REQUIREMENTS: “ASC” THE AI STUDY COMPANION

CMSC 4900 – Senior Project

TEAM MEMBERS:

Cameron Calhoun	CMSC // Specs & Implementation
Gage Keslar	CMSC // Presentation
Jonathan Buckel	CMSC // Analysis
Seth Morgan	CMSC // Design

Instructors Comments/Evaluations

Keep Empty!

Abstract

The A.I Study Companion, ASC, program will be a browser-based application that will serve the user as a study aid. The intended users are school staff, students and parents of students that want to either better their own knowledge or give a helping hand to those who need it. The user will be able to evaluate their knowledge by inputting a subject and receiving quiz questions back. The users can track their progress between sessions, print flashcards, and receive feedback on subjects. The program will be able to: generate questions and answers by using an LLM, give suggestions on where the user needs to focus their attention, be able to generate files that work as flashcards or save data to client side. This document defines ASC and functions as a plan for how the development of this project will continue.

Introduction

Background

A common issue for the average student is studying: how to begin studying, development of good studying techniques, and lack of guidance without a knowledgeable source around. Those close to the student, like a parent, become disheartened, hearing of their student's struggles. They may seek the help of a tutor, but that can be an expensive task. Instead, ASC will be able to provide insight and knowledge that can improve the user's knowledge, enforce good study behaviors, and provide guidance to those seeking it.

Objectives

The main objective of the ASC team is to create an application that uses an AI Large Language Model , LLM, that will serve as a study partner. Using JavaScript, with React framework, we will be able to create the GUI for our browser-based application. The user's data will be saved in local storage to keep track of their current subjects of interest and subjects that need improvement. As for the backend, we will be using Python to communicate with an LLM. It will first prompt for a subject and difficulty, output a question to be responded, feed that answer back into the LLM and it will then determine if the answer given was correct or false. Depending on what the LLM has decided, that will affect the subject's score in a positive or negative way. Depending on the subjects score and the number of questions and answers exchanged, ASC will prompt the user to improve the difficulty of the subject, change focus of a subject, or determine that the user needs more improvement before working on harder questions.

In addition, the application will also be able to generate flash cards for a subject that may be printed out. The flash card features will prompt a subject and will then generate keywords, with definitions, and frequent questions to be formatted onto a document to be printed. The user may also specify questions and answers to be on the printout. It will format the data into reasonable portions to fit standard printer paper size, 8" x 11", that the user can cut into flash cards to study on the go or for term memorization.

Team Details

The team consists of four CMSC majors, each of them have been assigned a phase of leadership for each step in the development process, as seen in Fig 1.1. Responsibility will fall on the team members of the current phase while all other team members work towards their overall goal and ideas of current phase. To ensure that the best work is done in a timely manner, we have created a Discord server with channels to allow the sharing of documents and helpful resources and direct communication between all members online. In addition, the version control software Git will be used in the development of our codebase and will be hosted on GitHub.

Team Member:	Major:	Assigned Leadership Phase:
Cameron Calhoun	CMSC	Specs & Implementation
Gage Keslar	CMSC	Presentation
Jonathan Buckel	CMSC	Analysis
Seth Morgan	CMSC	Design

Fig 1.1 - Team Member Majors working on the project and their leadership phase.

Application Domain

Project Context

The point of ASC is to provide study aid to those who need help studying, no matter what subject, grade level, school, or background. Few people are able to afford a tutor if needed, and there are some who cannot leave the house to seek tutoring elsewhere due to medical ailments. While “ASC” may not replace a tutor, it would be a good application for those who are seeking help. With the developments of artificial intelligence growing, it has the potential to give insight into any subject it has acquired. We can use that potential knowledge to increase our own understanding of how the world works in ways that can stimulate the brain and promote learning.

Glossary: SEE APPENDIX A

Initial Business Model

Operational Environment

ASC will be designed to be used by not only students, but also parents, teachers, and other academic influences in a student's life such as tutors. Given the broad domain of users, we need to ensure that ASC is as easily accessible as possible. This directly influences our decision to host the application on the web, as most families and individuals have access to an internet connection. If even that is not available, users will still be able to access ASC through a public internet connection. Given the portability of the software, carrying over local files is an important feature. This will be accomplished by allowing users to export their current progress in a portable format, which can then be imported in a separate instance of ASC.

Description of Data Sources

ASC will retrieve it's data directly through the LLM it is interfacing with. The user will provide a subject to ASC, and a problem in the subject will be provided to the user based on the specified subject, and the user's current skill level in the subject. The user will then respond to the question, and their answer will influence their skill level, which in turn influences the difficulty of questions they will receive. This data is not sensitive, and thus no encryption will be required. The user's skill rating will then be saved locally once practice is complete and can be exported for use in separate instances of ASC or loaded the next time the program is opened on the current device.

UML Diagrams W/ Descriptions

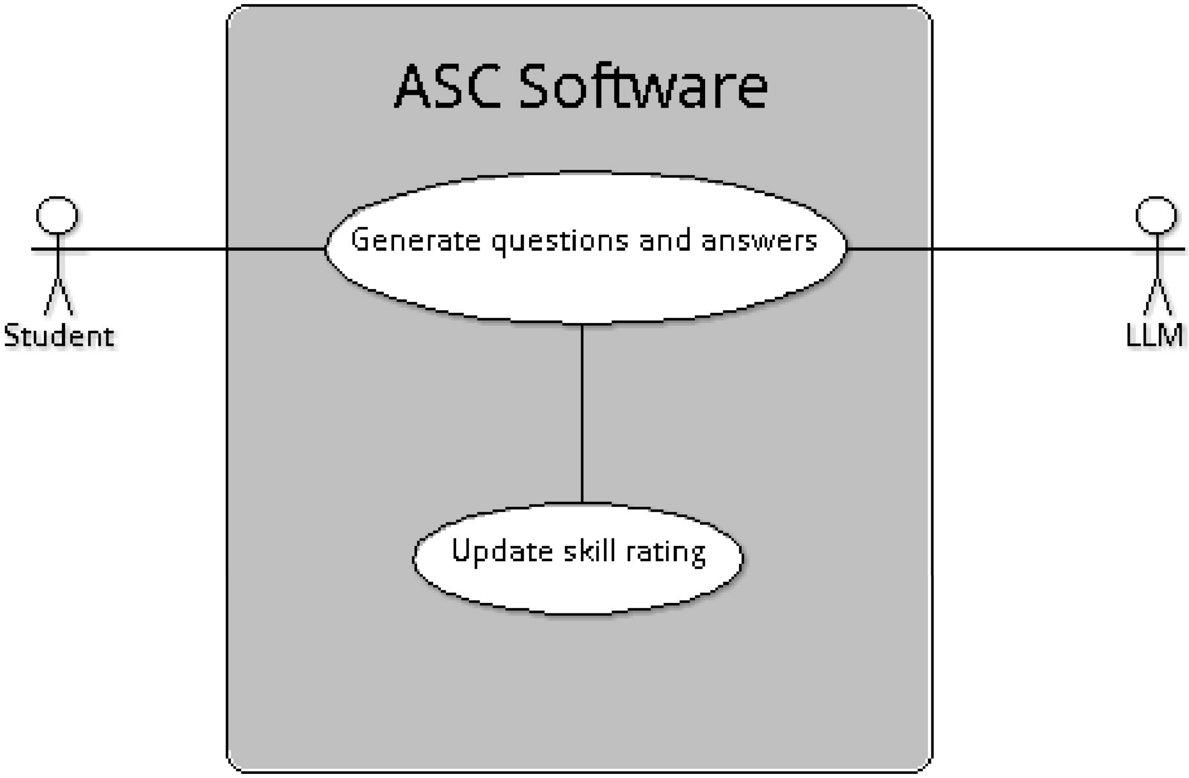

Figure 1: Use case UML diagram for user generating questions and receiving answers.

Description: The user will supply ASC with a subject, and ASC will interface with an LLM to generate a question that ASC presents to the user. ASC will then compare the users selected answer to the answer generated by the LLM and update the user's skill rating based on whether they answered the question correctly or not (Figure 1).

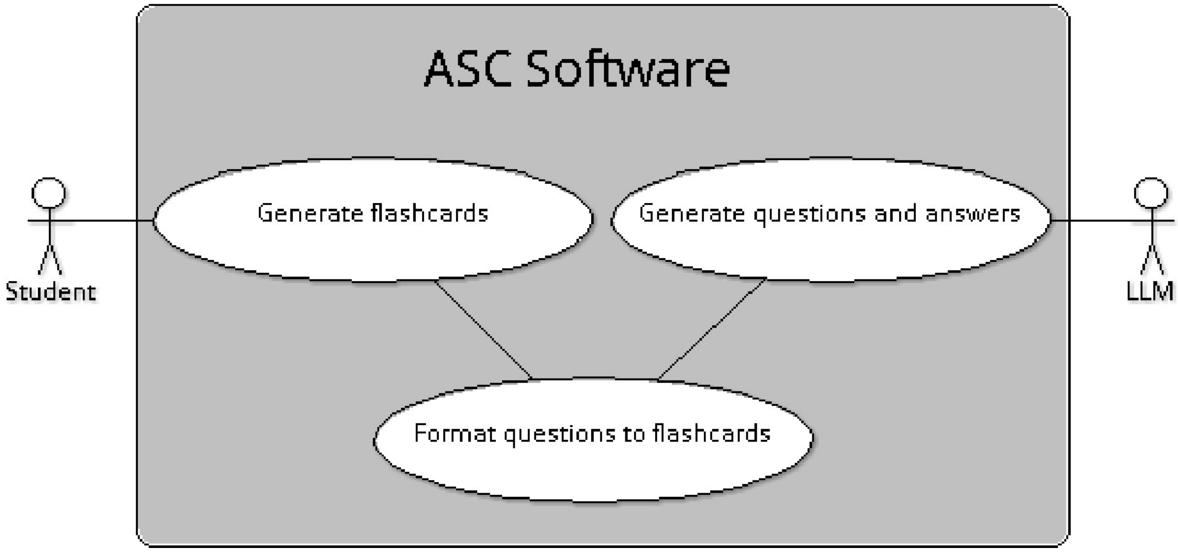
ASC Software
Generate flashcards
Generate questions and answers
Format questions to flashcards
Student
LLM

