GNG2101

Design Project Progress Update

“Gee-Gees Access Devs” GNG2101 Group 1.5

Submitted by:

Jonathan Soliman, 300345528

William Levesque, 300352375

Andrius Avenido, 300344515

Aidan Nikolaus, 300309124

Jacob Cearns, 300361219

September 29, 2024

University of Ottawa

1. Table of Contents

[Table of Contents ii](#_Toc178507847)

[List of Figures iii](#_Toc178507848)

[List of Tables iv](#_Toc178507849)

[List of Acronyms and Glossary v](#_Toc178507850)

[1 Introduction 1](#_Toc178507851)

[2 Sustainability Report and DFX 2](#_Toc178507852)

[2.1 Sustainability report 2](#_Toc178507853)

[2.2 Design for X 4](#_Toc178507854)

[3 Problem Definition, Concept Development, and Project Plan 5](#_Toc178507855)

[3.1 Problem definition 5](#_Toc178507856)

[Problem Statement: 6](#_Toc178507857)

[3.2 Concept development 7](#_Toc178507858)

[3.3 Project plan 9](#_Toc178507859)

[4 Detailed Design and BOM 10](#_Toc178507860)

[4.1 Detailed design 10](#_Toc178507861)

[4.2 BOM 10](#_Toc178507862)

[4.3 Project plan update 10](#_Toc178507863)

[5 Conclusions 11](#_Toc178507864)

[6 Bibliography 12](#_Toc178507865)

2. List of Figures

Figure 0. Decision Matrix.................................................................................................. 11

Figure 1. Home Page of OpenAI Chatbot Concept........................................................... 12

Figure 2. Tabbed Home Page Concept.............................................................................. 12

3. List of Tables

Table 1. List of Acronyms ....................................................................................................v

Table 2. Glossary ..................................................................................................................v

Table 3. List of Client Needs by priority...............................................................................5

Table 4. List of Inspired Metrics ...........................................................................................6

Table 5. Target Specifications................................................................................................7

List of Acronyms and Glossary

Table 1. Acronyms

|  |  |
| --- | --- |
| **Acronym** | **Definition** |
| LMS | Learning management service |
| LTI | Learning tool integration |
| WCAG | Web Content Accessibility Guidelines |
| AI | Artificial Intelligence |
| Bom | Bill of Materials |

Table 2. Glossary

|  |  |  |
| --- | --- | --- |
| **Term** | **Acronym** | **Definition** |
| Ally | N/A | A tool that integrates with LMS to improve content and accessibility |
| Accessibility | N/A | Ensuring content is usable by people with disabilities. |
| Web Content Accessibility Guidelines | WCAG | A set of guidelines for making web content accessible. |

# Introduction

This project aims to develop a web-deployed tool that analyzes course content on canvas, the learning management system (LMS) used at Rutgers University to help professors improve accessibility for students with disabilities. The US Department of Justice recently introduced new rules on web and mobile accessibility requiring a review of each course to ensure compliance. This software will leverage the outputs of ally, an accessibility advisory tool, to provide simple and easy to follow recommendations for content improvement. The scope of this document includes sustainability, consideration problems, definition concept, development, and project planning. Ally outputs a file in .CVS format which our software will then receive as input and gives out step-by-by step instructions on how to increase their accessibility score, making the classroom a space for everyone.

# Sustainability Report and DFX

## Sustainability report

The proposed software promotes long-term sustainability by reducing barriers faced by students with disabilities and enabling more equitable access to education. By integrating accessibility into course development, it guides and encourages inclusive training environments contributing to both educational and social sustainability.

**Social Impact:**

* + **Positive:** Increased accessibility in education promotes equal opportunities for students with disabilities, fostering an inclusive academic environment.
  + **Positive:** Educators become more aware of best practices to enable a more inclusive environment.
  + **Positive**: More accessible courses means that everyone can learn more efficiently.
  + **Negative:** The additional workload on professors to modify and update their course materials could lead to stress or burnout.
  + **Negative:** Professors with limited technical knowledge may need more help to make their courses more accessible

**Environmental Impact:**

* + **Positive:** The software reduces paper use as more content is transitioned online and optimized for digital formats.
  + **Positive:** Less time spent using a computer to organize a class’s content will lead to a smaller carbon footprint since less electricity will be needed.
  + **Negative:** Increased digital content may lead to higher energy consumption due to cloud storage and data processing requirements.
  + **Negative**: Since the tool is web-deployed, the server it is running on will constantly be using electricity.
  + **Negative:** Using a chatbot on our website will significantly increase our carbon footprint since AIs utilize a lot of power.

**Economic Impact:**

* + **Positive:** Improved accessibility compliance reduces the risk of legal penalties for institutions, leading to cost savings.
  + **Positive:**  No need to get external hires to help make professors courses more accessible, using our tool professors can do it themselves
  + **Negative:** Initial costs of implementing the software and training faculty may strain educational institutions' budgets.
  + **Negative:** Maintaining the software past the initial deployment dates comes with monthly costs such as keeping licenses that are used, which will add more financial burden to the client.
  + **Negative:** In the future, the technologies that were used in the creation of the software may be deprecated, leading to another cost and problem that will impact the client

Overall, our proposed software offers significant social benefits as aiding professors to improve their online course accessibility will enable inclusivity, awareness, and learning efficiency for students and professors alike. Environmentally, the software will let more content be efficiently held online, reducing physical waste. But there are negative effects such as power consumption and carbon footprint with the certain technologies being used in the software. Economically, the software provides cost savings by improving standard compliance and external hires, but ongoing maintenance poses financial concerns. Therefore, the success of the software will depend on balancing its positive impacts and improving them, as well as specific implementation to mitigate negative aspects that may arise in topics such as energy consumption and economic sustainability.

**Life Cycle Assessment (LCA) Framework:**

**Objective and Scope:**

The focus of this section is to assess the environmental impact of the Rutgers accessibility tool throughout its lifecycle, focusing on energy consumption, emissions, material usage, and overall digital infrastructure requirements. We can create an LCA based on existing software tools and their impacts such as the Ally Canvas tool used at Rutgers.

We specifically define the boundaries of our software product life cycle as such: development phase, distribution phase, use phase, and end of life phase. This will allow us to focus our assessment and analysis on the direct impact of the software product from its implementation to its end of life. See inventory section for our breakdown for the scope and analysis of our included boundaries.

We also define our comparison criteria, which will be solely focus on environmental impact metrics such as our carbon footprint, energy consumption, and resource depletion Furthermore, our time for assessment will cover its software development, which will be the time period of 1 semester. Based on if we can reach deployment, then this will extend to the software’s lifetime, and this assessment would possibly last years. There are a few specific limitations to our assessment such as geographic limits as we can only assess the impacts on our environmental location, and system limits as we can only analyze the software tool itself, and no other websites or software that will be used with the tool.

