

# Analysis

## Contents

Analysis .....	1
1. Introduction .....	2
1.1. Client identification.....	2
1.2. Define the current system.....	2
1.3. Describe the problems.....	2
1.4. Section appendix .....	3
2. Investigation .....	4
2.1. The current System .....	4
2.2. The proposed system.....	9
3. Objectives.....	15
3.1. General objectives.....	15
3.2. Specific objectives .....	15
3.3. Core objectives .....	15
4. E-R Diagrams and descriptions .....	16
4.1. E-R Diagram.....	16
4.2 Entity Descriptions.....	16
5. Object analysis.....	17
5.1. Object listing.....	17
5.2. Relationship diagrams .....	18
5.3. Class definitions.....	20
7. Constraints.....	22
7.1.Hardware .....	22
7.2. Software.....	22
7.3. Time .....	22
7.4. User knowledge .....	22
7.5. Access restrictions.....	22
8.Limitations .....	23
8.1. Areas which should not be included in computerisation.....	23
8.2. Areas considered for future computerisation.....	23
9. Solutions.....	23
9.1. Alternative solutions .....	23
9.2. Justification of chosen solution.....	23

## **1. Introduction**

### **1.1. Client identification**

My original client was Heather Chambers, she was my maths teacher. Heather chambers is very busy as she is vice principle and is no longer my teacher. I have chosen a new client who is equally interested in my project. His name is Paul Brown, like heather, he teaches math mechanics at Long Road Sixth Form College, he is however new to the college. He teaches two groups of approx 25 students and runs a forum group. Although he teaches A2 maths, my initial project was designed for Heather for AS maths. This is not such a problem because I may cover both courses with my software.

### **1.2. Define the current system**

On a lesson to lesson basis most of the work is carried out from content in text books. Although the classroom dynamic is not recognised as a system, it is a system. Paul would set specific questions to be answered in a period of time. The students would then attempt the questions and look at the back of the book to check the answers. If the student does not understand then the Paul would try to make the student understand. The same can be applied to past paper materials.

The main method of computational teaching is through a browser on the site MyMaths([www.mymaths.co.uk](http://www.mymaths.co.uk)). MyMaths offers a lot of flexibility in the following ways. It has portability as it is web based, on a flash plug-in. The students can test themselves, or the teacher can set work for them. The tests can be taken again and again until the student is satisfied with the result. MyMaths is clearly laid out and is simple to use. It is clearly an established program as it has some sophisticated features that bespoke software would not have.

Autograph is also used for coursework and graph demonstrations in class. It allows the user to enter an equation and the software will plot a graph on a virtual scale. The user can perform functions on the graph for analysis. It is useful for different areas of maths such as algebra, calculus, data manipulation.

### **1.3. Describe the problems**

The problem with books is that they aren't as interactive and the portability isn't as good as if it were web based or in downloadable application form. Maths text books are often very full of content, without whitespace or pictures to save on space. Books for classroom activities aren't as good as for teaching the students initial concepts. The Books are designed more as a repetition reinforcement method. The teacher teaches the initial concepts but the teacher is only available in class for help.

MyMaths is quite repetitious and samey in its ways. The questions do change, for example, the coefficients of terms are randomised each attempt to prevent cheating. The school must pay £460 + VAT per annum which is a reasonable amount to pay for the content, however it is still a cost. The users must have flash installed, in order to run the scripts as well as a web connection. It does not have very good cross platform capabilities. Also the pictures that MyMaths constantly display on the site are childish, distracting and irrelevant.

The problem with Autograph is that it isn't a student learning tool, it's more of a research tool. For example, studying how the  $1/x$  graph never actually touches the x axis.

## 1.4. Section appendix

What would you like to see in the maths teaching software?

- Exiting
- Easy to use
- Instant response
- Varied activity

Is there a specific area of maths that you would like the maths to be based on?

- Algebra and calculus
- Transformations of graphs
- log graphs
- Trig functions

Should there be a student submission from the software?

Yes.

Should the software be fun and optional or monitored and formal?

Fun and monitored

Should the scores be compared with others if implemented?

Yes, average calculated

Would the teaching tool be used in the classroom or at home?

Both

Are there any key requirements?

- Large font
- Uncluttered screen
- Accepting of equivalent answers

Are there any elements of current teaching tools that you like?

- I like the layout of MyMaths and scores
- For Autograph I like the output of graphs and functions but not as intuitive to use.

Are there any system requirements?

The standard PC at college screen resolution

Is there any student data stored with the current software?

- Name
- Tutor
- Teacher
- Scores
- GCSE grades

What is the current method for creating a new account?

- Tutor set up
- Password generated, given to students

If the learning tool is to be based on a level, what level of maths would you like the software to be based on? (introductory, AS, A2, Coursework)

-AS

To what extent are the current maths teaching tools useful?

Not essential but adds variety.

What would you like the aesthetic and layout style to be like? Existing software may be implemented.

- Clear
- Lots of white space
- Easy to input answers

## 2. Investigation

### 2.1. The current System

#### 2.1.1. Data sources and destinations

For MyMaths

Source	Data	Example data	Destination
Tutors account	Setting up new student account	Name: John Smith Password: apple Email: js@ntlworld	server
Student account	Score for the test	Total: 67% Sub1: 78% Sub2: 56%	server
Return values from request for answers	Page content Integers, strings, etc	$10^3$ , simultaneous	Client: Flash page
Request for page content	Module, topic, page number	Core 1, Integration, 6	Server SQL search

For classroom

Source	Data	Example data	Destination
Department office	Teaching plan	Week1: expanding the bracket polynomials	teacher
Test papers from students	Answers for marking	253, 66, 3.141	teacher
teacher	questions	Integrate: $5x^3$	students

For autograph

Source	Data	Example data	Destination
student	Equation string	$4x^4 - 2x^3 + x$	Input form
AutoGraph	graph	Vector graphic: X:130,Y:90	GUI output form
Users hard drive	Image data	Parabolic curve	Software memory
Request for graph initiation GUI	Mouse click	True	Software memory

### 2.1.2. Algorithms

There are hundreds of algorithms in the MyMaths system so to show them all would be futile. These are the most important algorithms.

