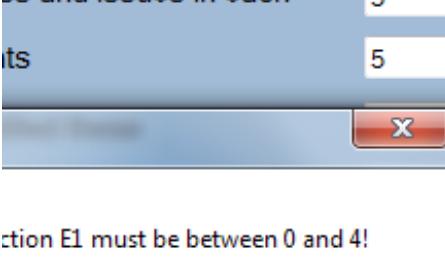
Inputting Marks A2	Steps	Expected Results	Evidence	Changes / Notes
23)	5	Data accepted		
24)	11	Data accepted		
25)	12	Error message		
26)	a	Error message		System crashed. I decided to limit all textboxes to integer only as shown below. <pre>// If statement to check if number has been entered. if (!char.IsControl(e.KeyChar) && !char.IsDigit(e.KeyChar)) { e.Handled = true; }</pre>
27)	-4	Error message	Minus cannot be entered as the textbox is limited to integer only. Code shown below: <pre>// If statement to check if number has been entered. if (!char.IsControl(e.KeyChar) && !char.IsDigit(e.KeyChar)) { e.Handled = true; }</pre>	Textbox is limited to one so minus cannot be entered.

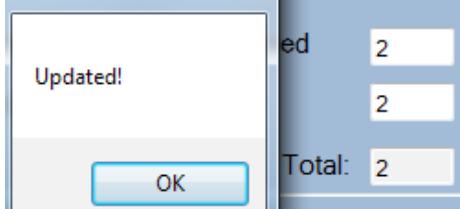
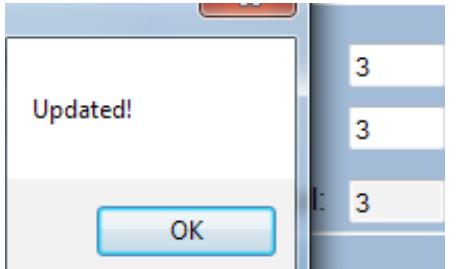
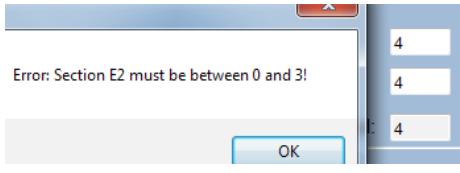
Inputting Marks B1	Steps	Expected Results	Evidence	Changes / Notes																
28)	3	Data accepted	<p>on User Requirements)</p> <table border="1"> <tr><td>3</td></tr> <tr><td>3</td></tr> <tr><td>3</td></tr> <tr><td>3</td></tr> <tr><td>3</td></tr> <tr><td>3</td></tr> <tr><td>Total: 3</td></tr> </table>	3	3	3	3	3	3	Total: 3										
3																				
3																				
3																				
3																				
3																				
3																				
Total: 3																				
29)	6	Data accepted	<table border="1"> <tr><td>6</td></tr> <tr><td>6</td></tr> <tr><td>6</td></tr> <tr><td>6</td></tr> <tr><td>6</td></tr> <tr><td>6</td></tr> <tr><td>Total: 6</td></tr> </table>	6	6	6	6	6	6	Total: 6										
6																				
6																				
6																				
6																				
6																				
6																				
Total: 6																				
30)	7	Error message	<table border="1"> <tr><td>7</td></tr> <tr><td>7</td></tr> <tr><td>7</td></tr> <tr><td>7</td></tr> <tr><td>7</td></tr> <tr><td>7</td></tr> <tr><td>7</td></tr> <tr><td>Total: 7</td></tr> </table>	7	7	7	7	7	7	7	Total: 7	<p>No error message shown. I edited the code so that it would result in an error, this is the code:</p> <pre>if (Variables.allStudentGrades[Variables.studentUniqueNumber, 13] > 6 { MessageBox.Show("Error, section B1 must be between 0 and 6"); return; }</pre> <p>This is the error message:</p> <table border="1"> <tr><td>7</td></tr> <tr><td>7</td></tr> <tr><td>7</td></tr> <tr><td>7</td></tr> <tr><td>7</td></tr> <tr><td>7</td></tr> <tr><td>7</td></tr> <tr><td>Total: 7</td></tr> </table>	7	7	7	7	7	7	7	Total: 7
7																				
7																				
7																				
7																				
7																				
7																				
7																				
Total: 7																				
7																				
7																				
7																				
7																				
7																				
7																				
7																				
Total: 7																				
31)	a	Error message	<p>'a' cannot be entered as the textbox is limited to integer only. Code:</p> <pre>// If statement to check if number has been entered. if (!char.IsControl(e.KeyChar) && !char.IsDigit(e.KeyChar)) { e.Handled = true; }</pre>																	
32)	-4	Error message	<p>Minus cannot be entered as the textbox is limited to integer only. Code shown below:</p> <pre>// If statement to check if number has been entered. if (!char.IsControl(e.KeyChar) && !char.IsDigit(e.KeyChar)) { e.Handled = true; }</pre>																	

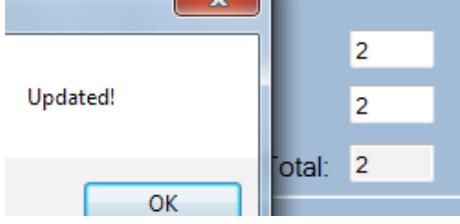
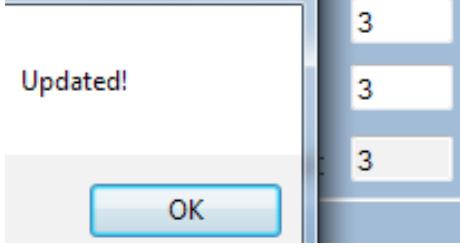
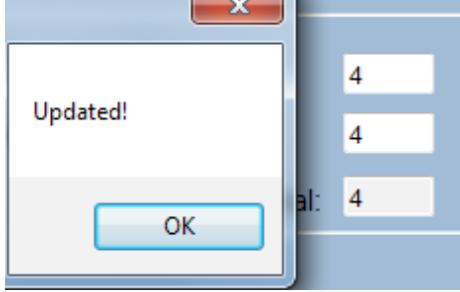
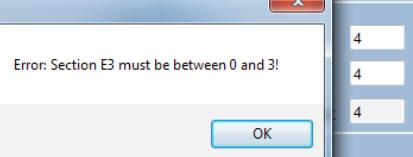
Inputting Marks B2	Steps	Expected Results	Evidence	Changes / Notes
33)	3	Data accepted	<p>Updated!</p> 	
34)	5	Data accepted	<p>Updated!</p> 	
35)	6	Error message	<p>Updated!</p> 	<p>There were no error messages. I went back into the code and realised that data was accepted till 6. This should not be the case, it should only accept up to 5. Edited code:</p> <pre>if (Variables.allStudentGrades[Variables.studentUniqueNumber, 18] > 5) { MessageBox.Show("Error, section B2 must be between 0 and 5"); return; }</pre>
36)	a	Error message	<p>'a' cannot be entered as the textbox is limited to integer only. Code:</p> <pre>// If statement to check if number has been entered. if (!char.IsControl(e.KeyChar) && !char.IsDigit(e.KeyChar)) { e.Handled = true; }</pre>	
37)	-4	Error message	<p>Minus cannot be entered as the textbox is limited to integer only. Code shown below:</p> <pre>// If statement to check if number has been entered. if (!char.IsControl(e.KeyChar) && !char.IsDigit(e.KeyChar)) { e.Handled = true; }</pre>	

Inputting Marks B3	Steps	Expected Results	Evidence	Changes / Notes
38)	2	Data accepted		
39)	5	Data accepted		
40)	6	Error message		Error message did not appear. Similar to B2, I realised the code was the same. The edited code: <pre>if (Variables.allStudentGrades[Variables.studentUniqueNumber, 22] > 5 { MessageBox.Show("Error, section B3 must be between 0 and 5"); }</pre>
41)	a	Error message	"a' cannot be entered as the textbox is limited to integer only. Code: <pre>// If statement to check if number has been entered. if (!char.IsControl(e.KeyChar) && !char.IsDigit(e.KeyChar)) { e.Handled = true; }</pre>	
42)	-4	Error message	Minus cannot be entered as the textbox is limited to integer only. Code shown below: <pre>// If statement to check if number has been entered. if (!char.IsControl(e.KeyChar) && !char.IsDigit(e.KeyChar)) { e.Handled = true; }</pre>	

Inputting Marks D	Steps	Expected Results	Evidence	Changes / Notes
38)	7	Data accepted		
39)	10	Data accepted		
40)	11	Error message		
41)	a	Error message	'a' cannot be entered as the textbox is limited to integer only. Code: <pre>// If statement to check if number has been entered. if (!char.IsControl(e.KeyChar) && !char.IsDigit(e.KeyChar)) { e.Handled = true; }</pre>	
42)	-4	Error message	Minus cannot be entered as the textbox is limited to integer only. Code shown below: <pre>// If statement to check if number has been entered. if (!char.IsControl(e.KeyChar) && !char.IsDigit(e.KeyChar)) { e.Handled = true; }</pre>	

Inputting Marks E1	Steps	Expected Results	Evidence	Changes / Notes
43)	2	Data accepted		
44)	4	Data accepted		
45)	5	Error message		
46)	a	Error message	'a' cannot be entered as the textbox is limited to integer only. Code: <pre>// If statement to check if number has been entered. if (!char.IsControl(e.KeyChar) && !char.IsDigit(e.KeyChar)) { e.Handled = true; }</pre>	
47)	-4	Error message	Minus cannot be entered as the textbox is limited to integer only. Code shown below: <pre>// If statement to check if number has been entered. if (!char.IsControl(e.KeyChar) && !char.IsDigit(e.KeyChar)) { e.Handled = true; }</pre>	

Inputting Marks E2	Steps	Expected Results	Evidence	Changes / Notes
48)	2	Data accepted		
49)	3	Data accepted		
50)	4	Error message		
51)	a	Error message	'a' cannot be entered as the textbox is limited to integer only. Code: <pre>// If statement to check if number has been entered. if (!char.IsControl(e.KeyChar) && !char.IsDigit(e.KeyChar)) { e.Handled = true; }</pre>	
52)	-4	Error message	Minus cannot be entered as the textbox is limited to integer only. Code shown below: <pre>// If statement to check if number has been entered. if (!char.IsControl(e.KeyChar) && !char.IsDigit(e.KeyChar) { e.Handled = true; }</pre>	

Inputting Marks E3	Steps	Expected Results	Evidence	Changes / Notes
48)	2	Data accepted		
49)	3	Data accepted		
50)	4	Error message		<p>Error message did not show. I went back to the code and realised there was no 'return' statement. So it was going through all the code and prompting 'Updated'. I added the return with the code below:</p> <pre>if (Variables.allStudentGrades[Variables.studentUniqueNumber, 47] { MessageBox.Show("Error: Section E3 must be between 0 and 3!"); return; }</pre> <p>Error message displayed:</p> 
51)	a	Error message	'a' cannot be entered as the textbox is limited to integer only. Code:	
52)	-4	Error message	Minus cannot be entered as the textbox is limited to integer only. Code shown below:	

Spreadsheet Total	Steps	Expected Results	Evidence	Changes / Notes																												
53)	Enter a set of marks & check the total	Produce the correct total	<table border="1"> <tr> <td>A1</td><td>A2</td><td>B1</td><td>B2</td><td>B3</td><td>C1</td><td>C2</td><td>D1</td><td>E1</td><td>E2</td><td>E3</td><td>TOTAL</td> </tr> <tr> <td>3</td><td>2</td><td>5</td><td>4</td><td>6</td><td>7</td><td>0</td><td>0</td><td>5</td><td>0</td><td>0</td><td>32</td> </tr> </table>	A1	A2	B1	B2	B3	C1	C2	D1	E1	E2	E3	TOTAL	3	2	5	4	6	7	0	0	5	0	0	32					
A1	A2	B1	B2	B3	C1	C2	D1	E1	E2	E3	TOTAL																					
3	2	5	4	6	7	0	0	5	0	0	32																					
54)	Enter marks under 0	Error message	<p>Cannot be done through the system as all data have been validated. It can be edited through the spreadsheet though. I added a code to make sure it creates an error message.</p> <pre>f2c =IF(M2>=67,"A",IF(M2>=58,"B",IF(M2>=49,"C",IF(M2>=41,"D",IF(M2>=33,"E",IF(M2<=32,"F",IF(M2<0,"ERROR"))))))</pre> <p>Screenshot of the error:</p> <table border="1"> <tr> <td>A1</td><td>A2</td><td>B1</td><td>B2</td><td>B3</td><td>C1</td><td>C2</td><td>D1</td><td>E1</td><td>E2</td><td>E3</td><td>TOTAL</td><td>%</td><td>GRADE</td> </tr> <tr> <td>-1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>-1</td><td>-1.25</td><td>ERROR</td> </tr> </table>	A1	A2	B1	B2	B3	C1	C2	D1	E1	E2	E3	TOTAL	%	GRADE	-1	0	0	0	0	0	0	0	0	0	0	-1	-1.25	ERROR	
A1	A2	B1	B2	B3	C1	C2	D1	E1	E2	E3	TOTAL	%	GRADE																			
-1	0	0	0	0	0	0	0	0	0	0	-1	-1.25	ERROR																			