**Inventory Analysis:**

Gather data on energy use during software development, hosting, and maintenance. Include usage of cloud services, data storage, and professor engagement time. We can analyze our energy and other costs through our set boundaries:

1. Development:
   1. Energy consumption during research, coding, testing, and implementation.
   2. Other byproducts of energy consumption, such as emissions and waste from power usage
   3. Resources used for developer workstations
2. Distribution:
   1. Energy consumption for hosting the software on physical or cloud servers
   2. Energy consumption for distributing adjacent resources for the software such as video guides and online documentation
3. Use:
   1. Energy consumption of end user devices and clients, cloud servers and infrastructures, data transfers, and maintenance/upkeep.
   2. A notable impact is the possible use of a large language model, which would be a significant energy cost
4. End of life:
   1. Disposal of development hardware
   2. Decommissioning and disposal of cloud services
   3. Management of post-use data and storage (for deletion)

**Impact Assessment:**

Evaluate the energy and resource consumption of the software, focusing on CO2 emissions from cloud computing and user interaction. Consider positive social impact (accessibility for disabled students) and any environmental trade-offs.

**Interpretation:**

The analysis will guide design decisions to reduce the environmental impact (e.g., using more energy-efficient servers, minimizing data processing needs) while enhancing the tool’s accessibility features.

## Design for X

Although this design focuses on many potential DFX points, the focus will be a “Design for Accessibility” an approach that ensures the software focuses on the ease of use for educators while maximizing the ability for students to access high-quality education. This principle guides the development of a product that prioritizes accessibility and improves based on the latest web standards and regulations.

**Such factors will include:**

1. **Accessibility**

Objective/Need:

Ensure all course materials can be easily analyzed for professors, but students also having ease to access the courses information.

Metrics: WCAG 2.1 compliance score.

Constraints: Content must meet all Level AA standards for accessibility.

Design Criteria: The graphical user interface design must accommodate screen readers and provide alternative text for images.

2. **Usability**

Objective/Need: Professors should easily navigate and use the tool with ease.

Metrics: Time to complete an accessibility review.

Constraints: Tool should require minimal technical knowledge.

Design Criteria: A user-friendly chatbot interface with a simple chat box providing 1 on 1 conversations.

3. **Data Efficiency**

Objective/Need: Reduce the amount of energy used when processing and analyzing data.

Metrics: Energy use per analysis.

Constraints: All processing should not have significant impact on server load.

Design Criteria: Optimize algorithms to ensure no unnecessary processes.

4. **Security and Privacy**

Objective/Need: Ensure that all student and professor data is securely handled.

Metrics: Number of security breaches (aim for zero).

Constraints: Must comply to the necessary privacy regulations.

Design Criteria: Use secure encryption and data handling protocols.

5. **Scalability**

Objective/Need: The tool must handle increased user load as more professors begin to use the tool for courses.

Metrics: Reaching maximum user capacity without performance degradation.

Constraints: Must function efficiently regardless of the amount of course resources loaded into the site.

Design Criteria: Cloud-based infrastructure with scalable resources.

# Problem Definition, Concept Development, and Project Plan

## Problem definition

With the introduction of new accessibility rules by the US Department of Justice, Rutgers University must review its courses on Canvas to meet compliance standards. Professors often face challenges in interpreting Ally output files, an extensive accessibility report, which can be extremely complex and hard to read. These complexities make it difficult to prioritize which issues should be addressed first. Our task is to create a software tool, specifically a chatbot or other forms of website software, that simplifies this process by providing recommendations, making the content process more manageable and easier to process. Below is a list of client needs based on analysis of their statements:

**Table 3. List of Client Needs by priority**

|  |  |  |  |
| --- | --- | --- | --- |
| Rank | Client Statement | Need Statement | Priority Score (Out of 5) |
| 1 | I want the software to condense information by Ally, and present it to faculty with specific feedback recommendations | Software can scan and condense Ally feedback, and provide information and recommendations to user | 5 |
| 2 | I want faculty to easily understand how to use the software | Software is easy to use and is accessible for users | 5 |
| 3 | I want the software in the form of a chatbot, or other interactive formats | Software is in an interactive format, emphasis on chatbot | 4 |
| 4 | I want the software to be reliable and fast | Software interactions and processes are fast, and has minimal downtime | 4 |
| 5 | I want the information presented to be easy to understand. | Information is concise and simple; easy to understand. | 4 |
| 6 | I want the application to be web hosted. | Chatbot is accessible through a URL. | 2 |
| 7 | I want it to switch between dark and light mode depending on the time of day | Chatbot is easy on eyes and has a good appearance. | 1 |

## Problem Statement:

The client requires a user-friendly website to assist professors in improving the accessibility of their courses based on feedback from a CSV report generated by the Ally LMS tool. Currently, the CSV files provided by Ally are too complex for professors to interpret, making it difficult for them to understand where to begin in making their courses more accessible. The website will be able to simplify the information, offering clear, actionable steps that guide professors through the process of enhancing course accessibility in an intuitive and supportive manner. In addition, proposed impacts such as environmental sustainability and impacts shall be considered during concept planning,

**Table 4. List of inspired metrics**

|  |  |
| --- | --- |
| Need | Metric(s) |
| Software can scan and condense Ally feedback, and provide information and recommendations to user | Feedback Accuracy (scoring based on user), response time |
| Software is easy to use and is accessible for users | Accessibility Compliance and Website Responsiveness |
| Software is in an interactive format, emphasis on chatbot | User Interface Accessibility score based on users |
| Software interactions and processes are fast, and has minimal downtime | Response Time, Uptime |
| Information is concise and simple; easy to understand. | Intent Recognition Accuracy (using Chat bot or other service, see concepts below) |
| Chatbot is accessible through a URL. | URL Availability |
| Chatbot is easy on the eyes and has a good appearance. | Interactive UI and design, based on user score |

Note: User score metric can only be confirmed with external testing, which will be a goal of this project to do in the future. It is an important metric for a website which cannot be measured by other units.

