ASC

AI STUDY COMPANION

SPECIFICATION DOCUMENT

Team Members:

Seth Morgan

Cameron Calhoun

Jonathan Buckel

Gage Keslar

Professor:

Dr. Weifeng Chen

Course:

Senior Project I: Software Engineering

University:

Pennsylvania Western University, California Campus

Instructors Notes:

Table of Contents:

Abstract

The AI Study Companion (ASC) is a browser-based application designed to function as a personalized study aid. Its primary users include school staff, students, and parents who aim to enhance their own knowledge or provide academic support to others. ASC allows users to assess their understanding of a topic by entering a subject and receiving relevant quiz questions. The application enables users to track their progress over time, print flashcards, and receive targeted feedback on specific areas. Key features include the use of a Large Language Model (LLM) to generate customized questions and answers, provide focused study recommendations, and produce downloadable flashcards or locally stored progress data. This document outlines ASC's design and serves as a strategic guide for the project's ongoing development.

Document Description

Purpose:

This document outlines the specific requirements for the ASC project, detailing the necessary system and functionality specifications. It serves as a reference for both the developers and the client to establish and agree upon the terms of their collaboration. The client retains the right to review and dispute any aspect of this document to ensure it accurately captures their desired requirements. Once both parties accept the terms stated here, this document will serve as a binding agreement.

Intended Audience:

This document is intended for use by both the client and the development team. Its purpose is to define the product's final design, functionality, and associated costs in clear terms, allowing the client to fully understand the scope and deliverables of the project. For the development team, this document serves as a comprehensive framework, establishing priorities and ensuring alignment with project objectives throughout the development process. Any ambiguities within this document must be resolved promptly, as it will serve as a binding contract between the client and developers.

System Description:

Overview:

The AI Study Companion (ASC) is an innovative browser-based application designed to address the common challenges faced by students, parents, and educators in developing effective study habits and accessing affordable academic support. Leveraging a Large Language Model (LLM), ASC functions as a personalized study partner, capable of generating quiz questions tailored to the user's specified subject and difficulty level. Users can track their progress, receive feedback on areas needing improvement, and obtain suggestions for advancing their skills. ASC also provides tools for creating printable flashcards that feature keywords, definitions, and user-specified questions, formatted for standard 8" x 11" paper. With a React-based front end for an intuitive user interface and a Python-driven backend for efficient communication with the LLM, the system ensures seamless operation. By saving user data locally, ASC supports ongoing improvement and enables users to focus on their knowledge gaps. This is the outlines the of how is ASC, aiming to revolutionize how students study.

End User Profile:

Those intended to use the system are students or those close to a student that want to improve the knowledge of a certain topic. It will also be used to improve self-studying practices through using the AI features or generating flashcards. Users will need the capability to navigate and operate webpages through their browser choice and reading comprehension to fully use this application. To use the flashcard feature, the user will need access to a printer that can print on the specified paper dimensions. To save their data and send it to another system, the user must be able to send data from one device to another, such as the use of a USB storage device, email, or other means of data transportation.

User Interaction:

All interactions between the software and user will take place on a web browser. Upon access to the site, the user will be presented with options they would like to take. If there is no user data available on the device, there will be the ability to load data from an external file. In addition, the user can still generate study questions and flashcards without any user data, this will then create a new user data file. Generating questions will prompt the user to choose one subject that is provided by ASC, if the user has not been prompted to do so before, and a skill level from 0 – 1600. Level 0 being elementary questions to 1600 being closer to college level. The user will be presented with a multiple-choice question. Each question will have only one correct answer to choose from. After a set number of questions answered, ASC will determine if the user is ready to go to a higher skill level, remain the same level, or to go back to a lower level. Generating flashcards will follow the same set of instructions for picking a subject as generating questions.

ASC will generate questions or terms with their respective answers or definitions on a text document that the user can print out.