Sorting algorithm

```
FUNCTION BubbleSort( listToSort: list of strings)
  lengthOfList ← length(listToSort)
  REPEAT
    newn ← 0
    FOR i ← 1 to lengthOfList -1 DO
      IF StringVal(listToSort [i-1]) > StringVal(listToSort [i]) THEN
        SWAP(listToSort [i-1], listToSort [i])
        newn ← i
      END IF
    END FOR
    lengthOfList ← newn
  UNTIL lengthOfList ← 0
```

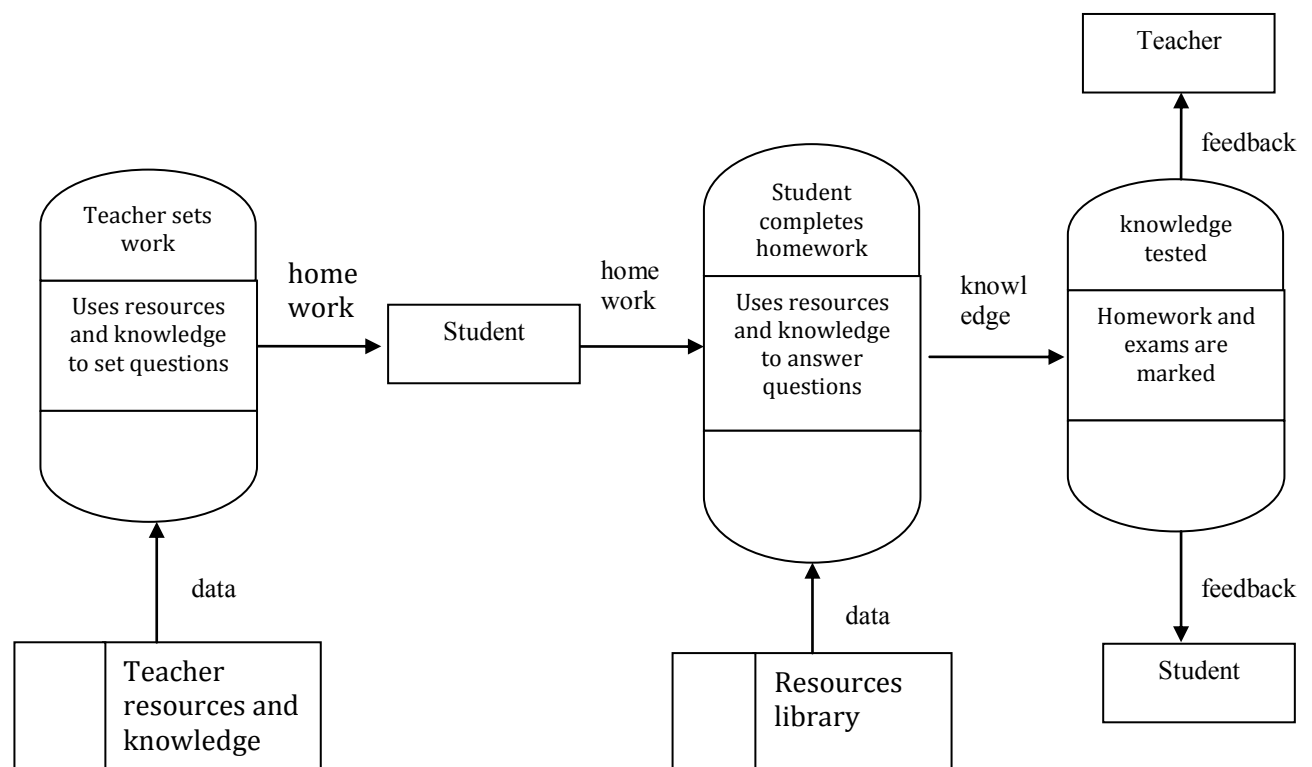
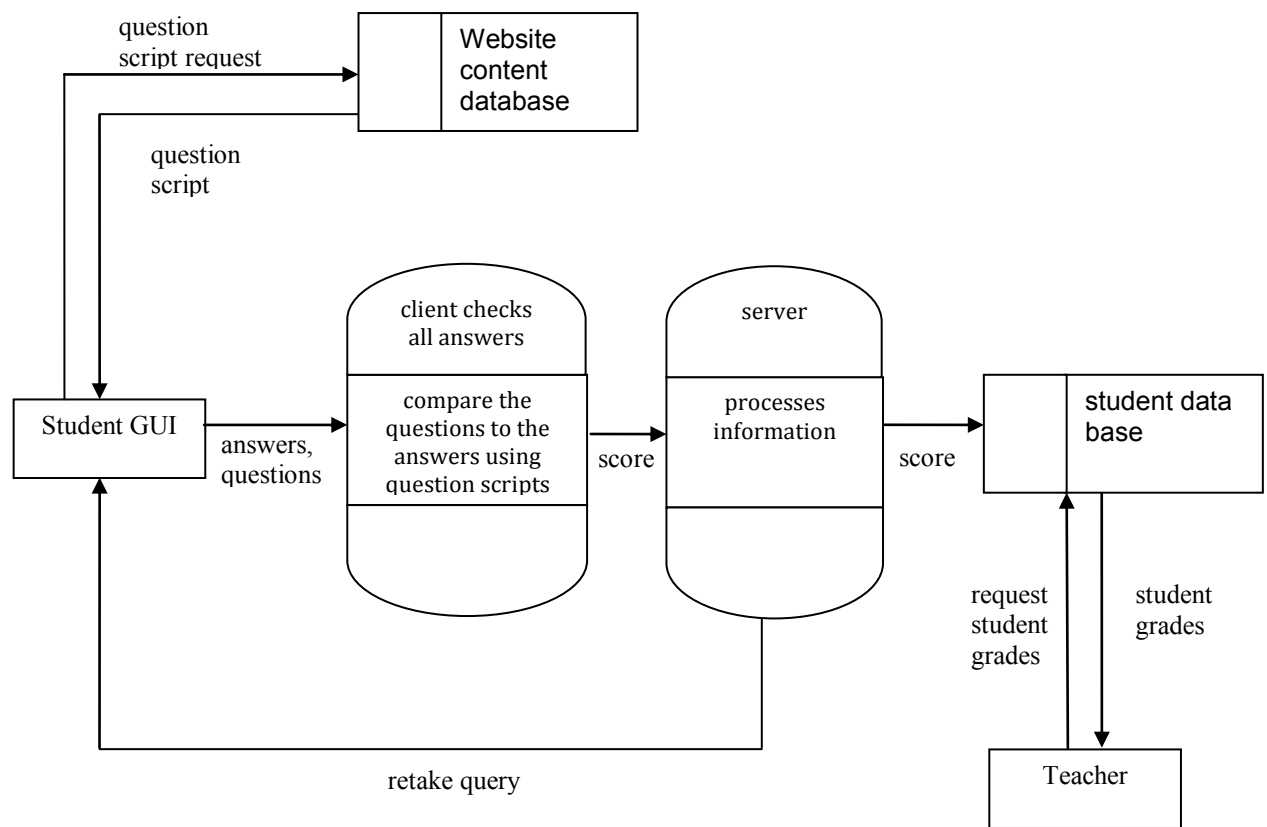
```
FUNCTION StringVal(stringToValue: string)
  Totalvalue ← 0
  lengthOfString ← length(stringToValue)
  FOR i ← 1 to lengthOfString -1 DO
    Totalvalue ← (ord(Totalvalue+ stringToValue[i]))/(10^i)
  RETURN TotalValue
```

