Candidate Num: 5462

Center Num: 29319

Aaran Tukhar

Thomas Telford school

Computer Science Revision Aid

Component 3 - Project

Contents

[Analysis 3](#_Toc180054375)

[Problem Identification 3](#_Toc180054376)

[Why use a computational approach? 4](#_Toc180054377)

[Identifying Stakeholders 5](#_Toc180054378)

[Interview 7](#_Toc180054379)

[Interview Questions 7](#_Toc180054380)

[Results 10](#_Toc180054381)

[Analysis 12](#_Toc180054382)

[Research & Investigation 13](#_Toc180054383)

[Seneca 13](#_Toc180054384)

[Physical Revision Guides 22](#_Toc180054385)

[Physics and Maths Tutor 27](#_Toc180054386)

[Essential features for my product 31](#_Toc180054387)

[Solution 31](#_Toc180054388)

[Limitations of Proposed Solution 32](#_Toc180054389)

[Software & Hardware Requirements 33](#_Toc180054390)

[Success criteria 34](#_Toc180054391)

[Design 36](#_Toc180054392)

[User Interface Design 36](#_Toc180054393)

[Home page 36](#_Toc180054394)

[Revision notes 36](#_Toc180054395)

[Quiz 37](#_Toc180054396)

[Usability Features 37](#_Toc180054397)

[Decomposing Problem with Structure Diagram 39](#_Toc180054398)

[Prototype 1 40](#_Toc180054399)

[Pseudocode 40](#_Toc180054400)

[Test table 42](#_Toc180054401)

[Login & Sign-up UI-Design 43](#_Toc180054402)

[Prototype 2 45](#_Toc180054403)

[Database 45](#_Toc180054404)

[Test table 46](#_Toc180054405)

[Prototype 3 47](#_Toc180054406)

[Test table 48](#_Toc180054407)

[Variable table 49](#_Toc180054408)

[Login Variables 49](#_Toc180054409)

[Revision Material Variables 50](#_Toc180054410)

[User Progress Variables 50](#_Toc180054411)

[Quiz Variables 50](#_Toc180054412)

[Class Variables 51](#_Toc180054413)

[Other Databases 51](#_Toc180054414)

[Development Cycle 51](#_Toc180054415)

[Bibliography 52](#_Toc180054416)

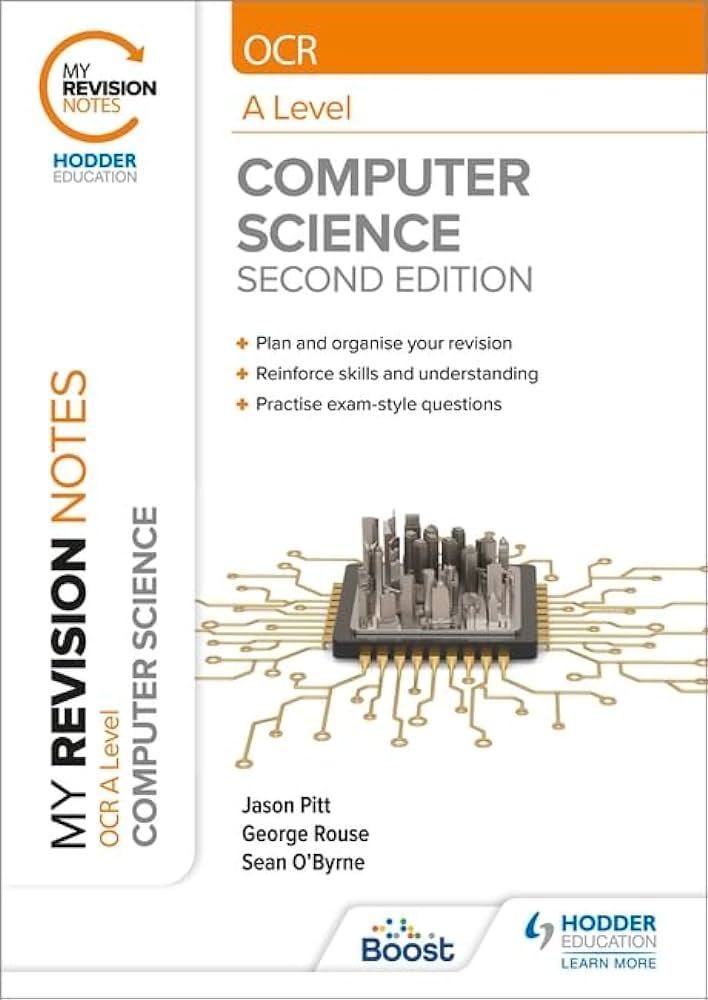
# Analysis

## Problem Identification

Tracking your progress whilst revising can be extremely difficult, especially for computer science where component one contains a lot of content. A revision aid has many benefits such as organising content in an efficient manner, being able to test users on the content and showing areas of improvement for the student.

A normal revision aid is usually designed for students to use independently, which is a problem because it hard for teachers to intervene and help the student. For example, a teacher may want to give tasks based on the topics their students are struggling on. Physical revision aids lack a sense of personalisation, for example, if a teacher had limited revision guides that students had to share it between students it would be hard to annotate some notes without ‘hindering other students’, however for the digital revision guide, users would have different accounts to log into and annotate how they see fit. Additionally, most revision aids leave it up to the consumer to decide when they have mastered a topic, which is a problem because they may find one topic easy leading them to think they have mastered the topic when they haven’t and there are more complexities to the topic than the user thinks. A huge problem with revision aids is that the quiz questions for each topic do not have variations, it is usually the same questions every time the consumer sits the quiz, which can make the user memorise answers for specific questions rather than actual course content.

When my project is complete, the user will be able to have a successful digital revision aid, which they can use to prepare themselves/their students for their examination.



[1] **Figure 1 – A typical OCR A-level Revision Guide.**

### Why use a computational approach?

The hardware that a computer offers, e.g. a hard drive as it can store vast quantities of digital data, can help reduce physical clutter for the revision aid and improve organisation. For example, 1.1.1 Structure and Function of the CPU is a topic that has many complexities like the fetch, decode, execute cycle and a lot of information about individual registers. This means that revision aids like flashcards can easily get lost and disorganised. The revision aid will store both content and questions on a hard drive, which can be used to make quizzes or be used for learning.

A computational method that can be used for creating the revision aid is decomposition, breaking down a complicated problem into small tasks. This project can be decomposed into sections such as content management, user interface design, quiz functionality, progress tracking, and database management. Each of these components can then be tackled individually, making the development process more organized and manageable.

Furthermore, to get the project completed in time I would have to be thinking ahead, carefully planning how I complete the project. For example, it may not be feasible to flashcards and notes for ALL computer science topics, so I should only consider only covering a few topics, thus having time to create the revision guide.

Pipelining, to allow things to be done simultaneously, can also be used within the revision aid, as answering questions could take time, meaning whilst waiting for an input, the program could also be calculating how long the user takes to answer the question, how much of the quiz they have completed, etc.

The task of automatically outputting topic notes can be programmed by storing the name of the file as a variable, reading the file, and then outputting line by line the text document. However, the questions will have to be stored within a database so the user can filter out specific topics, and it is easier for the program to randomise the questions the user is given during the quiz. For the questions it will require the use of SQL statements to grab the specific questions from the databases so that quizzes can be randomised.

A digital computer science revision aid will have random quizzes so that it can offer a different level of personalisation that a physical one may not have. For example, if a user is weaker in one topic, then the product will then have a quiz available for the user based on their weaknesses and the weaknesses can be found based on which quiz they have performed the lowest on. This means that the user can easily identify any weak areas of their knowledge in computer science and intervene. This can be done using a program and assigning an integer value to the score and using selections statements to determine weak areas compared to others. Also, if users do not score over a benchmark score it will also be classified as a weak area.

Physical revision aids typically do not give teacher enough room to intervene and boost student progress, which is why a computational approach is suitable. Using a digital computer science revision guide means there can be login systems which several types of user accounts. Students and teachers will see different layouts of the product, e.g., a teachers will be able to see a class leaderboard, as well as topic areas which the class are weakest at. This can be done through using databases and selections statements to verify whether they are a teacher or student. If it were a physical revision guide, they would have to manually check a student’s self-reported progress, which may not always be accurate and would be time-consuming. The use of abstraction, focusing on the important and needed parts of the revision aid, will be particularly important when making the revision guide to only focus on specific specification content rather than additional info that may not be necessary, which is another computational solution that the solution lends itself to.

Digital revision aids include log in systems which means it is easier to have the revision aid personalised to each individual student. If it were done with a physical revision aid, it would be hard for students to do the same piece of work/look at the same revision guide without purchasing multiple copies, and it would be less personalised to that specific student.

## Identifying Stakeholders

The clients and demographic targeted would be the teachers or students of an OCR A-level Computer Science Class. Since there are individual differences between each student it means that there will be a representative sample which can range from a student who struggles to understand the content of component one to another student who is good at component one but is weaker at component two content.

The stakeholders for this software are mostly teachers, as it helps monitor class progress and can identify when they need to intervene, so a student is “on track.” This aids teachers significantly because if students were to do a typical class test, it may not include all the content, which means it may not be able to identify weak areas, whereas, doing quizzes on every topic can help identify all weak areas.

The content and flash cards can all be stored on the digital revision aid. For example, 1.1.1 Structure and Function of the CPU has a lot of content on individual registers and each stage of the fetch-decode execute cycle, which means it can be a struggle to organise and store and then collect all the information. Using a digital revision aid means that flash cards or revision notes can be brought out at any time using the computer, rather than finding page numbers in revision books which could take time, and most revision books do not contain easily accessible flashcards unlike the product.

A physical revision aid means that it cannot be put on a larger interactive whiteboard for students to see, whereas a digital revision aid can be displayed on one for all students to view and interact with. An application of this would be to put a quiz on the board for all students to partake in and verify class knowledge of the content. This is an advantage to teachers as not only does it help students learn content, but also encourages class participation and engagement.

The product will also be appropriate for teacher usage as it will allow them to intervene in weaker topic areas that the class struggles on, as well as being able to set tasks that students can do, which means it will aid the teaching experience in general. The stakeholders for this demographic will be Mrs McAndrew. Mrs McAndrew is an experienced teacher who has worked in the education system for 4+ years and has used comparable products that aid students in computer science revision and exam preparation, which is useful because it means she can provide a lot of feedback going forward.

Another Stakeholder would be an A-level Computer Science student as they will use the revision aid to revise for their exam next year. Another client that I have is Filip Haase, who scores well on component 2 topics but not as well on Component 1 topics like ‘Structure and Function of the CPU’. My third client is Joesph Bond who scores well on Component 1 but not on component 2 topics. These two clients will give me useful feedback on my revision aid as they will try and use the revision aid to improve their weaker areas.

## Interview

### Interview Questions

My questions for Mrs McAndrew (as a computer science teacher) are:

1. Are you satisfied with your current teaching resources for computer science?

* This will help identify Mrs McAndrew’s current situation and the need for a revision aid for use in the teaching of computer science.

1. Have you used other digital revision guides in the past? e.g. Seneca, Smart revise.

* This will showcase expectations of the revision guide and past usage of other revision guides. This is particularly useful because it gives me a benchmark of what should be expected from a revision guide.

1. If yes, how was the experience and how did they aid teaching?

* This will give me an idea of what issues they had with other revision aids in the past, but also allow me to implement what was good about the revision aids in my product.

1. What do you think of having flashcards, quizzes and revision notes all stored on one application?

* The questions are more focused on the actual functions of the product, which will tell me if the client is actually happy with the main idea of the product (the product being a revision aid)

1. What do you expect to be in a computer science revision aid?

* This shapes an idea of what functions the revision aid should of have and what the client wants from a typical revision aid.

1. When would you use a revision aid?

* This gives an idea of what situations the client would use the product in and help tailor the product to have functions that are specific towards those situations.

1. Do you have anything else to add?

My questions for Filip and Joesph (as Computer Science students) are:

1. Are you satisfied with your current revision resources for computer science?

This will help identify Filip’s and Joesph’s current situation and the need for a revision aid for use in the revision aid of computer science.

1. Have you used other digital revision guides in the past? e.g. Seneca, Smart revise.

This will showcase expectations of the revision guide and past usage of other revision guides. This is particularly useful because it gives me a benchmark of what should be expected from a revision guide.

1. If yes, how was the experience and how did they aid revision?

This will give me an idea of what issues they had with other revision aids in the past, but also allow me to implement what was good about the revision aids in my product.

1. What do you think of having flashcards, quizzes and revision notes all stored on one application?

The questions are more focused on the actual functions of the product, which will tell me if the client is actually happy with the main idea of the product (the product being a revision aid)

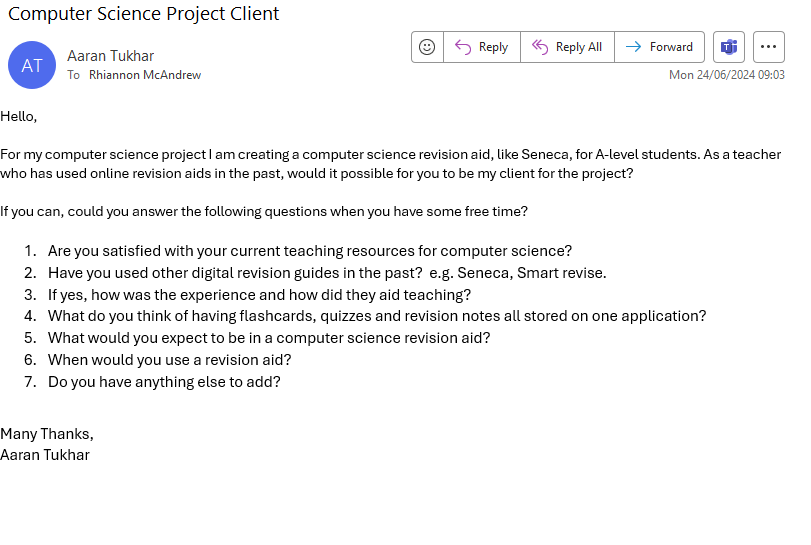
1. What do you expect to be in a computer science revision aid?

This shapes an idea of what functions the revision aid should of have and what the client wants from a typical revision aid.

1. When would you use a revision aid?

This gives an idea of what situations the client would use the product in and help tailor the product to have functions that are specific towards those situations.

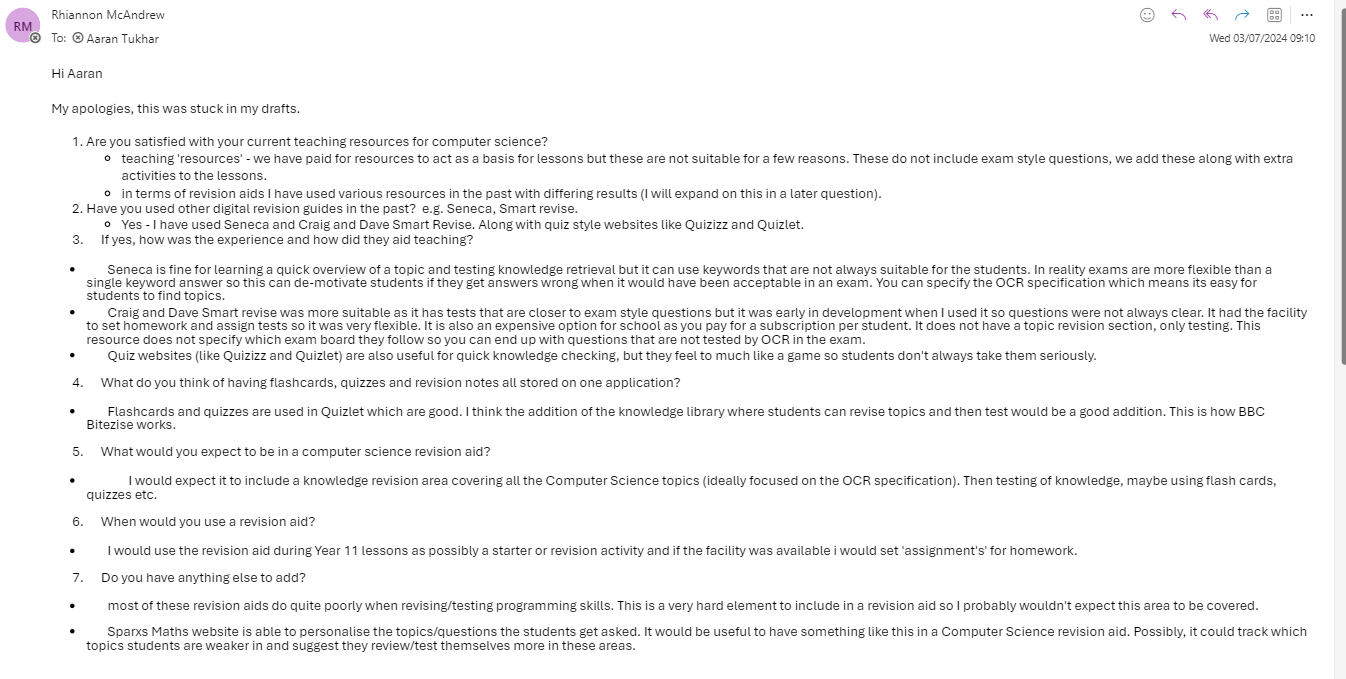
1. Do you have anything else to add?



[2] **Email to Client**

* Verbally asked Filip and Joesph and they responded to all questions.

### Results



[3] **Response from Client**

**Full transcript**

1. Are you satisfied with your current teaching resources for computer science?

Teaching 'resources' - we have paid for resources to function as a basis for lessons, but these are not suitable for a few reasons. These do not include exam style questions; we add these along with extra activities to the lessons.

In terms of revision aids I have used various resources in the past with differing results (I will expand on this in a later question).

1. Have you used other digital revision guides in the past? e.g. Seneca, Smart revise.

Yes - I have used Seneca and Craig and Dave Smart Revise. Along with quiz style websites like Quizizz and Quizlet.

1. If yes, how was the experience and how did they aid teaching?

Seneca is fine for learning a quick overview of a topic and testing knowledge retrieval, but it can use keywords that are not always suitable for the students. Exams are more flexible than a single keyword answer so this can de-motivate students if they get answers wrong when it would have been acceptable in an exam. You can specify the OCR specification which means it is easy for students to find topics.

Craig and Dave Smart revise was more suitable as it has tests that are closer to exam style questions, but it was early in development when I used it, so questions were not always clear. It had the facility to set homework and assign tests, so it was very flexible. It is also an expensive option for school as you pay for a subscription per student. It does not have a topic revision section, only testing. This resource does not specify which exam board they follow so you can end up with questions that are not assessed by OCR in the exam.

Quiz websites (like Quizizz and Quizlet) are also useful for quick knowledge checking, but they feel too much like a game, so students do not always take them seriously.

1. What do you think of having flashcards, quizzes and revision notes all stored on one application?

Flashcards and quizzes are used in Quizlet which are good. I think the addition of the knowledge library where students can revise topics and then test would be a good addition. This is how BBC Bitesize works.

1. What would you expect to be in a computer science revision aid?

I would expect it to include a knowledge revision area covering all the Computer Science topics (ideally focused on the OCR specification). Then testing of knowledge, maybe using flash cards, quizzes etc.

1. When would you use a revision aid?

I would use the revision aid during Year 11 lessons as possibly a starter or revision activity and if the facility were available, I would set 'assignments' for homework.

7. Do you have anything else to add?

Most of these revision aids do quite poorly when revising/testing programming skills. This is a hard element to include in a revision aid so I probably would not expect this area to be covered.

Sparxs Maths website can personalise the topics/questions the students get asked. It would be useful to have something like this in a Computer Science revision aid. Possibly, it could track which topics students are weaker in and suggest they review/test themselves more in these areas.

**Filip and Joesph Interview key points**

* They want a digital revision guide which they can access locally, and one that can monitor their progress.
* They want it to include past papers, revision notes and flash cards like physics and maths tutor
* Joesph specifically highlighted the need for it to be interactive and to include component 2 topics.

### Analysis

**Overview**

This interview highlights Mrs McAndrew’s and the students view on effective and ineffective computer science revision resources. This will help build a picture of what the desired features are and their undesired features. In this questionnaire, Mrs McAndrew has offered her experience as a computer science teacher, as well as Joseph’s and Filip’s concerns in their questionnaire.

