4.2 Design Exploration

4.2.1 Design Decisions

Due to the scale and details behind this project, there have been many important decisions for developing "Ask Captain Cyber" to ensure the highest response accuracy, query response, and user experience.

The most important decisions have been...

Deciding which AI model to use—We have decided to use OpenAI's model that powers Microsoft Copilot. We chose this model because it offered important features such as access to real-time data and web searches and the best balance of cost and LLM capabilities. These aspects are important because they enable "Ask Captain Cyber" to have access to live information and minimize AI generation response times.

Deciding which frontend design language to use - We have decided to use React to implement our frontend design for "Ask Captain Cyber." We chose this since it seamlessly integrates with WordPress, and some of our group members have previous experience using it. This helps us by promoting ease of implementation, allowing us to design the front end to cater to a good user experience.

Deciding how to categorize questions - One aspect of our project we still have to decide upon is how we will categorize user questions. We will need to ensure that our backend implementation can associate similar questions with each other so that if it is similar, a response is quickly generated by the LLM rather than waiting to be vetted by an Expert. This will help decrease response times for user prompts, a critical aspect of the user experience. Another important feature will be to make sure that the vetted questions can be sorted and organized in a clean and efficient way. We do not just want a random assortment of questions with no rhyme or reason. We can also implement some sort of priority if a question or similar questions are being asked a lot and can be answered by the ambassadors.

4.2.2 Ideation

The bulk of the ideation process for the project lies in deciding which LLM to use. There were many critical aspects we needed to consider such as the number of parameters, if it has access to real-time information, cost, and its response time. Ultimately, we decided that OpenAI's GPT-3.5 solution would work the best for our use case and evaluated other models such as Gemini by Google, Llama by Meta, Claude by Anthropic, and GPT-4o by OpenAI. We identified these potential options by researching the most popular LLMs and evaluating our previous experiences with each one. The other part was how to design user interaction along with how the front and backend will interact with each other. You can see in the documents that we provided that this is an important feature that we did before we even started to develop the product. We are also continuously coming up with ideas and implementations as we gain more information from our advisor.

4.2.3 Decision-Making and Trade-Off

To make this decision, we developed a weighted decision matrix to compare and contrast each LLM and gain a better understanding of their strengths and weaknesses.

Model	**Provider**	**Context**	**Pricing**	**Total**
GPT-3.5	OpenAI	6	7	**13**
Gemini 1.5 Pro	Google	6	6	12
Llama 3	Meta	3	8	11

Claude 3 Opus	Anthropic	8	1	9
GPT-4o	OpenAI	7	3	10

This decision matrix evaluated two different critical components of our LLM to provide context and pricing. Pricing related to the cost of operation of our "Ask Captain Cyber" project, with context defining the amount of text data it can consider simultaneously. By examining each criterion, we determined that GPT-3.5 would be the best option for our use case, offering a relatively cheap price while still maintaining its capability to process large amounts of text.

4.3 Proposed Design

4.3.1 Overview

"Ask Captain Cyber" is a cyber-security-focused chatbot that will be able to answer users' questions relating to cybersecurity, with support for all levels of complexity. Questions that have not been answered will be generated by an LLM and vetted by experts to ensure accuracy. Our current high-level design is illustrated in Figure 1:

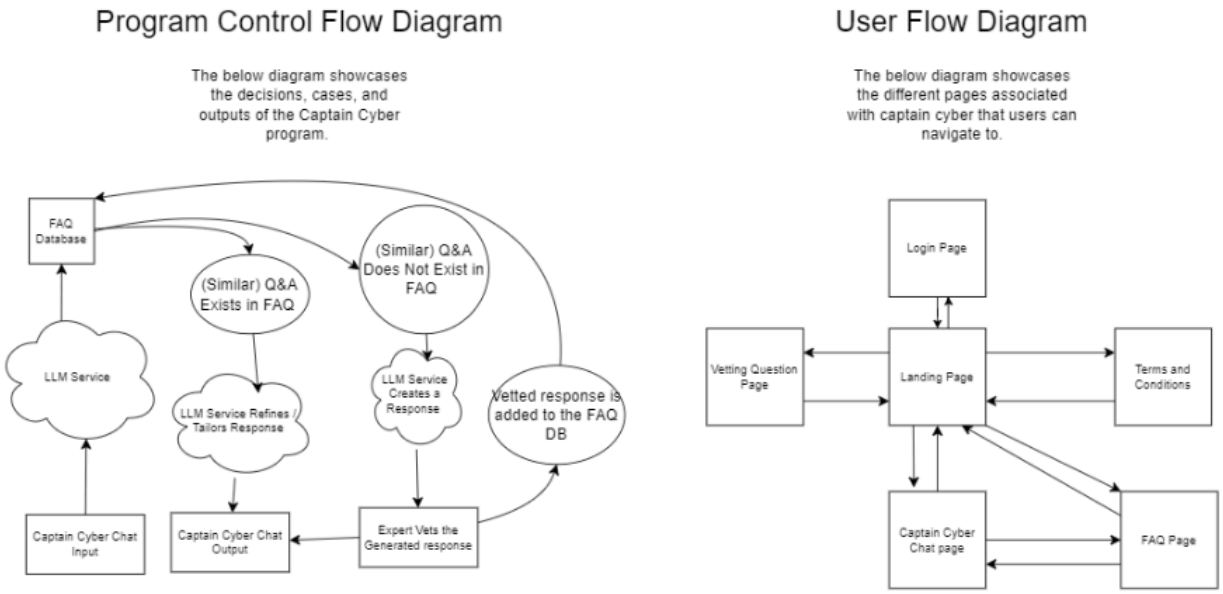

Figure 1 - "Ask Captain Cyber" Architecture

In the diagram depicted in Figure 1, we have the program control flow and the user interface flow. The program control flow diagram outlines the backend implementation of our project, where prompts are checked to see if an answer exists in our FAQ, and if they don't, they are AI-generated and then vetted by experts. The user flow diagram shows how the frontend implementation of our project will work, with six total screens allowing users to log in, vet questions, read the terms and conditions, view the FAQ page, query "Ask Captain Cyber" with their prompt, and a landing page explaining what the tool does. This high-level view of the front and back end of our project is representative of our ongoing implementation. The various subsystems, such as the FAQ and generation, are critical to this process as well, ensuring a good user experience for our final version.

4.3.2 Detailed Design and Visual(s)

For the backend of the design, we will have three or so databases. One will be for users. This user database will authenticate users, and entries must be manually entered. Each user will actually be a user for each organization. For example, School A will have a School A user, and School B will have a School B user, etc. This must be managed and queried when the user tries to log in. The backend will use email authentication as a MFA for an extra layer of security. Regular users will have no login page, which will only be for the ambassadors.

The frontend aspect of Ask Captain Cyber is rather simple. Since most user interaction will happen within the chatbot environment, we will design it so that the information is clearly presented with little to no other distractions. This will allow the user to focus on learning more about cybersecurity and less about how to explicitly interact with the site. Furthermore, when they land on the website, there will be a short summary of the project and a disclaimer so that they are aware of potential pitfalls or consequences of malicious use of Ask Captain Cyber. The experts will have an intuitive dashboard to view questions waiting to be vetted, questions that have recently been vetted, and a note service to collaborate with other expert ambassadors as needed.

We will also need a Question database where questions that the AI can't answer will get forwarded to a database that ambassadors can query and answer. This database will also have to be somewhat sorted or displayed in a sorted manner. This is because we do not want questions to be completely random when the ambassadors try to answer them. So, we need to create some sort of mechanism to organize or sort the questions in the database itself, or when it is displayed. This database must also be scalable because we do not know how many questions could be simultaneously. This database of questions might need to be encrypted. This is because if a user types in a question that includes sensitive information, we do not want that question sitting in plaintext. We also need to make sure only ambassadors can query the database. This might require some token or extra layer of authentication on top of the authentication required to log in. We must still determine how this process will work to balance security and usability. Finally, this database will have to be able to update questions -> give it back to the user -> and send it to the LLM for future reference.

The final database will be the LLM. This will be the dataset the AI uses when querying questions. This database will be ever-growing and must be scalable for a large amount of data. The AI will also have to be able to check if questions are similar and alter pre-existing questions if they are similar. It will then have to update the LLM with new answers and questions and be able to sort them effectively. Questions will probably have to be sorted by type, such as networking, password, security, etc.... It will also have to connect with the Question database that ambassadors will use.

An outline for the system design can be found below in Figure 2:

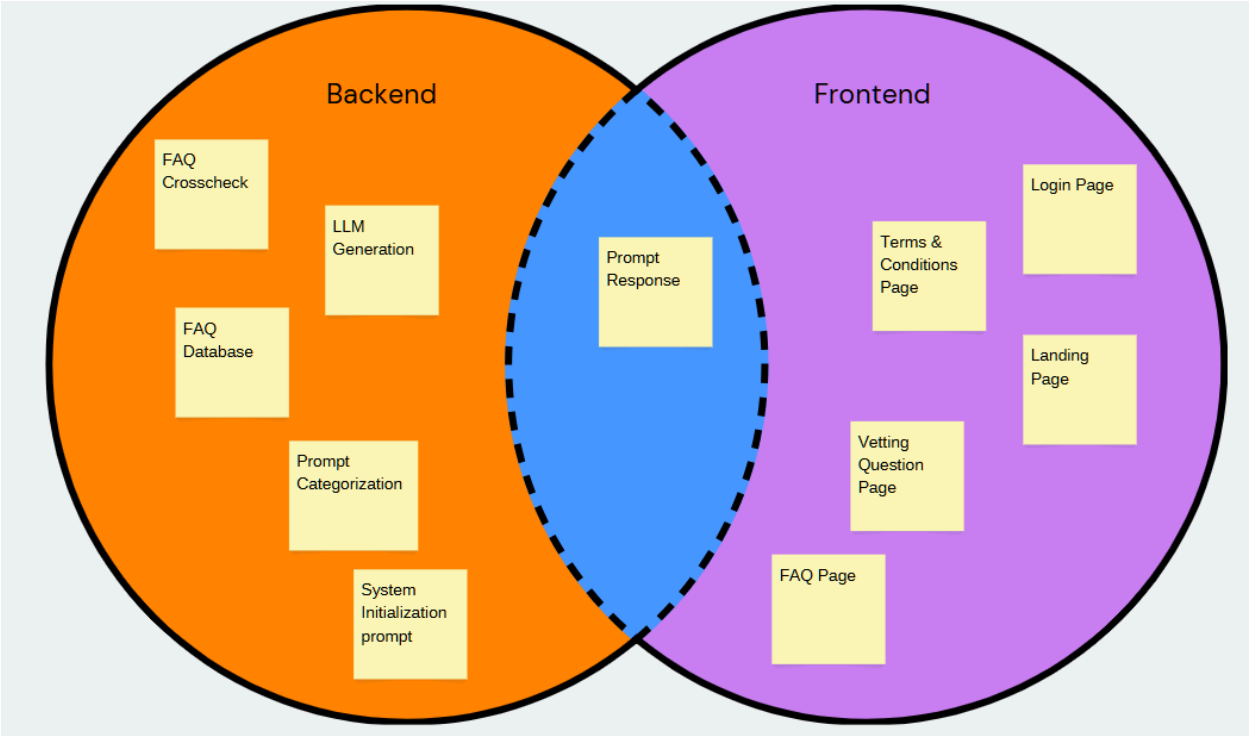

Figure 2 - "Ask Captain Cyber" Front & Backend implementation

4.3.3 Functionality

The design we are going with will wait for a user to "Ask Captain Cyber" a question. It will then go and check the current database of questions to see if an answer matches the question. If it does, it will just return that; otherwise, it will go and generate a response to the question and have a Cyber Ambassador check it, giving the user a basic "come back later" response. Once the question is answered, the person will either get an email telling them there is an answer, or the user can come back any time to ask since the question will now be in the database.

4.3.4 Areas of Concern and Development

Currently, our design and design process is focused on user accessibility. As we progress through the development of Ask Captain Cyber, if we keep a user-centric approach, we will ensure that user needs will be taken care of without an extra effort to implement them later. As stated throughout our design documentation, ensuring that Ask Captain Cyber is intuitive and easy for customers to interact with is a key feature. This requires our site to be useful to a wide range of experts. Keeping this in mind, we must establish an efficient frontend-backend communication protocol and an accurate response system that does not leave the user with more questions.

This does raise some concerns that we must take into account to deliver a sustainable and efficient product. We are required to host Ask Captain Cyber on WordPress. In order for the user interface to be customizable and dynamic, we will have to develop workarounds that would be necessary for other databases. We also must make sure the responses are pulled from the already vetted answers before relying on the AI implementation so that we know what information is being provided to the user. If there is no readily available answer, our site must pivot to a potential notification system or instead utilize the AI API to provide an answer despite it not being vetted by an expert. After our previous meeting with Dr. Jacobson, we gained a deeper understanding of how to design the expert login. We must provide one login for each chapter of ambassadors that they can use to vet the answers. This could make it easier than we initially

thought, as there won't be as many experts to adhere to. Finally, we must provide a list of related questions for each question a user asks. This will require an efficient and accurate system to find related topics to each question.

To address these concerns within our solution, we will have to focus on both user needs and requirements provided by our advisor. We will continue to develop a frontend that is intuitive and easy to use that leaves no room for confusion when interacting. Our backend and security team will also build efficient plugins that adhere to our goal of a secure and impenetrable backend that can be efficiently searched to provide accurate and detailed answers to many potential questions. Since we just recently met with our advisor, we had a couple of questions to ask. These ranged from what kind of login system he wants, whether or not a multifactor authentication is necessary, and what disclaimers must be shown to users. As we implement the answers to our questions, more questions will surely arise; however, we have built a positive relationship with Dr. Jacobson that will allow us to be open and honest with the roadblocks we face.

4.4 Technology Considerations

One of the main technologies we are using is ChatGPT 3.5. One of the main strengths of using this is that it has a large support backing compared to other AIs. But it does have a weakness that we will have to work around, and this weakness is being confidently wrong. One of the ways we are getting around this is by using ambassadors and our own data set. Another technology consideration we are looking at is WordPress. The main advantage to using this is it would help provide a start to the website and has a lot of plugins that we can use. One disadvantage to using WordPress is that our project might require things that WordPress doesn't have plugins for. We can overcome this by simply making our plugins. WordPress also has a lot of built-in features, such as databases, which makes it a lot easier to develop. WordPress also has built-in react programming, which makes it easy to add react code for the front and backend.

4.5 Design Analysis

So far, we have tested a few different AI models available online and conducted a plethora of research to ensure a seamless implementation of our "Ask Captain Cyber" application. We have also begun building the frontend for multiple web pages used in the project, alongside implementing a few backend plugins. So far, our proposed design has been working well and should represent the final project, given our current progression. One issue was setting up our local environments, but this issue has largely been resolved. For our future design, we may change which plugins we use in case any conflicts arise during our development. As a whole, our design is not only feasible but on track for a successful launch by the anticipated deadline.