Figure 2: Use case UML diagram for user generating flashcards.

Description: The user can prompt ASC for flashcards. ASC will then communicate with the LLM to generate questions and answers. ASC will then take the generated questions and answers, and format them to a printable sheet of flashcards that is given to the user (Figure 2).

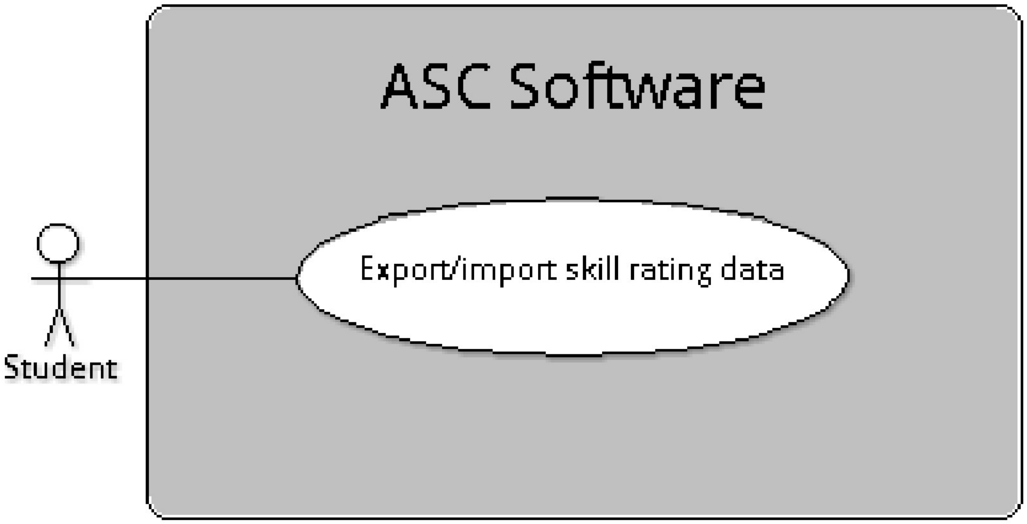

Figure 3: Use case UML diagram for exporting/importing saved skill rating data.

Description: User can prompt ASC to export saved skill rating. ASC will take the users skill rating in respective subjects, compile them to a single portable file, and return it to the user.

On a separate instance of ASC, users can then import their skill rating file generated by ASC to share their progress across multiple devices (Figure 3).

Initial Requirements

Functional Requirements

The following requirements outline the essential features of ASC:

1. **Subject Input:** Users must be able to input a subject to receive relevant quiz questions.
2. **Quiz Generation:** The application must generate quiz questions and answers using a language model.
3. **Progress Tracking:** Users must be able to track their progress across multiple sessions, including scores and completed quizzes.
4. **Flashcard Generation:** The program must allow users to create printable flashcards based on quiz questions.
5. **Feedback Mechanism:** Users must receive feedback on their performance and suggestions for areas needing improvement.
6. **Data Storage:** The application must be able to save user data on the client-side for future access.

Nonfunctional Requirements

The following list details the inessential features of ASC:

7. **Performance:** The application must respond to user input within a specified time limit (e.g., under 2 seconds for generating quiz questions).
8. **Scalability:** The system must manage a growing number of users and increasing amounts of data without significant performance degradation.
9. **Usability:** The application must possess an intuitive interface that accommodates users of varying technical skill levels.
10. **Compatibility:** The application must function seamlessly across various web browsers and devices (e.g., desktops, tablets, and smartphones).
11. **Reliability:** The system must maintain high availability and be resilient to failures, ensuring minimal downtime.
12. **Maintainability:** The codebase's structure and formatting must facilitate easy updates and maintenance.

Documentation

Our documentation will be based on the required materials listed below:

13. **Proposal**

14. **Requirements**

15. **Specification**

16. **Design**

17. **Project Log**

Testing and Revisions

All group members completed revisions and testing by consistently updating documents. Communication, including document sharing, was conducted through Discord. This document, along with others, was drafted and edited collaboratively by the team. When a team member had the opportunity to contribute new content, it was submitted in the document section of the Discord chat. Each revision of this specific document was assigned a number to indicate the most current updates.

References

CloudFlare. (2024a). *What is a large language model (LLM)? | cloudflare*. Cloudflare. https://www.cloudflare.com/learning/ai/what-is-large-language-model/

Cloudflare. (2024b). *What is Artificial Intelligence (AI)? | cloudflare*. https://www.cloudflare.com/learning/ai/what-is-artificial-intelligence/

Jacobs, M., Casey, L., & Kaim, E. (2022, November 28). *What is Git? - azure DevOps*. Azure DevOps | Microsoft Learn. https://learn.microsoft.com/en-us/devops/develop/git/what-is-git

Lakshita. (2024, October 9). *Introduction to JavaScript*. GeeksforGeeks. https://www.geeksforgeeks.org/introduction-to-javascript/#

Python Software Foundation. (n.d.). *What is python? executive summary*. Python.org. https://www.python.org/doc/essays/blurb/

React. React Blog RSS. (n.d.). https://react.dev/

Appendix A: Technical Glossary

Artificial Intelligence (A.I) - “Artificial intelligence, A.I, is the ability of a constructed machine, such as a computer, to simulate or duplicate human cognitive tasks. A machine with AI can make calculations, analyze data in order to create predictions, identify various types of signs and symbols, converse with humans, and help execute tasks without manual input“ (CloudFlare, 2024b).

Discord – An online messaging platform typically used for activities done over the internet.

Git – A open-source version control system that makes it easier to collaborate on projects. (Jacobs et al., 2022)

Graphical User Interface (GUI) - A method of letting users interact with an application using graphics instead of a console.

JavaScript - “A lightweight, cross-platform, single-threaded, and interpreted compiled programming language. It is also known as the scripting language for webpages. It is well-known for the development of web pages, and many non-browser environments also use it” (Lakshita, 2024).

Language Learning Model - “A large language model (LLM) is a type of artificial intelligence (AI) program that can recognize and generate text, among other tasks. LLMs are trained on huge sets of data — hence the name "large." LLMs are built on machine learning: specifically, a type of neural network called a transformer model” (CloudFlare, 2024a).

Python -” Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it appealing for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together” (Python Software Foundation, 2024).

React - “A JavaScript framework that lets you build user interfaces out of individual pieces called components. Create your own React components like Thumbnail, Like Button, and Video. Then combine them into entire screens, pages, and apps” (*React,* 2024).

Appendix B: Team Details

This document was written in collaboration with all team members. The contributions of each group member are described as follows:

Gage Keslar: Abstract, Background, Objective and Overview, Team Details and Dynamics, Technical Glossary, Writing Center

Jonathan Buckle: Table of Contents, Project Context, UML, and testing and revisions

Cameron Calhoun: Initial Business Model, Operational Environment, Description of data sources, UML Diagrams w/ descriptions.

Seth Morgan: Abstract, Workflow Authentication, Team Details and Dynamics, Initial Requirements (Functional, Nonfunctional, and Documentation)

All other contributions to the requirement document were developed in coordination with members of the group during an in person meeting at the California Campus. These contributions include:
· In Person Discussion
· Planning
· Formatting
· Proofreading
· Meeting with the Writing Center

Appendix C: Workflow Authentication

I, Gage Keslar, confirm that the details defined in this document represent functional requirements of "ASC." Also, I agree with all the above information provided in the requirement document.

Signed:_ Date:_10/21/2024________

I, Cameron Calhoun, confirm that the details defined in this document represent functional requirements of "ASC." Also, I agree with all the above information provided in the requirement
document.

Signed: Cameron Calhoun Date: 10/21/2024___________

I, Seth J. Morgan, confirm that the details defined in this document represent functional requirements of “ASC.” Also, I agree with all the above information provided in the requirement document.

Signed:_ Seth Morgan Date: 10/22/2024

I, XXXXXX, confirm that the details defined in this document represent functional requirements of “ASC.” Also, I agree with all the above information provided in the requirement document.

Signed:_ Jonathan Buckel Date:10/22/2024___________________

Appendix D: Writing Center Report

See Email From FoundryWritingCenter@Pennwest.edu 10/22/24 at 8 AM.