Spreadsheet Total	Steps	Expected Results	Evidence	Changes / Notes																												
55)	Enter 40 marks	50%	<table border="1"> <tr> <td>A1</td><td>A2</td><td>B1</td><td>B2</td><td>B3</td><td>C1</td><td>C2</td><td>D1</td><td>E1</td><td>E2</td><td>E3</td><td>TOTAL</td><td>%</td> </tr> <tr> <td>3</td><td>3</td><td>3</td><td>3</td><td>3</td><td>3</td><td>3</td><td>10</td><td>3</td><td>3</td><td>3</td><td>40</td><td>50</td> </tr> </table>	A1	A2	B1	B2	B3	C1	C2	D1	E1	E2	E3	TOTAL	%	3	3	3	3	3	3	3	10	3	3	3	40	50			
A1	A2	B1	B2	B3	C1	C2	D1	E1	E2	E3	TOTAL	%																				
3	3	3	3	3	3	3	10	3	3	3	40	50																				
56)	Enter 80 marks	100%	<table border="1"> <tr> <td>A1</td><td>A2</td><td>B1</td><td>B2</td><td>B3</td><td>C1</td><td>C2</td><td>D1</td><td>E1</td><td>E2</td><td>E3</td><td>TOTAL</td><td>%</td> </tr> <tr> <td>3</td><td>11</td><td>6</td><td>5</td><td>5</td><td>16</td><td>14</td><td>10</td><td>4</td><td>3</td><td>3</td><td>80</td><td>100</td> </tr> </table>	A1	A2	B1	B2	B3	C1	C2	D1	E1	E2	E3	TOTAL	%	3	11	6	5	5	16	14	10	4	3	3	80	100			
A1	A2	B1	B2	B3	C1	C2	D1	E1	E2	E3	TOTAL	%																				
3	11	6	5	5	16	14	10	4	3	3	80	100																				
57)	Enter 0 marks	0%	<table border="1"> <tr> <td>A1</td><td>A2</td><td>B1</td><td>B2</td><td>B3</td><td>C1</td><td>C2</td><td>D1</td><td>E1</td><td>E2</td><td>E3</td><td>TOTAL</td><td>%</td> </tr> <tr> <td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td> </tr> </table>	A1	A2	B1	B2	B3	C1	C2	D1	E1	E2	E3	TOTAL	%	0	0	0	0	0	0	0	0	0	0	0	0	0			
A1	A2	B1	B2	B3	C1	C2	D1	E1	E2	E3	TOTAL	%																				
0	0	0	0	0	0	0	0	0	0	0	0	0																				
58)	Enter over 80 marks	Error message	<p>Cannot be done through the system as it is validated. I created a code for Excel so that it would say 'ERROR' if this happens.</p> <pre>,IF(M2<0,"ERROR",IF(M2>80,"ERROR"))))))))</pre> <p>This is the screenshot:</p> <table border="1"> <tr> <td>A1</td><td>A2</td><td>B1</td><td>B2</td><td>B3</td><td>C1</td><td>C2</td><td>D1</td><td>E1</td><td>E2</td><td>E3</td><td>TOTAL</td><td>%</td><td>GRADE</td> </tr> <tr> <td>3</td><td>11</td><td>6</td><td>5</td><td>5</td><td>16</td><td>14</td><td>10</td><td>4</td><td>3</td><td>5</td><td>82</td><td>102.5</td><td>ERROR</td> </tr> </table>	A1	A2	B1	B2	B3	C1	C2	D1	E1	E2	E3	TOTAL	%	GRADE	3	11	6	5	5	16	14	10	4	3	5	82	102.5	ERROR	
A1	A2	B1	B2	B3	C1	C2	D1	E1	E2	E3	TOTAL	%	GRADE																			
3	11	6	5	5	16	14	10	4	3	5	82	102.5	ERROR																			

Bar chart results	Steps	Expected Results	Evidence	Changes / Notes																										
59)	Enter a random set of marks and check the bar chart is correct	Bar chart is at correct level	<p>These were the results I inputted:</p> <table border="1"> <tr> <td>Candidate Name</td> <td>A1</td> <td>A2</td> <td>B1</td> <td>B2</td> <td>B3</td> <td>C1</td> <td>C2</td> <td>D1</td> <td>E1</td> <td>E2</td> <td>E3</td> <td>TOTAL</td> </tr> <tr> <td>John Smith</td> <td>3</td> <td>2</td> <td>5</td> <td>4</td> <td>4</td> <td>7</td> <td>0</td> <td>10</td> <td>4</td> <td>3</td> <td>2</td> <td>44</td> </tr> </table> <p>This was the graph, it showed the correct data:</p>	Candidate Name	A1	A2	B1	B2	B3	C1	C2	D1	E1	E2	E3	TOTAL	John Smith	3	2	5	4	4	7	0	10	4	3	2	44	
Candidate Name	A1	A2	B1	B2	B3	C1	C2	D1	E1	E2	E3	TOTAL																		
John Smith	3	2	5	4	4	7	0	10	4	3	2	44																		
60)	Enter full 80 marks	All bars are at maximum level																												
61)	Enter 0 marks	Bars do not appear / rise to 0 marks																												

Centre Number	Steps	Expected Results	Evidence	Changes / Notes
62)	Enter 5 digits	Data accepted		
63)	Enter digits other than 5	Error message		
64)	Enter 5 alphabets and numbers	Error message	<p>Cannot be entered due to this code which stops any input except for numbers:</p> <pre>// Limits any alphabets from being entered in the Centre Number if (!char.IsControl(e.KeyChar) && !char.IsDigit(e.KeyChar)) { e.Handled = true; }</pre>	
65)	Enter alphabets and numbers that's not 5	Error message	<p>Cannot be entered due to this code which stops any input except for numbers:</p> <pre>// Limits any alphabets from being entered in the Centre Number if (!char.IsControl(e.KeyChar) && !char.IsDigit(e.KeyChar)) { e.Handled = true; }</pre>	

With this in mind, below will be the evidence that will be shown to the end user of all their requirements. Following this, all changes will be done depending on the user's feedback. This will then mean that the system has been fully developed after any appropriate amendments have been

Acceptance Testing

This was the acceptance testing that I agreed on originally, which can be found on page 49 but has also been shown below. I gave the tests to my user and asked them to fill in the table to show if there were any problems:

- Menu button links up to all appropriate modules.
 - Group will link up to a class full of students
 - 'LIST ALL STUDENTS' will direct the user to an Excel file
- Class full of students will contain buttons with different students
 - Buttons with no data will transfer the user over to a form so they can add a new student
 - Button with data will transfer the user over to a form which allows the user to input grades
 - Close button will close the page; therefore the main menu will still be visible.
- Adding a new student will open a form, the following details will need to be filled in:
 - Candidate Name, Candidate Number, Centre Number, Target Grade, Project Title
 - All the details will need to be validated as shown in the Validation section.
 - Submit button will need to save all the data that has been input, unless there are missing / unacceptable data.
- For profiles that have already been created, the user will be able to input grades.
 - Grades will need to be validated – this has been shown in the validation section.
 - Comments are saved upon input, if this does not work, a save button will be created.
 - Student profiles contain two buttons at the bottom which will allow the user to view the next / previous page.
 - There will also be a graph button which allows the user to view the excel file but the data will be represented in a graph / bar chart form.
 - An edit button will be there to allow the user to edit the information of the student.
- 'LIST ALL STUDENTS' will direct user to an Excel file and data that the user has input will be transferred over to the spreadsheet with the use of a Database.
- Printing will happen in the Excel file and the student input page.

Each of the tests have been split into the modules:

Module 1

- 1a.** Click 'GROUP 1' button to display the list of students
1b. Click 'LIST ALL STUDENTS' to display the spreadsheet full of students

Module 2

- 2a.** Create a new student
2b. Edit the students information
2c. Input valid grades into the students details
2d. Input invalid grades into the student details, this should result in an error
2e. Click 'save all' and ensure that all the totals are correct
2f. Enter a comment and save
2g. Highlight a student
2h. Un-Highlight a student
2i. Click on graph button
2j. Check to see if the graph is accurate from the grades that you have inputted

Module 3

- 3a.** Check the colour scheme to see whether they have been correctly displayed
3b. Check to see if the correct students are displayed

- 3c.** Input some grades into the spreadsheet
3d. Check to see if the totals are correct
3e. Check to see if the percentage is correct
3f. Check to see if grade is correct
3g. Print

Acceptance Testing Results

Module	Test Number	Target met?	Comments
1	1a	Yes	None
	1b	Yes	None
2	2a	Yes	None
	2b	No	Copy info into the textbox would be ideal
	2c	Yes	None
	2d	No	Allowed 4 to be accepted in A mark – should only allow 3. Validation of A2 is weird – Error for < 3 but should be 11.
	2e	Yes	None
	2f	Yes	None
	2g	No	Highlight student information in yellow to be clearer on screen as colour not very easily distinguishable.
	2h	Yes	None
	2i	No	Can it be moved to the main screen?
	2j	Yes	None
3	3a	Yes	None
	3b	Yes	None
	3c	Yes	None
	3d	Yes	None
	3e	Yes	None
	3f	Yes	None
	3g	Yes	None

SIGNATURE: _____ **DATE:** _____

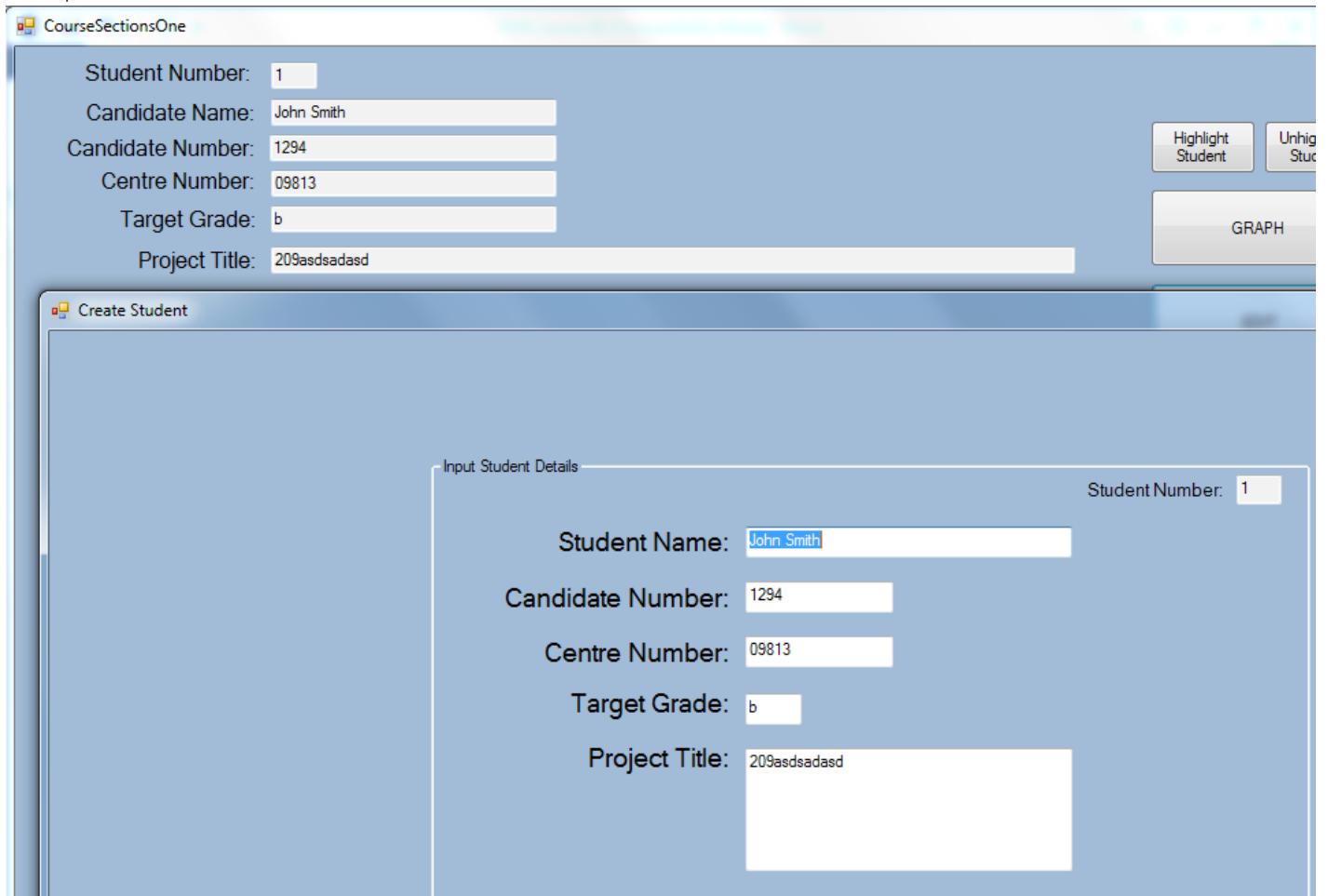
Acceptance Testing Response

I have gathered the results sent by my client and I have created amendments which are shown below:

Editing Student

My client did not like the way when he clicked 'Edit' as the student's current information were not displayed. Therefore, he would like the current student's information pre-written so that he can just edit small parts of the detail instead of having to re-write all the textboxes.