**Table 5. Target Specifications:**

|  |  |  |  |
| --- | --- | --- | --- |
| Metric | Units | Marginal Value | Ideal Value |
| Feedback Accuracy | User Score (out of 5) | 3 | 5 |
| Response Time | Seconds | ≤ 3 | ≤ 1 |
| Accessibility Compliance | WCAG Compliance Level | WCAG 2.0 AA | WCAG 2.1 AA + Nondiscrimination on the Basis of Disability (USA Government Regulations) |
| Website Responsiveness | User Score (out of 5) | 3 | 5 |
| UI Accessibility | User Score (out of 5) | 3 | 5 |
| Uptime | Percentage (%) | 99 | 99.5 |
| Intent Recognition Accuracy | Percentage (%) | 85 | 95 |
| URL Accessibility | URL Availability (Hours/Days) | 23/7 | 24/7 |
| Interactive UI Design | User score (out of 5) | 3 | 5 |

## Concept Development

Professors will have to obtain the .CVS files from the ally LMS tool. Using this file as input, our website will give back actionable feedback in a concise step-by-step message. This feedback will be given in a prioritized manner, based on the impact on the students and will make sure that the professors’ courses meet WCAG guidelines. The software is a website so that professors and users do not need to download additional software, and no setup is needed.

**Final Prototype Concepts and Analysis:**

1. Chatbot Website powered by OpenAI:

A website chatbot that provides real time interactions with users, which uses OpenAI’s public API to implement a large language model to act as the chatbot logic and file analysis

System Layout:

* The core system uses OpenAI's API access a large language model for real-time user interaction
* Key features include user friendly design, ability to analyze upload .csv files based on Ally’s feedback and return the user personalized feedback and recommendations to improve their canvas course accessibility. Possible features include chatbot, custom tailored chatbot responses, user login, chat logs, and accessibility standards appendix.

Analysis against Target Specifications:

* Feedback Accuracy: Expected to be the highest due to OpenAI’s model capabilities as known in existing implementations (such as ChatGPT)
* Accessibility Compliance: Chatbot had simple implementation and key features, so accessibility compliance will be easy to meet
* Response Time: OpenAI offers fast response times, but may have varying times depending on our external requests over the internet
* Intent Recognition Accuracy: Extremely Advanced with OpenAI, expected to meet ideal values

Justification:

* This concept offers high reliability and quick implementation due to having access to OpenAI’s language models. Having chatbot logic and behavior abstracted will allow the development team to focus on implementing accessible front-end website designs.
* However, this option is less customizable as we cannot directly access the chatbot functions. This concept prioritizes fast deployment with moderate customizations.

2. Custom OpenAI Chatbot with “in-house” bot and scraper implementation

A website chatbot that provides real time interactions with users. This concept includes chatbot and .csv scraper and analysis implementation to be proprietarily made by the development team. This will allow more custom integration and more personalized behavior of the chatbot to be in line with the client’s organization.

System Layout:

* Concept focuses on a fully customized and implemented chatbot, using in house file scrapers for .csv files and proprietary chatbot logic
* This system allows for a more personalized layout tailored to client needs
* Key features include user friendly design, ability to analyze upload .csv files based on Ally’s feedback and return the user personalized feedback and recommendations to improve their canvas course accessibility. Possible features include much more custom tailored chatbot responses, concrete chat logic implementation, user login, chat logs, accessibility standards appendix.

**A**nalysis Against Target Specifications:

* Feedback Accuracy: Likely high, as custom scrapers will target specific requirements with tailored accuracy (ideal 5/5 user score).
* Response Time: With a fully in-house system, this can be optimized for speed (ideal ≤ 1 second).
* Customizability: Much greater control over the chatbot logic, making it a better choice for deep integration with the organization’s needs.
* Accessibility Compliance: Easier to meet specific accessibility standards, such as WCAG 2.1 AA.
* Development Time: Longer due to the need to develop both a proprietary chatbot and a custom scraper.

Justification:

* This concept offers greater control over chatbot behavior, but performance is dependent on the in-house implementation
* Higher development Effort is justified for clients seeking long term scalability and personalization. But, depending on the team, deployment would take much longer than the scope of the allowed development time.

3. Pop Up Extension

A web browser extension that injects an analysis overlay of the canvas website providing real time feedback. It can also provide popup feedback if user chooses to upload Ally .csv feedback. This allows the user to simultaneously update their course content while also seeing recommendations on their screen.

System Layout:

* The core function of this system is injecting an analysis overlay directly on the user’s canvas interface, providing real time feedback as professors modify their course content
* Based on Ally’s .csv analysis, can provide pop-up feedback
* Features and sub-systems include web overlay, voice assistant, and real time visual metrics

Analysis Against Target Specifications:

* Feedback Accuracy: Provides direct integration with course content, enabling real-time, highly accurate feedback.
* Response Time: Instant feedback as the extension runs client-side with minimal latency
* Customizability: Offers a unique, real-time interface but lacks depth in chatbot interaction compared to other concepts.
* Accessibility Compliance: Can comply with WCAG but requires additional design work to integrate voice assistants and visual metrics.
* Development Complexity: Requires both extension development and API integration with Ally’s feedback.

Justification:

* Concept provides real time visual feedback at a much faster response and feedback rate than other choices. It is more intuitive but less flexible and requires substantial custom development to meet accessibility standards.

4. Accessibility Dashboard

A website that provides an analytical dashboard which provides metrics based on Ally’s .csv feedback.

System Layout:

* A dashboard website that presents course accessibility metrics and analysis based on Ally’s .csv feedback
* Key features and subsystems include .csv analysis, direct course accessibility metrics, feedback and resources. Possible features include a forum posting function for instructors to help each other with course feedback and canvas templates, guides etc.

Analysis Against Target Specifications:

* Feedback Accuracy: Very high due to focus on data from .csv analysis
* Response Time: dashboard concept may not provide instant feedback, but data retrieval and visualization can be optimized
* Intent Recognition: Does not rely on chatbot interactions, so this target is not relevant, which can reduce implementation time.

**Justification:**

* Concept is good for users who want to visually explore the canvas metrics of their website for accessibility, but it is not as interactive as the other choices.

**Figure 0. Decision Matrix:**

A white table with black text

Description automatically generated

**Chosen Concept and Global Design:**

Based on the updated decision matrix, the OpenAI Chatbot is the clear winner, scoring 27 points, making it the best option for the following reasons:

* Lower Implementation Complexity: OpenAI offers a ready-made, powerful solution that can be easily integrated, which significantly reduces the development time and technical complexity.
* Cost Efficiency: Since running and maintaining a custom chatbot requires constant resource allocation and technical expertise, OpenAI’s cloud-based solution is more cost-effective in terms of upfront and ongoing expenses.
* Modern, Scalable Solution: It meets the need for a modern, scalable, and user-friendly interface, as preferred by both customers and professors. The software is scalable as implementation is based on OpenAI functionality and front-end design, both of which can be highly adaptable to future changes.