Search algorithm

```
ListOfResults ← [0 ]
NameToGet ← ""
NameToGet ← INPUT
```

```
Count ← 0
WHILE (Count < SIZEOF(ListOfSchoolStudents)) DO
  IF (ListOfSchoolStudents[Count].FirstName= NameToGet) THEN
    ListOfResults[SIZEOF (ListOfResults)+1] ← Count
  ELSE IF (ListOfSchoolStudents[Count].LastName= NameToGet) THEN
    ListOfResults[SIZEOF (ListOfResults)+1] ← Count
Count ← 0
WHILE (Count < SIZEOF(ListOfResults)) DO
  OUTPUT ListOfSchoolStudents[ListOfResults[Count]]. FirstName
  OUTPUT ListOfSchoolStudents[ListOfResults[Count]].LastName
  OUTPUT ListOfSchoolStudents[ListOfResults[Count]].ID
  OUTPUT "\n"
```

### 2.1.3. Data flow diagram



## 2.1.4. Input forms, output forms and report forms

### input form

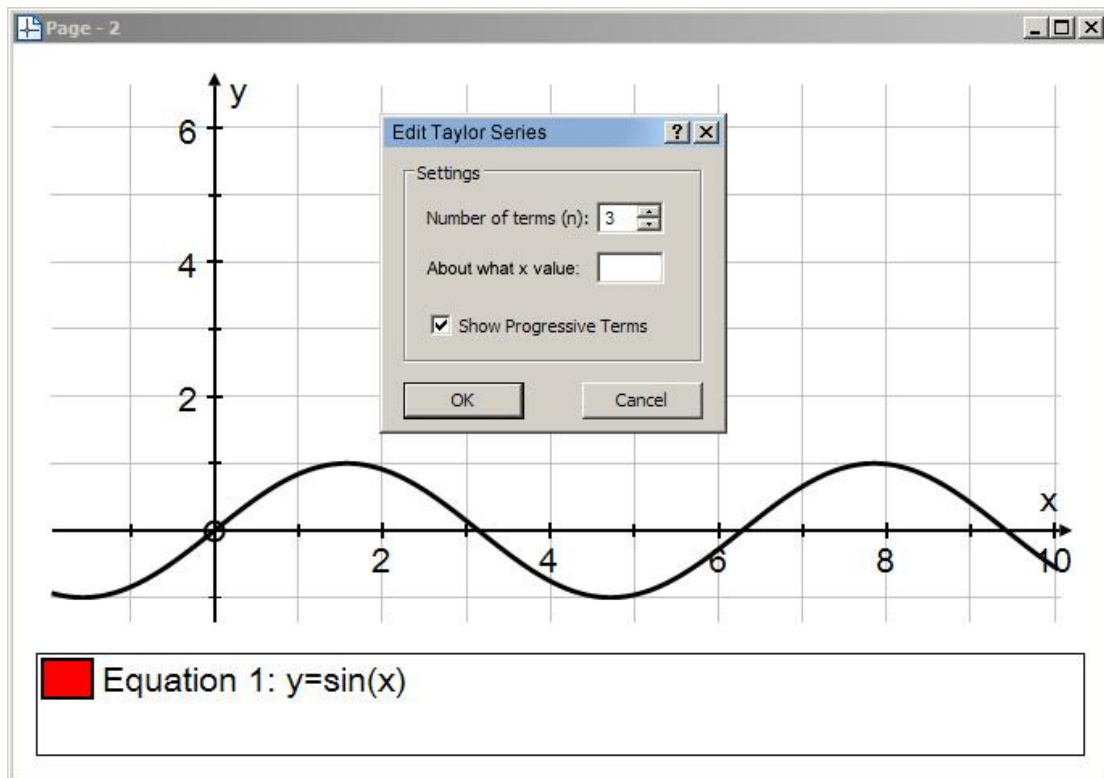
The user enters data into borderless text boxes, sliders and buttons. Each page of questions has a 'Markit' button which stores the data in working memory. When the user completes the whole quiz, the data from all the pages is compiled and sent to the 'MyMaths' servers to be stored on the database.

The screenshot shows the MyMaths.co.uk website interface. At the top, it says "2nd Sep 2012" and "Midpoints and Line Length". Below this, there is a "Summary and question" section. The main content area contains a question: "M is the midpoint of the line joining A (0, 1) and B (-4, -4). Find M." The question is followed by a formula for the midpoint of a line segment:  $M = \left( \frac{x_1 + x_2}{2}, \frac{y_1 + y_2}{2} \right)$ . There are input boxes for the coordinates of M. A "Markit" button is located below the input boxes. On the right side, there is a cartoon character of a person wearing a graduation cap and holding a diploma. The interface also includes a sidebar with a list of topics and a bottom navigation bar with buttons for "CALC", "OVERLAY", and "INDEX".

This is another input form from the GUI. This page is where the student will find a quiz to complete. It is broken down into resources on the left, module across the top, topic in the central white area, then quizzes when the student clicks on the topic.

The screenshot shows the MyMaths.co.uk website interface for A Level. At the top, it says "A Level" and "Bringing Maths Alive". Below this, there is a navigation bar with buttons for "Core 1", "Core 2", "Core 3", "Core 4", "Decision", "Stats 1", and "Mech 1". The main content area is divided into three sections: "Resources" (Library, Booster Packs, Statistics GCSE, A Level, Games, Toolkit), "My Portal" (Login, Password, View), and "Admin" (Help, Contact, News, Documents, Integrate, AssessmentManager). On the right side, there is a list of topics: Graphs, Indices, Sequences, Series, Factor Remainder, Differentiation, Integration, Trigonometry 1, Trigonometry 2, and Logs. At the bottom, there is a "Boards" section with buttons for "OCR", "MEI", "AQA", "EdEx", "WJEC", "All", and "?".

This is a typical input form for Autograph. It appears small and in a separate GUI.



### Output form

This is simply the retrieval of data from the database that is displayed in a presentable manner. This page is displaying the results of a specific quiz completed at an earlier date.