In response, I have created this as shown below:



The image on the previous page shows that the same information had been copied over to the edit student screen. To do this, I used the code shown below:

```
case 1:
{
    // Writing to textfile from array
    studentNameInput.Text = Variables.allStudentNames[0, 0];
    candidateNumberInput.Text = Variables.allStudentNames[0, 1];
    centreNumberInput.Text = Variables.allStudentNames[0, 2];
    targetGradeInput.Text = Variables.allStudentNames[0, 3];
    ProjectTitleInput.Text = Variables.allStudentNames[0, 4];
    break;
}
```

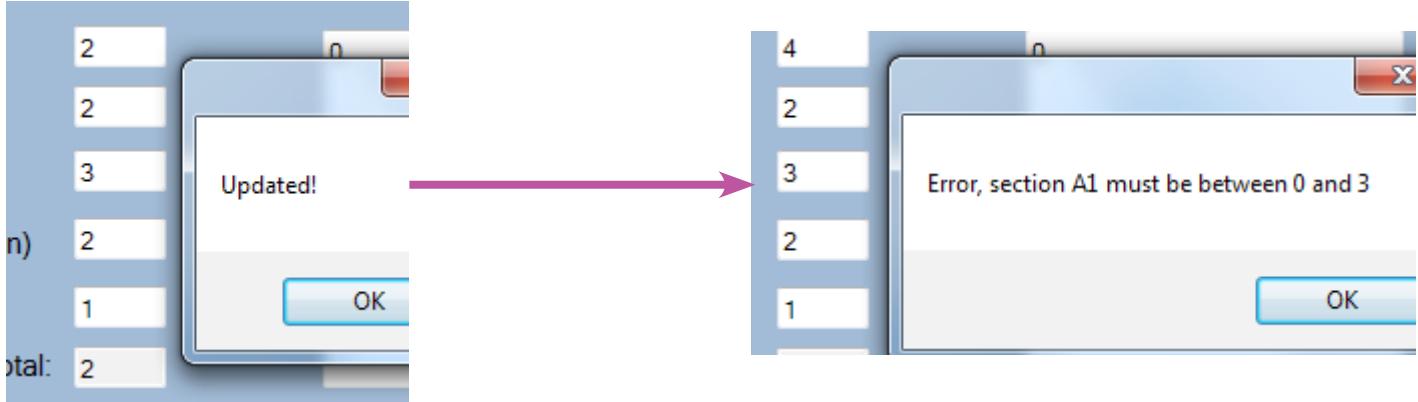
Validation

When the user tested to see if invalid data was accepted, it did. Section A1 was accepting data that was above 3 and section A2 was not accepted data above 3. Through reviewing the code, what was happening was that the code was saving, but once there was an error, it would no longer save. Therefore, I created a read instruction in order to scan through all the student grades so that it can detect which grades are the latest one.

```
private void sectionOneSaveAll_Click(object sender, EventArgs e)
{
    // Opens the reader
    StreamReader gradesRead = new StreamReader(@"S:\studentgrades.txt", false);

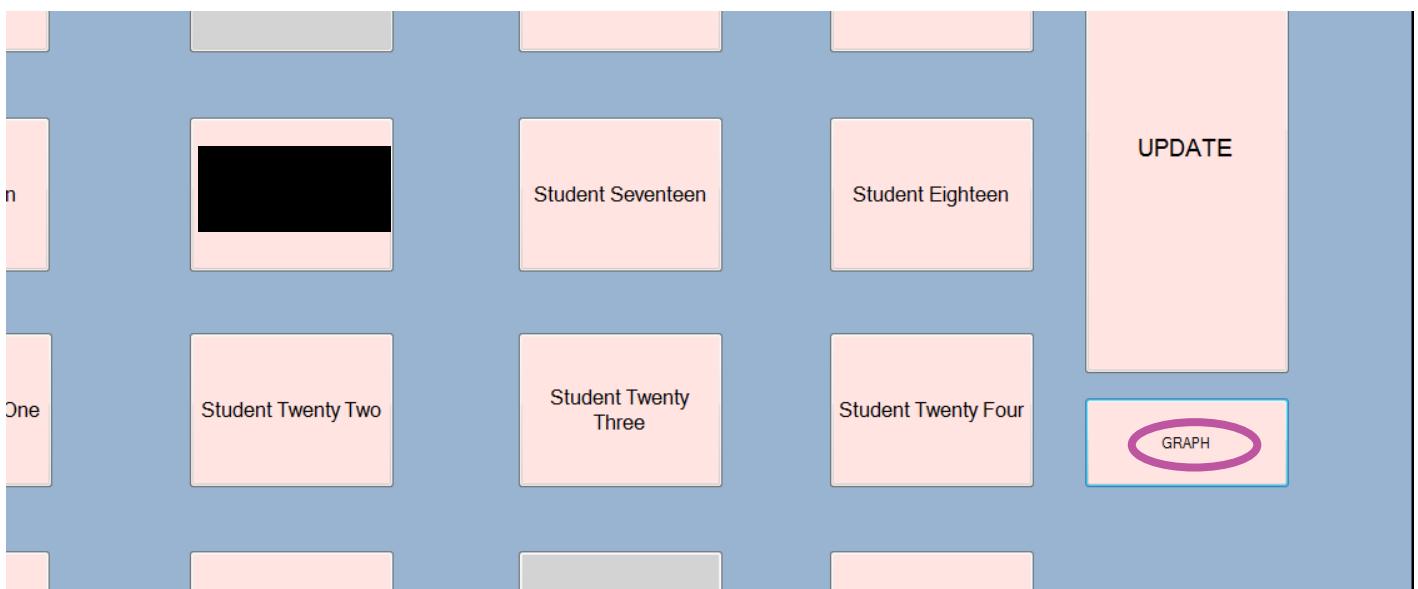
    // For loop to go through each of the students.
    for (int x = 1; x < 31; x++)
    {
        for (int y = 1; y < 49; y++)
        {
            Variables.allStudentGrades[x, y] = Convert.ToInt32(gradesRead.ReadLine());
        }
    }
}
```

Below shows the saving working correctly:



Graph

My client requested that a graph button was placed on the main page of all students in order to easily access it without having to click on a student. A button was created on the page and the same code was added, image shown below:



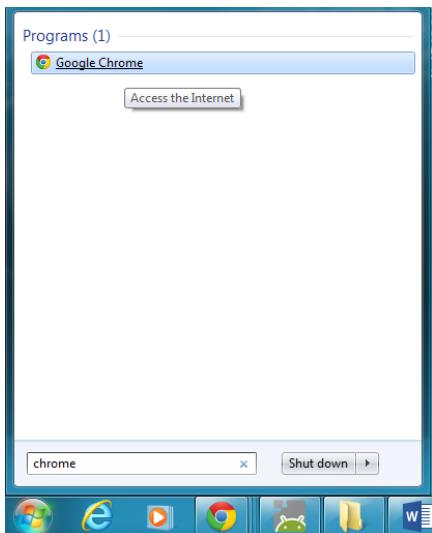
D1: Documentation

User Guide

Download and Installation

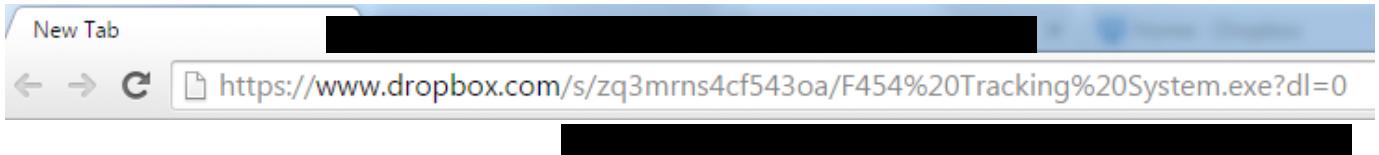
To download the F454 tracking system, you must do the following:

1. Open your browser:



2. Enter the following link:

<https://www.dropbox.com/s/zq3mrns4cf543oa/F454%20Tracking%20System.exe?dl=0>



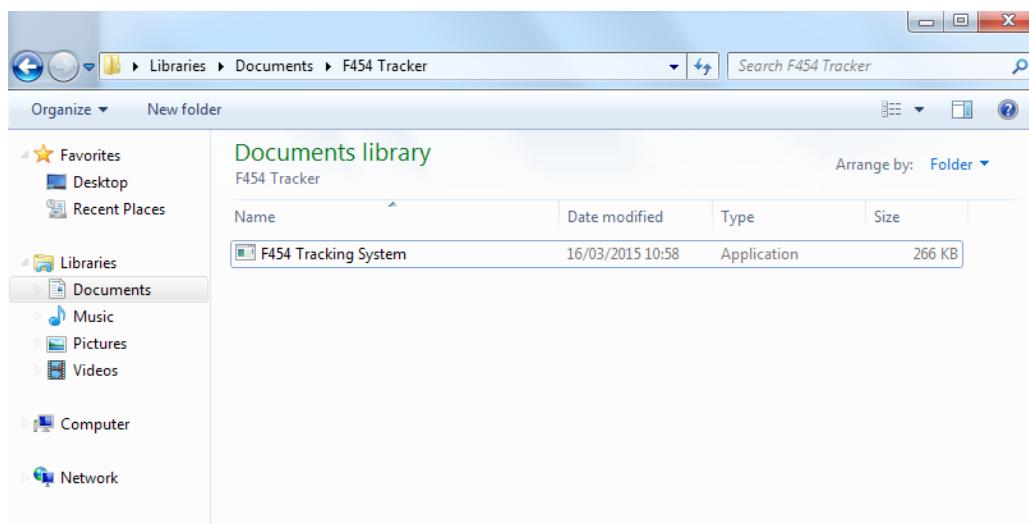
3. When on the website, click on 'Download'.



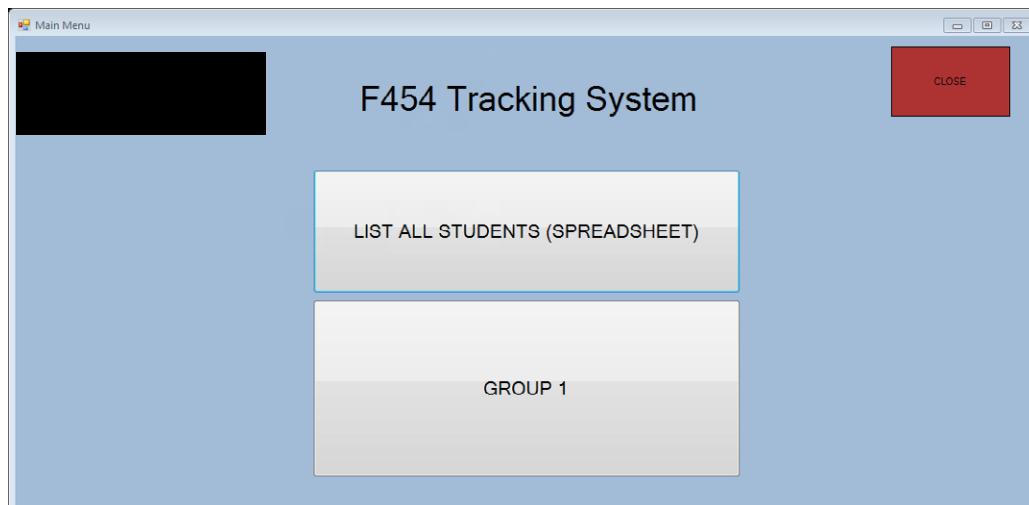
F454 Tracking System.exe
18 mins ago · 266 KB

[Download](#) [Save to my Dropbox](#)

