Quiz Qrafter

COSC 4P02 Project Proposal

Member	Student Number	Role(s)
Jason Parsons	6586861	Scrum Master, Product Owner, Project Manager
Nikos Dritsakos	6995021	Full-Stack Developer
Gideon Oludeyi	7333586	Back-End Developer
Akshay Rash	6830236	Full-Stack Developer
Kelvin Odinamadu	7063571	Full-Stack Developer
Jingyi Ma	6897185	Full-Stack Developer
Luke Reinaerts	6955421	Back-End Developer

GitHub page: https://github.com/QuizQrafter/quiz-qrafter

High-Level Proposal

This proposal is for the software project Quiz Qrafter. Quiz Qrafter will be an Artificial Intelligence-based learning tool that would exist to help students succeed in a variety of different subjects. The basic idea is as follows: a user selects or creates a course on Quiz Qrafter. The user can then upload their notes, lecture materials and past exams, tests, quizzes, assignments, or other assessment materials available to them. Then, Quiz Qrafter can generate one or many sample questions, formatted as an assignment or exam, to allow the user to practice and improve on the given topic. Quiz Qrafter would also provide a grading and answers system to allow users to see how well they did, however, this would come with a warning that due to the still emerging Artificial Intelligence technology, these answers may not be one hundred percent accurate.

Quiz Qrafter would be a powerful and useful tool to help encourage student success across a variety of different fields and subjects. Not only would it provide a way to help students prepare for exams, but it will also help students to think critically regarding their answers, as Quiz Qrafter will not be entirely accurate when it comes to assessing them, so students will be expected to analyze why their answer was marked “wrong” and determine if they made a mistake, or if there was an error on the software’s side. Quiz Qrafter will be a positive force for artificial intelligence in education and will make an excellent project for this term using relevant technology to create a powerful and useful tool.

Background Research

The field of student success has many studies that support the usefulness of the proposed software, ranging from its general effectiveness (Cogliano et al. 2019) to comparisons between different forms of practice tests (Naujoks et al. 2022) as well as comparisons to other forms of studying and learning (Adesope et al. 2017). In these fields, it is often referred to as “retrieval practice”, which yields many more works related to its effectiveness. Practice tests unfortunately can be quite difficult to find when not provided by the course, however, as any course will have a unique pace and style, as well as potentially unique syllabus patterns. Often students will be able to get access to a previous year's examination which can help greatly, but rarely will they be able to get more than one or two. We believe with the use of artificial intelligence this process can be made much more effective, with practice exams generated by a software as many times as the student feels they need. After some light research through regular searches for practice exams and practice exams integrating artificial intelligence, we were unable to find any sources indicating a project like this is currently underway. Additionally, when searching for similar tools, we were unable to find any sources of exams available in more than specific subjects, such as a certification exam or similar one-time examinations.

This research shows that the idea is relatively unique and has not yet been approached publicly, while also showing the value such a project could have on many students' education.

Team

Our team consists of 7 members, Jason will be taking on the roles of Project Manager, Product Owner and Scrum Master. This was decided due to the unique experiences of the team, as Jason has direct experience in these roles, while the remaining six individuals wanted to focus on development. We split the development team into two teams of three, as we decided it would be easier to work on features in smaller teams rather than having six team members trying to work on the same thing at the same time. Two individuals wanted to focus on the back-end more, Gideon and Luke, while the remaining developers wanted to do full-stack. We elected to make our sub-teams as follows: Nikos, Gideon and Akshay as team one, and Kelvin, Jingyi and Luke as team two. These are mostly informal teams, as there will be overlap and shared work, but this allows us to organize our development on individual features more effectively.

Features

There are a variety of features planned for this software both necessary to its functionality and more optional features for ease of usage or additional functionalities not critical to the software's primary purpose. The critical features include:

- Users will have the ability to upload and view notes
    - These notes will potentially be handwritten, so the software will need to be able to detect handwritten text. Because this may not always be perfect, it will also need to convert that into text and allow the user to modify it as needed.
- Users will have the ability to log in and access notes and courses on multiple devices
    - This is a critical feature as it will allow users' courses, notes and assessments to save, so they will not have to re-upload everything each time they wish to use the software.
- Users will have the ability to upload assessment materials
    - These will be used primarily in the generation of new assessment materials and will allow the software to understand the formatting that can be expected.
    - An optional sub-feature would be to make note of the professor who gave out the assessment material if exams or assignments from previous years are available. This would allow the AI to weigh the value and accuracy of generated assessments based on whether or not the professor is the same, as different professors may have different styles of assessment questions.
    - Another sub-feature would be making note of when in the term the assessment was given, this allows the assessment generator to generate test material based only on the knowledge expected up to a certain point, rather than the entire year. It would be more useful this way, as a user would not expect to see questions on topics covered in the final exam for their first test.
- Users will have the ability to generate assessment materials using Artificial Intelligence
    - This would be the primary value of the software, the ability to have the software generate assessment materials based on uploaded notes and

previous assessment materials would serve as the main feature to interest users.
    - These sample assessments would also feature a way to submit a completed version of the assessment, which the software would then grade using AI and provide feedback on whether answers are correct or not. This would come with a warning that the software will not be entirely accurate in all cases and subjects and that users are expected to analyze their answers critically.
    - Filters could be used to exclude or weight different materials uploaded to ensure a more accurate assessment, such as only using notes before a certain date, or excluding assessments from different professors.
- The software will be browser-based and will feature a landing page for users, as well as a user hub with functionality available once logged in.
    - This landing page would provide first-time users with information regarding Quiz Qrafter's function and purpose to help users decide whether they would like to create an account and take advantage of its features.