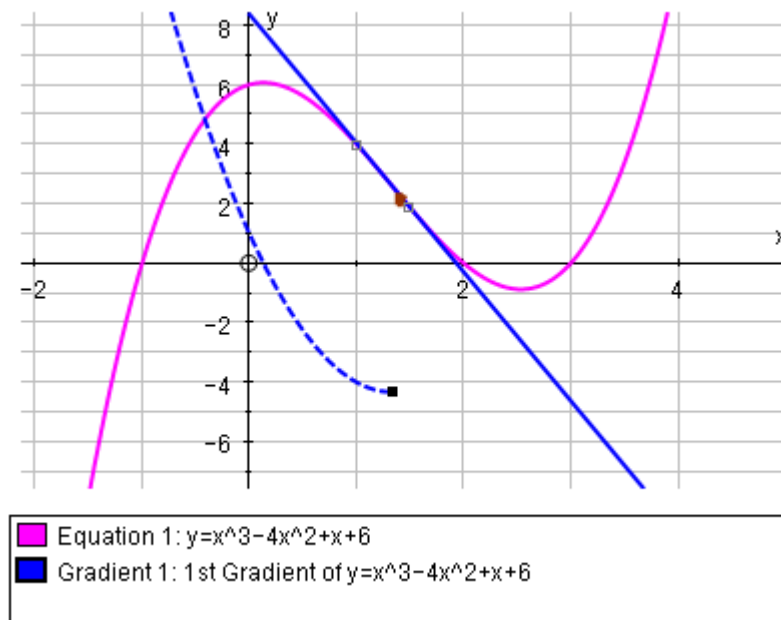
The screenshot shows the "MyMaths.co.uk Online Homework" checkout page. The header includes the MyMaths.co.uk logo and the text "Online Homework". Below the header, a message reads: "The Checkout - Here is a round up of all of your scores on this worksheet". A "Score Sheet" section displays the following information:

- Homework: Least Squares Regression
- Question 1:  $\frac{0}{13} = 0\%$
- Exam Marks
- Question 2:  $\frac{0}{9} = 0\%$
- Mothers and Daughters
- Overall:  $\frac{0}{22} = 0\%$

Below the scores, there is a red link "Want to try again?" and a "Next" button. On the left side of the page, there is a vertical "Checkout" button. On the right side, there is a cartoon character of a man with a large head, wearing a green shirt, blue pants, and a gold chain, pointing upwards. At the bottom of the page, there is a "Least Squares Regression Line" button.



This is an output form from autograph. It plots the graph to different accuracies depending on how far you zoom in on it.



## 2.2. The proposed system

### 2.2.1. Data sources and destinations

My proposed system is software by the name of 'Equation and Calculus Quizzer'. This covers many of core aspects of AS C1 mathematics in an area which I feel hasn't been taught well by software yet. It is essentially a specific teaching tool for the area of equations and calculus with an interactive interface.

I feel like my client does not want an exact replacement of the current system teaching. The reason is because the system is very good at what it does, for example it is easily accessible from home, it covers a vast amount of math topics, it gives good feedback and help if needed, and it informs the teacher if the student is progressing. What Paul wants is something different from what the current system does. Some features will be copied from the existing system like the homework setting ability and some of the topics.

The style of aesthetics will be very clean, spaced, clear, and will have a constant colour theme throughout. The font face may be Arial or Verdana or similar, the colours should be black, white, blue.

The areas of math that I want it to cover are all the common equations in AS maths, this includes, straight line, quadratic, cubic, quintal, circle, sine cosine, tangent equations. Each equation will have several different, interesting and interactive questions based on them.

There will be a GUI that displays a graph with axes. This graph can display a static graph of a function or the user can create a graph by clicking on the points on the graph. This is important because graph modelling is a big feature of AS maths.

I will be using a database to store all necessary information. This will include student data, teacher data, as well as class and school data. It will also logon details.

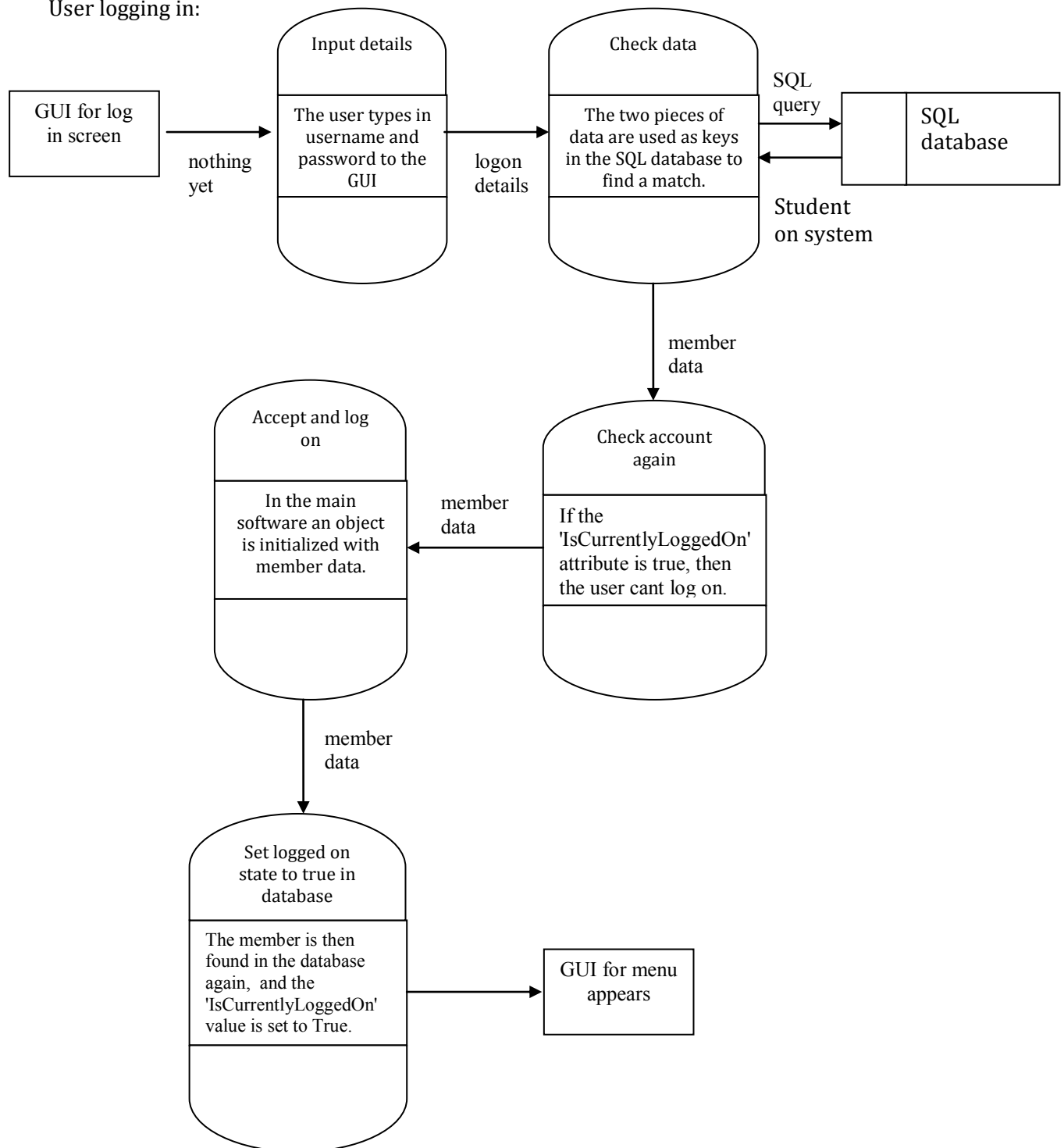
The system may have the scripts for producing the questions/answers on the database, this makes it difficult to code, however it means that new question/answer templates can be created on demand without having to change the software code and compile it. If not they will be hardcoded.