**Client’s Desired Features**

* Exam style questions
* Any resources should ideally include exam-style question practice; in virtually all cases, these are slightly different from actual exam questions.
* Topic-specific materials
* It should be possible to target smaller groups of concepts, e.g. those that correlate with the OCR specification.
* Flexible answer acceptance
* It should accept a range of 'correct' variations of wording or synonyms, avoiding frustration for students who do not hit on the exact keywords.
* Integrated Learning and Testing
* All-in-one platform that integrates revision notes, flashcards, and quizzes so that students can learn and practice in one place.
* Progress Tracking
* Tools whereby teachers can administer and track student testing. Although these are like those offered by Craig and Dave, Seneca, and Quizlet

**Undesired Features**

* Over-reliance on keywords
* Tools that penalize students for minute differences in wording will discourage students.
* Game-like quizzes.
* Can be encouraging but making the quizzes too game-like may make the students take the material less seriously.
* Lack of Specificity
* Questions and resources that end up foggy or not specific to the exam board may confuse students.

## Research & Investigation

I am looking at other revision aids because it will give me a view of already successful products that people utilise on a day-to-day basis and thus, I will be able to pick out the basic features that make all these revision guides/aids successful. I will be looking at both physical and digital revision aids.

### Seneca

**Overview**

A screenshot of a computer

Description automatically generated

[3] **A picture of the Seneca homepage.**

Seneca is an online revision aid, which not only offers a revision resources for computer science, but for other subjects such as psychology as well. Seneca offers a plethora of courses which means not only is just specialised for OCR Computer Science but other exam boards such a AQA as well. This means that it can target multiple demographics of students rather than just Computer Science students.

As you can see at the top of the homepage it offers different modes, which means it can be specialised for use in certain situations. e.g. Cram mode can be used the day before an exam. This is particularly useful as the typical mode only tests on the specific topic you requested and does not allow you to test yourself on multiple topics.

A screenshot of a computer

Description automatically generated

[4] A **screenshot of the assignment system.**

Teachers can add students to the class, monitor progress and set homework on different topics. Also, at the bottom of the homepage [2] you can see that there is a tab called “assignments” which means teachers can set tasks as homework and can even put deadlines for when it is due. The percentage on the right-hand side indicates how many questions the student got correct on the homework which is useful to the teacher because it can help monitor progress. Furthermore, teachers can schedule for when the assignments are released so they do not have to actively “be present” when the homework is set.

A screenshot of a computer

Description automatically generated

[5] **Leaderboard system on Seneca.**

Figure 4 shows a leaderboard system within a class, which can help teachers reward students for demonstrating high work ethic and overall engagement within their classes. This means teachers can also identify students who need to put in more effort in their revision and in class. Furthermore, the leaderboard system not only functions within classes but also functions as a friendly regional competition for schools in general which can motivate teachers and students to set more assignments and tasks on Seneca.

A login screen with blue and white text

Description automatically generated

[6] **Login System on Seneca.**

Seneca being an online revision aid means that it can be accessed on any device at any time, provided it has internet access. This means that you will not have to worry about downloading software nor worrying about “saving work.” In Figure 5, you can see the login system which has a “continue with google option” which eases the whole sign in/log in process. The login system means that users can have different accounts, which allows them to add different courses and offer an entirely different level of personalisation. Furthermore, if the teacher is provided emails they can add them to classes. *This is something that could only be done digitally, as with a physical revision aid it would be hard to separate each student work from each other’s without buying multiple copies.*

Additionally, the sign-up process is extremely easy as all users must do is input their first name and last name, as well as the email they want to use to log onto Seneca and then enter the password. The sign-up, like the login, includes a continue with google option meaning users only must click one button. As a result of this, Seneca benefits from new users being enticed and encouraged by the website to sign up and utilise the website.

A screenshot of a computer

Description automatically generated

[7] **Seneca’s XP system**

This XP System tracks how many questions/assignments the user has done and rewards them a certain amount of XP for each question done. This can help students monitor how effective their revision has been compared to the amount of time they have spent on the website; this means that the student can get an idea of how effective their revision has been. Furthermore, the XP can be used to compete between students, which allows friendly competition and can motivate students to use the website more and complete more questions whilst revising.

The XP system directly ties into the leaderboard system that they used; however, the difference is that the XP system is more personalised to the user and can individually track their own progress.

A screenshot of a computer

Description automatically generated

[8] **AI Assistant on Seneca**

The AI Assistant allows the user to ask specific questions about the topic, which allows them to understand the content better and it prevents them having to waste time to ask teachers or try find the answer on google. The AI can also simplify topics which means that users can quickly scan through concise information, which is especially useful in cram situations e.g. the day before an exam. *This feature is something that could only be done digitally.* The third feature that the AI offers is that it can explain notes in more detail, this will especially help users as most of Seneca’s revision resources are in bullet point form which may not have enough detail, especially for nine markers in the A-Level Paper 1 Computer Science Exam. Unfortunately, the AI assistant that Seneca offers is behind a paywall, so I could not find out more about the features that the AI offers.

Seneca offers a range of different questions which allow the user to memorise questions in different ways. This stops the user from building up tedium and getting tired of doing the same “A, B, C or D” type questions that most revision guides use. This allows the user to focus for longer and memorise more information. A screenshot of a computer

Description automatically generated

[9] **Multiple choice question on Seneca**

Seneca offering multiple choice questions allow the users to recall information and then answer the question in quick succession. This is especially useful for cramming as the user will not have time (especially near the exam) to waste time answering long answer questions that will not invoke recall of enough topics. Another benefit of Seneca using multiple choice questions is that it is versatile as they can be used for any topic and in most situations, which means the format can be used for revision across the specification therefore Seneca would have wasted less time developing each individual question. Additionally, multiple choice questions can be used repeatedly to assist with knowledge retrieval and retention.

However, there are limitations of using multiple choice questions, for example as I mentioned earlier, is that it can lead to tedium if they are done consecutively. This means students can lose motivation to revise and lose momentum which means that they will lack preparation for their exam. Another limitation of multiple-choice questions is that students will often gloss past the questions that they got wrong and will not fully reflect on why the answer is incorrect, this could mean that the student can make the same mistake in the actual exam. Furthermore, multiple choice questions allow the student to simply guess the correct answer which means that they are not actively recalling information. This means that content that is not ‘known’ will simply be classified as learnt, which means that Seneca will not detect the content they guessed as a ‘weaker area.’

A screenshot of a computer

Description automatically generated

[10] **Correct order type question on Seneca**

These ‘correct order questions’ are especially good for computer science as it can be used for multiple topics, such as learning the fetch decode execute cycle, how a web browser can connect to a website through a URL and many more topics. A benefit of this question is that it allows the user to actively recall multiple steps of a ‘process’ or ‘action’ that is in the specification which means they will have better practice for the exam. Furthermore, if the question is difficult to answer because they do not know the content, they are more likely to reflect on it, unlike multiple choice questions, because it involves active thinking and deciding where the steps go. Additionally, these questions are harder to guess than multiple choice questions, which means that users are forced to understand and memorise the content in order to do the question properly.

However, a disadvantage to using these questions, that although that they can be used for a number of topics in computer science, they cannot be used for all topics, e.g. 1.5 Legal, Moral, Cultural, and Ethical issues. This means that they lack versatility. Additionally, these questions can be quite time-consuming which means that users may not be able to revise enough content in time, which means that doing many of these can be inappropriate for ‘cramming’ before an exam.

A screenshot of a computer

Description automatically generated

[11] **Fill in the gap question on Seneca.**

These fill in the gap questions have the important benefit of encouraging active recall by forcing students to retrieve material from memory without the use of prompts. Fill-in-the-blank questions also encourage more information processing, which is another advantage, as students will have to do the same in the A-level exam. In addition to memorization, students must also comprehend the content being studied when completing the question. Seneca is able to encourage critical thinking in their students by including important terms or specifics about topics covered in the computer science specification when creating tasks. This task forces students to consider their responses more carefully rather than just taking in the information that is put in front of them. Another advantage is that Seneca automatically marks the answer as soon as you put the correct answer which means it can save time for the student when revising.

A disadvantage of using these fills in the gap questions, specifically on Seneca, is that the content is right above the question, which means the user does not have to be critically think and instead can just find the answer. The sentences that they use for the fill in the gap are within the content above the question as well which means that a student may not be doing effective revision for the exam. Another disadvantage of using these questions is that the actual scoring of the question is complicated because you have to decide what counts as a correct answer or whether something deserves ‘half marks or partial credit. Luckily, Seneca counteracts this by auto correcting any incorrect spelling/synonym of the correct answer. However, it does eradicate the chance for partial credit as there is no option for the automatic marking to give the user ‘half a mark’ or partial credit. Another disadvantage of these questions is that it automatically enters and marks the correct answer, this means that users do not have to input incorrect answers and make a mistake, they are able to guess until they get the correct answer. For example, if a user entered ‘program’ in figure 10, the website would count it as the correct response for the second box, which means that even if they thought it was ‘program’ for the top box they would still get the answer correct. This means that users could easily gloss over topics that they are weak at simply because they got the answer accidentally correct.

A screenshot of a computer

Description automatically generated

[12] **Toggle question on Seneca**

In my opinion, using these toggle questions are ineffective because a user could blindly click on the toggles until the answer become correct. This means that the student is not actively engaging in the critical thinking, nor revising the content effectively. As a result of doing these questions, it means that they could lack actual preparation for the exam.

A screenshot of a computer program

Description automatically generated

[13] **Seneca’s revision notes**

At the top of figure 12 you can see that any major titles are highlighted in bold. This grabs the student’s attention at makes sure that they know that the content that they are revising relates to title. This is effective, especially for computer science, as topic names/information are quite similar e.g. Structure of the Central Processing Unit and Function of the Central Processing Unit, Memory Address Register and Memory Data Register.

The little description underneath describes what the actual content is about and gives the user an overview of what the notes are going to review. This means that the user will have a better understanding of what the notes are about, which can mean that there is less chance that they get confused. The font being smaller is effective because it immediately suggests to the student that the description is not as important as the main content.

In figure 12, they use an image of a diagram which shows the components of a CPU and how they are all connected. This gives the user a visual representation of the topic, and allows students, who may be visual learners, a better understanding of the topic. Furthermore, the image is colourful in comparison to the black text around it, to give it more attention and perhaps make the user look at the image first before reading the notes.

Seneca’s revision notes are also in bullet points which help the revision notes be more concise and specific to the specification. Additionally, this allows the user to be able to read the notes better. Additionally, Seneca use a font style known as ‘Seneca,’ with a size thirteen text to make sure students can read the information easily and quickly. In the top right corner, you can see a speaker icon, which will allow text-to-speech to start. This will help people with visual impairments, e.g. blindness. It also allows consumers to listen to the revision aid whilst doing another task.

#### **Suitable features from Seneca for my product**

* Login system
* Students’ revision aids can be personalised.
* Recognise a difference between teacher and student users when using the revision guide.
* Allow users to sign up and start using the product.
* Being able to give students tasks.
* Teachers can set homework for students.
* Can monitor students’ progress based on how many answers they got correct.
* Can set deadlines for these tasks.
* Multiple choice questions
* Good for quizzes
* Can help consolidate knowledge.
* Can be used alongside revision notes.
* Correct order questions
* Great for 1.1.1 Structure and function of the CPU questions
* Organised homepage layout
* Allows users quickly go to where they want e.g. revision notes or quiz.
* Gives good impression of the revision aid.
* Notify users on what the revision aid is about/for what it is used.
* Revision notes layout
* Including images and bullet point layout
* Concise information and specific to specification
* Easy and quick to read.

**Unsuitable features**

* Leaderboard
* Will not have enough users to make a leaderboard worthwhile.
* XP system
* Since I will not be including a leaderboard system in the product, this makes the XP system less useful, as it is harder to make a comparison to about a user’s progress to another user’s progress.
* AI Assistant
* Will be hard to implement within the time constraints of the project, as training of the AI is required to make it accurate enough to use.
* Fill in the gap questions.
* User’s may use different vocabulary to describe their answer.

### Physical Revision Guides

A stack of books on a table

Description automatically generated[14] **Computer Science Guides**

#### **OCR A-level Computer Science – PG Online**

A page of a computer program

Description automatically generated

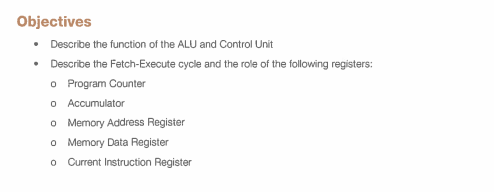
[15] **Table of contents**

**Table of contents**

A table of contents is a useful feature that every physical revision guide uses. It is particularly useful to readers as they can quickly get to the topic that they need to revise, and for new students to A-level Computer science they are able to get a general overview and gist of what the specification covers. Furthermore, a table of contents helps increase the overall perception of quality that the reader has off the revision guide, as it feels more organised and professional, thus it feels like they are compensated for what they paid for. It also makes long revision guides like the one in figure 13, more reader friendly as it can be complicated to sort through all the information without a guide.

In figure 13, we can see that this revision guide is laid out in chapters. This gives a clear division between topics and helps users distinguish topics from each other as they can cover similar information. Furthermore, readers can have a mental break after reading every chapter, as they can always come back to what chapter they were reading. Also, if readers were to communicate with each other about where a specific computer science topic was in book, they would be able to reference it by chapter number rather than telling the other person the whole name which can take time. Additionally, the revision guide uses page numbers and sections to further organise the document and make the reader experience as efficient as possible.

The use of bold text to outline the chapters gives the book a professional layout and grabs the reader’s attention to make sure that they know what the chapter includes and what the chapter number it is. Each section also has a subheading, which gives the reader a general idea of what computer science topic the section relates to. The alternating colour scheme that each section helps the document be less bland and mundane to read, which can increase concentration when revising. Furthermore, the font style that PG Online utilise is unique and easy to read which helps readers to understand what each topic is about.



[16] **Objectives at the start of every topic**

A diagram of a bus

Description automatically generatedPG Online include objectives at the start of each topic to notify the user that after reading this chapter they should be able to do what the objective says, which helps give a clear direction to the student when revising. Furthermore, the use of bullet points, when outlining the objectives, gives the student some sort of order to learn the content in and concisely describes what the student needs to be able to do. This also allows students to check their progress when revising and check how many objectives have been achieved.

[17] **Images**

The use of images in the physical revision guide allows information to be represented as a picture to help the user to understand the concept and content of the computer science specification better. The use of images also grabs students’ attention which helps keep them focused rather than having to read lines of lines of text, which can be monotonous. Like in Seneca they use bright colourful images, as their target audience is students are usually 15-18 years old and they may not have the attention span to read for long, to refresh the student’s interest in the topic. The text at the bottom describes the image so students are not confused at what the image is trying is trying to show them.



[18] **Questions**

By asking students questions at the end of each topic/sub-topic, they could test their knowledge of the content to see if they met the objectives that were listed at the topic of the page. Additionally, asking students questions allows the author to see if their revision notes were sufficient. Furthermore, when the student gets the question right, they get a sense of accomplishment which means that they feel more motivated to revise.

The yellow box around the question clearly distinguishes the question from the actual revision content that they need to learn, this means that students will not get confused and take longer to learn content. The bold text of “**Q2”** clearly tells the student that it is a question and that they may answer it. However, there are disadvantages with the layout of the question, one being that it uses vague words like “this process,” which may not be specific enough and the student may get confused as to which process it is talking about. Another being that there are two questions on the same line, the student may think that the answer will be similar for both questions, and completely miss out the first/second question which means that it will not be effective to test the student’s knowledge on the topic.

#### **OCR GCSE Computer Science Revision Guide– CGP**

The CGP Computer Science guide serves as a wonderful resource with features that I am applying to my own project. Firstly, I noted how its content is well structured. For example, complex topics are divided into clear and concise sections in such a way that key concepts and exam relevant material are dealt with. This approach eases comprehension as well as retention of the material by students. Therefore, I will incorporate this in my revision aid where information will be organized into manageable portions focusing on vital points for easy navigation and revision by students.  
  
Furthermore, another strength with this guide is that it uses student friendly language. Often, this guide speaks to readers in a conversational manner and at times humour is added to make it more engaging as well as less scary. This way of addressing things really works when trying to clear up difficult ideas while making it easier for learners to absorb the content being taught. I intend to adopt a similar writing style in my own revision aid document that uses simple language together with relatable examples.

The CGP guide often contains diagrams, flowcharts, and illustrations that go hand in hand with the text to explain complicated processes as well as render abstract concepts concrete. In computer science, where visualizing algorithms and data structures can significantly improve comprehension, these visuals become increasingly useful. Just like in my project where I would incorporate the same type of visual tools such as diagrams and infographics to complement the writing so that students would understand better.

Another important aspect of the CGP guide is its practice questions and exercises which I find very important. The guide has a variety of questions ranging from multiple-choice to longer exam-style ones that are key to reinforcing learning and preparing students for their exams. Thus, I will also include different types and levels of complexity in this revision aid. Furthermore, I will provide immediate feedback on answers, together with explanations for such answers students’ errors through which they could learn more about them thereby improving their understanding.

#### **OCR A-level Computer Science – ClearRevise**

ClearRevise’s A-Level Computer Science revision guide by OCR serves as a valuable model for my own project. One of its standout features is its precise match with the OCR specification. The guide guarantees that all its content is highly relevant to the actual exams, making it a powerful revision tool. In my project, I aim to stick closely to the exam board's requirements to ensure the content is suitable for students' study needs.

The simplicity and clarity of the information presented in ClearRevise is another point of inspiration. It divides challenging subjects into straightforward, bite-sized pieces, often using bullet points and key terms for emphasis. This helps students to quickly grasp the essentials without feeling overwhelmed. In my work, I will use a comparable approach, keeping information clear and concise with bullet points and highlighting to make key facts easily noticeable.

The visual aspect of ClearRevise’s guide is also quite impressive. It uses a neat layout with a lot of diagrams and flowcharts to show difficult concepts in a simple way. This is particularly useful in computer science, where seeing how things like algorithms and systems work can really help understanding. For my project, I aim to have a similar visual style, using diagrams, flowcharts, and other visuals to explain the material effectively.

ClearRevise is also excellent for practice. It has a wide selection of practice questions that are just like the ones on the real exams, which helps students get used to what they will face. Plus, it gives detailed explanations for each answer, which helps students learn from their mistakes and get better at answering questions. I will include practice questions with explanations in my project to help students reinforce their learning and feel more prepared for exams.

Furthermore, ClearRevise has a feature that lets students check their progress. It has sections where students can rate their understanding of different topics and see how they improve over time. This helps them figure out where they need more work. For my project, I will think about adding something like this, so students can keep track of their progress and know where to put more effort. The guide has summary sections that gather the main points of each topic into brief notes. These are great for quick review before exams and help students to remember important facts. In my project, I will include summaries like these to give students a fast and easy way to go over the main information.

### Physics and Maths Tutor

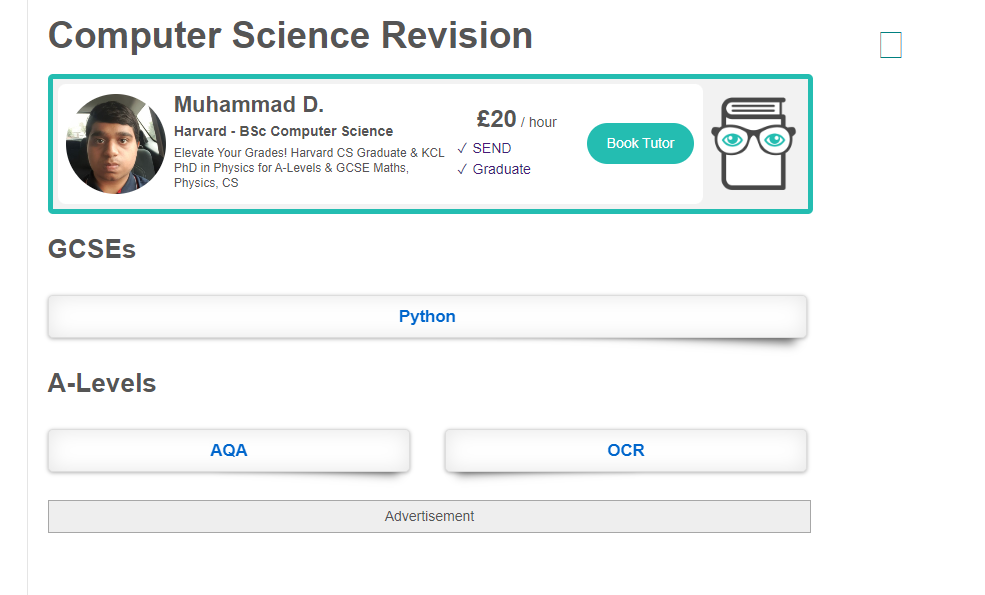
**Overview**

A screenshot of a computer

Description automatically generated

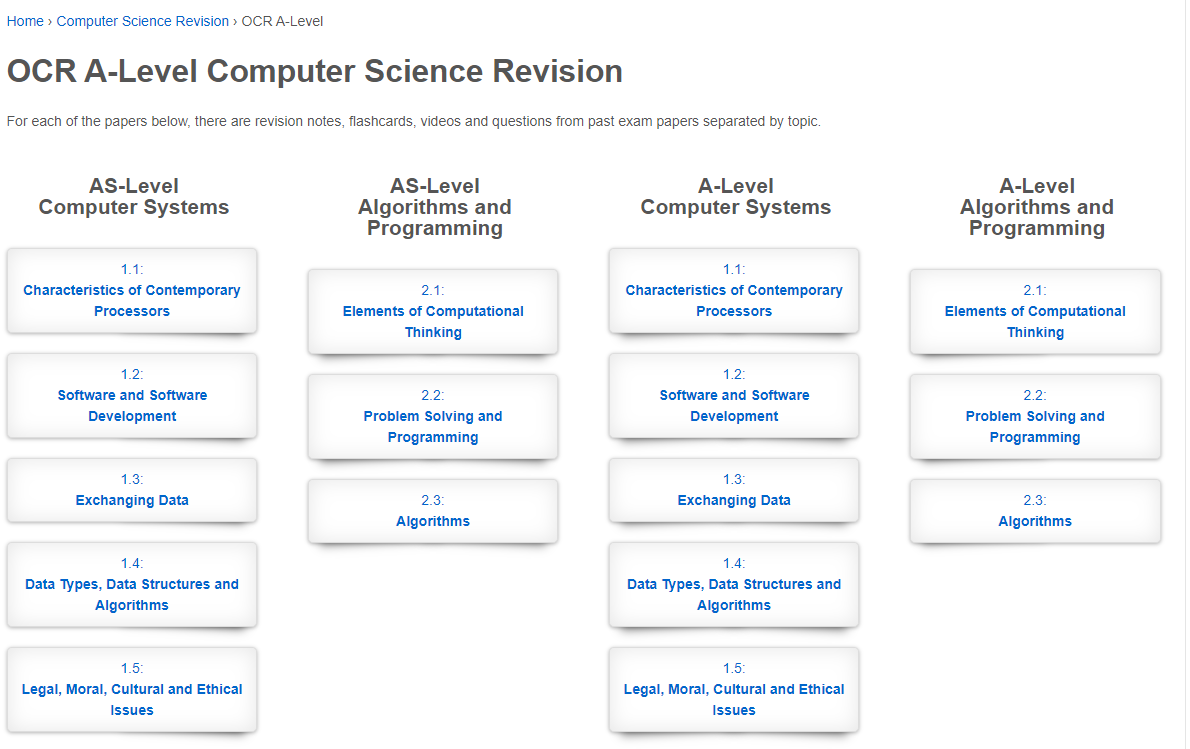
[19] **Physics and Maths Tutor**

Physics and Maths tutor is a prestigious website the students across the UK use for revision. The reason it has so many users is because they have revision notes, flashcards, and exam papers all on one website. The revision notes and flashcards are something I want to implement in my product. As you can see the website has a neat layout, with the main purpose of the website, the revision section, being in the middle so students can access revision notes quickly. Physics and maths tutor also offer revision aids for multiple subjects which is why it has garnered many users. It also targets GCSE students, AS level Computer Science, A-Level Computer Science. The advantage of having so many subjects for the students is that they do not have to look on separate websites and waste time. PMT also offer a ‘find a tutor’ service where tutors can advertise their service which can help overcome the limitations of independent learning. Additionally, unlike physical revision guides, most of, if not all, resourced on physics and maths tutor are free.



[20] **Physics and Maths Tutor Computer Science Revision**

This part of the website is dedicated to resources for the computer science revision part of the resource, where there is clear mention of supporting students from as early as GCSE to A-Level. It directly indicates to the user that that is what this page is all about: — Computer Science Revision—developing learning resources and activities in relation to computer science. It closely aligns with the idea of helping students understand the subject better and doing well in their exams. At the top of the page, there is a tutor profile: introducing "Muhammad D." with a Bachelor of Science in Computer Science from Harvard University. These types of qualifications will add credibility to the tutor, which will help to attract students in need of very high standards and expertise. Along with the name and qualifications of the tutor, a clear statement of hourly rate at £20 is quoted to reveal transparency about the real cost of the tutoring service. The "Book Tutor" button is a good trigger action to click for learners and parents to directly engage with the tutoring service from the page. The website is thoughtfully organized into sections that cater to different educational levels. The first row is marked for GCSEs. It contains one button called Python. It means that in this row, there are resources for the Python programming language because it is part of the core content for GCSE Computer Science. The second row is for A-Level and it groups two buttons marked AQA and OCR.

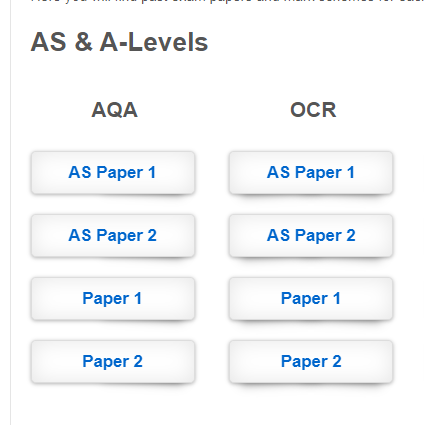
[21] **Physics and Maths Tutor Computer Science Revision**

This OCR A-Level Computer Science revision page is well-presented in structure, which will allow the students to study through the curriculum with an effective approach. The page will be divided into four key columns: AS-Level Computer Systems, AS-Level Algorithms and Programming, A-Level Computer Systems, and A-Level Algorithms and Programming. The large categories generalize to small specific topics, and each point is drawn up with a code, for example, "1.1" or "2.1," to show its position or level in the syllabus. The constant level of labelling and structure ensures that this will be easy for a student to pinpoint exactly what they are looking for in review content.

The AS-Level sections are developed around themes like "Characteristics of Contemporary Processors," "Software and Software Development," "Exchanging Data," and "Legal, Moral, Cultural, and Ethical Issues." This means that the same topics reappear in the A-Level sections but are discussed more in-depth, for students to realize the heavier expectations that go from AS to A-Level. Ease of use is enhanced, as the similar layout can guide the student through different topics and levels easily, which is so critically important during successful revisions.

Design-wise, the page is neatly laid out, and each topic presents itself rather nicely as a click-button, suggestive of the fact that a student would be able to open detailed resources or revision materials related to an exact topic. The uniformity of the buttons' layout and labelling of the same guarantees that students can relate to and locate the materials they need without confusion. Above all, students benefit most from this ease since they shall get the focus to their studies without being disoriented by a maze of a website layout.

One of the most significant pros for the educational value of the page is that these are very important topics for OCR A-Level Computer Science. The page includes topics from both AS and A-Level. By including them on the same page, it is evident how the ideas introduced in the AS topics are built upon at the A-Level. Alongside the columns, it was mentioned that a mixture of resources was suggested to each topic, which includes revision note, flashcards, video clip, past exam questions, and so forth. This way, the students of all kinds, like those who learn better by reading the written note, by practical flashcards, or through printed questions, all can easily find the revision process helpful for them.



[22] **Physics and Maths Tutor Computer Science Revision Past Papers**

Above is a part of the website to help students navigate through the AS and A-Level exam papers for the AQA and OCR Exam Boards. It has two clear columns for AQA and OCR, respectively, each with buttons labelled "AS Paper 1," "AS Paper 2," "Paper 1," and "Paper 2 which are linked directly to the appropriate examination papers. When the user clicks on any of the text it will take them to a page where they can see past years exam papers.

The layout is basic; students can find what they need: the exact papers required, as quickly and painlessly as possible. In fact, the simplicity of this site allows for the focus to be exactly where it should be allowing students easy access to all their revision materials.

### Essential features for my product

* Topic-Specific Resources
* PMT organizes materials by specific topics, allowing students to focus on areas where they need the most help.
* Past Papers and Mark Schemes
* The site provides extensive past papers and mark schemes, helping students practice exam techniques and understand how marks are awarded.
* Variety of Resources
* PMT includes revision notes, flashcards, and videos, catering to different learning styles and making it easier to grasp complex concepts.
* Navigation
* The website is user-friendly, with a clear layout that allows students to quickly find the resources they need.

#### **Unsuitable features**

* Limited Interactivity
* The resources on PMT are mostly static, with few interactive elements like quizzes or practice tests that provide instant feedback.
* Basic Design
* The website's design is quite plain and can feel outdated, which might not engage students as much as more modern, visually appealing platforms.
* Lack of Advanced Content
* PMT mainly focuses on core topics, so students looking for deeper, more challenging material might find it lacking in advanced resources.
* Overwhelming for New Users
* The sheer amount of content can be overwhelming for first-time users, making it difficult to know where to start.

**so that they may get to the needed resources and revise efficiently for their exams.**

## Solution

1. Making the revision guide interactive.

In my product I will integrate a variety of interactive features into the platform, such as quizzes, practice tests, and flashcards with instant feedback. This approach will actively engage students and enable them to evaluate their comprehension in real-time, allowing for the adaptation of their study methods accordingly. Additionally, these elements will contribute to a more engaging and less repetitive learning environment. Reminders and study goals will be included.

1. Effective Design

I will conduct a thorough design of the app with a focus on aesthetic appeal and usability. Incorporate responsive design principles to ensure compatibility across different devices, enhance the visual experience with compelling graphics, and create an intuitive interface. This redesign will not only make the platform more appealing to students but also facilitate smoother navigation, leading to increased user engagement and satisfaction.

1. Content

I will broaden the resource selection to include advanced and rigorous materials, such as comprehensive revision notes, challenging practice papers, and flashcards. These resources will cater to students pursuing more ambitious academic objectives, providing them with the depth of preparation they seek. This helps me get closer to the desired features of both Filip and Joesph as my stakeholders.

1. Support for new users

I am going to be implementing an onboarding process that is both accessible and informative as well as offer detailed tutorials to help new users become accustomed to the platform with ease. This is a weakness of Seneca which I have decided to improve in my own revision guide.

1. Personalized Recommendations

In my revision aid I am going to create a recommendation system that uses algorithms to suggest pertinent resources and study materials tailored to individual user progress, preferences, and educational needs. Personalized suggestions will improve content discovery and tailor the learning journey to the specific requirements of each student. For example, the last resources that the user used should show up in the home page.

## Limitations of Proposed Solution

Since I need to complete this project on my own and quickly, I will need to think about what is possible and make some practical changes. Making the app interactive with quizzes, practice tests, and flashcards that give feedback right away is hard because of how much I know about coding and the time I have. Instead, I will work on simpler interactive parts, like basic quizzes, that I can handle with my skills. I will also look for tools and libraries that have some of these features already, so I do not have to build everything from scratch. This way, I can keep the app interesting without spending too much time on it.

Designing the app to look good and be easy to use is also a challenge. It needs to look modern and work well on all devices. To manage this, I will go for a clean and simple look that is easy to use. I will use templates and frameworks to make the design process quicker. This will help me meet the basic design standards without spending too much time on it.

Adding lots of advanced learning materials like long articles and hard practice tests makes the project more complicated. So, I will start with a smaller set of important stuff and add more later when I have more time. I will use simple ways to handle the content, so the app stays easy to use and works well. Creating a detailed guide for new users with lots of tutorials and a big help section would take too long. To keep it simple, I will make a basic guide with quick tutorials. This will help users get started without getting lost or confused.

Lastly, making a smart system that suggests resources based on what the user likes is pretty complex and needs a lot of coding knowledge. For now, I will stick to a basic system that recommends content based on what users choose. It will not be as fancy, but it will still give some personalized options without all the extra work.

## Software & Hardware Requirements

Since I will be making the project in Java, for someone to run my project the requirements are modest.

The software where I will be making the product will be in Apache NetBeans.

<https://netbeans.apache.org/front/main/download/>

To run the software the user will need to install the Java Virtual Machine (JVM):

[Java Downloads for All Operating Systems](https://www.java.com/en/download/manual.jsp" \o "d)

**Processor:**

* Intel Core i3 or equivalent

**Memory (RAM):**

* 2 GB DDR3

**Storage:**

* 1 GB or more (to allow for additional space for logs, temporary files, and future updates)

**Operating System:**

* Windows 10 or later, or Windows 7 with updates.
* Linux

**Screen resolution**

* 1280 x 800 minimum screen resolution.

**Java Runtime Environment**

* The JRE version required by my application needs to be installed (typically Java 8 or later).

**Mouse and Keyboard**

* Need to a basic USB/Wireless Keyboard & Mouse to interact with the GUI.

## Success criteria

1. The app must give users topic-specific resources.

**Justification**: It should have materials sorted by subject or theme, so users can focus on what they need help with. Also, a client-desired feature (seen on Pg 9)

1. Must have a Graphical User Interface

**Justification:** The users need the GUI to use for users to not build up tedium

and for the revision aid to qualify as being interactive. Successful in online revision guides like Seneca (seen on Pg 11)

1. The app needs to have at least 3 past papers and mark schemes.

**Justification:** This helps users practice for exams and understand how they are marked. I will check that the papers are new and cover many topics. Physics and maths tutor includes this feature and has is well received by users.

1. The app should have a mix of resources, like revision notes, flashcards etc.

**Justification:** Allows the user to learn through multiple methods, which is something that my clients wanted.

1. The app must have a navigation system.

**Justification:** User must be able to navigate around. It should have a design that makes it quick to find what you need. Seneca has a good navigation system which I want to take inspiration from.

1. The app will have basic interactive features like simple quizzes or practice tests.

**Justification:** These might not be super advanced because of time available, but they will still be helpful and give users some feedback. I will make sure these parts work well and add to the learning experience. One of my clients, Filip, explained how he wanted this feature to practice his Component 1 skills.

1. The app will have a quick start guide and help section.

**Justification:** There will be simple tutorials and a FAQ area to help inexperienced users learn how to use it. This way, users can find help without any trouble.

1. My revision aid must allow the user to create individual accounts.

**Justification:** The user must be able to create accounts in order monitor individual progress and be able to get specific recommendations for weak topics. Most online revision aids include this feature because it can be really beneficial to students to have this level of personalisation.

1. The app must have a log in screen.

**Justification:** The user must be able to access their account. Essential for the clients to access their accounts.

1. The app must have a personalized dashboard.

**Justification:** Provide a simple dashboard that shows basic information like the user's name, recent activities, and quick links to resources. This helps users stay organized. Mrs McAndrew mentioned this issue when talking about Physics and Maths Tutor, where it is easy to get lost in the vast content they have.

1. Include a basic search functionality.

**Justification:** Implement a simple search bar that lets users quickly find specific resources or topics by typing in keywords. This makes navigation more efficient. One of the downsides of using PMT as it is difficult to find specific resources without going through each and every single topic, which I wanted to work on.

1. Allow users to set study goals and reminders.

**Justification:** Let users set basic goals with simple notifications. This helps them stay on track without needing complex scheduling. In Joseph’s interview, he outlined this as one of the strengths of the revision aid he was using called ‘Smart Revise’.

1. Enable integration with simple calendar features.

**Justification:** Allow users to mark important dates directly in the app, like exam dates or deadlines. This keeps them aware of key events without needing full calendar integration. Basic necessity for students using a revision aid to keep up time management whilst revising.

1. Incorporate a progress tracker.

**Justification:** This gives users feedback on their progress without requiring advanced analytics. Seneca include this in their website which I thought was something that I should include in my revision guide.

1. Teachers can set homework for students.

**Justification:** Set tasks for students to monitor their progress. One of the desired features that the client (Mrs McAndrew) wanted in order to test students.

1. Offer bookmarking of important resources.

**Justification:** Let users’ bookmark or favourite key resources they want to return to quickly. As a computer science student myself, I thought this would be useful as it is difficult trying to access specific resources on online revision aids especially for Component 2 topics.

1. Allow the user to have account management.

**Justification:** Provides options for users to update their profile, change passwords, or log out, they need to be able to manage their profile. Basic necessity if accounts and log-in systems are to be incorporated into the revision guide.

# Design

## User Interface Design

### Home page

A screenshot of a computer

Description automatically generated

This is the home page of the program. This will allow the user to access any area, through the navigation search bar on the top allows users to access any area. Every part of the program is easily accessible through the headings at the top, so the user can quickly and efficiently use the guide.

### Revision notes

A screenshot of a computer

Description automatically generatedThe revision notes button will take you to another page which will look like this:

This page will allow you to access the revision notes which are required for the student’s revision. Easy access for the student/teacher. The recently looked at notes will help the user track what they have already revised in the past.

### Quiz

The quiz button will take you to the quiz page where users can start a quiz which will look like this:A screenshot of a quiz

Description automatically generated

The mark button will immediately allow users to get feedback for their work. It will show the user a time count to make sure they aren’t spending too long on some questions. Lifetime shows what the user has done in total so far. They will be allowed to choose which topics their questions come from as you can see in the top right corner, also, they will be able to see what percentage of the questions they have gotten right to see how accurately they have been answering the project.

## Usability Features

For my revision aid, I will ensure that all user inputs are thoroughly verified via input validation. This will help reduce mistakes and improve the user experience by averting errors before they arise. I want the system to gently guide users and ensure they provide accurate information at each step without feeling overwhelmed.

I am going to create a user interface that is clean and inviting. By implementing a minimalistic yet appealing design, I create an environment that welcomes users while minimizing unnecessary distractions. My objective is to maintain an interface that is intuitive and easy to navigate, allowing users to concentrate on their revisions without being distracted by cluttered visuals.

To enhance usability further, I will introduce a streamlined registration process that demands minimal effort from users. Because I am simplifying the sign-up process, it will ensure that users can begin using the tool immediately without facing a lengthy or complicated setup.

My revision aid will be both practical and user-friendly. I am designing it with a focus on the user’s needs, ensuring it is intuitive to navigate and effective for learning. The objective is to develop a tool that enables users to quickly understand concepts and effortlessly review topics, making it a valuable aid in their study journey.

I will ensure that the design is responsive and compatible with various screen sizes. The resolution and layout will automatically adapt to suit the user's device, whether it is a computer or a mobile device. This ensures users will experience seamless functionality, regardless of their method of accessing the tool, guaranteeing clarity and optimal performance across all screens.

The menu system within the revision aid will serve a role in being able to navigate the app effortlessly. With a navigation bar on the top, and each item on the menu being clearly labelled, users can efficiently navigate the revision guide.

## Decomposing Problem with Structure Diagram

A diagram of a company

Description automatically generated

The main subsystems of the revision aid which I have chosen decomposed:

Login system: This subsystem handles user authentication. Breaking it down allows for secure handling of sensitive data like passwords, session management, and user credentials, which will be stored in a separate database as it would be too congested and complex to do this in the same subsystem. If I isolate the sign-up feature, the subsystem can focus on user data validation, error checking (e.g., weak passwords), and ensuring that user data is stored securely.

Main Program: The main program subsystem serves as the core interface for users. It includes navigational elements to guide users through the different areas of the revision aid. Breaks down the navigation from content and save functionality simplifies tracking the user’s progress and offering a smooth experience. (Broken down into smaller subprograms – some below)

* User statistics: Users need feedback on their progress to assess improvement areas, also there will be data from all over the program, e.g. time spent revising, quiz questions got right, so it makes sense to decompose it as its own sub-system
* Navigation system: The navigation system will be decomposed as it will be used everywhere in the program, so it needs to be worked on independently so I can ensure consistency across the revision aid as users move between topics and across the program.

Database: I will have a separate database module, as data can be efficiently managed and can be easily scaled, backed up, and accessed by the various subsystems (Main program e.g. Quiz, Login, User statistics, etc.). This separation also enhances security and performance.

## Prototype 1

In this prototype I will be focusing on creating the log-in and sign-up screen, making sure it is fully usable. I won’t be focusing on the database part of log-in system yet, but in prototype 2 I will complete the log in-system. The log-in screen will be the first thing the user will see when launch on the desktop. One feature I may add is a keep signed in feature, so users don’t have to keep signing in.

* The first subproblem I will encounter is creating an appealing login and signup interface which provides a smooth user experience. This will be by making sure the interface is easy to understand and is responsive
* The second sub problem is going to be handling user inputs. This is because I will need to capture user inputs accurately by handing form submission and managing navigation between different application sections. For this I will implement username/email and password fields and a login and signup button to allow the user to initiation these two processes.
* The last sub problem is going to be visual feedback and user interaction. The problem is providing visual feedback and ensuring a smooth interaction within the login page to the user. The challenges with this will be implementing animations to display user interactions and show the outcome of login attempts. I will solve this by including visual icons such as icons to show ongoing actions. I will also provide immediate feedback when users click the login button such as changing the button’s appearance or showing a brief animation.

### Pseudocode

**Login System**

A computer screen shot of a computer program

Description automatically generated

This procedure functions as the primary controller for the login and sign-up system. Initially, it presents the login interface and subsequently enters a loop, awaiting the user's decision to either log in or register. Based on the user's choice, it invokes the corresponding procedure, either Login or SignUp. This loop persists, guaranteeing that the system remains attentive to user interactions.

This pseudocode essentially covers what the first prototype will be, but I will provide some pseudocode for the functions like log-in and sign-up

**Login procedure**

A computer screen shot of a computer code

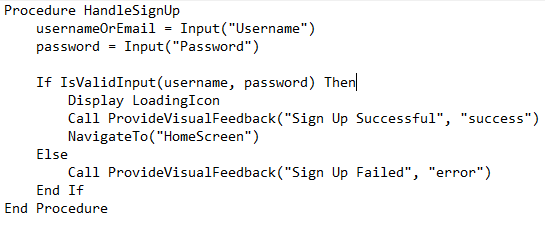
Description automatically generated

This procedure oversees the login process. It collects the user's information from the username/email and password fields, verifies the input, and offers visual indications regarding the success or failure of the login attempt. If the "keep me signed in" option is selected, it invokes a subroutine to preserve the session. Upon successful authentication, the user is directed to the home screen.

**Sign up procedure**

A close up of a sign

Description automatically generated



This procedure outlines the process for user registration. It gathers the necessary sign-up information, verifies the provided data, and offers feedback regarding the success of the registration attempt. Upon successful registration, the user is directed to the home screen, like the login procedure. This is just an outline of a basic sign-up procedure so exam date and class code are not on here, but they will be added in the main program.

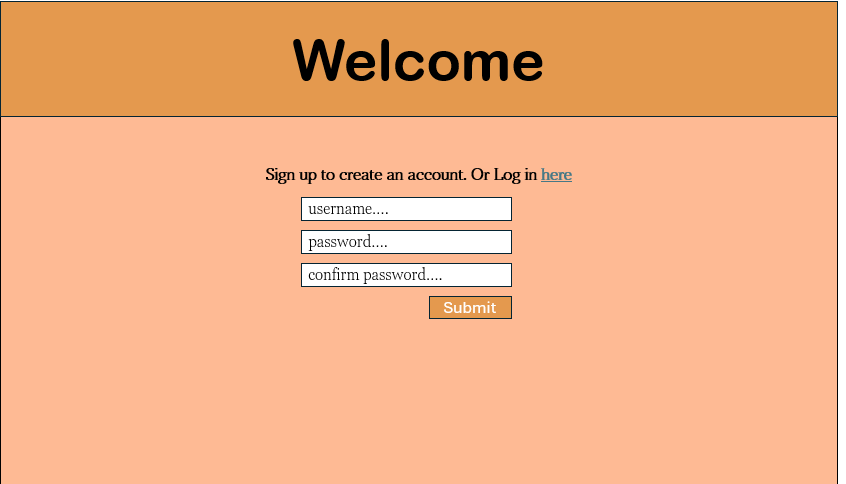
### Test table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test No. | Test Description | Input Data | Expected Outcome | Actual Outcome | Pass/Fail |
| 1 | Test if the login screen loads successfully | N/A | Login screen should display all elements (fields, buttons) correctly |  |  |
| 2 | Test login with valid username and password | username: user1  password: Password123 | Successful login, user is directed to the homepage |  |  |
| 3 | Test login with invalid username | username: invalidUser  password: Password123 | Error message "Invalid username or password" appears |  |  |
| 4 | Test login with incorrect password | username: user1  password: wrongPassword | Error message "Invalid username or password" appears |  |  |
| 5 | Test sign-up with a new username | username: newUser  password: NewPassword123 | User account is created, and user is redirected to login screen |  |  |
| 6 | Test sign-up with a username that already exists | username: user1  password: Password123 | Error message "Username already exists" appears |  |  |
| 7 | Test password field validation (min length) | username: newUser  password: short | Error message "Password must be at least 8 characters" appears |  |  |
| 8 | Test "Keep me signed in" feature functionality | username: user1  password: Password123  "Keep me signed in": checked | User stays logged in even after the app is closed and reopened |  |  |
| 9 | Test navigation from login to sign-up screen | N/A | User is directed to the sign-up screen upon clicking "Sign-up" |  |  |
| 10 | Test visual feedback on login button | username: user1  password: Password123 | Button shows loading animation until login completes |  |  |

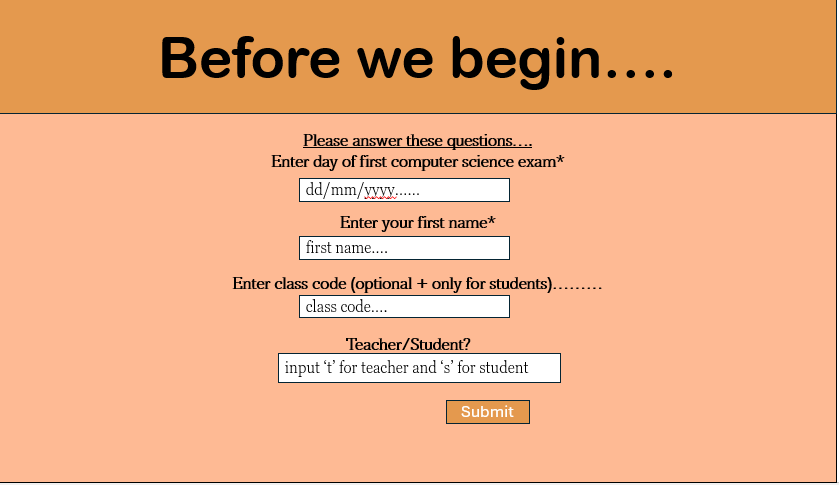
### Login & Sign-up UI-Design



Simplistic Log-in screen. Allows users to understand what to do when they get to this page. The ‘welcome’ display at the top in invites user into the revision aid and motivates them to revise. There is sign-up hyperlink which will take you into the sign-up page below:



Initial sign up details to get user to enter the details they will use to log-in. Submit button stands out against background, which will then take the user to a screen to submit extra details:



## Prototype 2

Prototype two will incorporate the storage of login credentials within the database, ensuring that passwords are hashed and that user permissions are assigned as either basic students/teachers or admin based on authentication.

* The first subproblem involves the creation of a functional database that securely stores user information. The challenges associated with this task include designing a database that can accommodate user data, such as usernames and hashed passwords, along with their respective permission levels. This will be achieved by developing a database structure that effectively stores user details and establishing a secure connection between the application and the database.
* The second subproblem pertains to password hashing. The challenges here involve selecting a strong hashing algorithm that can generate, and store hashed passwords, as well as validate these hashes during the login process. This will be addressed by opting for a secure hashing algorithm and modifying the login procedure to compare hashed passwords during user authentication.
* The third subproblem focuses on user authentication and permissions. The challenges in this area include the need to create a reliable authentication process that differentiates between basic students and teachers and admin users while enforcing the appropriate permission levels. This will be accomplished by implementing a user role or permission level within the database and assigning the corresponding permissions to users based on their designated roles.

### Database

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: User** |  |  |  |  |
| **Field** | **Key** | **Data Type** | **Validation** | **Notes** |
| userID | Primary | Number |  |  |
| username |  | String | 1-20 |  |
| password |  | String | >8 |  |
| userAccess |  | String |  | Determines whether user is ‘student’, ‘teacher’ or ‘Admin’ |
| examDate |  | Date | >=TODAY() |  |
| classID | Foreign | Number |  |  |

### Test table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test No. | Test Description | Input Data | Expected Outcome | Actual Outcome | Pass/Fail |
| 1 | Test if the database connection is established successfully | N/A | Application connects to the local database successfully |  |  |
| 2 | Test if user credentials are stored securely in the database | username: user1  password: Password123 | Username and hashed password are stored in the database |  |  |
| 3 | Test if passwords are hashed before storage | username: user1  password: Password123 | Password stored in the database is hashed and not in plaintext |  |  |
| 4 | Test login with valid username and password (hashed) | username: user1  password: Password123 | Successful login; hashed password is correctly validated |  |  |
| 5 | Test login with incorrect password (hashed) | username: user1  password: wrongPassword | Error message "Invalid username or password" appears |  |  |
| 6 | Test role assignment for a basic student | username: student1  password: Password123 | Student logs in and is assigned basic permissions |  |  |
| 7 | Test role assignment for a teacher | username: teacher1  password: Password123 | Teacher logs in and is assigned basic permissions |  |  |
| 8 | Test role assignment for an admin | username: admin1  password: AdminPass123 | Admin logs in and is assigned admin permissions |  |  |
| 9 | Test if permissions restrict access to certain features (student) | username: student1  password: Password123 | Student cannot access admin features |  |  |
| 10 | Test if permissions allow admin access to modify users | username: admin1  password: AdminPass123 | Admin can modify users and assign roles |  |  |
| 11 | Test login attempts with non-existent user | username: invalidUser  password: Password123 | Error message "Invalid username or password" appears |  |  |
| 12 | Test password change (hashing remains consistent) | username: user1  old password: Password123  new password: NewPassword123 | Password is updated in the database and hashed correctly |  |  |

## Prototype 3

At this phase of the project, the emphasis will be on developing an intuitive home page that enables users to access study materials and resources, incorporating features that enhance both content navigation and management. I am going to let admins be able to modify revision aid features.

* The first sub-problem will be designing the layout for the home page has already been accomplished; however, accurately coding it will require additional time and further research into Swing tutorials.
* The second sub problem will be the integration of the revision guide with a database is essential for the secure storage of study materials and tracking user progress. The primary challenge lies in ensuring that the system can efficiently retrieve and update content.
* The third subproblem will be that the system must also accommodate various user roles, including students and administrators, to effectively manage access to materials. This necessitates the creation of a secure framework that permits authorized users to modify content.

### Test table

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test No. | Test Description | | Input Data | Expected Outcome | Actual Outcome | Pass/Fail | |
| 1 | | Test if the home page layout for the revision guide loads correctly | N/A | Home page displays all elements (navigation buttons, topic sections, progress tracker) correctly |  |  |
| 2 | | Test navigation from home page to specific Computer Science topics | User clicks "Start Revision" button | User is redirected to the chosen Computer Science topic page |  |  |
| 3 | | Test if study topics are retrieved from the database | N/A | Topics and related content are retrieved and displayed on the home page |  |  |
| 4 | | Test if progress tracking updates correctly for students | username: student1  topic: Algorithms completed | User's progress for the "Algorithms" topic is updated and displayed on home page |  |  |
| 5 | | Test if students can access their revision progress | username: student1 | User can view their progress (completed topics, remaining topics) on the home page |  |  |
| 6 | | Test if administrators can add new Computer Science topics | username: admin1  new topic: Data Structures added | Admin successfully adds "Data Structures" as a new topic, stored in the database |  |  |
| 7 | | Test if administrators can modify existing topics | username: admin1  existing topic edited: Networking | Admin modifies the "Networking" topic, and changes are reflected in the database |  |  |
| 8 | | Test if students are restricted from modifying content | username: student1  attempt to edit topic: Networking | Error message or restricted access; student is unable to modify content |  |  |
| 9 | | Test if progress is tracked correctly after completing a topic | username: student1  topic: Databases completed | System updates the student's progress for the "Databases" topic in the database |  |  |
| 10 | | Test role-based access control (student vs admin) | username: admin1  username: student1 | Admin can modify or add topics, students can only view and complete topics |  |  |

## Variable table

These are the main variables that will be used in the program.

### Login Variables

|  |  |  |
| --- | --- | --- |
| **Variable Name** | **Data Type** | **Description /Justification** |
| username | String | The username of the user, allow users to log-in |
| password | String | The user's password (hashed), allow users to log-in |
| role | String | The role of the user (e.g., "student", "admin"), configures their permission level. |

### Revision Material Variables

|  |  |  |
| --- | --- | --- |
| **Variable name** | **Data Type** | **Description/Justification** |
| title | String | Title of revision material, lets user know what they are actually revising |
| content | String | Actual content of revision, lets user revise |

### User Progress Variables

|  |  |  |
| --- | --- | --- |
| **Variable name** | **Data Type** | **Description/ Justification** |
| userID | int | The unique identifier for the user, makes it easier to locate them in the database. |
| materialID | int | The unique identifier for the revision guide, makes it easier to locate it in the database |
| completed | boolean | Whether the user has completed the material already, stops users from backtracking and wasting time. |
| score | float | Score achieved by user on practice quiz, lets them monitor their progress. |

### Quiz Variables

|  |  |  |
| --- | --- | --- |
| **Variable Name** | **Data Type** | **Description/Justification** |
| quizID | int | The unique identifier for the quiz |
| totalQuestions | int | The total number of questions in the quiz |
| questionID | int | The unique identifier for the question |
| questionText | String | The text of the question |
| options | String[] | An array of answer options for the question |
| correctAnswer | String | The correct answer for the question |

### Class Variables

|  |  |  |
| --- | --- | --- |
| **Variable Name** | **Data Type** | **Description/Justification** |
| classID | int | The unique identifier for the class |
| totalAssignments | int | The total number of assignments |
| completedAssignments | int | Number of assignments completed |
| dueAssignments | String | Assignments Due |
| dateDue | Date | The date that the assignment is due |

## Class Database

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** |  |  |  |  |
| **Field** | **Key** | Validation |  | Notes |
| classID | Primary |  |  | This will be the code that the user will join through as well. There will be one class created when a teacher creates an account, they cannot have multiple classes. Which is a limitation of my project. |
| classList |  |  |  | List of students and the teacher of the class. Teachers will automatically be added to the list, |
| classSize |  | >0 |  |  |
| assignment |  | 1-200 |  |  |
| dueDate |  | >TODAY() |  |  |
| completed |  |  |  | For ‘revision note’ assignments, students should be able to click it for it to become true, and for ‘quiz assignments’ it will automatically become true when the quiz is completed |
| maxMarks |  |  |  | Total number of marks available for all the quizzes set by the teachers |
| Marks |  |  |  | Mark available for the individual  quiz |

### Entity Relationship Diagram

A screenshot of a computer

Description automatically generated

This entity relationship diagrams shows the link between the two tables. My primary key is the userID, and every userID can have one classID associated with it, whereas it is the opposite for classes. ClassID can be associated with many userIDs. The foreign key classID connects the user with the associated class, which will be useful for making the class system in the development stage as I can easily access the class database through the user database.

## Main program

A diagram of a flowchart

Description automatically generated

This is a flowchart that breaks down the main algorithm. My program will be more complex than this, which is a weakness of using the flowchart to decompose the main algorithm, but it has simplified the whole layout of the design which will make it easier for me in the development cycle.

## Final Test Table

|  |  |  |  |
| --- | --- | --- | --- |
| **Test No** | **Input** | Input | **Expected Output** |
| 1 | Test successful user registration (student role) | Username: "student1"  Password: "Revise446CS" | Registration confirmation message, user saved in the database |
| 2 | Test successful login with correct credentials (student role) | Username: "student1"  Password: "Revise446CS" | User is logged in and redirected to the main page |
| 3 | Test choosing Computational Thinking topic | Select: "Computational Thinking" | Computational Thinking topic page displays relevant revision content |
| 4 | Test choosing Algorithms topic | Select: "Algorithms" | Algorithms topic page displays with key revision material from the spec |
| 5 | Test quiz initiation for the Algorithms topic | Select: "Take Quiz" on "Algorithms" | Quiz on Algorithms starts, displaying the first question |
| 6 | Test quiz completion and score feedback for Data Types | Submit answers for "Data Types" quiz | Quiz result displayed, showing correct and incorrect answers |
| 7 | Test tracking progress after completing a quiz on Algorithms | Complete a quiz on "Algorithms" | User’s progress updated, showing quiz score and completion |
| 8 | Test error handling for invalid sign-up data (e.g., password too short) | Username: "newstudent"  Password: "short" | Error message: "Password must be at least 8 characters long" |
| 9 | Test scheduling a revision session for Networking topic | Select date/time for revision session | Booking confirmation message, session saved |
| 10 | Test that only admin users can add or edit revision content | Admin attempts to add new content for "Programming Techniques" | Admin is allowed to add content to the system |
| 11 | Test searching for a specific topic or keyword | Input: "Object-Oriented Programming" | Search results display relevant content from the H446 specification |
| 12 | Test logout functionality | N/A | User is logged out and redirected to the login screen |
| 13 | Test handling of incomplete quiz submission | Attempt to submit quiz without answering all questions | Error message: "Please answer all questions before submitting" |
| 14 | Test completion of progress tracking for System Software | Complete a quiz on "System Software" | Progress is updated and displayed on the user dashboard |

# Development Cycle

# Bibliography

1. Hodder Education. (2021). *My Revision Notes: OCR A-level Computer Science: Second Education*. [Online]. Hodder Education. Last Updated: 27 August 2021. Available at: https://www.hoddereducation.com/subjects/general/products/a-level/my-revision-notes-ocr-a-level-comp [Accessed 21 May 2024].

[2] Aaran Tukhar. (2024). Email to Rhiannon McAndrew. Computer Science Project. 24 June 2024.

[3] Seneca. (2024). *Homepage*. [Online]. Seneca. Available at: app.senecalearning.com [Accessed 4 June 2024].

[4] Seneca. (2024). *Assignments*. [Online]. Seneca. Available at: app.senecalearning.com [Accessed 4 June 2024].

[5] Aaran Tukhar. (2024). Email to Rhiannon McAndrew. Computer Science Project. 24 June 2024.

[6] Seneca. (2024). *Leaderboards*. [Online]. Seneca. Available at: app.senecalearning.com [Accessed 4 June 2024].

[7] Seneca. (2024). *Login.* [Online]. Seneca. Available at: app.senecalearning.com/login [Accessed 4 June 2024].

[8] Seneca. (2024). *Dashboard* [Online]. Seneca. Available at: app.senecalearning.com [Accessed 4 June 2024].

[9] Seneca. (2024). *Assignments*. [Online]. Seneca. Available at: app.senecalearning.com [Accessed 4 June 2024].

[10] Seneca. (2024). *Assignments*. [Online]. Seneca. Available at: app.senecalearning.com [Accessed 4 June 2024].

[11] Seneca. (2024). *Assignments*. [Online]. Seneca. Available at: app.senecalearning.com [Accessed 10 June 2024].

[12] Seneca. (2024). *Assignments*. [Online]. Seneca. Available at: app.senecalearning.com [Accessed 10 June 2024].

[13] Seneca. (2024). *Assignments*. [Online]. Seneca. Available at: app.senecalearning.com [Accessed 11 June 2024].

[14] moleculea. (2013). *Books*. [Online]. flickr. Last Updated: January 20, 2013. Available at: https://www.flickr.com/photos/moleculea/8396104239 [Accessed 25 June 2024].

[15] PM Heathcote & RSU Heathcote. (2016). *OCR AS and A Level Computer Science*. Dorset: PG Online. pp. IV-V.

[16] PM Heathcote & RSU Heathcote. (2016). *OCR AS and A Level Computer Science*. Dorset: PG Online. pp.2.

[17] PM Heathcote & RSU Heathcote. (2016). *OCR AS and A Level Computer Science*. Dorset: PG Online. pp.3.

[18] PM Heathcote & RSU Heathcote. (2016). *OCR AS and A Level Computer Science*. Dorset: PG Online. pp.4.

[19] Physics and Maths Tutor. (2014). *Computer Science Revision*. [Online]. PMT. Available at: https://www.physicsandmathstutor.com/computer-science-revision/ [Accessed 26 August 2024].

[20] Physics and Maths Tutor. (2014). *Computer Science Revision*. [Online]. PMT. Available at: https://www.physicsandmathstutor.com/computer-science-revision/ [Accessed 26 August 2024].

[21] Physics and Maths Tutor. (2014). *Computer Science Revision*. [Online]. PMT. Available at: https://www.physicsandmathstutor.com/computer-science-revision/ [Accessed 26 August 2024].

[22] Physics and Maths Tutor. (2014). *Computer Science Revision*. [Online]. PMT. Available at: https://www.physicsandmathstutor.com/computer-science-revision/ [Accessed 26 August 2024].