Hardware Constraints:

The user must be able to connect to connect to the internet and have access to printer for all the functions to work properly. With access to the internet, the user should navigate to the webpage and begin using its functions. For the flashcard generations to work properly, the user must have access to a printer to create the cards. The user will also benefit from owning a pair of scissors for cutting the paper.

Software Constraints:

The software constraints are the ability to access an LLM, creating a webpage that can hold the source code and execute effectively, and how user data will be stored. The choice of software language will affect our choices. For this reason, the use of Python for backend development with the FastAPI framework, JavaScript with React components for frontend development, and the use of Docker images will help us go forward. Python will allow the communication between the source code and a LLM with the use of FastAPI. JavaScript and React have the ability not only to stylize and create web pages, but also to hold a Docker image that will hold Python code and execute it accordingly.

Time Constraints:

This project has a deadline towards the end of the spring 2025 semester and is expected to be done by then. The team consists of four CMSC students who will be

splitting their time between this project and other academic commitments. The team members will split their time and responsibilities accordingly to ensure that deadlines are met. Said deadlines will be discussed through online and in-person meetings to determine who is assigned to what task, current issues were facing both individually and collectively, and possible solutions to said issues. Without broadening our scope, it is possible to complete this project in a timely manner.

Cost Restraints:

The cost we face will come from the LLM that we plan to use. There are multiple LLMs that we have access to. Each one having their own pitfalls, the largest one being cost. LLMs like ChatGPT require payment for each token parsed as a response. Other LLMs are free but are not as powerful, not able to produce the functions needed, or will not integrate with our chosen method. If we use OpenAI's ChatGPT, we are looking to spend 1.25 dollars per 1 million input tokens, and 5 dollars per 1 million output tokens. To ensure cost stays effective using ChatGPT, we would need to keep our Application Programming Interface (API) key safe, and limit testing to ensure we are not wasting input or output tokens.

Other Concerns:

There are concerns about the user access to the webpage and how the LLM will provide the generated questions. The user will need to be able to access the webpage through any browers and hence need an internet connection. There are also the concerns of the ability to interact with the LLM. While some LLMs are available to be downloaded on to a single device,

they are more resource heavy than accessing them over the internet. Finally, the LLM must be able to produce accurate questions with their answers in a way that is reliable and consistent. That may take time to figure out what prompt is needed to provide the best possible questions to answer. Some subjects may prove to be more work than others. For instance, math has proven itself to be a challenge to most LLM's, being unable to do even simple math questions. Our solution is to make the LLM create a math problem for the program to solve. Once the problem is solved, we will include the answer as one of the multiple choice answers for the user to select.

Acceptance test criteria

Testers:

Each team member will be responsible for testing and quality assurance throughout the process. All members will have individual testing criteria when testing their own source code and when testing another team member's. All testing will be consistent to the Python back-end, React based front-end, and ensuring that the LLM will provide reliable and consistent data to feed into the program. Testing will be done often and will ensure that the team will finish on time with a program that will work effectively and consistently.

Criteria for user acceptance:

User acceptance is achieved when ASC can achieve the following features consistently and reliably:

- The ability to accept or create user data
- Provide subjects of study: Math, Computer Science, History, Geography, and English
- Correctly generate questions with correct difficulty and correct answers to said question via LLM
- Adjust the users score accordingly to reflect the users understanding of chosen subject
- Export user data when requested and correctly loads user's saved data
- Properly display the users' stats on the webpage
- The webpage is accessible and functional to users
- Can generate flashcards accurately for user download
- Format text file for ready to print and cut flashcards

Integration of separate parts and installation

ASC is a web-based application, users will have to visit the webpage in order to have access to the project. Users will require an internet connection and a device capable of running an internet browser. Visiting the webpage will provide the users with a Graphical User Interface (GUI) that will provide access to the program's functionalities. No installation of Python or the software is needed since the source code is encapsulated in a Docker image capable of executing Python code independently from the machine used to access the webpage. Users that wish to use the flashcard generation capabilities of the program will require a printer to print the formatted text files.

System Modeling

Functional: Use cases and scenarios:

There are seven use cases of the ASC app, described in the figure below (Figure 1).