We have also identified a list of optional features that we may include in the final design or that could be added as additional functionality in future updates to the software. These include:

- Users could have the ability to see other users notes and assessment materials for the same or similar courses
    - This could be done by allowing the user to create their course under their school, selectable from a list or input by the user if the school is not in the list. Then, they could see notes and assessments from other students who are registered in the same course at the same school.
    - Additionally, there could be keyword-based search features to allow users to find similar courses covering similar content from other schools for more material.
    - This feature would also require the implementation of privacy settings for users' uploaded materials, sharing would be disabled by default, to allow users to keep their notes to themselves if preferred.
- The software could be able to provide summaries of notes or lecture slides
    - This feature would use artificial intelligence to create short summaries of the topics covered in different notes or uploaded materials like lecture slides. This would be useful to more quickly determine which sets of notes are important when studying specific topics.
- Users could access a personal settings menu to enable or disable different features
    - This could include the aforementioned privacy settings for uploaded notes
    - This would also include quality-of-life features, such as an optional "dark mode" to make the software easier to look at for extended periods.
- Admin users could have access to an admin console
    - This feature would allow admin users to access a special dashboard which could include analytical data about the use of the software or other useful information.
- Subject analytics could be available from the landing page or logged-in hub page
    - This would work with the optional feature for multiple users sharing data, it would allow users to know how many other users are uploading content that

is similar to their interests, and help them gauge whether it is valuable to use the shared data feature for those courses.

- Users may have access to a mobile app that could scan notes
    - This optional feature would allow users to scan notes on the go much more easily. While this mobile app wouldn't feature the full power of the software, it would allow the user to log in and scan notes to their courses, rather than having to get photos onto their computers and upload them manually.

Planning

We will be using a variety of tools to create Quiz Qrafters, most of which are free and easily accessible. This will include:

- OpenAI API
    - This will be our primary artificial intelligence system, responsible for compiling the uploaded user data and creating assessment materials, as well as handling features such as the grading system and summarizing content.
- Google Vision API
    - This will be used to read and translate handwritten notes and assessments into a format more easily understood by the rest of the software.
- React
    - React will be used as our front-end library.
- Firebase
    - Firebase will be used as our database and authentication service.
- Typescript
    - TypeScript will be our primary programming language.

We will be using GitHub as our primary hub for managing the software and related files, and we will be using the Scrum methodology for our software engineering process. We will be meeting twice a week to go over goals and current progress, broken up into Sprints. Sprints will last for two weeks, allowing for four meetings in every Sprint.

Sprint 1 will begin on Monday, January 22nd, and will end on Monday, February 5th. Because we have selected an Agile method for our development cycle, we have elected to leave future sprints intentionally vague to allow for a more reactive approach as we encounter difficulties or complete features faster than expected. The timeline is laid out as such:

Sprint	Goals	Milestones
Sprint 1 (Jan 22 - Feb 5)	-Landing Page -Project Setup	Release Planning Document
Sprint 2 (Feb 6 - Feb 19)	-Authentication/Login	
Sprint 3 (Feb 20 - Mar 4)	-User Dashboard	Progress Report 1
Sprint 4 (Mar 5 - Mar 18)	-Upload Documents	

	-Google Vision Implementation	
Sprint 5 (Mar 19 - Apr 1)	-Generate Assessments	Progress Report 2
Sprint 6 (Apr 2 - Apr 15)	-Optional Features/Catch Up	
Sprint 7 (Apr 16 - Apr 29)	-Optional Features/Catch Up	Final Presentation + Final Report

Conclusion

Quiz Qrafter is a software project that utilizes artificial intelligence to improve students' education and ability to learn without the drawbacks of allowing students to “cheat” their way through courses by simply providing the answers for them. Quiz Qrafter would encourage critical thinking regarding students' responses to questions, which would provide a more in-depth learning experience while providing a resource that students would be able to use to see an improvement in grades. There have been multiple studies that support the usage of practice tests and exams having a direct correlation with final grades, however, there exists minimal online resources to help students in this way. Through Quiz Qrafter, students would be able to generate their own, working on as many practice tests as they need until they believe they are prepared to handle the real test or exam. Quiz Qrafter will encourage student success and incorporate artificial intelligence in education as a positive resource, rather than being used as a source of cheating. Quiz Qrafter could have a strong positive impact on the educational community and we believe it is the perfect project to work on throughout the term.

References

1 - Cogliano, M. C., Kardash, C. A. M., & Bernacki, M. L. (2019). The effects of retrieval practice and prior topic knowledge on test performance and confidence judgments. *Contemporary Educational Psychology, 56*, 117–129.

2 - Naujoks, N., Harder, B. & Händel, M. Testing pays off twice: Potentials of practice tests and feedback regarding exam performance and judgment accuracy. *Metacognition Learning* **17**, 479–498 (2022).

3 - Adesope, O. O., Trevisan, D. A., & Sundararajan, N. (2017). Rethinking the Use of Tests: A Meta-Analysis of Practice Testing. *Review of Educational Research*, *87*(3), 659-701.