4. Upon the download of the program, a folder will open which has been shown below.



5. Double click on this file and the system will load as shown below:

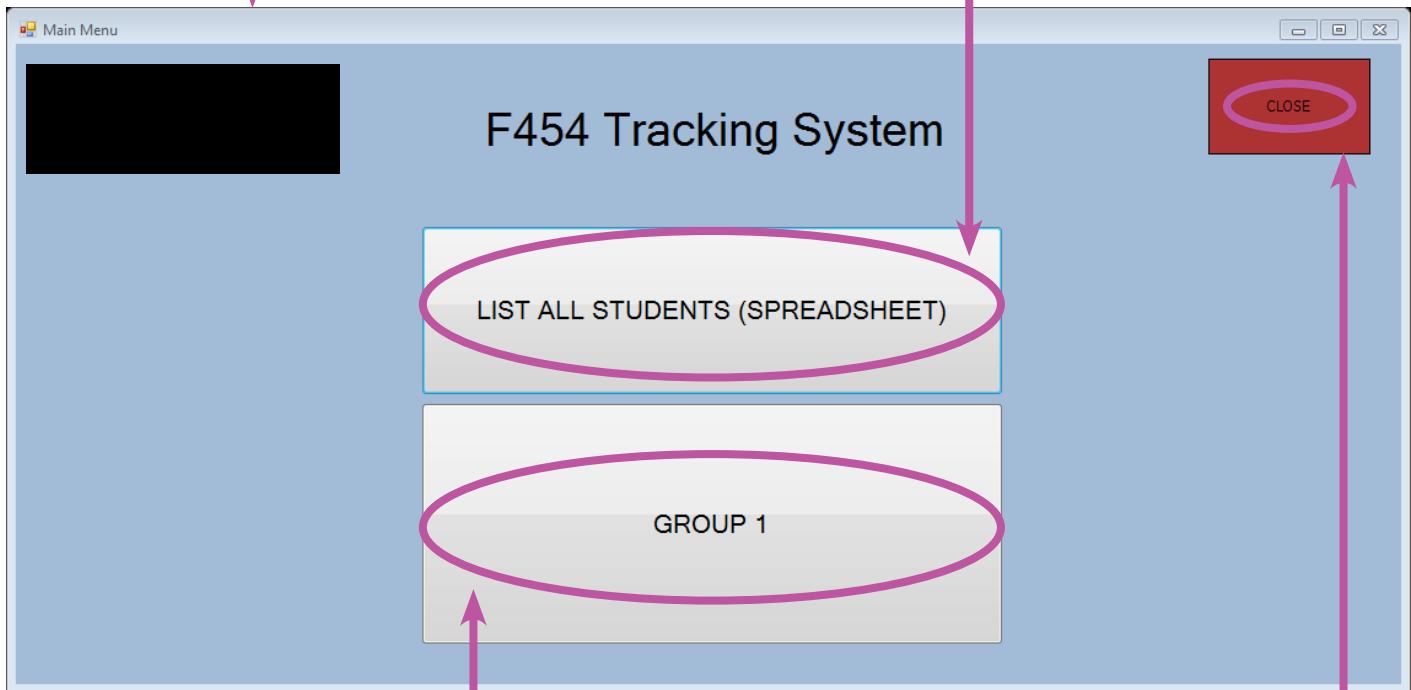


Navigation

Main Menu

This main menu has two options named 'LIST ALL STUDENTS' and 'GROUP 1'.

'LIST ALL STUDENTS' will load a spreadsheet with all the student's details including name, current grade, target grade, percentage and total marks.



GROUP 1 contains the students in the class. This is where you can select students to add grades.

This will close the system

Group 1

Click on the 'Group 1' button and the page shown below will appear:

All the buttons in the main page display each student. The students which have been labelled as 'Student (number)' are students which have not been added.

CLOSE

Updates the page for: Changing student names and highlighting them.

UPDATE

Graph

Open graph

GroupProfile

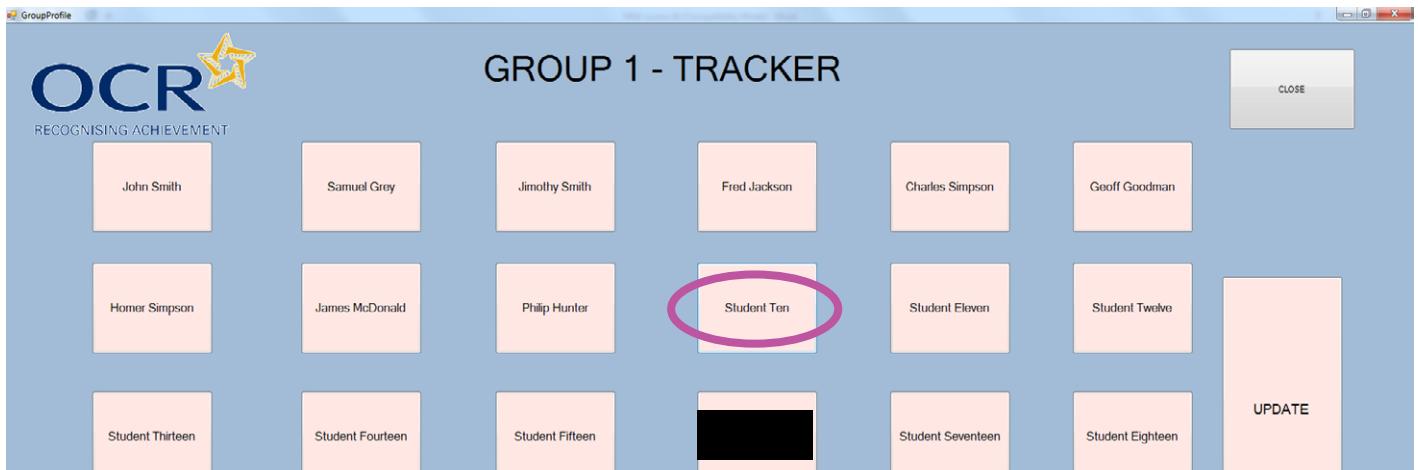
UP 1 - TRACKER

John Smith	Samuel Grey	Jimothy Smith	Fred Jackson	Charles Simpson	Geoff Goodman
Homer Simpson	James McDonald	Philip Hunter	Student Ten	Student Eleven	Student Twelve
Student Thirteen	Student Fourteen	Student Fifteen		Student Seventeen	Student Eighteen
Student Nineteen	Student Twenty	Student Twenty One	Student Twenty Two	Student Twenty Three	Student Twenty Four
Student Twenty Five	Student Twenty Six	Student Twenty Seven	Student Twenty Eight	Student Twenty Nine	Thirty

Creating a Student

The following set of instructions will show you how to create a student.

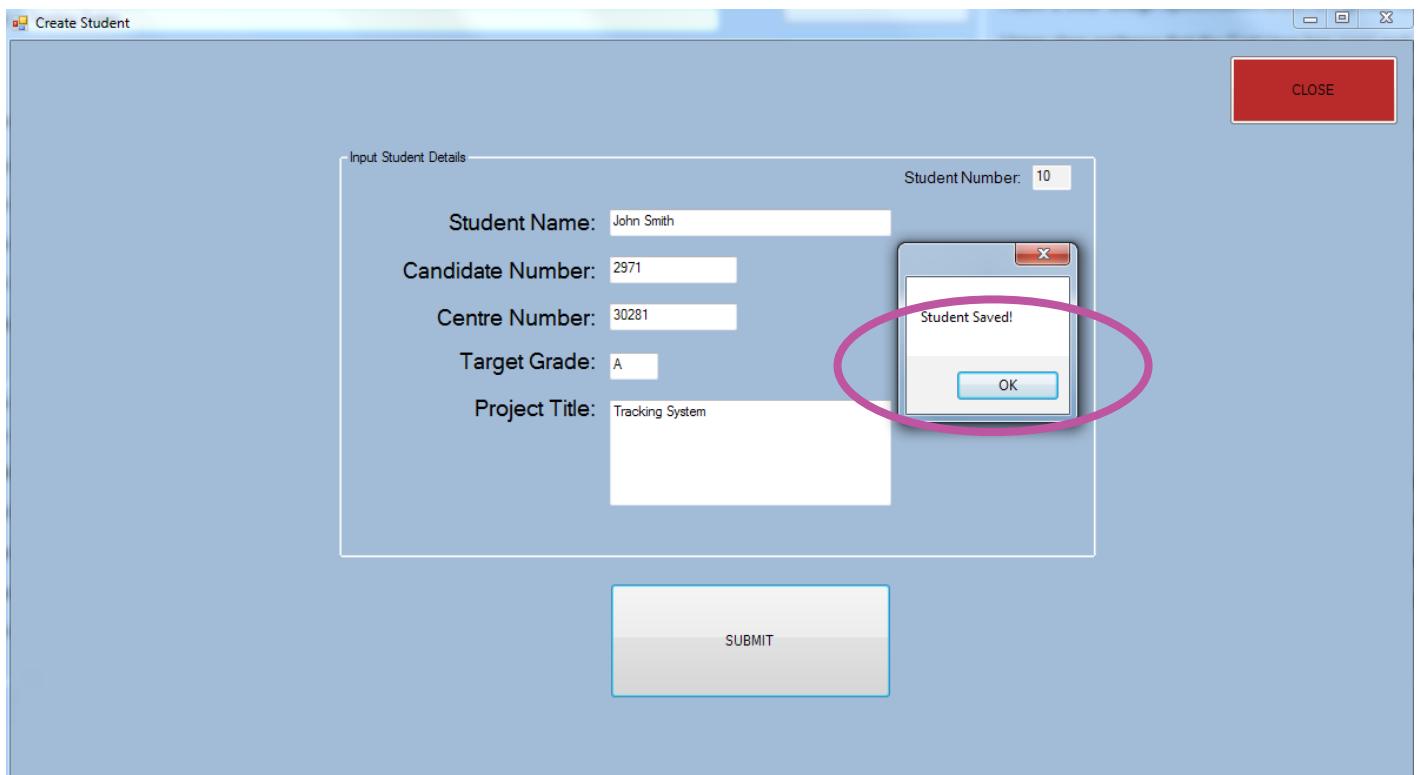
Firstly, click on a button which you would like the student to be in. For this example, student 10 has been chosen.



Upon clicking on the student, the page shown below will open:

A screenshot of a Windows dialog box titled "Create Student". The dialog has a light blue background. At the top left is the title "Create Student". In the top right corner is a red "CLOSE" button. Below the title is a section titled "Input Student Details". Inside this section, there are five input fields: "Student Name:" with a text box, "Candidate Number:" with a text box, "Centre Number:" with a text box, "Target Grade:" with a text box, and "Project Title:" with a large text box. Above the "Project Title" field is a "Student Number:" label with a text box containing the value "10". At the bottom of the dialog is a large grey "SUBMIT" button.

Begin to fill in the details for the student. Below shows an example of what could be entered.



Once the information has been entered, a message box will appear which will tell you that it has been saved.

Error messages – Please ensure you have done the following shown below:

- Used alphabets only for student name
- Candidate name has 4 numbers
- Centre number has 5 numbers
- Target Grade has letters are either: A, B, C, D or E.
- Project Title is more than 6 characters.
- All text boxes must be inputted.

If all the conditions above are met, your data will be accepted. If it does not, then an error message will appear telling you what you have done wrong.

Click 'OK' on the message box and this page will appear:

The screenshot shows a Windows application window titled "CourseSectionsOne". In the top left, there's a form with fields for "Student Number" (10), "Candidate Name" (John Smith), "Candidate Number" (2917), "Centre Number" (30281), "Target Grade" (A), and "Project Title" (Tracking System). To the right of these fields are buttons for "Highlight Student", "Unhighlight Student", "GRAPH", and "EDIT". A red "CLOSE" button is in the top right corner.

Section A1: Problem Definition

- Brief Description of end user (e.g. Firm of Business involved, location, turnover etc)
- Current description of problem
- Current description of methods used or area in need of development
- Clear statement of origins and form of any relevant data (e.g. paper based registration)
- Describe how further research is needed to define a problem and its intent

Total:

Section A2: Investigation and Analysis

- I have included my end user fully in all discussions and have evidence of this
- I have detailed evidence of all interviews, research and observations
- I have fully researched all avenues and opportunities for the problem
- I have evidenced and discussed a full analysis of the problem to be computerised
- I have a detailed user requirements list that is justified and signed by the end user
- I have justified the use of all hardware and software

Total:

Section B1: Design

- I have created a clear set of system objectives (based on User Requirements)
- I have a clear design specification which is logically correct
- I have clear evidence that the End User has seen and agreed all designs
- I have detailed/drawn descriptions of processes and modules
- I have clear definitions of data structures/tables of variables for each module etc.
- My designs could be built by someone else using the software stated

Total:

Section B2: Algorithms

- A complete set of algorithms has been produced
- I have described my algorithms to show they meet the requirements
- I have evidence that I have tested my algorithms for functionality
- I have shown that my algorithms produce a complete solution

Total:

GRAPH

- My test strategy is detailed and covers all types of testing and test data
- My test strategy covers all aspects of my system (input, processing, output)
- I have covered Normal, Extreme and Abnormal test data
- I have included how my end user will test the system (creating test examples)

Total:

Comments

SAVE ALL

Next Page

This contains all the information of the student that you have entered

Highlights and un-highlights student names in the main group page.

Upon clicking next page, this page will open:

CourseSectionsTwo

Student Number:	10	CLOSE																					
Candidate Name:	John Smith																						
Candidate Number:	2917																						
Centre Number:	30281																						
Target Grade:	A																						
Project Title:	Tracking System																						
		SHOW GRAPH																					
		EDIT																					
<p>Section C1: Software Development</p> <table border="1"> <tr> <td>I have clear and complete evidence showing system development</td> <td>0</td> <td>Comments</td> </tr> <tr> <td>I have used Alpha testing and refined my system, with discussion</td> <td>0</td> <td></td> </tr> <tr> <td>I have all code and it is well annotated and commented</td> <td>0</td> <td></td> </tr> <tr> <td>Each module used is described and completed</td> <td>0</td> <td></td> </tr> <tr> <td>Links between each part of the solution and clearly explained</td> <td>0</td> <td></td> </tr> <tr> <td>The solution is fully developed from the initial designs</td> <td>0</td> <td></td> </tr> <tr> <td>Total:</td> <td></td> <td></td> </tr> </table>			I have clear and complete evidence showing system development	0	Comments	I have used Alpha testing and refined my system, with discussion	0		I have all code and it is well annotated and commented	0		Each module used is described and completed	0		Links between each part of the solution and clearly explained	0		The solution is fully developed from the initial designs	0		Total:		
I have clear and complete evidence showing system development	0	Comments																					
I have used Alpha testing and refined my system, with discussion	0																						
I have all code and it is well annotated and commented	0																						
Each module used is described and completed	0																						
Links between each part of the solution and clearly explained	0																						
The solution is fully developed from the initial designs	0																						
Total:																							
<p>Section D: Documentation</p> <table border="1"> <tr> <td>I have spell checked and grammar checked my work - it is clear from errors</td> <td>0</td> <td>Comments</td> </tr> <tr> <td>Documentation is well presented, logical, headers, footers, page numbers and contents</td> <td>0</td> <td></td> </tr> <tr> <td>I have created a user guide including: Installation, Use, Troubleshooting and Backup</td> <td>0</td> <td></td> </tr> <tr> <td>Where possible I have on-screen help and tool tips etc.</td> <td>0</td> <td></td> </tr> <tr> <td>My use of technical language is accurate and at an appropriate level</td> <td>0</td> <td></td> </tr> <tr> <td>Total:</td> <td></td> <td></td> </tr> </table>			I have spell checked and grammar checked my work - it is clear from errors	0	Comments	Documentation is well presented, logical, headers, footers, page numbers and contents	0		I have created a user guide including: Installation, Use, Troubleshooting and Backup	0		Where possible I have on-screen help and tool tips etc.	0		My use of technical language is accurate and at an appropriate level	0		Total:					
I have spell checked and grammar checked my work - it is clear from errors	0	Comments																					
Documentation is well presented, logical, headers, footers, page numbers and contents	0																						
I have created a user guide including: Installation, Use, Troubleshooting and Backup	0																						
Where possible I have on-screen help and tool tips etc.	0																						
My use of technical language is accurate and at an appropriate level	0																						
Total:																							
<p>Section E1: Evaluation - Object Success</p> <table border="1"> <tr> <td>I have commented on all aspects of the project and the success and issues in each</td> <td>0</td> <td>Comments</td> </tr> <tr> <td>I have linked this back with Page Numbers to highlight my points</td> <td>0</td> <td></td> </tr> <tr> <td>I have discussed any issues I had in each section fully and justified these</td> <td>0</td> <td></td> </tr> <tr> <td>Total:</td> <td></td> <td></td> </tr> </table>			I have commented on all aspects of the project and the success and issues in each	0	Comments	I have linked this back with Page Numbers to highlight my points	0		I have discussed any issues I had in each section fully and justified these	0		Total:											
I have commented on all aspects of the project and the success and issues in each	0	Comments																					
I have linked this back with Page Numbers to highlight my points	0																						
I have discussed any issues I had in each section fully and justified these	0																						
Total:																							
<p>Section E2: Evaluation - User Response</p> <table border="1"> <tr> <td>I have evidence I have discussed faults with end user and how they can be improved</td> <td>0</td> <td>Comments</td> </tr> <tr> <td>OR discussed a successful project in agreement with end user + requirements and there are no known faults</td> <td>0</td> <td></td> </tr> <tr> <td>Total:</td> <td></td> <td></td> </tr> </table>			I have evidence I have discussed faults with end user and how they can be improved	0	Comments	OR discussed a successful project in agreement with end user + requirements and there are no known faults	0		Total:														
I have evidence I have discussed faults with end user and how they can be improved	0	Comments																					
OR discussed a successful project in agreement with end user + requirements and there are no known faults	0																						
Total:																							
<p>Section E3: Evaluation - Extensions</p> <table border="1"> <tr> <td>Good and Bad points of the system are fully discussed</td> <td>0</td> <td>Comments</td> </tr> <tr> <td>Limitations and how you could develop extensions are fully discussed</td> <td>0</td> <td></td> </tr> <tr> <td>Total:</td> <td></td> <td></td> </tr> </table>			Good and Bad points of the system are fully discussed	0	Comments	Limitations and how you could develop extensions are fully discussed	0		Total:														
Good and Bad points of the system are fully discussed	0	Comments																					
Limitations and how you could develop extensions are fully discussed	0																						
Total:																							
<input type="button" value="SAVE ALL"/>		<input type="button" value="Previous"/>																					

This page is similar to the previous first page. The only difference is that there are different criteria. All other features work the same.

Editing Student Information

To edit a student's details, click on 'Edit' which can be found on the first page of the criteria as shown below:

The screenshot shows a software interface for managing student projects. At the top, there are input fields for Student Number (10), Candidate Name (John Smith), Candidate Number (2917), Centre Number (30281), Target Grade (A), and Project Title (Tracking System). To the right of these fields are three buttons: 'Highlight Student', 'Unhighlight Student', and 'GRAPH'. Below these is a large 'EDIT' button, which is circled in pink. To the left of the 'EDIT' button is a section titled 'Section A1: Problem Definition' containing five input fields for problem descriptions and a 'Comments' area. To the right of the 'EDIT' button is a section titled 'Section B1: Design' with several statements about design objectives. Further down on the left is a section titled 'Section A2: Investigation and Analysis' with a statement about including the end user. On the right side, there is another section titled 'Section B2: Algorithms' with statements about algorithm development.

After clicking the edit button, this page will appear. This will now allow you to overwrite any information already entered for this student and will replace it with information that will be entered.

The screenshot shows a 'Create Student' dialog box. It contains a form titled 'Input Student Details' with fields for Student Name (John Smith), Candidate Number (2917), Centre Number (30281), Target Grade (A), and Project Title (Tracking System). Above the form, the Student Number is listed as 10. In the top right corner of the dialog box is a red 'CLOSE' button. At the bottom of the dialog box is a large grey 'SUBMIT' button.

Following this, you can now enter any details you wish for the student and it will automatically overwrite with the new information.

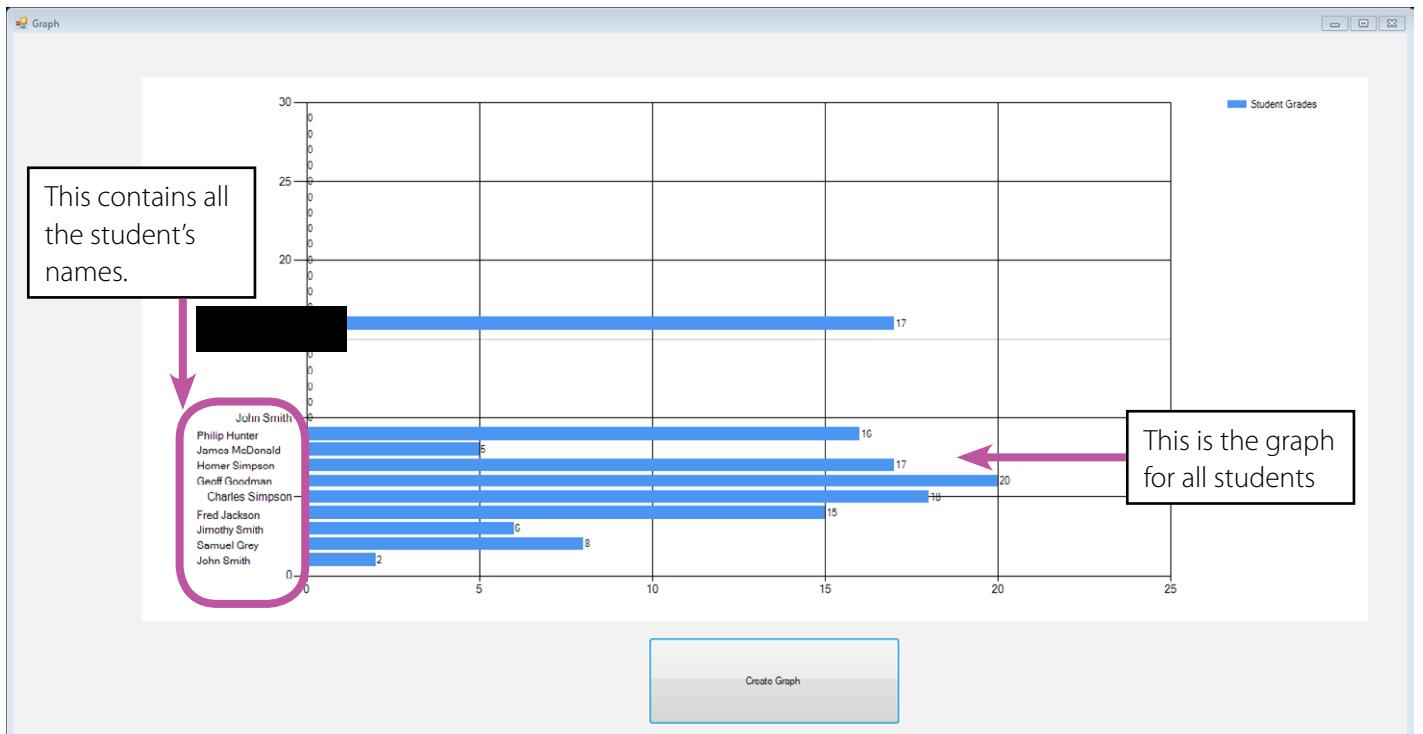
Viewing Graph

Graph feature represents each student and their grade. This will allow you to compare how much progress everyone is making.

Firstly, click on 'Graph'.

The screenshot shows the 'CourseSectionsOne' application window. At the top, there are input fields for 'Student Number' (10), 'Candidate Name' (John Smith), 'Candidate Number' (2917), 'Centre Number' (30281), 'Target Grade' (A), and 'Project Title' (Tracking System). To the right of these fields are 'Highlight Student' and 'Unhighlight Student' buttons, and a large 'GRAPH' button which is circled in pink. Below these are 'EDIT' and 'PRINT' buttons. The main area is divided into sections A1 and A2. Section A1 contains fields for problem definition, current descriptions, and methods, each with a numerical scale from 0 to 5. Section A2 contains a statement about including the end user, followed by a 'Comments' field. To the right of the main form are two columns of text under 'Section B1: Design' and 'Section B2: Algorithms', both starting with 'I have created...' statements.

Upon clicking 'Graph' this page will open:



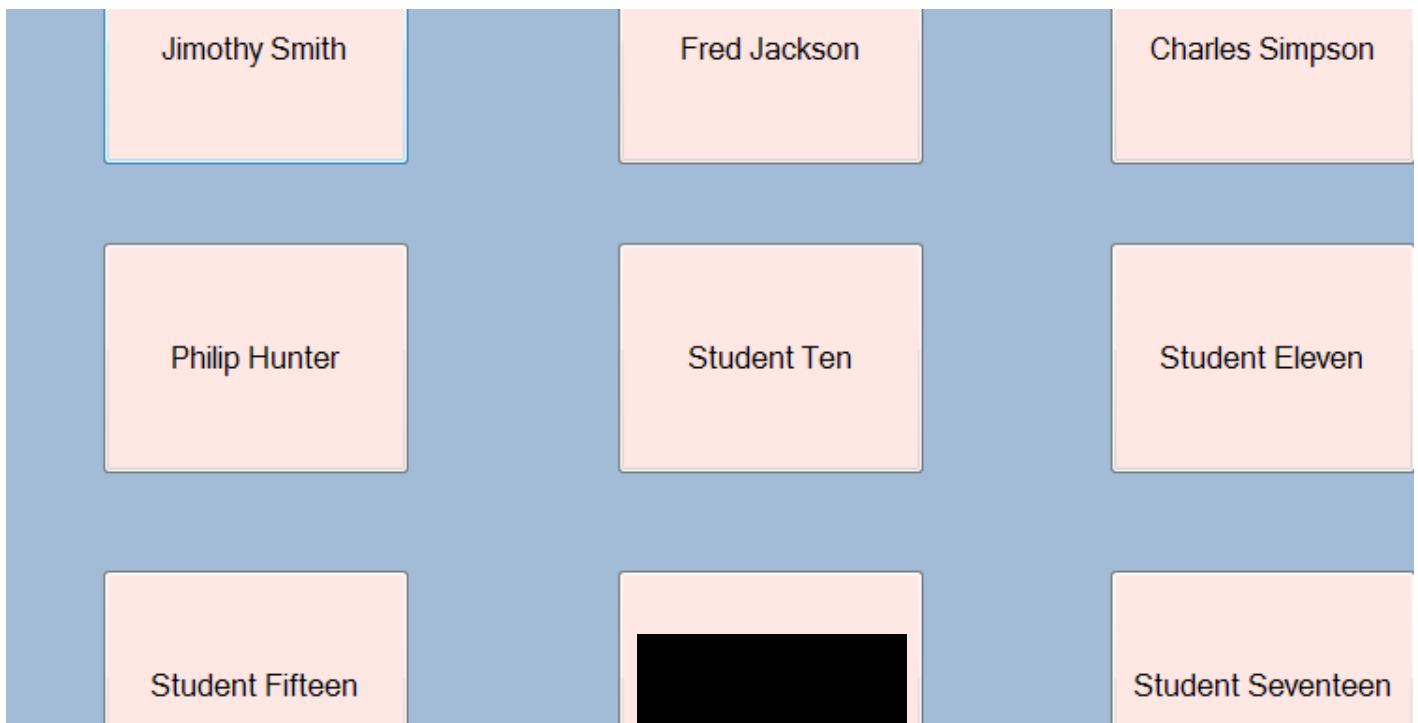
In the instance that you see an empty graph, click 'Create Graph' and it will generate a graph for all the students.

Highlighting / Un-Highlighting Students

In order to highlight or un-highlight a student, simply click on the button shown below.

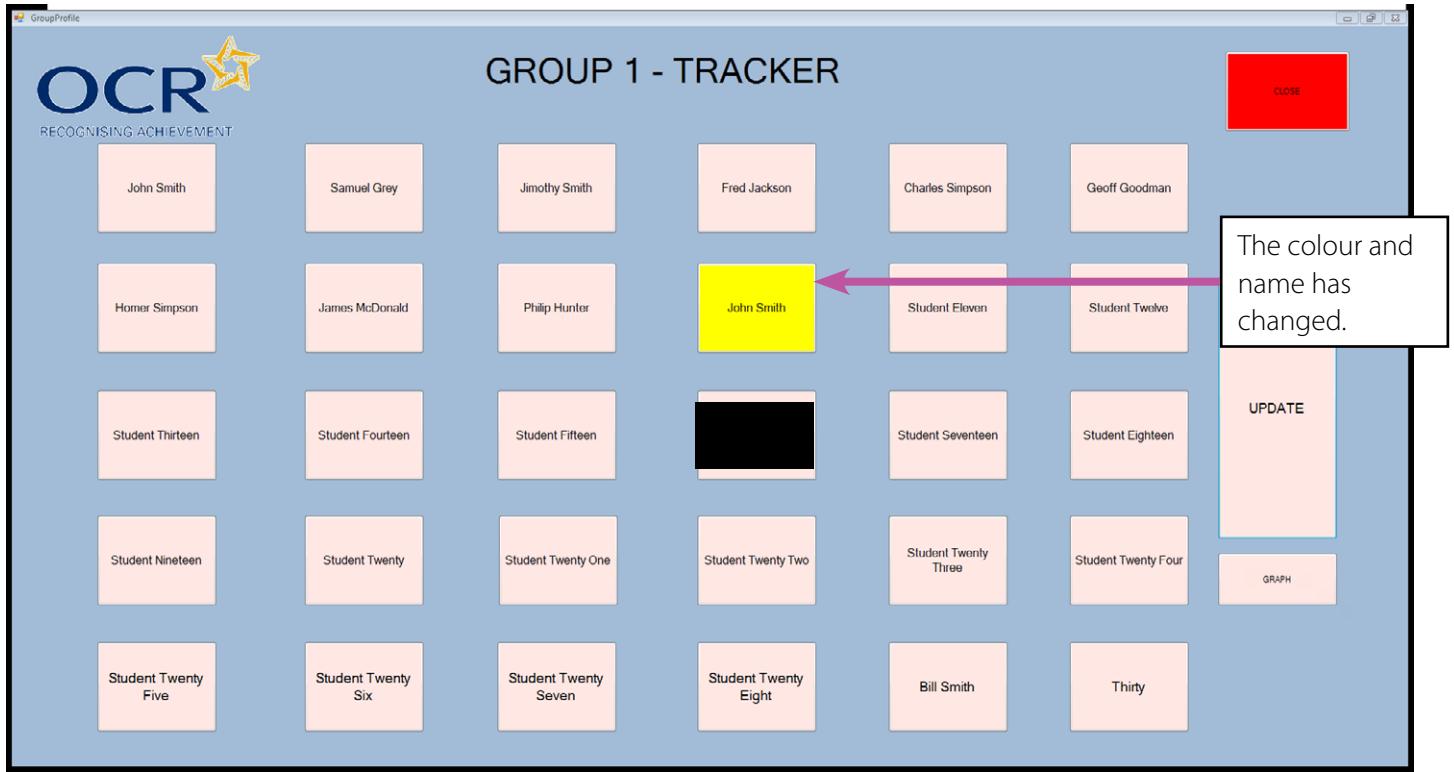
When clicking this button, it will change the colour of the student's button on the main group section.

Close or minimize the page and view the 'Group 1' page. When going back to the page, you will notice that the button name is still 'Student Ten', as shown in the image below.



Updating

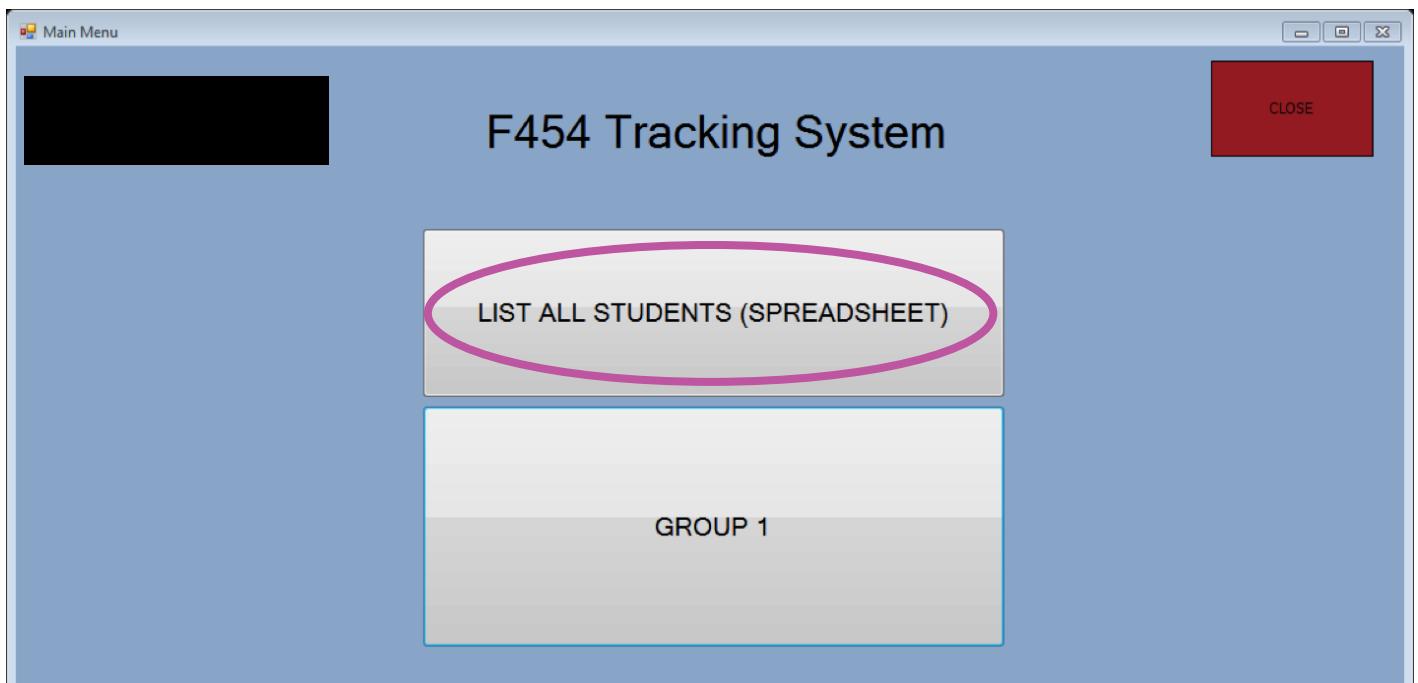
In order to update, you must click on the update button. This button will change the names on buttons and change the colour of the highlight if the user has selected the option.



You are not always required to do update. If you close the Group 1 page and reload it, the information will automatically update without the need of clicking the 'Update' button.

Viewing the Spreadsheet

In order to view the spreadsheet, go to the Main Menu and click 'List All Students (Spreadsheet)' as shown below.



Upon clicking this, another application will appear. This is Microsoft Excel, a software used to view data in a spreadsheet form.

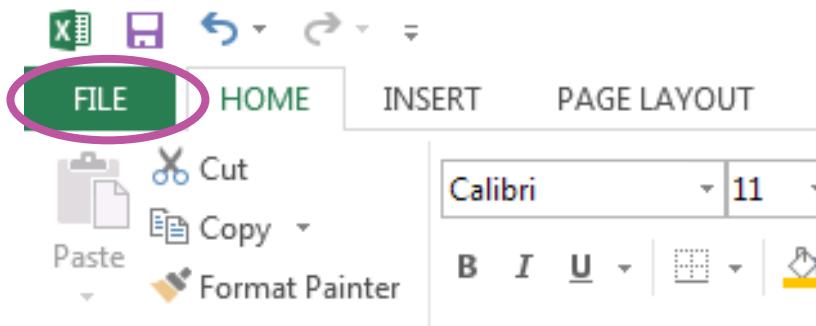
On the spreadsheet contains all the information on every student you have entered.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Candidate Name	A1	A2	B1	B2	B3	C1	C2	D1	E1	E2	E3	TOTAL	%	GRADE	TARGET
2	John Smith	2	0	0	0	0	0	0	0	0	0	0	0	2	2.5	F
3	Samuel Grey	2	2	2	1	1	0	0	0	0	0	0	0	8	10	b
4	Timothy Smith	3	0	0	2	1	0	0	0	0	0	0	0	6	7.5	F
5	Fred Jackson	3	3	0	3	3	0	0	0	0	0	0	0	12	15	b
6	Charles Simpson	3	3	3	4	5	0	0	0	0	0	0	0	18	22.5	F
7	Geoff Goodman	0	0	0	0	0	2	6	3	3	3	3	0	20	25	F
8	Homer Simpson	3	6	3	4	4	0	0	0	0	0	0	0	20	25	F
9	James McDonald	1	1	1	1	1	0	0	0	0	0	0	0	5	6.25	B
10	Philip Hunter	3	3	4	3	3	0	0	0	0	0	0	0	16	20	F

Each of the grades have been highlighted in a different colour to represent the progress of the student. For instance, if the student has reached a high amount of marks, then it will be represented as green. If they have an average amount of marks, it will be coloured as orange. Finally, if it is red, it means that the student is not achieving the required amount of marks.

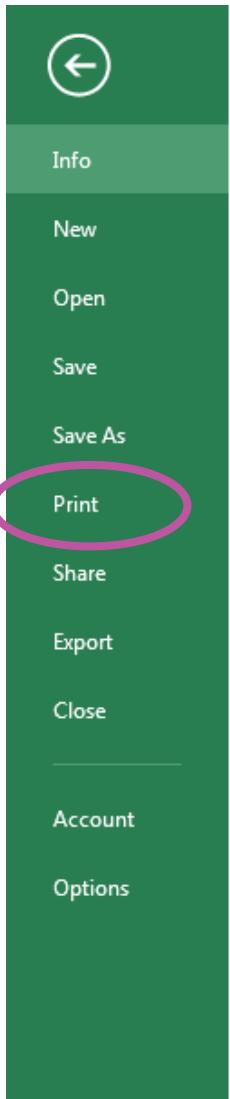
Printing

To print the spreadsheet, click on the top left button named 'File'.



After you click on 'File', the page below will open.

Click on 'Print' and another page will appear.

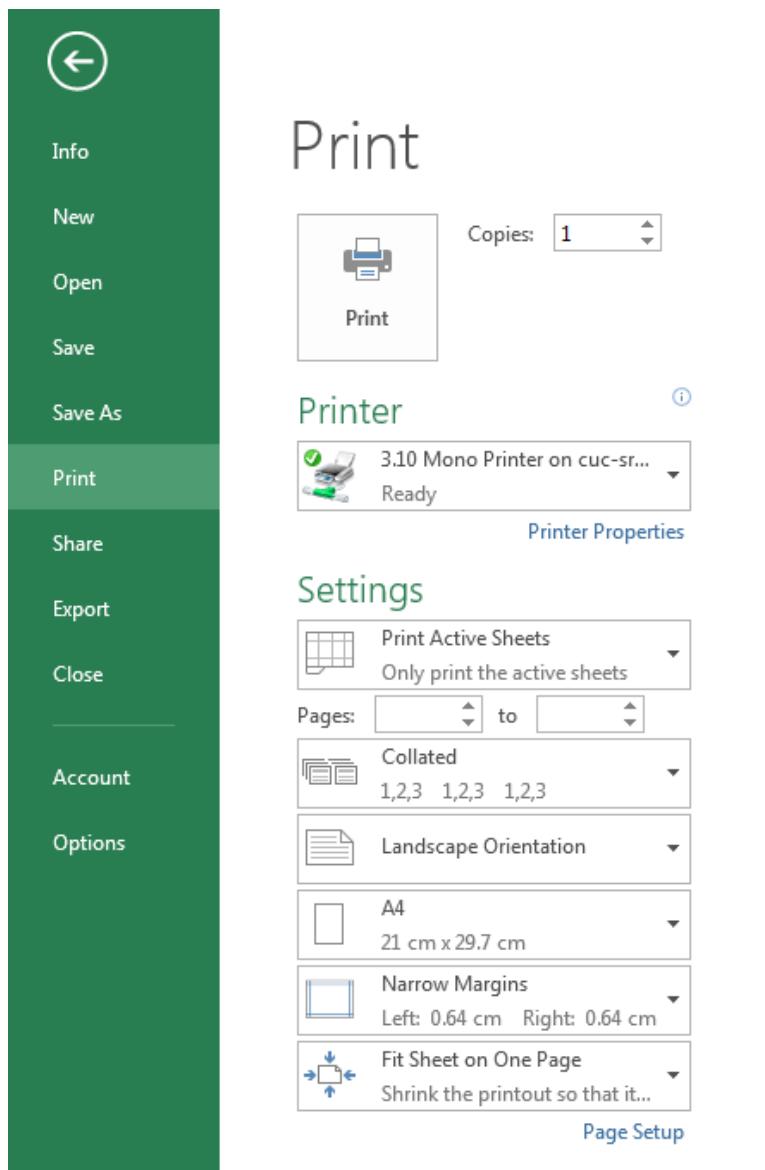


The screenshot shows the 'Info' ribbon tab selected. On the left, a vertical menu lists options: Info, New, Open, Save, Save As, Print (which is circled in pink), Share, Export, Close, Account, and Options. To the right of the menu, there are four sections: 'Protect Workbook', 'Inspect Workbook', 'Versions', and 'Browser View Options'. Each section has an icon and a brief description.

- Protect Workbook**
Control what types of changes people can make to this workbook.
- Inspect Workbook**
Before publishing this file, be aware that it contains:
 - Author's name and absolute path
 - Content that people with disabilities find difficult to read
- Versions**
There are no previous versions of this file.
- Browser View Options**
Pick what users can see when this workbook is viewed on the Web.

After clicking 'Print', the page that is shown will provide details of the printer you are connected to and general settings such as choosing the colour and the amount of copies. This all depends on your preference.

In order to print the whole spreadsheet on one page, you must select the following settings:



Once this has been selected, click 'Print' in order to print.

Troubleshooting

Most of the errors that you have may be just general validation errors which will prompt you what must be done. However, there are some errors that may be resulted with the computer.

Download and Installation Issues

Issue	Possible cause	Solution
This webpage is not available	Dropbox may be down for maintenance.	Try accessing the website at a later time.
DNS lookup failed	You may not be connected to the internet or you may not have a secure connection.	Contact your internet service provider.
Error 522, connection timed out	Dropbox may be facing a heavy amount of users trying to connect to the website.	Try accessing the website at a later time.
Program will not save	There may not be enough space in your hard drive.	Ensure that you have enough space. If you do not, consider deleting any unnecessary data from your computer.

General Issues

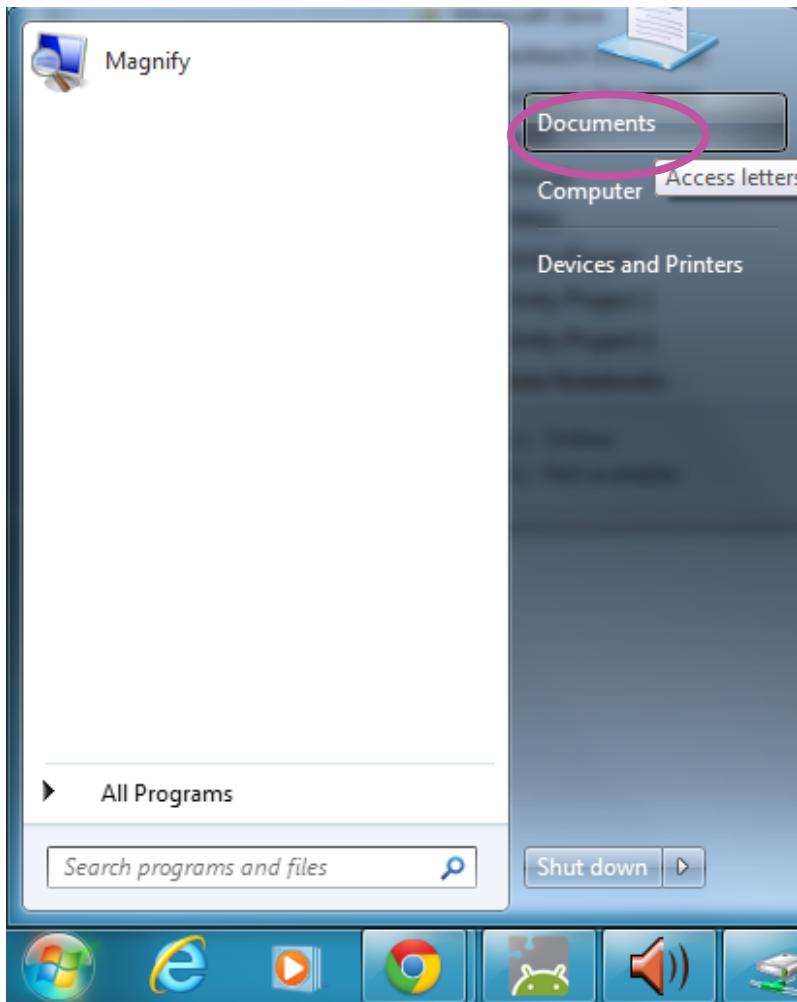
Issue	Possible cause	Solution
Clicking spreadsheet does not open.	You may not have installed Microsoft Excel properly.	Try reinstalling Microsoft Excel.
Not enough paper	You do not have enough paper in your printer.	Add more paper into your printer.
Cannot print due to insufficient ink	There is not enough ink in your printer.	Add ink to your printer or consider printing in black and white if you do not require it in colour.
System slows down while saving.	You may not have sufficient RAM and processing power.	If it a major issue, consider purchasing RAM with more memory and replace your current processor with faster gigahertz.

Backup

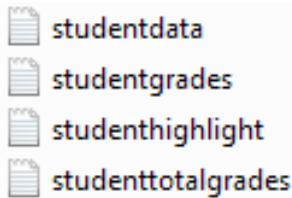
Upon launching the application, there will automatically be the creation of five text files. Each of these text files contain data of the students in the class.

In order to locate these files, go to your main 'Documents' page. There will be five files, named 'studentdata', 'studentgrades', 'studenthighlight', 'studenttotalgrades' and 'usercomments'.

To navigate to your documents, click 'Start'. Then select 'Documents' as shown in the screenshot below:



Once documents is open, scroll down until you find the textfiles, it may appear like this:



If you cannot find the textfiles, then search the documents in the top right side of the page, shown below:



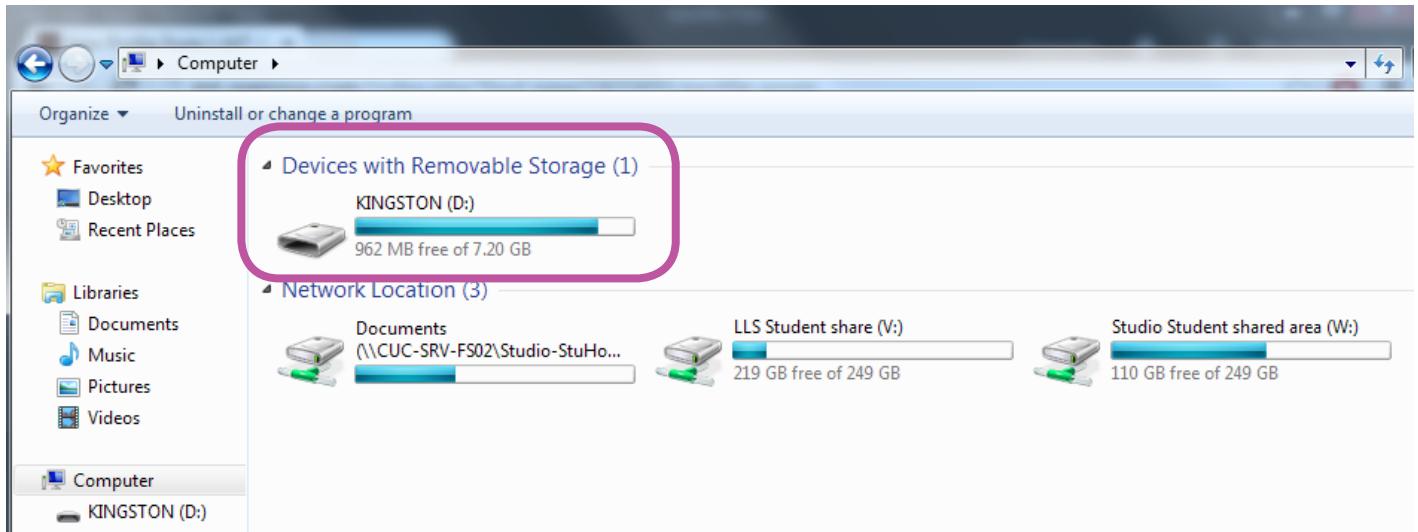
Once you have located the text files, there are several ways to save your data which will be shown below.

Storage Device

You can back up the data using a storage device such as a flash drive or CD-RW. Firstly, to do this, insert your device into the system.

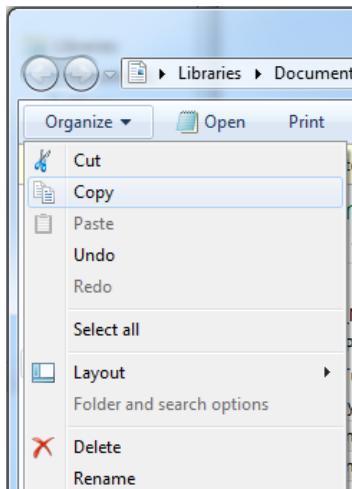
In this example, a flash drive has been used.

Firstly, navigate to 'Computer' where you will find a section in the page titled 'Devices with Removable Storage'.

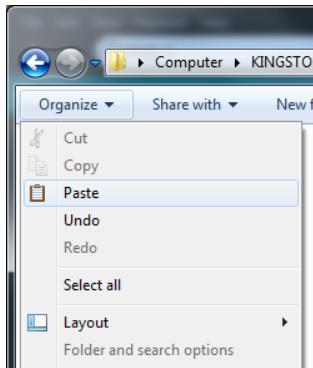


Click on the device you wish to back up your files.

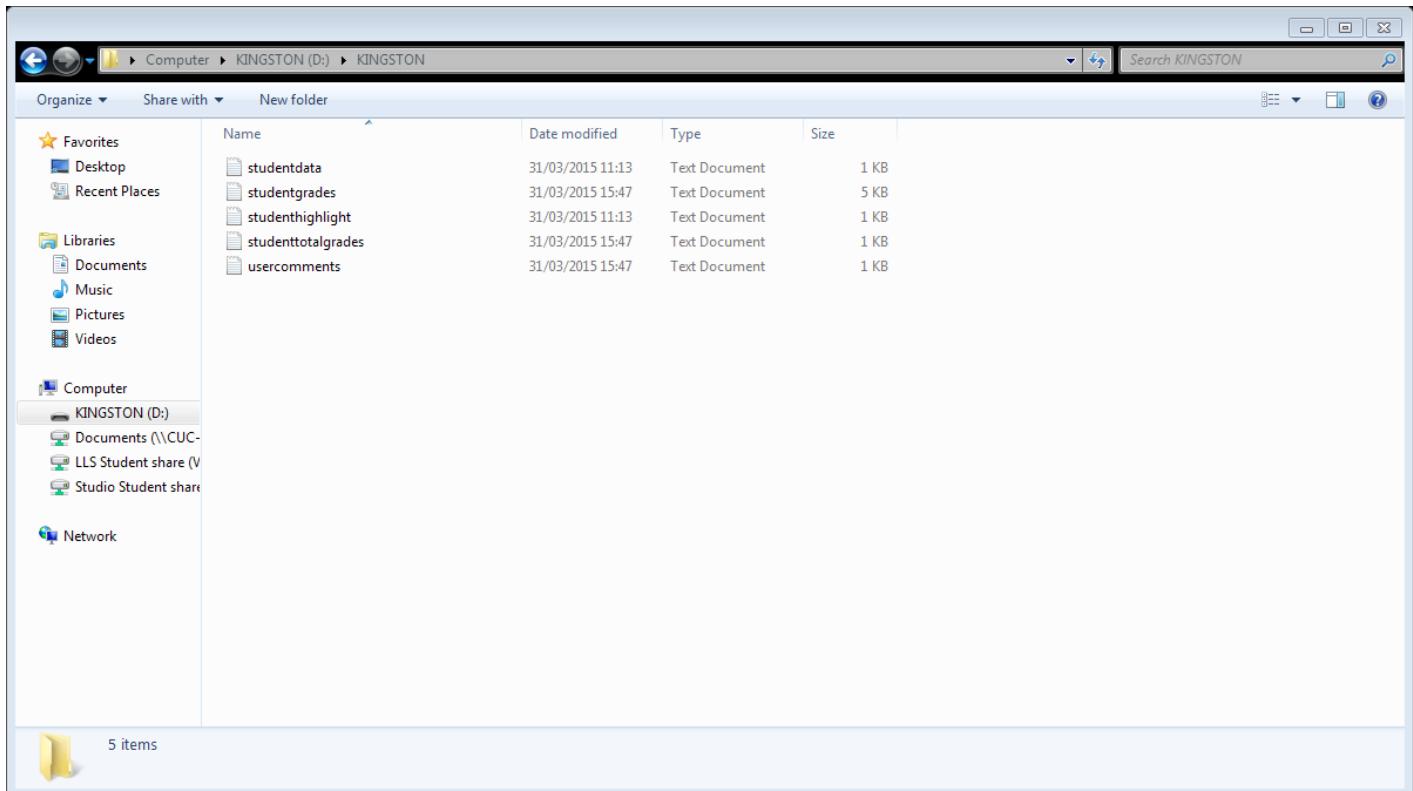
After you have done this, go back to your documents and highlight the text files. Once all of them are highlighted, right click and click 'Copy'.



After you have copied the text files, go to your flash drive and paste the contents in the folder. To do this, right click and select 'Paste'.



Once you have done this, all the contents will be pasted and the page should appear like this:



You have now backed up your files.

If you would like extra assurance that your files have been backed up, you have the option to store it in cloud storage areas such as Google Drive or Dropbox.

E1: Evaluation: Object Success

To determine whether or not I have met the user's requirements, I have set up a table providing information about the requirements.

Requirement	Explanation of why the requirement was met
The screen size will be (1920x1080pixels) as it is the monitor size if the school PCs that the system will be run on.	Every section has 1920x1080 dimensions apart from the main menu as it is not necessary. This can be seen on page 64.
Green indicates a student is on target Yellow is a student with one grade below Orange indicates the student is two grades below. Red indicates student is not on target and is three grades or below.	All the colours have been added to the spreadsheet. This can be seen on page 85.
Background colour of the system will be plain white or very light blue.	Background colour is light blue. This can be seen on page 85.
Highlighted students will be yellow.	The user changed their mind as grey was not clear enough. This can be seen in acceptance testing results on page 100 and page 116 shows the yellow buttons. In response to this change, the buttons were changed to yellow.
An exit button will be displayed on each page to exit to the main menu. Exit button colour will be red.	Yes, each button has a red exit button. This can be seen in page 106, 107, 108 and 109.
The client will have the ability to enter student's marks by selecting the spreadsheet system. Expanding onto this, they can also select the student's profile and edit the details in there.	This has been met as the spreadsheet can be edited by the user. This can be seen on page 116.
Client will have the ability to add notes to students using the note box provided on the student profile.	Client can add notes and it does save. This can be seen on page 77.
Client can easily edit data in two ways. They can open a student's profile and edit it there or can access the spreadsheet and edit the marks form there.	This has been met as the user can edit both the system and the spreadsheet. Seen on page 116 for spreadsheet and 110 for system.
Client will have the ability to highlight certain students who teachers may be worrying about. They will be able to do this by selecting the dropdown box.	The client can highlight students and a yellow colour will represent them. This can be seen on page 115.
The total marks is 80, therefore the percentage of completion will be based on how many marks a student gets per section.	The formula has been used to calculate the percentage. Page 96 shows the tests for the percentage.
Formula: Marks / 80 * 100	
<ul style="list-style-type: none"> • For total marks on each sub section, this will be: <ul style="list-style-type: none"> – Green – On target – Amber – One grade below – Red – Two or more grades below 	All of the colours have been added. This can be seen on page 116.
A database will be created so each student's progress saves. I can use the feature which is already on Visual Studio.	I have decided to use textfiles to create the database. This has saved the students data and therefore has met the requirement. This can be seen on page 120.

Requirement	Explanation of why the requirement was met
Every time a grade is inputted, the graph will increment or decrement a point (depending on the grade) on the graph so it joins together to the right grade.	This has been met as the graph displays correct results. This can be seen on page 113.
<ul style="list-style-type: none"> • My client will be able to print the tracker for a student so they can see their current progress. This will include, as stated above: <ul style="list-style-type: none"> – Name, candidate number, name and project title will be on the top of the page. – Section of coursework, marks and percentage of completion will be the main page. – Whether the student is on target or not will be displayed under each section. 	The client can print with the use of the excel printing. I have added instructions on how to print in the user guide. This can be seen on page 117.

Issues – Module 1 & Module 3

Module 1 is the menu that is opened as soon as the system starts. Although there were not much graphical features on the main menu, this did result in a lot of issues. The reason I merged both Module 1 & 3 to this description is because the programming for Module 3 was done in Module 1.

The main issue was the spreadsheet. Opening the spreadsheet did not have any problems, but learning to import the information I gathered from the textfiles into the spreadsheet took a while to learn. There were also issues with correctly linking each student's information on the spreadsheet as at times I was continuously forgetting which row and column the grades should have been in.

Issues – Module 2

Module 2 was the most time consuming in comparison to both Module 1 and 3 as there were many errors which can be seen in the software testing section. Module 2 firstly had the main problem of not saving correctly. The save file was continuously incrementing itself, meaning that some students grades were getting mixed up with others. In addition to this, I had indexed the arrays in a wrong format, meaning that I was continuously confused on which array to use. These were the two main problems in regards to grades. Other than that, there were also problems with creating a visual studio graph as previously I had never used this tool. The issue with the graph was that the student's names were not being displayed properly and instead were only posting results of every 5 students. Therefore, I had to use labels instead in order to list all the students.

Overall Summary

The project met the client's requirements and although there were a few issues that the client brought up, they were all amended to fit the user's requirements.

Furthermore, the system has been completely tested through both its input and output to make sure that the correct information is being displayed.

E2: User Response

To get the final input from the user regarding the change implemented, I created a questionnaire to ensure that all the parts of the system is correct.

Question	Answer / Comments
Are you happy with the changes created?	The problems I had previously were solved and the criteria were met. The changes I saw were highlighted and I am pleased with how it is.
What new feature would you like to see implemented in the future?	I would like to see a log-in page where students can view their progress. I would also like a cover sheet which can be used at the end of a project and finally encrypting student data.
Are you happy that the requirements have been fulfilled?	Yes
Do you think the user guide is helpful?	Yes, maybe some more on screen help and ability to hover over sections. Even an online guide would be nice.
Is there anything you would like to see in the user guide that isn't currently shown?	As I said, just ability to hover over sections online would be good.
Do you think the colours used in the system is suitable?	Colour scheme is fine.
Is there any problems with the current system?	No obvious problems within the system

SIGNATURE: _____

DATE: _____

Response to Questionnaire

The user has shown that he is happy with the final outcome of the system and has deemed all the requirements met. The previous problems that the user had highlighted during acceptance testing (shown on page 96) has been corrected and is fully functional. Therefore, the only changes with the system that is needed is future implementations of the user's wishes. These wishes will be explained upon in the next section.

E3: Evaluation: Extensions

To finally conclude with the system, I have wrote the good and bad points of the system and have discussed the limitations and how I could develop the system in the future.

Good and Bad Points

Good Points

- After talking to my client about the state of the system, most of the systems requirements were fulfilled and there were only some minor changes that had to occur such as the change of buttons and colours.
 - In response to the change, I have successfully managed to amend the requirements as shown on page 99.
- I have implemented an addition which was suggested by another teacher (highlight feature).
- The designs create originally are very similar to the actual system.
- The system itself is user friendly and very easy to navigate around.
- Through extensive testing, there is no bugs or glitches appearing. This can be seen on page 84.
- All the pages and buttons have a similar theme, light blue background and red buttons.
- The information on the spreadsheet loads correctly.

Bad Points

- Although there are no errors in the final system, there were errors when given to the client which should have been checked over before it was going through acceptance testing.
- There were some additions that could have been implemented if there were time however there were too many bugs to fix during the software testing phase.

Limitations and Improvements

Comments

There are some limitations to the system which can be developed further. The comment section only has the ability to save so that the client can refer to it at a later time. This feature can be developed into an email option where the client can click a button which will automatically send an email to the teacher to tell them of any comments by the head of department. This can save a lot of time for the client as they will no longer need to go through the time of having to find the email and needing to create a new email through the website. This also does mean that a new addition to the 'create a student' page which means that the client needs to add each school email address in.

Cover sheet

My client requested a cover sheet as a desirable addition to the system but this was not implemented due insufficient time. This can be a future improvement system which allows the user to click a button to open a cover sheet where all the information on the database can be imported onto this page.

Log-in system

Lastly, this is a further advancement to the system which can be implemented in the future. This feature will give a login details for each user who is accessing the system. There will be two different set of permissions one for teachers and one for head of department. This was something I wanted to do but I was not allowed to have access to the shared staff area for security issues. This meant that I could not develop the system properly.



We'd like to know your view on the resources we produce. By clicking on the 'Like' or 'Dislike' button you can help us to ensure that our resources work for you. When the email template pops up please add additional comments if you wish and then just click 'Send'. Thank you.

If you do not currently offer this OCR qualification but would like to do so, please complete the Expression of Interest Form which can be found here: www.ocr.org.uk/expression-of-interest

OCR Resources: the small print

OCR's resources are provided to support the teaching of OCR specifications, but in no way constitute an endorsed teaching method that is required by the Board and the decision to use them lies with the individual teacher. Whilst every effort is made to ensure the accuracy of the content, OCR cannot be held responsible for any errors or omissions within these resources. We update our resources on a regular basis, so please check the OCR website to ensure you have the most up to date version.

© OCR 2015 – This resource may be freely copied and distributed, as long as the OCR logo and this message remain intact and OCR is acknowledged as the originator of this work.

OCR acknowledges the use of the following content:
Square down and Square up: alexwhite/Shutterstock.com

Please get in touch if you want to discuss the accessibility of resources we offer to support delivery of our qualifications:
resources.feedback@ocr.org.uk

We will inform centres about any changes to the specification. We will also publish changes on our website. The latest version of our specification will always be the one on our website (www.ocr.org.uk) and this may differ from printed versions.

Copyright © OCR 2015. All rights reserved.

Copyright

OCR retains the copyright on all its publications, including the specifications. However, registered centres for OCR are permitted to copy material from this specification booklet for their own internal use.

ocr.org.uk/alevelreform

OCR customer contact centre

General qualifications

Telephone 01223 553998

Facsimile 01223 552627

Email general.qualifications@ocr.org.uk

OCR is part of Cambridge Assessment, a department of the University of Cambridge. For staff training purposes and as part of our quality assurance programme your call may be recorded or monitored. © OCR 2015 Oxford Cambridge and RSA Examinations is a Company Limited by Guarantee. Registered in England.

Registered office 1 Hills Road, Cambridge CB1 2EU. Registered company number 3484466. OCR is an exempt charity.

A
A DIVISION OF
CAMBRIDGE ASSESSMENT