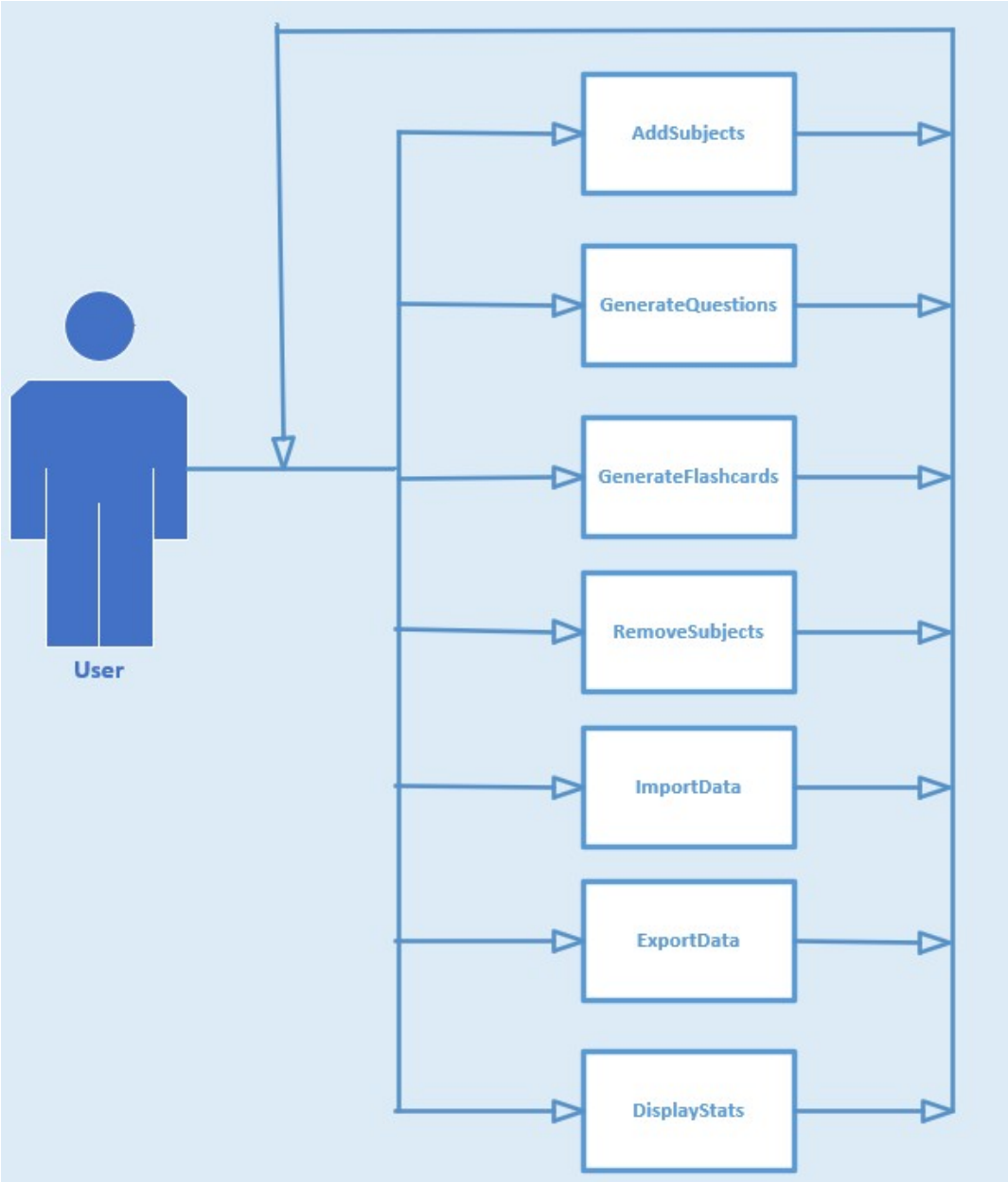

Figure 1: Use case diagram of ASC app

Figure 1 above shows a simplified Use Case Diagram as the user goes to the ASC web application. From this menu, they can add or remove subjects to be tracked by the program, import or export current or previously used saved data from tracking subjects, generate a series of questions to answer, generate flashcards in portable document format (PDF), or display their current progress to the screen.

RunScenario:

- User enters application
- User chooses one of the following options from the home screen
    a. AddSubjects
    b. GenerateQuestions
    c. GenerateFlashcards
    d. RemoveSubjects
    e. ImportData
    f. ExportData
    g. Display Stats

a. If AddSubjects is chosen, the user can choose from a list of predetermined subjects which subjects they would like to track.
b. If GenerateQuestions is chosen, two things can occur:
    a. If the user has no subjects currently tracking, they will be sent to AddSubjects to add a subject to track
    b. If the user has at least one subject they are tracking, a process will begin in which ASC will ask the user questions based on the selected subject. ASC will track the users progress by updating an Elo rating for the given subject
c. If GenerateFlashcards is chosen, two things can occur:

   a. If the user has no subjects currently tracking, they will be sent to AddSubjects to add a subject to track
   b. If the user has at least one subject they are tracking, they will be able to select a number of questions to be generated and formatted into a printable document, given to the user in PDF format.

d. If RemoveSubjects is selected, the user will be taken to a screen in which they can remove their progress for a given subject, under the condition that they have at least one subject being tracked
e. If ImportData is selected, the user will be taken to a screen in which they can provide a previously generated file tracking their progress, to resume where they left off.
f. If ExportData is chosen, so long as saved data currently exists in the program, the user will be able to export their current progress as a saved file to later be imported in a different instance of the program.
g. If DisplayStats is chosen, the user will be able to display their progress on all currently tracked subjects.

Entity: Class Diagrams:

The following figure displays the classes that will be needed in our program (Figure 2).

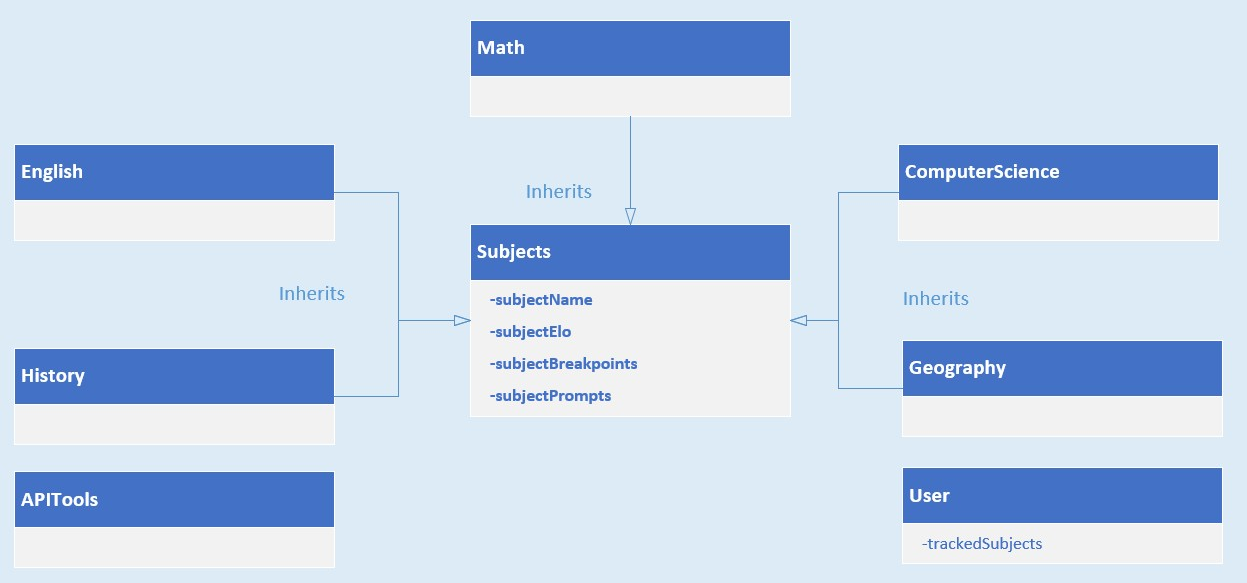

Figure 2: Class diagram

Class Descriptions

User Class

The User class only holds one attribute, that being a list of subjects currently tracked by the user. The User class will also hold the member functions needed to perform actions specific to a user, such as adding subjects, removing subjects, importing data, exporting data, and displaying data.

Subjects Class:

The Subjects class holds four attributes. The first, subjectName is a

value to store the name of the subject being tracked, this will hold a placeholder in the Subject class and will be overridden in the derived classes. The second, subjectElo, will hold a default value to start every subject's Elo when generated. The third, subjectBreakpoints, will hold default Elo values that will determine when a subjects difficulty needs to be increased. For example, if a breakpoint of 1200 is stored, when the subject's subjectElo value reaches that number, the difficulty of the subject's questions will change. This value will be overridden in

each derived class to allow for custom difficulty scaling for each subject. The final attribute, subjectPrompts, holds a list of default prompts that will be sent to the LLM when at a particular breakpoint. These prompts will be overridden by the derived classes and will allow for custom prompting to the LLM based on the subject being tracked, and the current difficulty level of the subject.

Math Class:

The Math class is a derived class of Subjects and inherits all the attributes of the Subjects class. These values will be overridden in Math to store unique values for subjectName, subjectBreakpoints, and subjectPrompts.

History Class:

The History class is a derived class of Subjects and inherits all the attributes of the Subjects class. These values will be overridden in History to store unique values for subjectName, subjectBreakpoints, and subjectPrompts.

Geography Class:

The Geography class is a derived class of Subjects and inherits all the attributes of the Subjects class. These values will be overridden in Geography to store unique values for subjectName, subjectBreakpoints, and subjectPrompts.

English Class:

The English class is a derived class of Subjects and inherits all the attributes of the Subjects class. These values will be overridden in English to store unique values for subjectName, subjectBreakpoints, and subjectPrompts.

ComputerScience Class:

The ComputerScience class is a derived class of Subjects and inherits all the attributes of the Subjects class. These values will be overridden in ComputerScienceto store unique values for subjectName, subjectBreakpoints, and subjectPrompts.

APITools Class:

The APITools class has no attributes, and will instead just house member functions for interfacing with the API's used to talk to the LLM's.

Dynamic: State Chart:

The following figures represent the dynamic flow of the program by means of a state chart. The key for the state chart is found in the figure below (Figure 3).

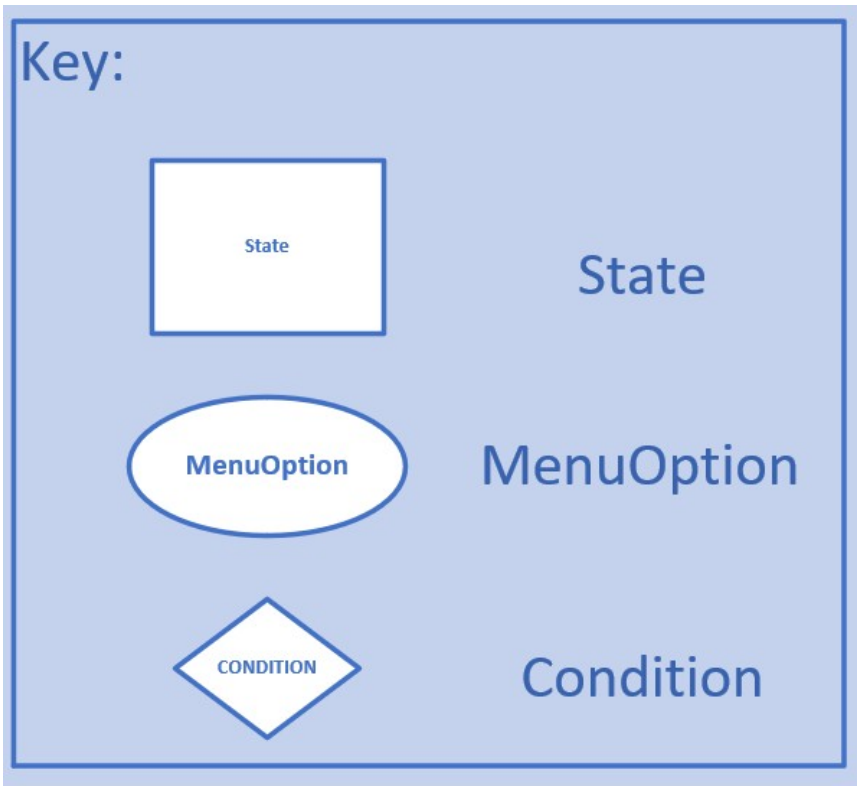

Figure 3: State chart key.

The State rectangles serve as a state accessible in the program. These states are accessed by selecting one of the ovular MenuOptions found in the main menu state. These MenuOptions effectively serve as the transitions from the main menu state to the selected state. Some entrances to states are locked under conditional logic, the logic of which can be found in the diamond shaped conditions.

The following figure serves as the state chart of our program (Figure 4).

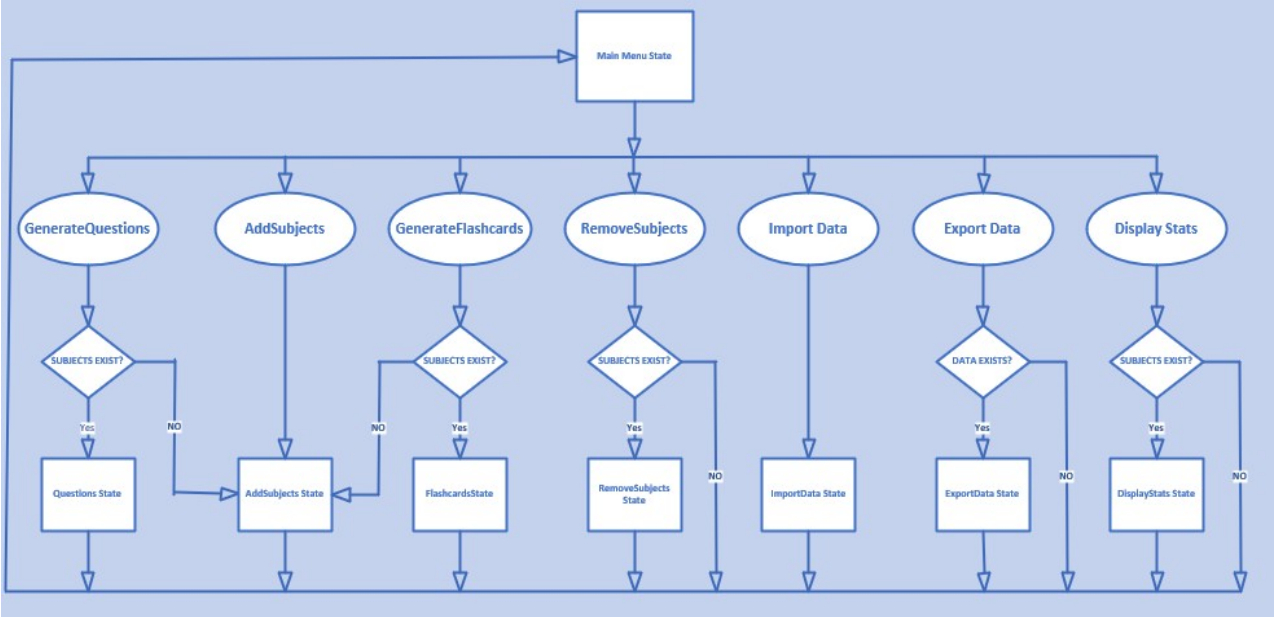

Figure 4: Dynamic State chart

States:

- Main Menu – This state functions as the homepage of our web application. From this state, the user is able to access all other states.
- Questions – This state serves as the state in which the user interacts with ASC and is able to ask questions on a given subject. This state can only be entered if the user has subjects currently in tracking.
- AddSubjects – This is the state that allows the users to track subjects. The state houses a list of available subjects to track and allows the user to choose subjects from the list.
- Flashcards – This state allows the user to choose a from a predefined list of numbers and generate that number of questions into a portable format the user can save to print out and study with. This state can only be accessed if the user has subjects currently in tracking

- RemoveSubjects – This state allows users to remove subjects from their list of currently tracked subjects and wipe their data of the given subject. This state can only be entered if at least one subject currently exists in tracking.
- ImportData – This state allows users to take a previously generated save file and import it into ASC. This allows users to track their progress amongst multiple sessions of ASC.
- ExportData – This state allows users to export their current data into a saved file which they can use in future sessions of ASC. This state is only accessible when the user has data they are tracking.
- DisplayStats – This state shows the user the progress they have made on their currently tracked subjects. This state is only accessible if the user has at least one subject currently in tracking.

Transitions:

A transition occurs every time a user selects a menu option from the main menu, and the given options conditional logic is met. When the webpage is first accessed, they will arrive in the initial state, which can also be referred to as the home page, or the main menu. From the main menu, the user can transition to the seven states listed above. When the user is finished in any state, they will be transitioned back to the main menu page where the process can begin again. This effectively creates a control flow of the program in which the user begins in the initial state. The user then makes a selection from the main menu and is transitioned to a different state given on the resulting conditional logic. Processing will occur in this state, (generating flashcards, generating questions, importing/exporting data, etc.), until the user is complete. When complete, the user will hit a back button and will transition back to the main menu to start the cycle again. The cycle breaks when the user exits the webpage.

Collaboration Diagram

The collaboration diagram below shows the interaction of the User class, and a given Subject child class in realizing the Add Subjects use case (Figure 5).

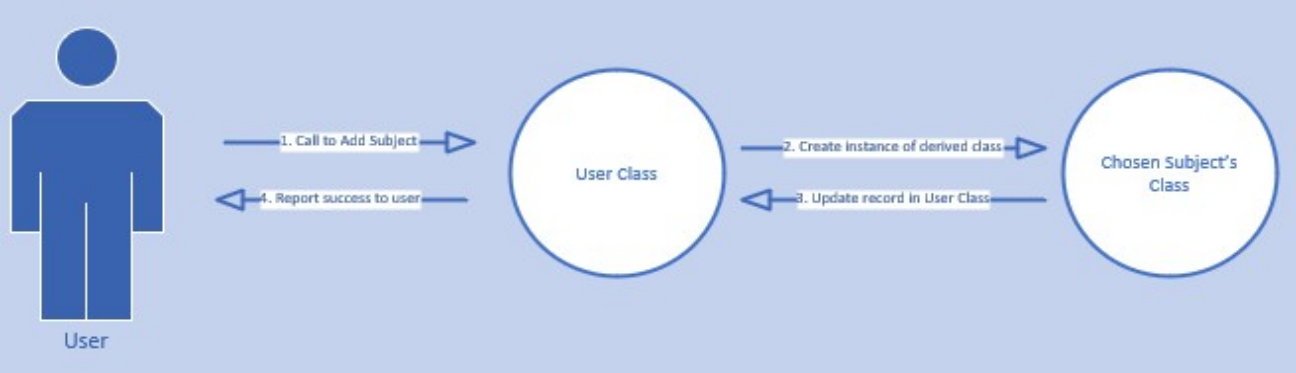

Figure 5: Collaboration Diagram

The collaboration diagram follows the following steps:

1. The User selects the Add Subject feature from the main menu. This will transition the program to the Add Subject state. In this state, the user will make a selection of which subject they would like to add, and the call will be issued to its user class.
2. The User class will call one of its member functions and create an instance of the class corresponding to whichever subject was chosen.
3. Once the instance of the class is created, the trackedSubjects record within the User class will be updated to include the instance of the class.
4. Finally, a message displaying the successful addition of the new subject will be presented to the user, and the program will return to the main menu state.

Components / Tools Needed:

Hardware needed consists of a device capable of connecting to the internet and using a web browser in order to visit and interact with the webpage. For software purposes, the team will

need an API key for accessing the LLM. It will allow the LLM to interface with the source code and will be used to track how many input tokens were fed to the LLM and how many output tokens were produced. Development versions are synced through a GIT repository. Docker images will be used to contain the final product on the webpage. Finally, FastAPI will allow the webpage function to be properly interfaced with the source code.

References

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

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

FASTAPI. FastAPI. (n.d.). https://fastapi.tiangolo.com/

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/

What is Docker?. Docker Documentation. (2024, September 10). https://docs.docker.com/get-started/docker-overview/#:~:text=Docker%20is%20an%20open%20platform,and%20running%20it%20in%20production.

Appendix A – Glossary of Terms

Attribute – A variable in an object oriented class.

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).

Class – A defined structure in object-oriented programming language that holds attributes and functions.

Docker - “Docker is an open platform for developing, shipping, and running applications. Docker enables you to separate your applications from your infrastructure so you can deliver software quickly. With Docker, you can manage your infrastructure in the same ways you manage your applications. By taking advantage of Docker's methodologies for shipping, testing, and deploying code, you can significantly reduce the delay between writing code and running it in production.” (*What is Docker?* 2024).

FastAPI - “FastAPI is a modern, fast (high-performance), web framework for building APIs with Python based on standard Python type hints.” (*FASTAPI*).

Class Function – A function that operates on members of a class.

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).

State -

Appendix B: Team Details

The leader of this paper is Cameron Calhoun. Each member was responsible for proofreading and making corrections to grammar and formatting. Each member was also responsible for their own segments in the paper:

Cameron Calhoun, has taken leadership role, creating UML and Collaboration Diagrams in addition to the System Modeling segment.

Gage Keslar, responsible for the System Description and Environment description segments as well as going to the Writing Center.

Seth Morgan, responsible for creating the document, formatting, table on contents, and additional research and editing.

Jonathan Buckel was responsible for additional research, finding software documentation, and additional editing.

Appendix C: Workflow Authentication

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

Signed:  Date: 11/18/2024

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

Signed:  Date: 11/18/2024

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

Seth Morgan

Signed: Date: 11/18/2024

I, Johnathan Buckel, confirm that the details defined in this document represent the specifications of "ASC." Also, I agree with all the above information provided in the requirement document.

Signed: Jonathan Buckel Date: 11/18/2024

Appendix D: Writing Center Report.

Dr. Chen,

Hello, my name is Emily Esposito and I am a writing consultant for the Foundry Writing Center. At 2:00 this afternoon, I had an appointment with Gage Keslar in the Foundry Writing Center to receive tutoring for the draft of their Senior Project for the CMSC 4900 Software Engineering class.

Today we focused on the overall structure and ensuring that the formatting and grammar were correct. In our last session we discussed formatting the headings for their paper. I noticed their group was still struggling and went over this concept again. We pulled up an example of how to properly add all of the different types of headings they needed and applied it to their project. We then went through the paper and read it aloud in order to find grammar mistakes as we went along. We looked at reworking some sentences to make them more concise and limit the receptiveness while maintaining the purpose of the sentences. I noticed improvement from our last session as we worked a lot with acronyms and how to format them into a paper. These

concerns were fixed for their session today. We also discussed to ensure that figures were being referenced properly by adding "(Figure #)." at the end of their sentence.

If you have any other questions about our meeting, please do not hesitate to contact me at esp2609@pennwest.edu.

Sincerely,

Emily Esposito
Pennsylvania Western University California
Secondary Education Biology, *Junior*
Alpha Lambda Delta Honors Society, *Member*
Vulcan Dance Team, Vice President
University Choir, Secretary
Foundry Writing Center, *Consultant*