**Below is a table for sources and destinations of data. Note that the conversion from database format to class or list format or visa versa, is not here because it distracts from the point of where the data actually ends up.**

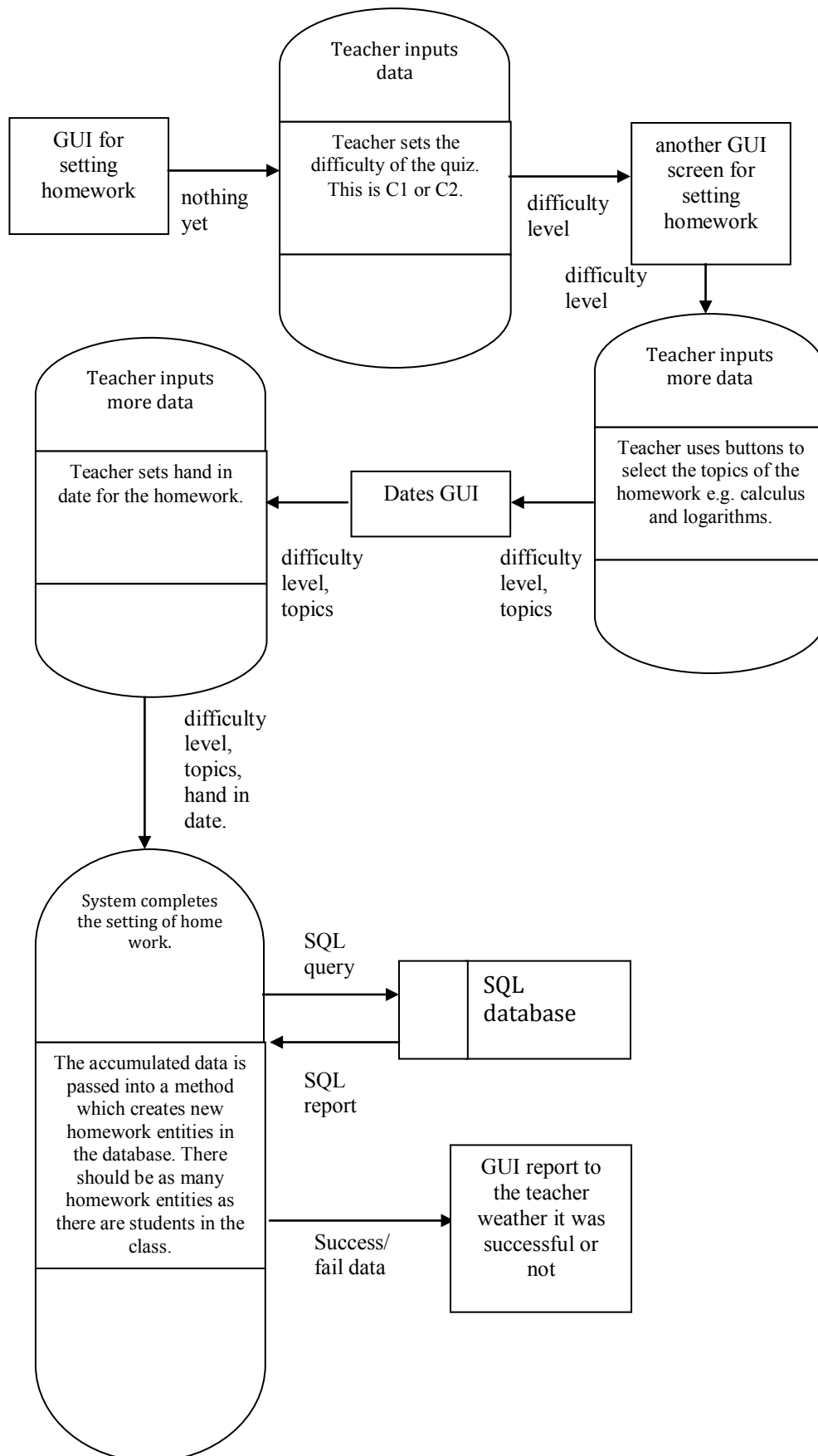
Source	Data	Data Type	Destination
Log in section	"Alastair", "password123", true	Strings and Boolean for 'isLoggedIn'	Database, changes student entity.
Log out section	false	Boolean	Database, changes student entity. The 'isLoggedIn' changes.
Student: Quiz section	"logs", 300,150,50,26	String for name and integers for scores	Database, adds test result entity.
Teacher: Homework creation	"Logs", "C1" "12/11/2012",	Strings	Database, adds homework entity
College administrator: teacher creation	"John", "Lancelot", "Lancy", "pass123",	Strings	Database, adds a new teacher entity.
Teacher: Student creation	"max", "smith", "username", "password"	Strings	Database, adds a new student to the database
Database: student entity	"max", "smith", 300	Strings and time spent	Ultimately Teachers GUI where there is a table of students data.
Database: TestResult	"logs", 300,150,50,26	String for name and integers for scores	Ultimately the students GUI where he/she can view all the completed work.

### 2.2.2. Data flow diagrams

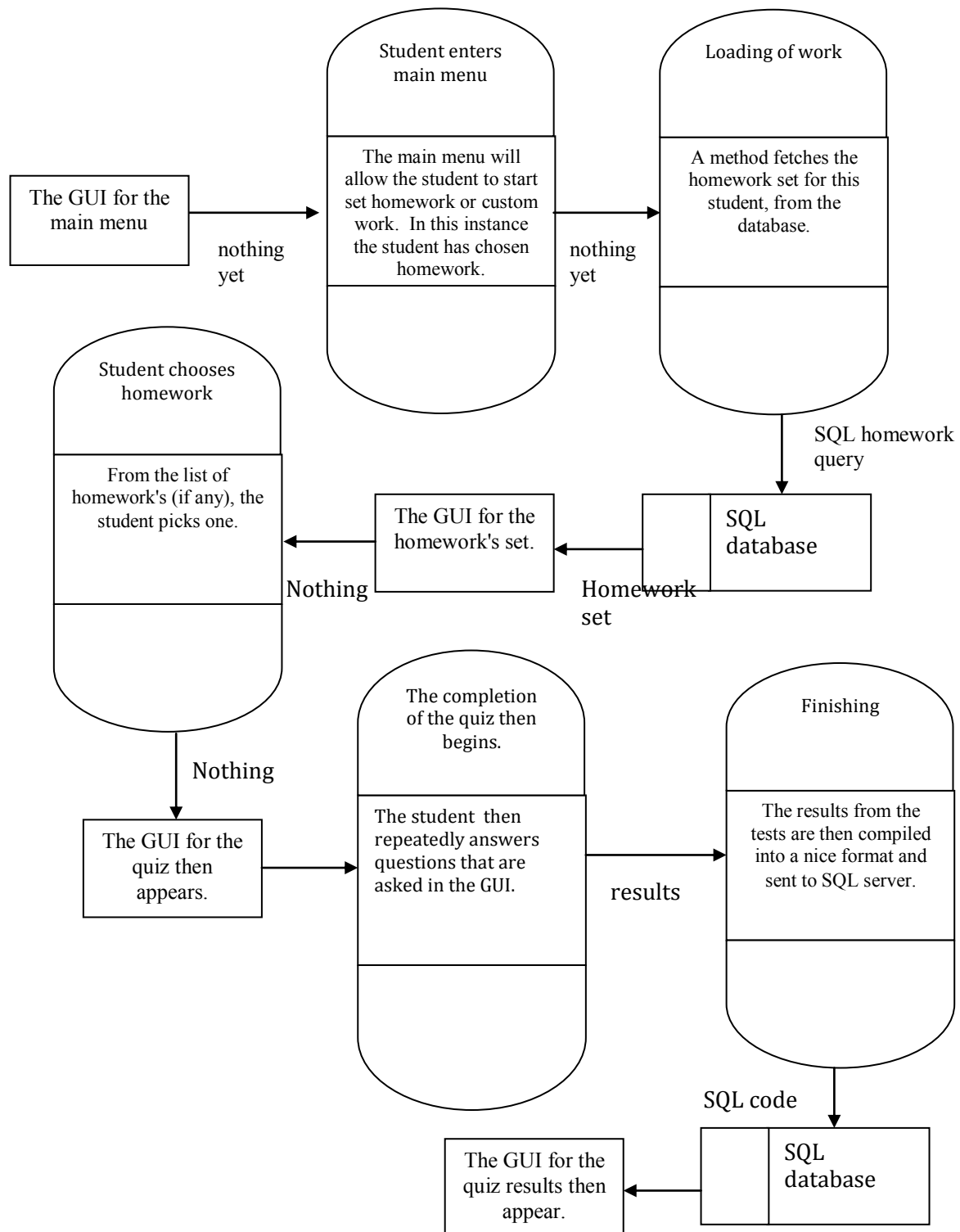
User logging in:



## Teacher setting homework:



The completion of a quiz:



### 2.2.3. Data dictionary

These values are for the objects in the software.

Name	Data type	Length	Validation type	Example data	Comment
TotalScore	integer	1	None	110	For result
averageScore	integer	1	None	50	For result
maxScore	integer	1	None	220	For result
dateSubmitted	Daytime/string	8	None	11/10/2012	For result
nameOfQuiz	string	1-15	Range check	"Time trial"	For result
studentFirst-NameOfBelonging	string	1-10	None	"John"	For result
studentSecond-NameOfBelonging	string	1-10	None	"Smith"	ForResult
username	string	1-20	Range check, format check	"extremeMaster360"	
password	string	1-20	Range check, format check	"Bespoke"	
dispalyNameFirst	string	1-10	Range check, format check	"Alastair"	
dispalyNameLast	string	1-10	Range check, format check	"Livingstone"	
accessLevel	integer	1	Presence check		
nameOfQuizMode	string	1-10	Presence check	"Time Trial"	For quiz mode
timeLeft	integer	1	Limit check	15	For quiz mode, in minutes
lives	integer	1	None	9	For quiz mode
skips	integer	1	None	3	For quiz mode
Coefficient_A	integer	1	Limit check	16	For equation
Coefficient_B	Integer	1	Limit check	2	
Coefficient_C	integer	1	Limit check	9	
equasionName	string	1-30	None	"Circle equation"	
timeSpent	integer	1	None	300	For student, in minutes
timeSet	String/datetime	1	Presence check	02/04/2013	For HomeWork
timeDue	integer/datetime	1	Presence check	20/05/2013	For HomeWork
isCompleted	Boolean	1	None	True	For HomeWork
maximumNoQuestions	integer	1	Limit check	30	For homework

#### **2.2.4. Volumetrics**

Any number of people could use the system as long as the memory capacity is high enough. My client teaches two classes which totals 45 students. The amount of memory that a single student would use would be approximately 1.5 kb. The amount of memory that would be taken up by the quiz results is very small because they use up 16 bytes each.

### **3. Objectives**

#### **3.1. General objectives**

- Simple interface
- Intuitive and easy to use
- Must give feedback
- Must provide variety
- Must be usable from different locations on the college network
- Helpful and informative graphics
- The software should cater to different levels of progress
- There must be accounts with data assigned to them
- The teacher must be able to create homework

#### **3.2. Specific objectives**

- The system must save student data
- The system must keep track of results
- The system must have a teacher account with special privileges
- The system must generate random questions to prevent cheating
- The software must produce a graph
- The teacher must be able to remove accounts
- The software must run on Windows, Mac and possibly Linux
- The software should not take up more than 4 GB of data
- The teacher should be able to view the scores of all the students in the class
- The teacher should be able to set homework with many customisable options
- The students should be able to complete a quiz that isn't specifically 'Homework'
- The teacher should set homework using different input methods

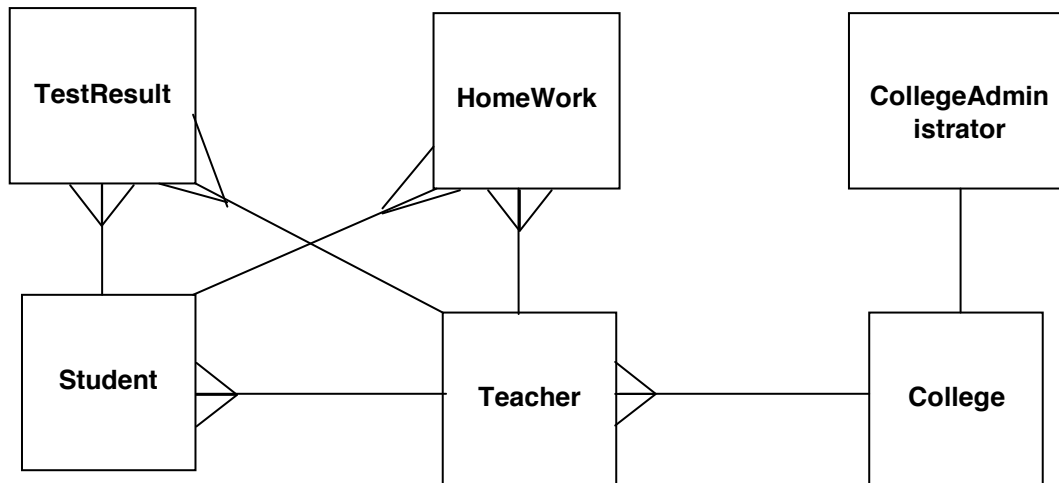
#### **3.3. Core objectives**

- The software must teach algebra and calculus
- Must teach AS equations and trig functions
- Must give test feedback
- Must be able to view student data
- Must be able to view teacher data
- Must be able to remove entities from database
- The actual software side: The design must allow for growth e.g. adding a new GUI.
- The actual software side: The design must reuse code

## 4. E-R Diagrams and descriptions

### 4.1. E-R Diagram

This is the entity relationship diagram for the SQL database. The reason why TestResult is a table in the database but not question is because the database will not store code.



### 4.2 Entity Descriptions

TestResult(TestResultID, TotalScore, ScorePart1, ScorePart2, ScorePart3, ScorePart4, ScorePart5, ScorePart6, ScorePart7, *StudentID*)

Student(StudentID, FirstName, LastName, Username, Password, Tutor, StartingDate, ExpiertyDate, IsCurrentlyLoggedOn, TimeSpent, *Teacher*)

Teacher(TeacherID, FirstName, LastName, Username, Password, IsCurrentlyLoggedOn, *CollegeID*)

CollegeAdministator(CollegeAdministatorID, FirstName, LastName, Username, Password, IsCurrentlyLoggedOn, *CollegeID*)

HomeWork(HomeWorkID, TimeSet, TimeDue, difficultyLevel, maximumNoQuestions, IsCompleted, *TeacherID*, *StudentID*)

College(CollegeID, Name, Address)



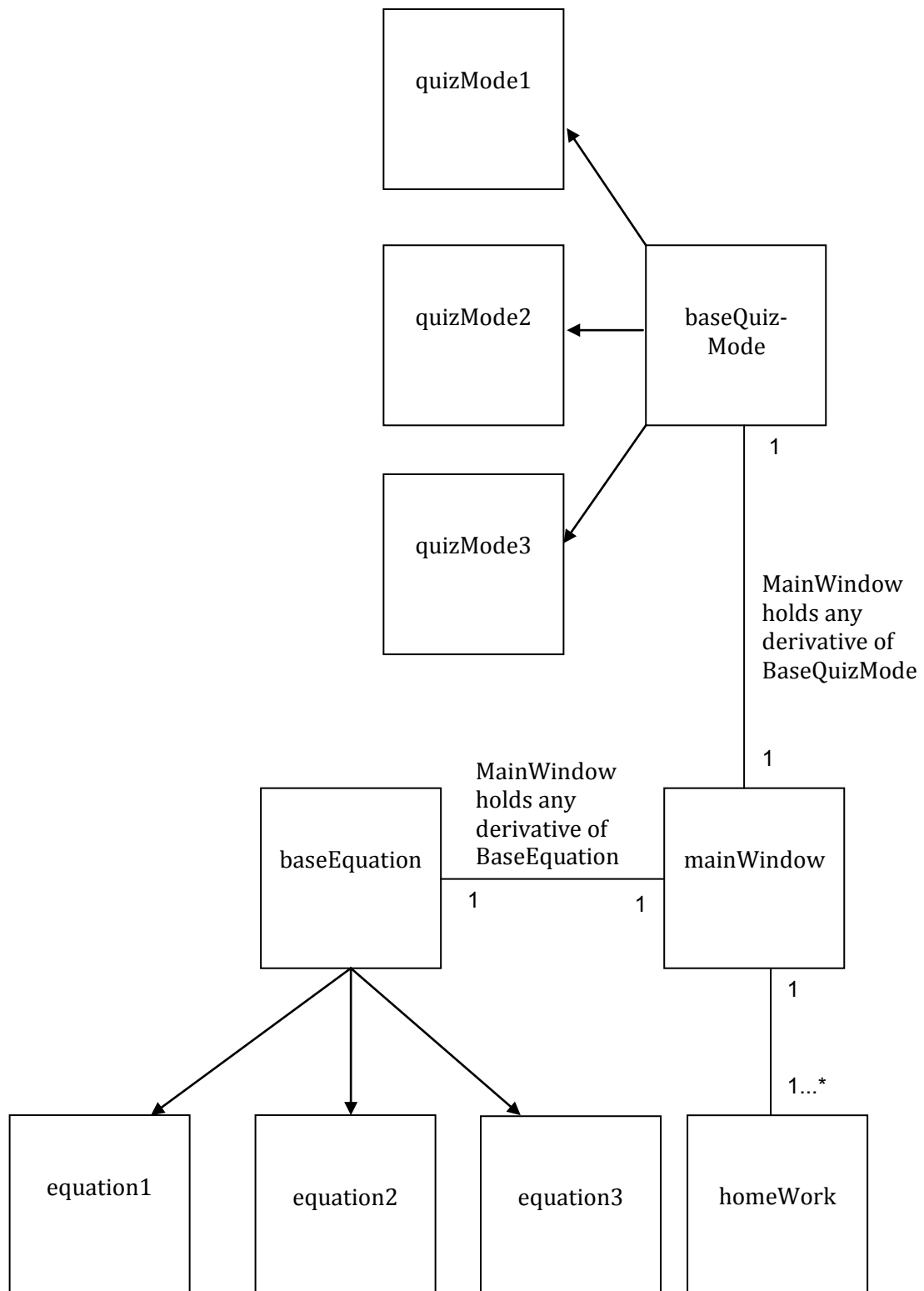
## 5. Object analysis

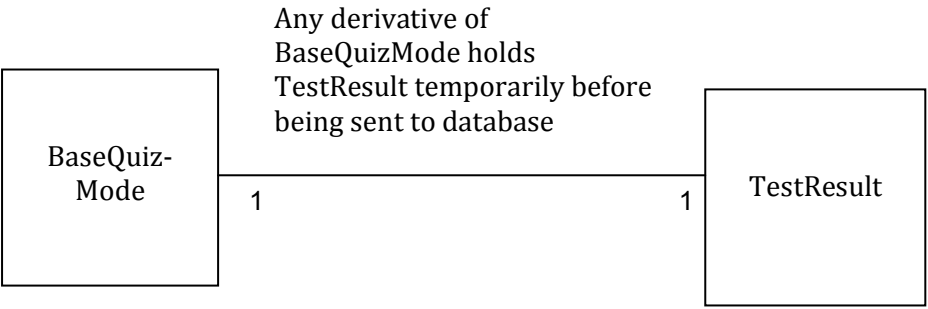
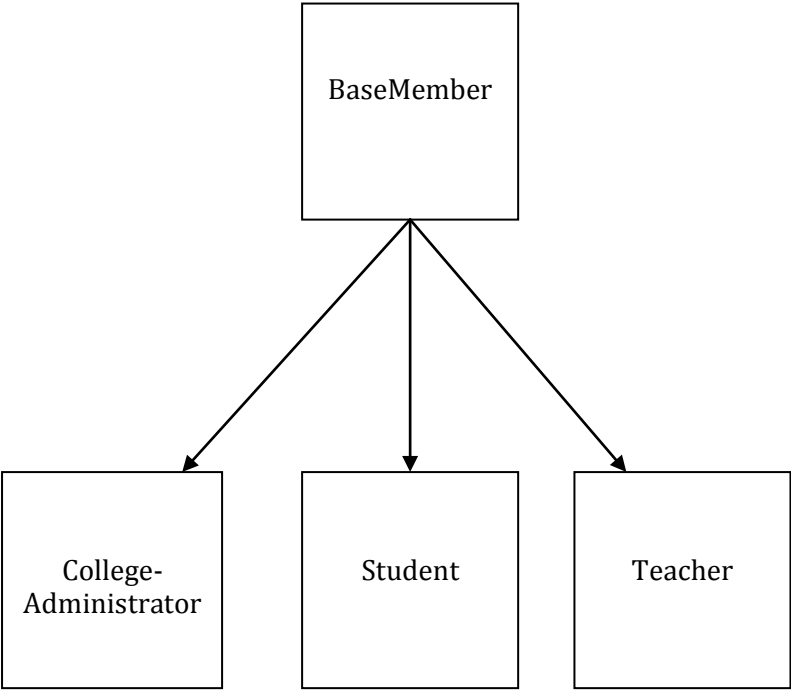
### 5.1. Object listing

These are the main classes that are essential to the core of the system. Other classes relating to the GUI and minor features will may not be included in this stage. Other classes that will not be gone in depth with are the Quiz modes and the types of equations. I feel that these will be gone in depth with in the design section.

- BaseEquation (The reason why I have not included child classes for this base equation is because this will come at the design stage. I do however, know that the BaseEquation methods are virtual. )
- MainWindow
- TestResult
- BaseQuizMode
- QuizMode1
- QuizMode2
- QuizMode3
- BaseMember
- Student
- Teacher
- CollegeAdministrator

## 5.2. Relationship diagrams





### 5.3. Class definitions

The reason why the members don't have many methods is because they have access levels which are used to access certain functions which are access level specific.

testResult
TestResultID nameOfQuiz totalScore maxScore averageScore dateSubmitted studentFirst- NameOfBelonging studentSecond NameOfBelonging storeSelfInDatabase()

baseEquasion
equasionName coefficient_A coefficient_B coefficient_C
getYCoord(Xpos)  getQuestion1() getQuestion2() getQuestion3() getQuestion4() getQuestion5() getQuestion6() getQuestion7()  getAnswer1() getAnswer2() getAnswer3() getAnswer4() getAnswer5() getAnswer6() getAnswer7()

Student: baseMember
<b>Inherited:</b> password username displayNameFirst displayNameLast accessLevel <b>Not Inherited:</b> timeSpent
submitTest() clearTestHistory()

baseMember
password username displayNameFirst displayNameLast accessLevel

homeWork
timeSet timeDue isCompleted quizModeName difficultyLevel maximumNoQuestions storeSelfInDatabase()

collegeAdministrator: baseMember
<b>Inherited:</b> password username displayNameFirst displayNameLast accessLevel <b>Not Inherited:</b>



teacher: baseMember
<b>Inherited:</b> password username displayNameFirst displayNameLast accessLevel <b>Not Inherited:</b>
getStudentDetails()

baseQuizMode
testResult nameOfQuizMode timeLeft lives skips
onSkipQuestion() answeredQuestion(score) checkEndOfQuiz()

mainWindow
baseEquation baseQuizMode
linkQuizMode1() linkQuizMode2() linkQuizmode3()

quizMode1:baseQuizMode
<b>Inherited:</b> nameOfQuizMode timeLeft lives skips <b>Not Inherited:</b>
onSkipQuestion() answeredQuestion(score) checkEndOfQuiz()

quizMode2:baseQuizMode
<b>Inherited:</b> nameOfQuizMode timeLeft lives skips <b>Not Inherited:</b>
onSkipQuestion() answeredQuestion(score) checkEndOfQuiz()

quizMode3:baseQuizMode
<b>Inherited:</b> nameOfQuizMode timeLeft lives skips <b>Not Inherited:</b>
onSkipQuestion() answeredQuestion(score) checkEndOfQuiz()

## 7. Constraints

### 7.1. Hardware

**This is the hardware spec for the computer I will be developing the system on:**

Operating system: Windows 7 Professional 64-bit (6.1, build 7601)  
Language: English  
System manufacturer: Dell Inc  
System model: OptiPlex 7010  
Processor: Intel (R)Core(TM) i5-3550S CPU @ 3GHz (4CPUs)  
Memory: 4096MB RAM  
DirectX version: DirectX \* 11  
Current display mode: 1920\*1080 (32bit) (60Hz)

**This is the spec of the lowest quality college computer:**

Operating system: Windows 7 Professional 64-bit (6.1, build 7601)  
Language: English  
System manufacturer: Viglen  
System model: VIG5805  
Processor: Intel (R)Core(TM) 2CPU 4400 @ 2GHz (2CPUs)  
Memory: 2048 RAM  
DirectX version: DirectX \* 11  
Current display mode: 1280\*1024 (32bit) (60Hz)

At the school there are no built in speakers, but there is an audio jack which is commonly used for headphones.

### 7.2. Software

The operating system the software will be running on is windows 7. The apple computers in the college only belong to media. The amount of memory being used by the execution python can't exceed the ram.

### 7.3. Time

The implementation must be completed by 25th of October 2012.

### 7.4. User knowledge

The users of the system will be students and teachers. The as far as I am aware, the system will not be intellectually demanding, it should be a simple to use quiz program. The installation may require some knowledge however.

### 7.5. Access restrictions

The system should be accessed from college simply because the software and all of its data would be installed on the college network.

## 8.Limitations

### 8.1. Areas which should not be included in computerisation

I do not wish to include maths content like mechanics and statistics. This is bad because the software is specialized to core maths. I do not need to encrypt the database as it is not important data.

### 8.2. Areas considered for future computerisation

Core 3 and Core 4 maths content should be added to the system in future. If I have spare time remaining I may wish to add some of this content in the main development section.

## 9. Solutions

### 9.1. Alternative solutions

Solution	Advantages	Disadvantages
(main solution) To write a quiz system in python for the school network.	I have good knowledge of python.	Lacks portability, runs when school network is up.
Online flash/JavaScript quiz.	Has portability.	Requires internet connection. Bad knowledge of PHP networking/ Action Script for flash. I must pay for a domain.
Alternative to hardcoded quiz structure. The quiz components are stored as scripts and can be run by the main python program.(Includes LUA and VBScript)	Increases longevity of system life. Allows students with scripting knowledge to add to the quizzes.	Requires knowledge of a script and how the main program communicates with the script.

### 9.2. Justification of chosen solution

I have chosen the top solution because it is the most solid and realistic solution in terms of time and knowledge, despite the fact that it lacks portability. The only relatively difficult part of the chosen solution is the GUI programming, especially the making of a graph.