**Global Concept Visualization**

Figure 1. Home Page of OpenAI Chatbot Concept

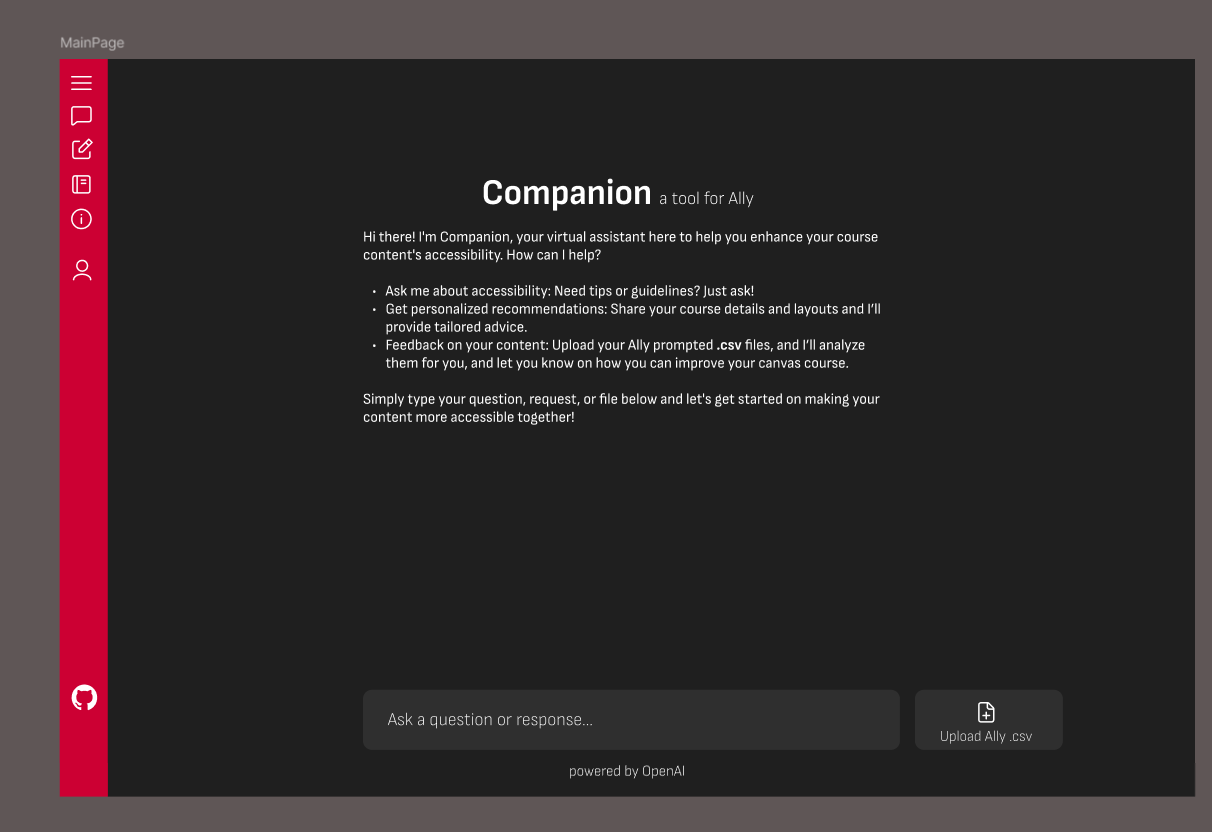
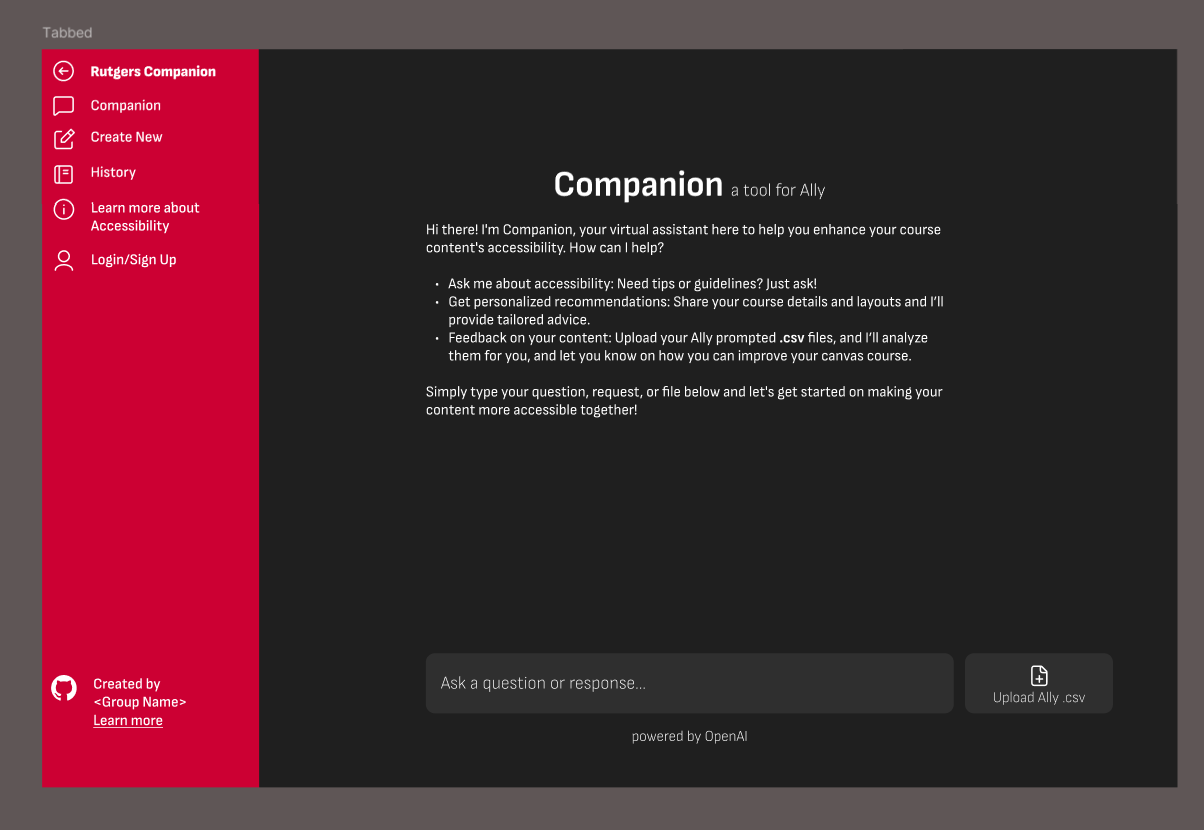


Figure 2. Tabbed Home Page Concept



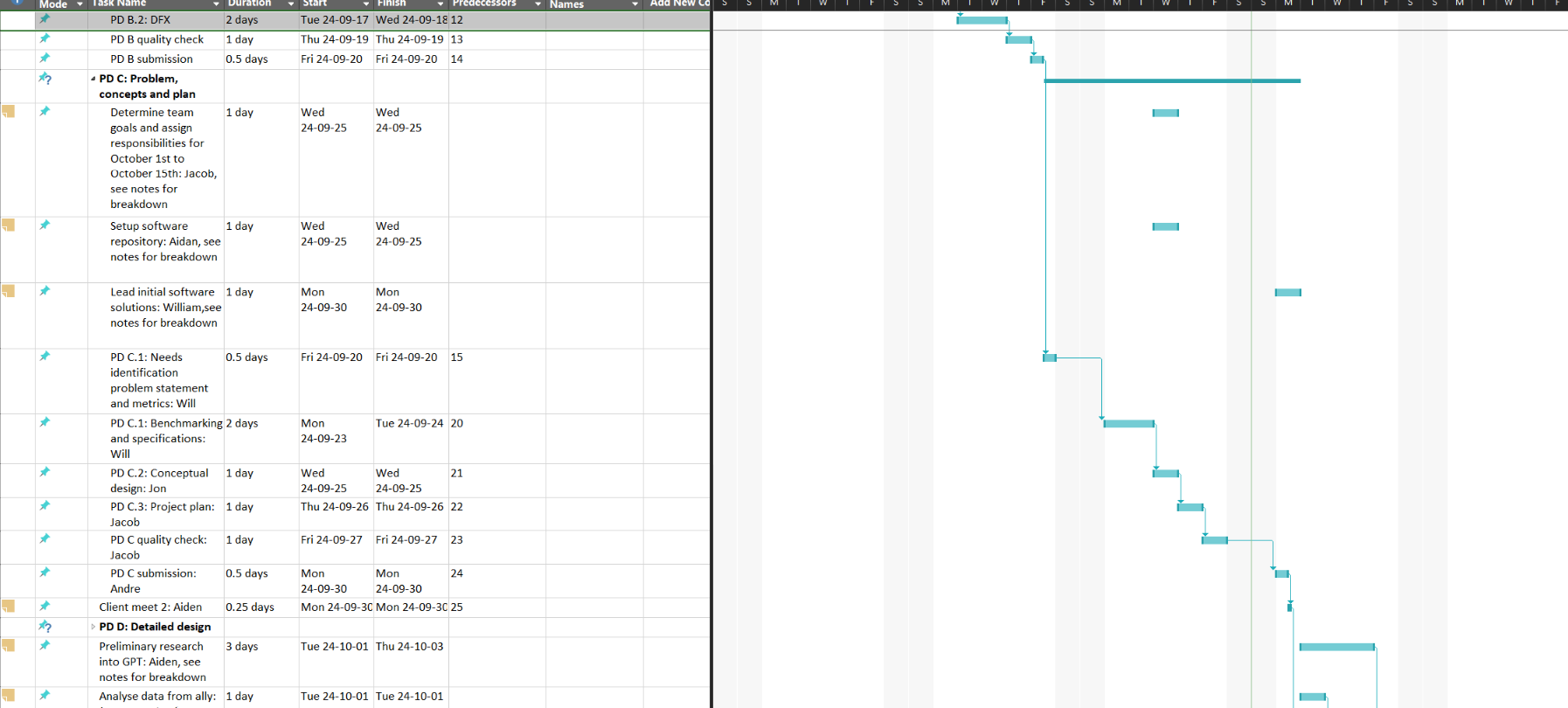
**Analysis of Concept and Target Specifications**

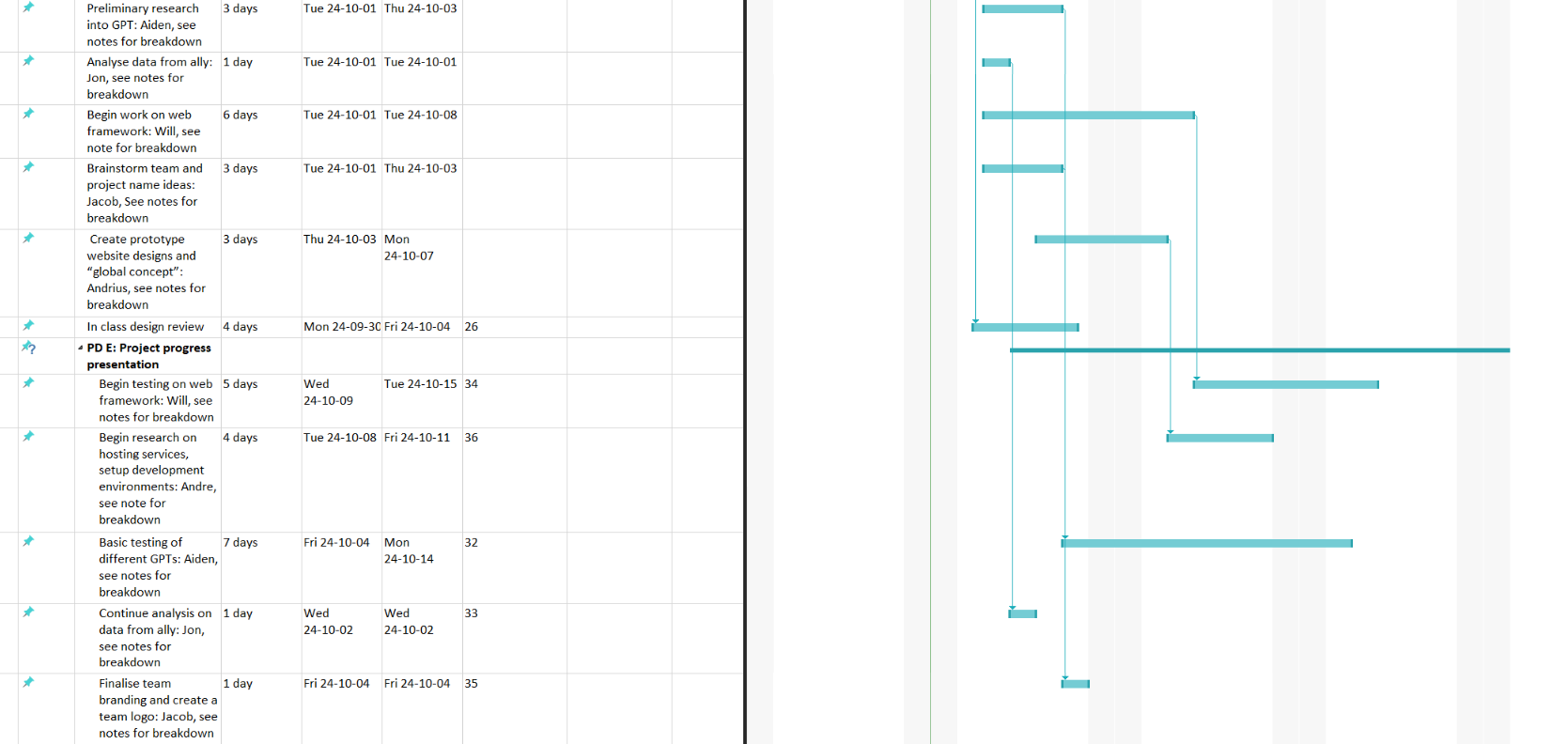
The OpenAI-powered chatbot solution meets the target specifications by ensuring WCAG 2.1 AA compliance, providing an accessible and user-friendly interface for professors and students. It enhances feedback accuracy with natural language processing to achieve a user score of 5/5, while its optimized cloud infrastructure supports fast response times (≤ 1 second) and improves website responsiveness and UI accessibility (targeting a score of 5/5). Additionally, it supports uptime of 99.5%, ensuring constant availability, and complies with privacy regulations through strong data security protocols, safeguarding both student and professor information.

**Analysis of Concept and DFX**

The solution aligns with the DFX factor as it is designed for accessibility, ensuring all users, including those with disabilities, can interact easily with the platform. It also emphasizes usability, requiring minimal technical knowledge, and energy efficiency, reducing server load through optimized processing. Security is prioritized with encryption and privacy measures, while scalability is ensured through a cloud-based infrastructure that can handle increasing user loads without performance degradation. This aligns the solution with key DFX principles, optimizing it for long-term performance and adaptability.

**Project Plan Below** (Descriptions for each task are in notes for task inside the MS project. Names of who each task is assigned to is under task name on the left.)







# Detailed Design and BOM

## Introduction and Client Summary

## This section describes our detailed design and bill of materials to be presented in our design review day. This is the final step in our project before our first prototype implementation. Based on our DFX and concept analysis, we have formulated an in-depth detailed design of both our software user interface and the related technology systems interactions the drive the software. During our second client meeting, the client was very satisfied with our presented concept in terms of user interface design and technologies. The client believed that our concept matched their needs and vision for the software. However, a large concern brought up: professors are concerned how their user data will be handled when interacting with an AI powered software. We intend to address this concern by choosing an enterprise model of OpenAI that does not collect user data. This will be focused on our design and future prototype and will be discussed in our next client meeting. Overall, we are satisfied with how our project progress is going and we are working attentively to improve and implement our designs.

## Detailed Design

Figure 3. Wireframe UI Design Overview

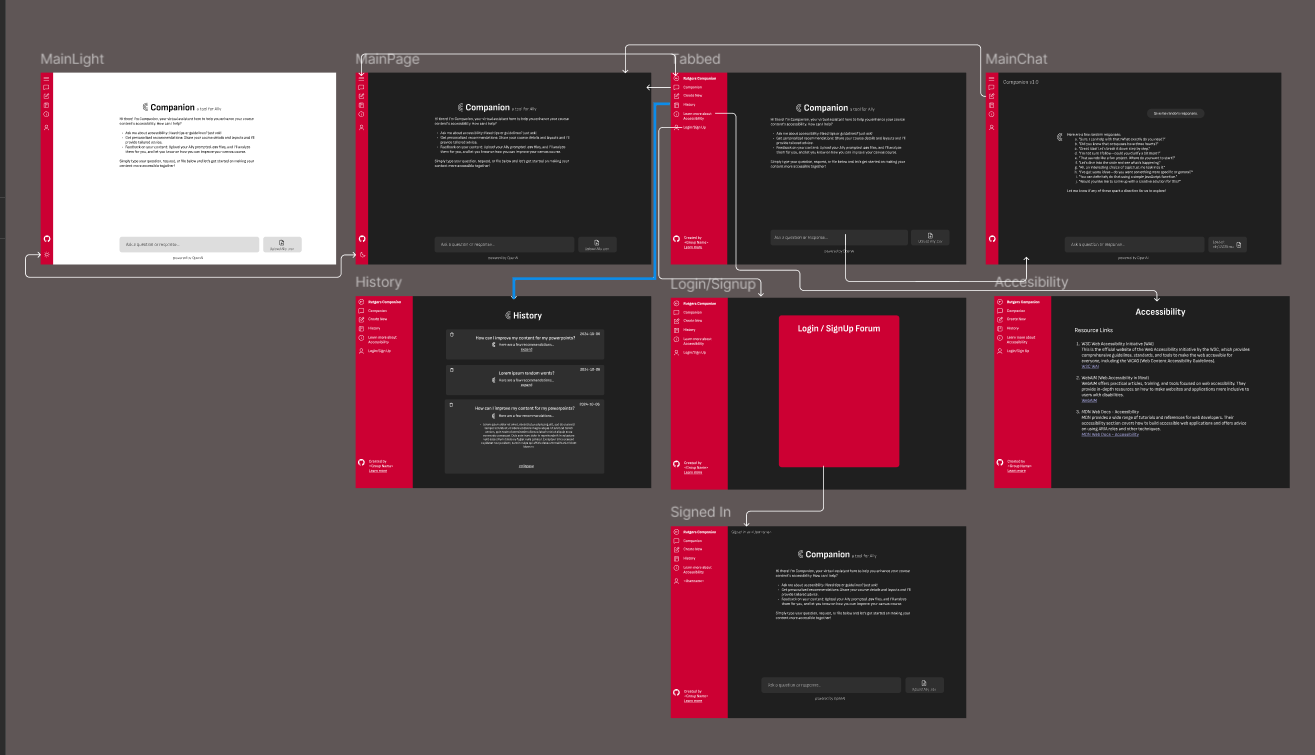


Figure 4. Wireframe UI Design Zoomed In 1.

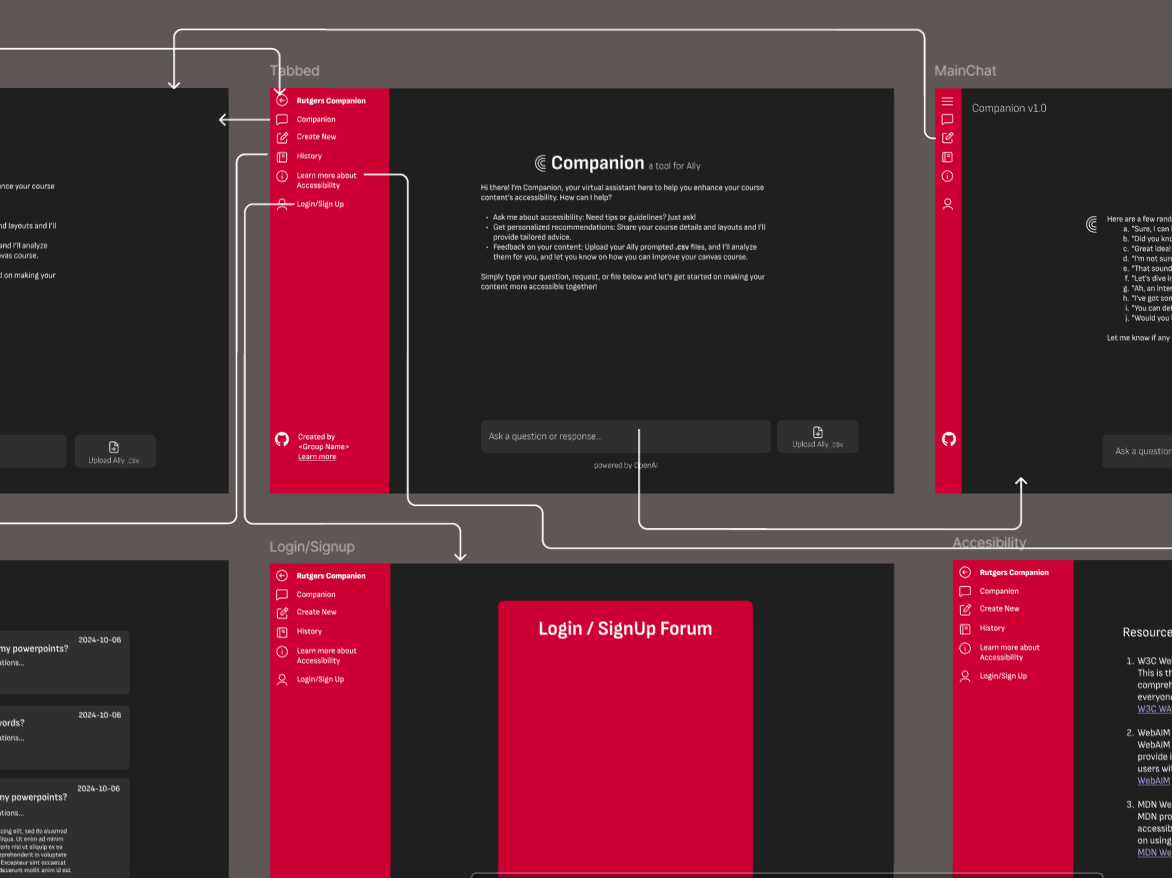
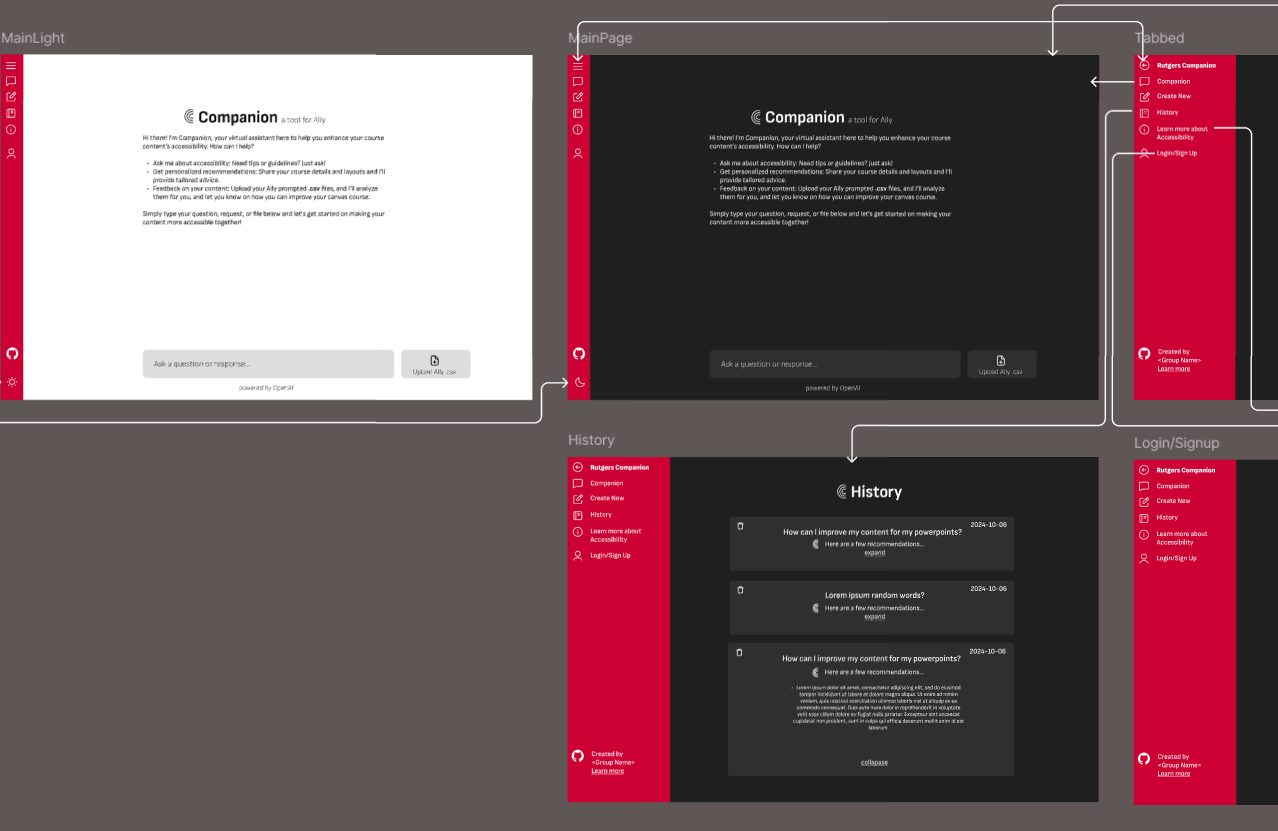
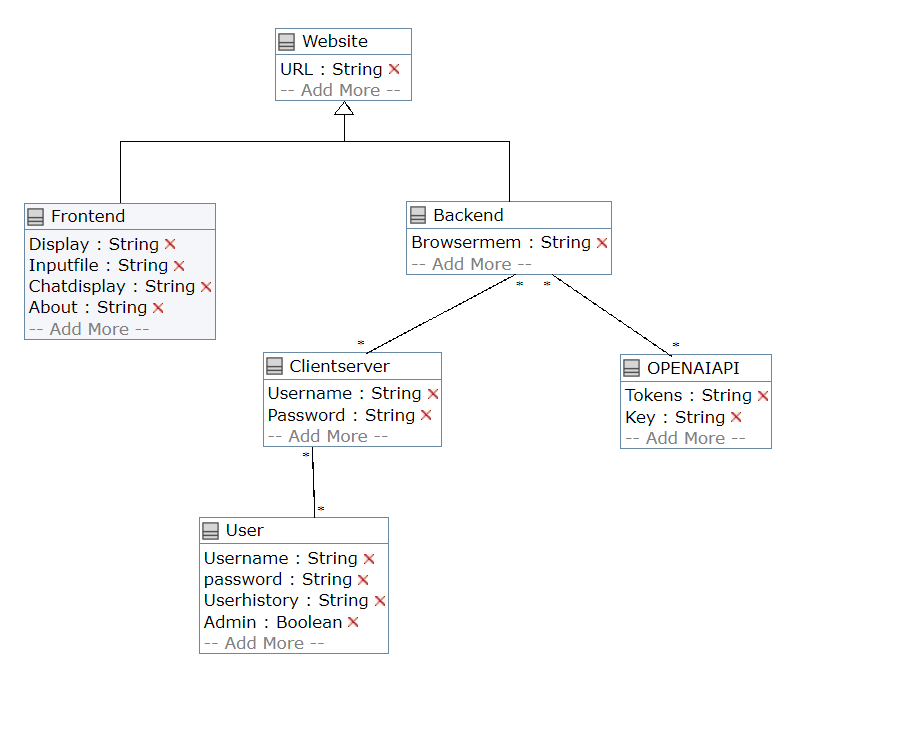


Figure 5. Wireframe UI Design Zoomed In 2.



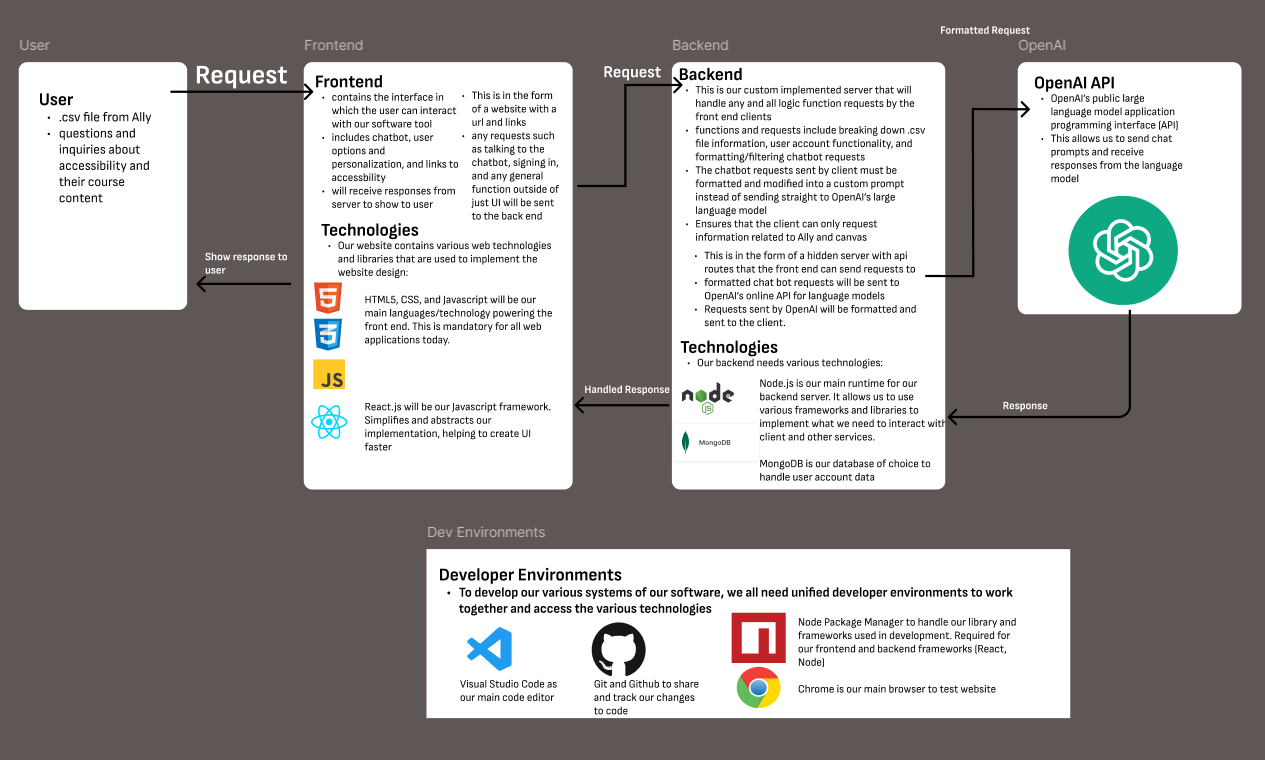
The figures above show our detailed UI design for the front side of our website and its various pages.

Figure 6. Initial Software Systems Design



This figure shows our preliminary design of our systems interaction for the entire software

Figure 7. Software Systems Interaction and Technologies Diagram



This figure shows our detailed systems interaction diagram with various technologies that we will be using to implement our software prototype.

Overall, our design was created with the main DFX factors in mind in terms of accessibility and usability. We went with a simple layout that is easily understood and with user-friendly icons. We also added many features that are staple in many other websites so that users can be familiar when accessing our software. The convern of security and privacy will be addressed through our implementation of our backend logic as we hope to use technologies that will help with authentication and storing user data safely. Finally, we want the software to be easily scalable and hosted easily, thus our chosen tech stack and database will help us ensure that target of scalability.

**Design Considerations**

Since our software is intended to be used by a lot of faculty users at Rutgers, we take note of a few considerations regarding our detailed design based on DFX and some other factors. Design for accessibility and usability are the main driving factors of our focus for this detailed design. We must consider that our UI layouts must appropriately follow accessibility and usability design conventions such as user-friendly icons, fonts, and simplistic functionality to ensure that our software is easy to use. Security and privacy were the main concerns during our client meeting; thus, it is an important consideration to take security steps when implementing the capabilities of storing user data.

**List of Skills and Resources of each Team Member**

Aidan:

* Knowledge of web applications, with a focus on the back end.
* Strong understanding of designing GUIs using various graphical applications.
* Experience primarily in back-end development.

Areas for Improvement:

* Gain a deeper understanding of developing web applications, particularly using React.
* Expand knowledge of OpenAI API services and their implementation.

These skills can be enhanced by utilizing tutorials on YouTube or referring to official documentation for the relevant tools and functions helpful to the development.

Andrius:

* Experience in creating full stack web applications using the aforementioned technologies
* Can provide pre-existing projects like this one as resource that can be used for implementation reference
* Knowledge of developer environment and how to install each technology on other member’s devices

Missing Skills and Resources:

* Unfamiliar with implementing websites into hosting services.
* Unfamiliar with using OpenAI API services.

A lot of research and time is dedicated to learning how to make our detailed design succeed by learning how to use and leverage these weaknesses in knowledge.

Jacob:

Current

* Experience in basic coding in C, which will allow for the quick acquisition of other coding skills.
* