Analysis

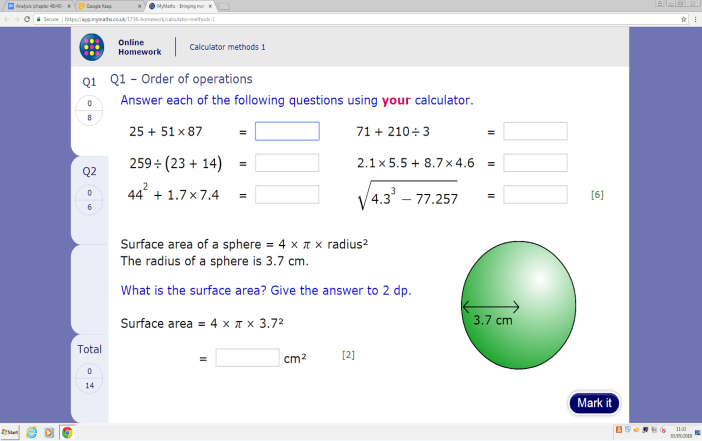
Idea

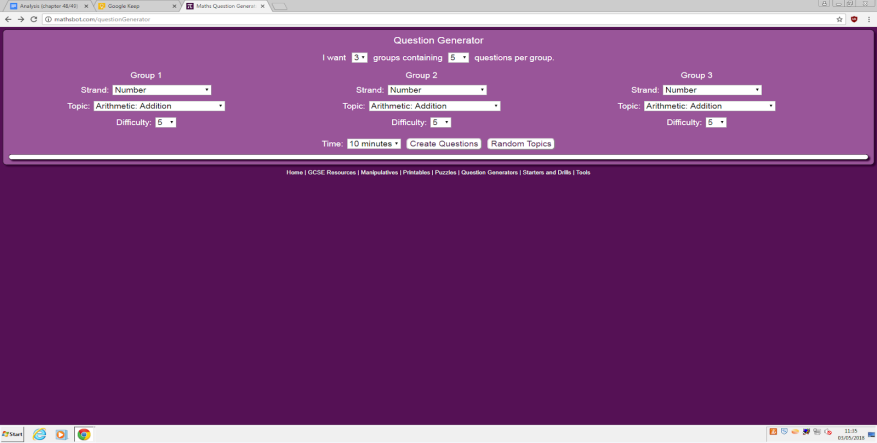
Although computers have made an entrance into essentially every school in the United Kingdom, methods of testing students on their knowledge of a subject is still typically done with the old method of paper and a pen. However, this has its drawbacks such as the teacher having to carry around a stack of papers with them whenever they wish to mark students’ papers, or if it’s a test that a student needs to complete at home, the student may lose the paper or forget to bring it to school on the deadline. There are many factors that make testing on pen and paper inconvenient, and so having a digital equivalent would make students’ and teachers’ lives a lot easier.

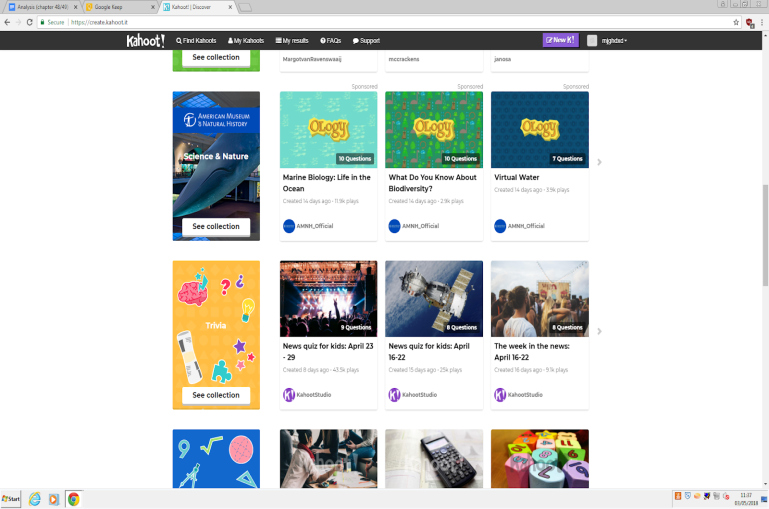
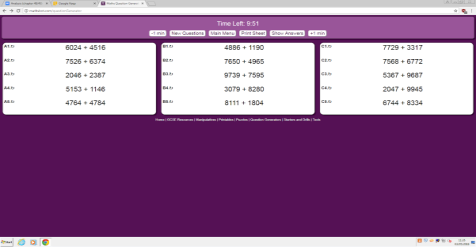
Research

What’s currently available?

There are multiple services available to provide different quiz experiences, however none fully fit the criteria and system that typical pen and paper testing usually ensues.

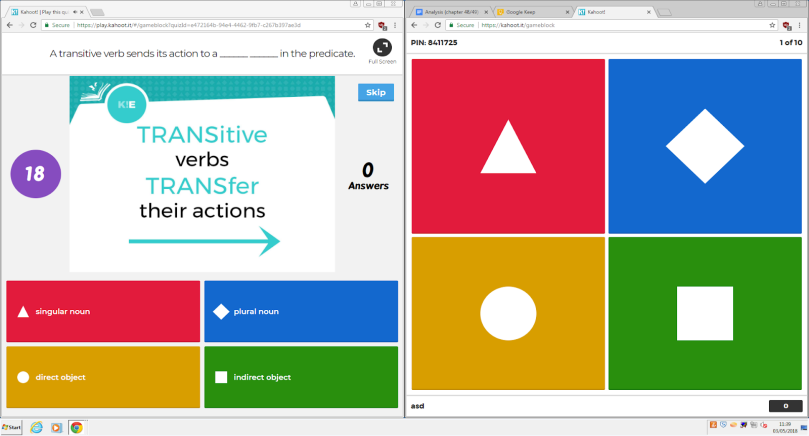
To start, possibly the most popular in the United Kingdom is MyMaths. This is used for maths lessons and consolidation of students’ knowledge, however it lacks the connection between teacher and student and the lack of ability to create custom quizzes means that a teacher may not be able to find exactly the correct quiz to give to their students. Perhaps most importantly, MyMaths only contains maths quizzes, so any other subject teachers cannot use their services, accentuated by the lack of custom quizzes. However, even with these limitations, it is the most sophisticated in terms of features for the teacher, such as being able to create classes to make homework for, as well as being able to see how many answers their students got right and how many attempts it took them. It also retains the pen and paper-like system of students answering a set of questions individually, which is uncommon in quiz websites or programs.

In a similar vein, MathsBot is a website in which a teacher can randomly select questions in relation to a topic that the teacher wishes to test their students on. Although this site lacks custom questions and the digital element that I wish to achieve, as once the questions are picked, the only option is to print it off to give to students physically, the element of randomly choosing questions in relation to a topic is one that will be useful to implement into my program, as it is a unique idea for a quiz program.



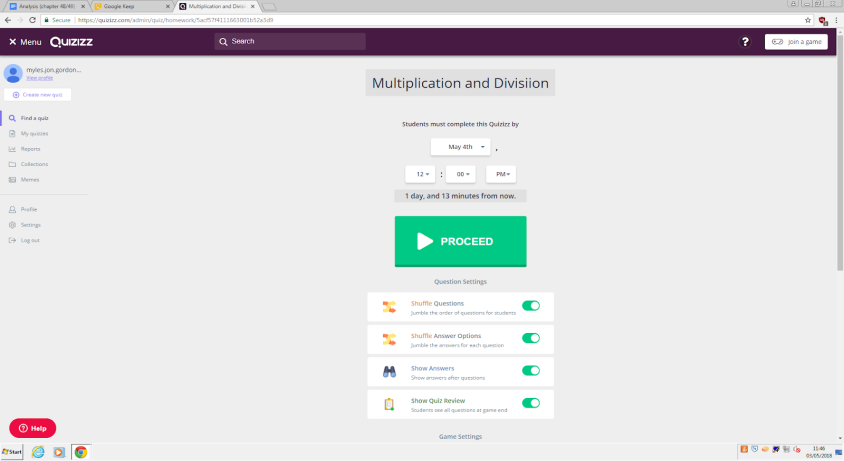
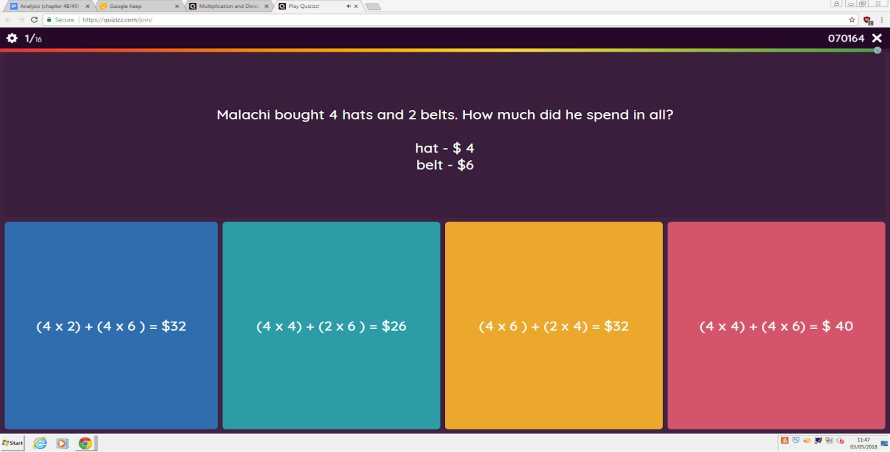
Choosing a quiz on Kahoot

Another popular quiz website used in schools is Kahoot. On Kahoot, a teacher is able to choose or create a quiz and display it on a screen, and then the students must answer the questions in a given amount of time. This creates excitement while answering, and is a great way for students to consolidate material covered in previous lessons, however for a private test to consolidate knowledge or provide exam style questions, this format is unsuitable, as well as it being too informal.

Kahoot also lacks text input answers, which are essential if a teacher wants to test a student’s exam technique.

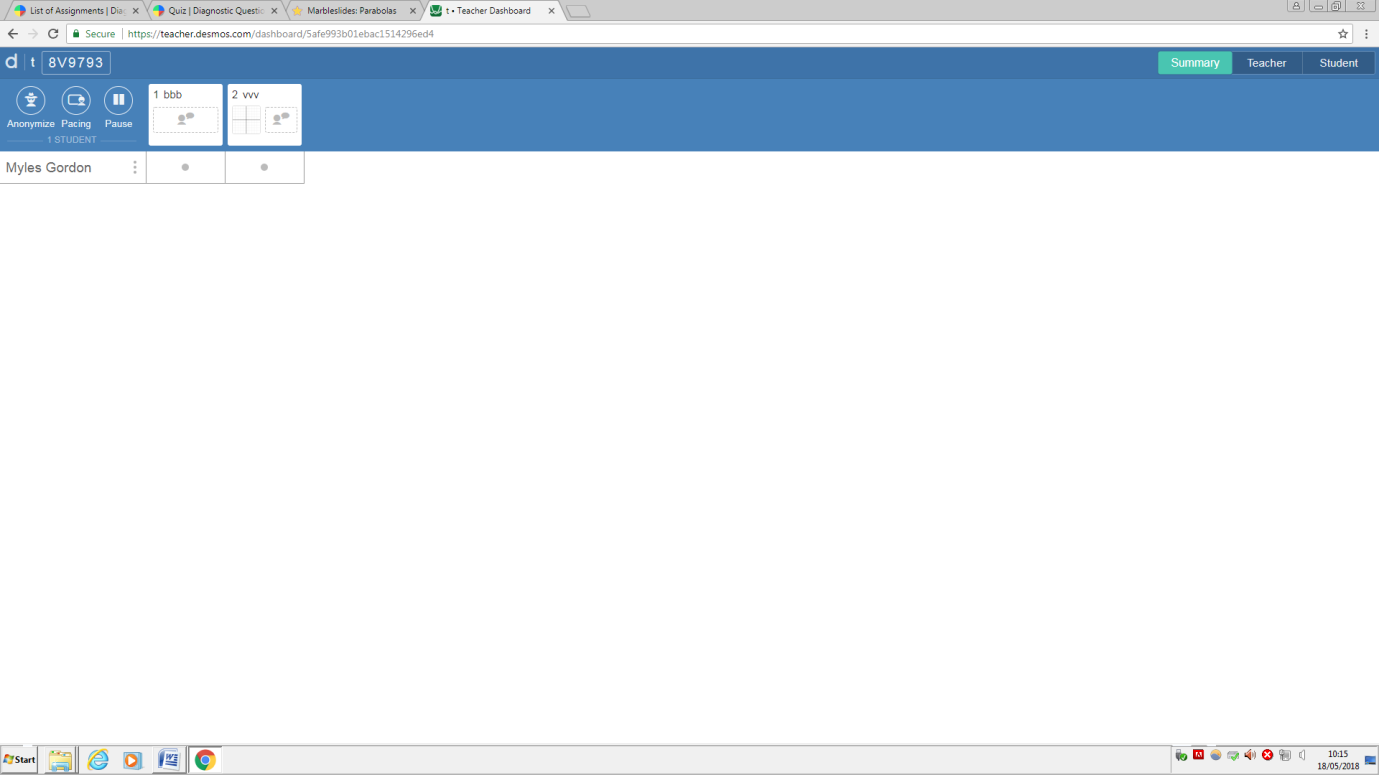
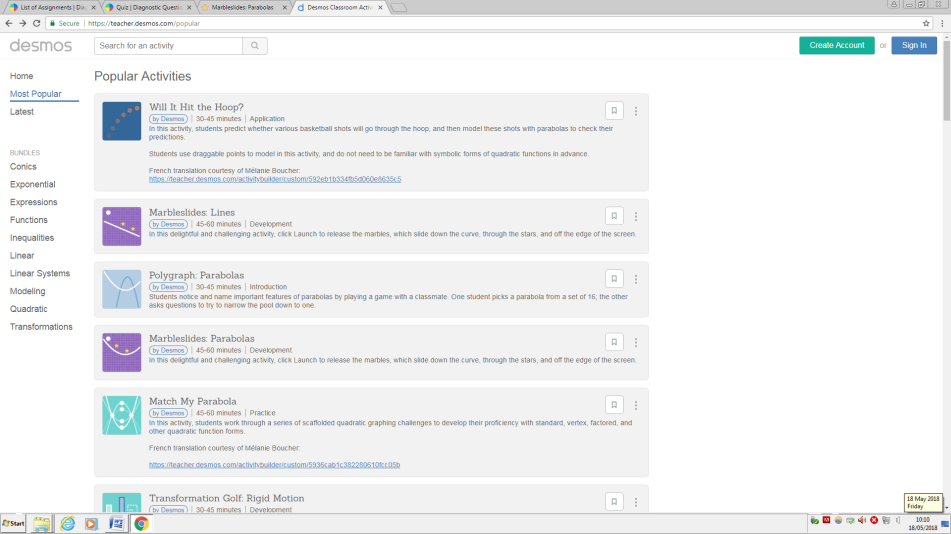
Doing a quiz on Kahoot

As well as this, Kahoot lacks the capability for a teacher to send a quiz to their students to do either at school or at home to do personally, and so the traditional style of testing that students will experience in exams will be lost.

Quizizz is similar to Kahoot, and does include homework tasks to a student to complete at home, which is definitely a step in the right direction. Once a student has finished a quiz, they are able to go over their incorrect answers and amend them, which is a good feature for consolidating knowledge and revision, however the teacher does not see this and so personal recommendations from the teacher cannot occur. Like Kahoot also, it lacks formality, especially with its inclusion of memes between questions and bright colour palette for the UI. It also lacks text input answers, which a teacher may prefer to give their students as it’s closer to exam style questioning.

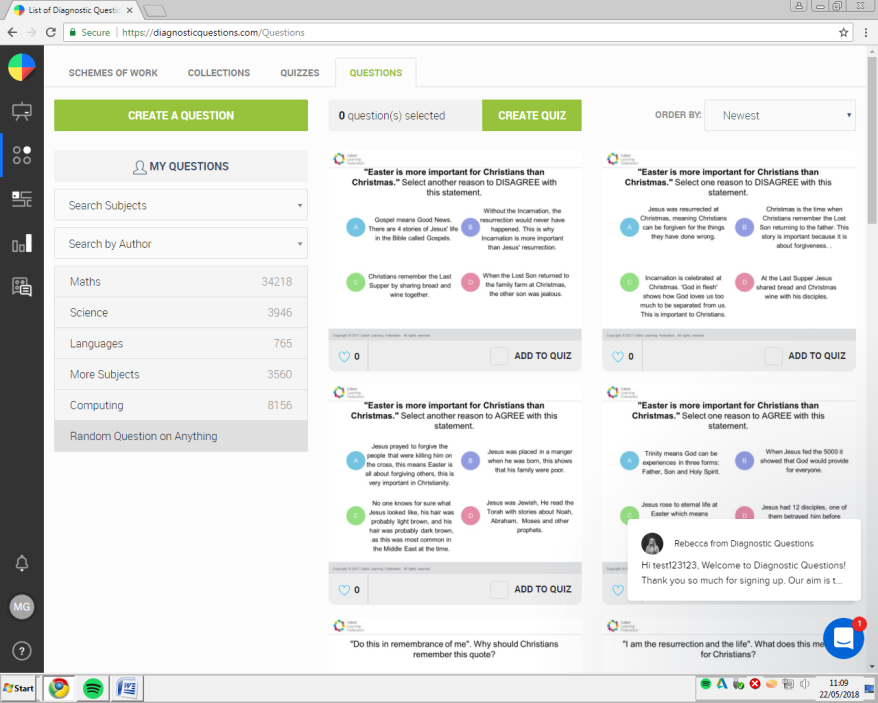
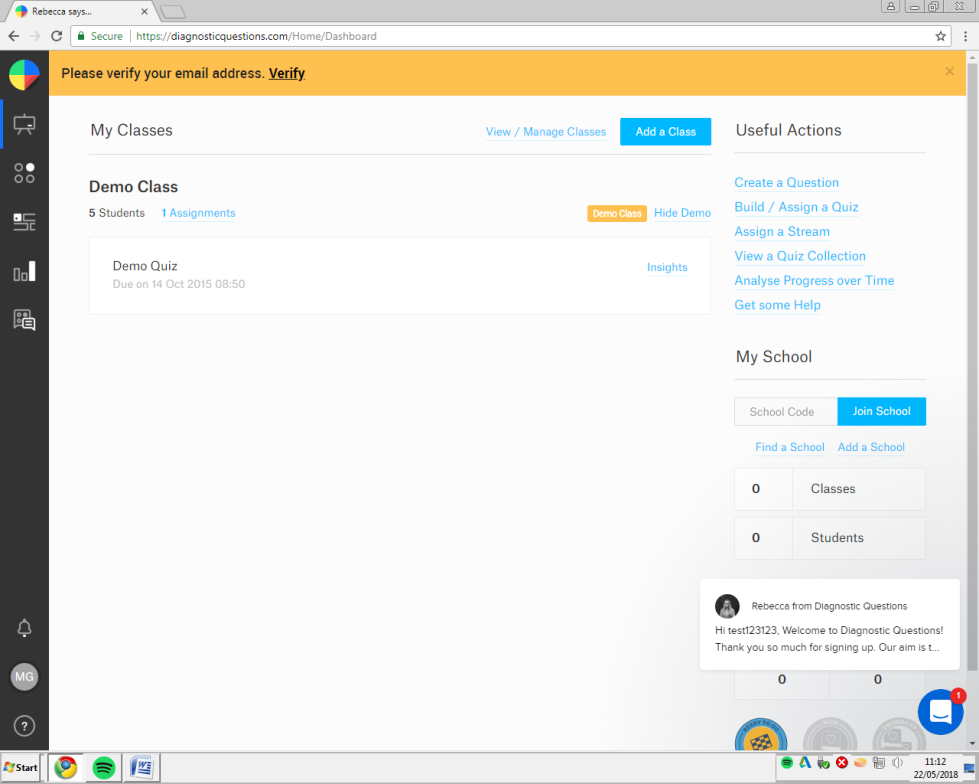
Doing a quiz on Quizizz

Setting a homework on Quizizz

Desmos is a website commonly used in maths lessons, with all of its activities being under mathematic categories. In this website, a teacher can create a class and invite students to particular games, where the teacher can track their progress in the activity. Custom quizzing is available, however instead of having a single class with all of the students to set quizzes to, the student has to join a quiz with a randomly generated code, as well as the teacher having no control over the name of the temporary class. This means that if a teacher wishes to set 3 quizzes to their class, the students will have to quit the quiz each time and then type in a randomly generated code, which is inconvenient for the student, and also the teacher has to keep giving out the quiz codes which is inconvenient for the teacher. However, this site does make good use of marking, with the students’ name being next to all of their answers, but it does lack instant marking in custom quizzes, and so if a teacher wishes to create a simple maths quiz where there’s only one acceptable answer, they have to waste a lot of time marking something that could be automatically marked.

Teacher's marking screen

Browsing the activities

Diagnostic Questions is a website that adopts the separation of teacher/student well, with a student being able to join a class that the teacher can send a quiz out on. A teacher can build a quiz and create questions, or search for a public quiz that they can send out to their classes.

Student's class screen

Browsing through questions

The main problem with this website, however, is that the teacher is unable to create a whole quiz with custom questions without creating each question and uploading it one by one, and then building a quiz with those previously created questions, which is extremely inefficient for a teacher who wishes to create a quiz from questions that they feel is appropriate to their class. Also, the only answering type is multiple choice, which is not always an appropriate way of answering questions, especially in an exam-like fashion for most subjects, which require the student to write a unique answer to a question (e.g. English exams)

User

The people who will benefit the most out of a new quiz program for schools will be the teachers creating the quizzes and looking at their students’ results and the students themselves, who will benefit hugely from a new way to consolidate their knowledge of a subject.

For the student, I asked Alexander Henderson, a student who takes Maths, Computing and Music:

1) Are you satisfied with how schools implement computers currently?

I'm satisfied, however from time to time I'd like to see a little bit more implementation.

2) How often do you use computers at home for homework/other consolidation of knowledge?

More or less every time I'm working at home, there's a computer open to use

3) How beneficial are quizzes to consolidation of knowledge for you personally?

They're fantastic as if you get something wrong you're more likely to remember it.

4) Would you say that a system of teachers sending out quizzes for you to do at any time like a normal homework would be beneficial?

Beneficial's an understatement, it would be very useful for exams, as teachers have their own ideas of what's going to be on, and the idea of being able to do it at any time would be helpful for exams.

5) What do you feel are the main drawbacks of using pen and paper for consolidation of knowledge?

It's slower to answer, you can't lose a digital copy of a quiz, pen and paper doesn't always give you the room to answer a question properly.

6) What would you like to see in a digital quizzing software as a student?

I'd like to see competition with other pupils as well as a digital notepad with space to work out the answers

For the teacher, I asked Mr. Frost, who teaches Maths and Computing:

1) Do you feel like computers are sufficiently implemented in schools?

No, not enough teachers know enough about programming. There's a lot of word processing and some schools do excellent in computers and computer science but there's not enough programming or teachers fully utilising software available

2) Do you think that students benefit from using websites and other software to consolidate knowledge?

It depends on a few things: the content, but also the user interface of the website is very important, like ease of data entry and feedback given

3) What do you feel are the main drawbacks of using pen and paper for consolidation of knowledge?

The time taken to mark them, there has to be feedback quickly otherwise students forget what they've done, and it's not as effective as having it given straightaway

4) What would you like to see in a quiz software geared towards consolidation of knowledge?

Design a quiz very quickly, minimise the time for questions being in my head and having it in a quiz, need to access student results in a way that's simple and as quick as possible, for example exporting to a spreadsheet

Objectives of the program

1. a) The user should be able to sign up with a username and password, and say whether they’re a teacher or a student

b) Once signed up, the user should be redirected their respective main menu (depending on whether they’re a teacher or student)

c) If the user tries to sign up with an existing username, they should be given an error message and stay on the login/signup page

d) The user should be able to log in with the details that they signed up with

e) If the user has entered incorrect credentials, then they should be told that what they’ve entered is incorrect

f) When someone has logged in, they should get the corresponding menu depending on whether they’re a teacher or student

1. a) If a teacher has logged in, they should be shown a main menu with the options to create a new quiz; open a draft quiz; view the quizzes that they have currently set to their classes as well as their quizzes set history; the quizzes they have yet to mark, view and amend the classes they’ve set up and to view all public quizzes.

b) If a student has logged in, then they should be shown a main menu with the options to view currently set quizzes; review their quiz history; request to join a class and to view all public quizzes.

1. a) A teacher should be able to create a quiz with as many questions as they’d like, custom timing, whether it’s an instant result quiz or not, different types of answering and custom questions.

b) Once the teacher has created a quiz, they should be able to send it to their students and/or be given the option to upload it to the public database of quizzes.

c) If a teacher hasn’t finished a quiz, they should be given the option to save as a draft

1. The teacher should be able to open their draft quizzes, and then edit them or delete them.
2. a) When taking a quiz, as soon as the quiz has started, the timer should start (displayed in the top right corner of the screen) and the user should be taken through the quiz one question at a time, with the question and corresponding answer type (such as multiple choice, text input box etc.). The current question number is also given in the top left corner alongside how many questions there are in total. When the user has finished a question, they can press “next” to go onto the next question. They are also given the option to go back to the previous question to amend their answer.

b) The user can quit at any point during the quiz with the “Quit” button, where they’re given a dialog asking whether they’re sure they want to quit or not.

c) When the user has pressed “Next” on the final question, they should be given a dialog asking if they are sure that they’re finished, and if they press “Yes”, then if it’s an instant quiz, their mark is displayed with their position on the leaderboard, and their mark is sent to the teacher. If it’s not, they’re taken back to the menu that they came back from and their answers are sent to the teacher to mark.

1. A student should be able to check if they have any homework that has been set by the teacher, and if they do, they should be able to take the quiz
2. a) A student or teacher should be able to browse and search the database of public quizzes

b) When a student clicks on a public quiz, if it’s an instant quiz then they should be shown a UI with the leaderboard and a “Play” button, where pressing “Play” should take them to the quiz.

c) When a teacher clicks on a public quiz, they should be given the options to “Play”, “Edit” or “Send to class”. If they press play, then they should play the quiz normally, if they press “Edit” a copy of the quiz can be modified by the teacher as if it were their own to put into their drafts or to send out to their students, and if they press “Send to class”, they are given a dialog as to which class they wish to send it out to.

1. a) A teacher should be able to view the quizzes with answers that they have yet to mark

b) The teacher should be able to click on the quiz that they wish to mark the answers to and be given a list of the questions and the students’ answers. Once they have gone through it, they should be able to give the student a mark, which is sent to the student.

1. a) A teacher should see their set quiz history

b) A student should see their quiz history with the amount of marks they received for it, and the ability to retake any quizzes

c) A teacher when looking at their quiz history should be able to export the student results as a CSV

1. a) A teacher should able to create classes, as well as amend any classes that they have

b) A student should also be able to request to join a class with the unique ID given to the teacher

Critical path of the project (going into more depth about the details in the objectives)

In the program, there is a system of logging out and signing up, which dictates what homework you get, what class you’re in etc. As a result, a database is needed. As a result, I will be running a server program which will be run externally which handles all commands of trying to log in and sign up. The server program will then take this information, and check it against a SQLite database. In the SQLite database, there will be a “User” table for anything regarding users, which will have columns for the username, the password and the type of user (teacher or student). Due to the handling of passwords, SHA256 encryption will be used to store the passwords, as well as for when the password is being sent to the server in case an illicit packet sniffer is able to read the serialised data stream going to the server. If the response for the signing up or logging in query shows that the username and password were correct, then a session is created to store the username (to be displayed at various parts of the program) and the type of user (teacher or student) so that certain parts of the program can be adapted for whether the user is a student or teacher. If the response shows that it was false, then the user is shown a dialog saying that their details were incorrect, where they can then retype the username and/or password.

In the program, the teacher is able to create a quiz and save it as a draft or upload it to the public database and/or share it with a class. For this, I will need to create a custom file type and a “Quiz” table. In the custom file type, there will be a section for metadata including the name of the quiz that the user will see and the unique ID that is randomly generated and a section for the questions themselves. Each question will be one line in the file, with a comma separating the type of values that each question will require - the question itself, whether it’s an instant answer question (and the answer if it’s an instant answer question) and the type of answer. This unique ID is put into the “Quiz” table when the teacher saves it as either a draft or for the public database, with rows for the ID, username of the person who made the quiz and whether it is public or not.

In order for the program to communicate with a database, I have decided to create a server side to the program to be hosted away from the client’s computer. This allows for server and client-side tasks to be done separately, and will allow for any server-side tasks such as accessing a database to be much easier. To send commands to the server, the client side will send the command via TCP which will then be received and interpreted by the server side, which will then send a response once it has completed the task. In order to allow multiple clients to send commands at once, this interpreting of commands and completing tasks will be done on different threads which will be created when there is a pending TCP request.

For any sort of UI, I have decided to use WPF with MVVM, using the Stylet framework (<https://github.com/canton7/Stylet/>). This was the clear choice for me, as MVVM means that multiple pages are able to be created without any sort of issue, and it allows for extremely to read and maintain code, especially when properties set to notify the UI when there is a change are created. MVVM also means that conductors can be used, so the initial viewmodel (known usually as the ShellViewModel) can essentially choose what page is being shown and when. In order to incorporate this into my program, I will have the ShellViewModel class incorporate the event handling interface and conductor interface of Stylet, which will mean that I can raise an event on one page saying to change page to a certain page when something has happened, and then the ShellViewModel will activate the new ViewModel due to the conductor, and so showing a new page. This will be used, for example, on a successful login attempt, where the LogInViewModel will raise an event for changing page to the page number of the main menu (corresponding to whether they’re a teacher or student). The ShellViewModel will handle this event and activate the ViewModel of the teacher/student page, showing the main menu.

In order for the quiz to be shown one question at a time, there will be a counter corresponding to what line of the quiz file that the program is currently on, which will increase when the “Next” button is pressed, and will decrease when the “Back” button is pressed. Then, whenever a question is to be displayed, all of the information for the question will be read from that line that the counter corresponds to, due to the file format for the quizzes being one line per question (apart from the metadata).

In order to have the results of a quiz exported as a CSV, I will need to create a class to handle CSV creation, which will create a CSV file with the headers for student name, mark and percentage. The reason for creating a CSV file is that it is easily opened in Excel and other spreadsheet programs. It will also be easy to create a CSV file compared to other file types like .xls, as all CSV requires is a string of text, a comma and another string of text, with a new line for each new entry, which can be easily created using a loop to get the data for each of the lines, the StringBuilder() class to collect all of the data together to write to the file, and then the File.WriteAllText function. This CSV processing will be done server-end, as it will be easier to make SQL calls from the server-side. The finished CSV file will be transferred over TCP to be received by the client.

For some of the features of the program such as having quizzes in your to-do section, I will need to create separate tables for each user. These tables will be created upon user creation, and having these separate tables for each user for a specific purpose such as all the quizzes that the teacher has set them will make it so that loading all of the required information will be faster than having an all in one set quiz table with every user’s set quizzes, which would take a while to sort through and would be inefficient.

# Design

In the program, the MVVM architectural pattern will be used, meaning that each “page” of the program (such as a main menu, or the log-in page) will have 3 elements to it – a Model, a ViewModel and a View. The Model is essentially just a class of properties that both the View and ViewModel are able to read without interfering with the MVVM pattern (as the View and ViewModel shouldn’t be able to interact directly with each other). The ViewModel is where most of the logic takes place and handles all of the events and other functions in the program, and the View is the actual UI itself and also dictates what should happen when a button is clicked, for example.

However, the MVVM pattern cannot function without a graphical system, and for this I will be using WPF in conjunction with the framework Stylet. WPF is the follow up to Microsoft’s WinForms, with the main difference being the introduction of XAML, which forces a separate file for the View to be made in the application. For general ease of programming, I will be using Stylet, which is similar to the popular Caliburn Micro framework where it essentially makes using MVVM and WPF a lot easier. For example, Stylet handles all of the binding between ViewModel and the View, meaning that when a new object of a ViewModel is made in a conductor (which in my case will be the ShellViewModel), the respective View is also called.

The quizzes in this program will all be stored as .zip files, with the name of them being the ID of the quiz. This is for the simple reason of there being a Microsoft library for opening and extracting the contents of .zip files, and it will be easier to transfer one file over TCP/IP than it will transmitting multiple files, as there is a “SendFile” function in the C# System.Net.Sockets library. Inside the .zip files, there’ll be a “questions.txt”, “answers.txt”, “metadata.txt” and “questiontypes.txt”. The questions.txt file will contain all of the questions, with each new line being a new question; the answers.txt file will contain all of the answers (if it’s an instant quiz) with each new line being an answer to the respective question; metadata.txt will contain the time allocated to the quiz and questiontypes.txt will dictate what type of question each question is, for example if it’s a multiple choice question or an integer input question.

The database for Tackle will be using SQLite. This is because of the speed of using SQLite, and it also alleviates the inconsistencies and latency of using MySQL, which runs as a separate server on the server side. All of the tables and rows will be contained in one .db file to ensure that, when necessary, the relational database model is used, which is not possible when multiple database files are used.

On the server side, whenever a command is sent to the server via TCP/IP from the client, if the command requires something to do with the database, such as logging in, a new database connection will be established to the SQLite database using System.Data.SQLite. This library ensures the easy use of reading of data from a SQLite database using SQLiteDataReader, and also can execute SQL queries to the database.

For logging in, to ensure that the user’s username and password hash is identical to the details they supplied when they signed up, this SQL query will be run: “SELECT \* FROM Users WHERE username = '{username}' AND password = '{password}'” (where {} are the C# variables to be queried).

For signing up, this simple SQL query is run: “INSERT INTO Users (Username,Password,UserType) VALUES ('{username}','{password}',{userType})”, with userType being either 0 for a student, and 1 for a teacher.

The way in which the server will check and execute commands has to be optimised for checking for and handling multiple requests at once, otherwise the server response time for clients will be extremely high, especially with more use. As a result, a new thread will be created that calls the function to handle the client’s request. The use of threads means that instead of the checking of whether a client is trying to connect and each individual request handling are separated and able to be executed at the same time, much like how asynchronous functions work by separating one specific task from the UI thread and other task threads.

Also in regards to the client-server communication, JSON will be used to transfer objects of classes which will be serialised at the client side and then deserialised on the server side. For example, when the quiz is submitted, there’ll be a QuizResults class:

public class QuizResults

{

public int quizID;

public string username;

public string quizType;

public string[] questions;

public string[] answers;

public int correctTotal;

public List<bool> correct;

}

When serialised to JSON on the client side, it will have this structure:

{

"quizID": 0,

"username": "teststudent",

"quizType": "Instant",

"questions": ["What is 2+3?"],

"answers": ["5"],

"correctTotal": 1,

"correct": [true]

}

This will be using the “Newtonsoft.Json” library due to its ease of use for this task. For example, on the server side, deserialisation from JSON to a C# class object is all one in one line:

##### var clientRequest = JsonConvert.DeserializeObject<ServerRequest>(jsonReceived);

This use of serialisation ensures consistency on both client and server side by using the same classes, and it also means that when deserialised on the server side, it is a lot easier to just read a variable from a class object than it is having some sort of array and having the values stored in that way.

The reason for using JSON is that JSON is able to handle the use of arrays (with the parameters for sending information to the server being enclosed in a string array), whereas XML is not able to. Also, JSON has the advantage of being faster than XML due to XML taking up more characters by being more verbose, and so JSON parsing libraries tend to be faster. Also, the storing of JSON due to it being less verbose in the database will take up less storage and will take less time to transfer from server to client.

THINGS HUGELY CHANGED:

Not using .zip anymore, now using the database’s “Quizzes” table and storing everything as JSON in QuizContent row due to the ability to store arrays

# Testing

### Test plan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test no. | Objective | Test | Type | Test Data | Expected Result |
| 1 | 1a | Test whether a student can sign up with the expected response | Normal | Username: “student1”  Password:  “student123” | The new user should appear in the Users table in the database |
| 2 | 1a | Test whether a teacher can sign up with the expected response | Normal | Username: “teacher1”  Password:  “teacher123” | The new user should appear in the Users table in the database |
| 3 | 1a | Test the response of the program if a username and/or password hasn’t been entered | Erroneous | Username:  null  Password:  null  Username:  null  Password:  “123”  Username:  “test”  Password:  null | There should be a dialog box informing the user that they need to enter a username and/or password |
| 4 | 1b | Test if a student sign up redirects to the student main menu | Normal |  | The student should be redirected to their main menu, with the contents seen in objective 2b |
| 5 | 1b | Test if a teacher sign up redirects to the teacher main menu | Normal |  | The teacher should be redirected to their main menu, with the contents seen in objective 2a |
| 6 | 1c | Test if the program rejects a pre-existing username when signing up | Normal | Username:  “student1”  Password:  “test123” | There should be a dialog box informing the user that that user already exists |
| 7 | 1d, 1f | Test if a student can log in with a correct username and password and be redirected to the student main menu | Normal | Username:  “student1”  Password:  “student123” | The student should be redirected to the student main menu after logging in |
| 8 | 1d, 1f | Test if a teacher can log in with a correct username and password and be redirected to the teacher main menu | Normal | Username:  “teacher1”  Password:  “teacher123” | The teacher should be redirected to the teacher main menu after logging in |
| 9 | 1e | Test if the program rejects an incorrect username and/or password | Normal | Username:  “wrong”  Password:  “student123”  Username:  “student1”  Password:  “wrong” | There should be a dialog box informing that the username or password is incorrect |
| 10 | 2a | The teacher main menu should all relevant options (see objectives) | Normal |  |  |
| 11 | 2b | The student main menu should all relevant options (see objectives) | Normal |  |  |
| 12 | 3a | Test to see if the teacher can create a quiz with a standard amount of questions, sufficient time allocated, an instant or non-instant quiz and different questions and question types | Normal |  |  |
| 13 | 3a | Test to see if the teacher can create a quiz with a large amount of questions | Extreme |  |  |
| 14 | 3b | Test to see if the teacher can send a quiz to their class | Normal |  |  |
| 15 | 3b | Test to see if the teacher can make a quiz public | Normal |  |  |
| 16 | 3c | Test to see if the teacher can save a quiz as a draft | Normal |  |  |
| 17 | 4 | Test to see if the teacher can open their draft quizzes | Normal |  |  |
| 18 | 4 | Test to see if the teacher can edit their draft quizzes | Normal |  |  |
| 19 | 4 | Test to see if the teacher can delete their draft quizzes | Normal |  |  |
| 20 | 5a | Test to see if the user can normally take a quiz (as per objective 5a) | Normal |  |  |
| 21 | 5b | Test to see if the user can quit the quiz | Normal |  |  |
| 22 | 5b | Test to see if the user can select “Quit” but change their mind | Normal |  |  |
| 23 | 5c | Test to see if finishing the quiz works normally (as per objective 5c’s requirements) | Normal |  |  |
| 24 | 6 | Test to see if a student can check their homework and take it | Normal |  |  |
| 25 | 7a | Test to see if a user can browse public quizzes | Normal |  |  |
| 26 | 7a | Test to see if a user can search for existing public quizzes | Normal |  |  |
| 27 | 7a | Test to see if a user can search for non-existing public quizzes | Erroneous |  |  |
| 28 | 7b | Test to see if a student can take a quiz from the public quizzes page | Normal |  |  |
| 29 | 7c | Test to see if a teacher can take a quiz from the public quizzes page | Normal |  |  |
| 30 | 7c | Test to see if a teacher can edit a quiz from the public quizzes page | Normal |  |  |
| 31 | 7c | Test to see if a teacher can send a quiz to their class from the public quizzes page | Normal |  |  |
| 32 | 8a | Test to see if the teacher can check their to-mark list | Normal |  |  |
| 33 | 8b | Test to see if the teacher is able to mark the student’s attempt question by question | Normal |  |  |
| 34 | 9a | Test to see if the teacher can see their set quiz history | Normal |  |  |
| 35 | 9b | Test to see if the student is able to see their quiz history with the amount of marks they got for each quiz | Normal |  |  |
| 36 | 9c | Test to see if the teacher can export student results of a quiz as a .csv | Normal |  |  |
| 37 | 10a | Test to see if the teacher can create a class | Normal |  |  |
| 38 | 10a | Test to see if the teacher can remove a member from their class | Normal |  |  |
| 39 | 10b | Test to see if a student can request to join a class given a correct ID | Normal |  |  |
| 40 | 10b | Test to see if a student can request to join a class given an incorrect ID | Normal |  |  |

### Test evidence

**1)**

**2)**

**3)**

**4)**

**5)**

**6)**

**7)**

**8)**

**9)**

**10**

**11**

**12**

**13**

**14**

**15**

**16**

**17**

**18**

**19**

**20**

**21**

**22**

**23**

**24**

**25**

**26**

**27**

**28**

**29**

**30**

**31**

**32**

**33**

**34**

**35**

**36**

**37**

**38**

**39**

**40**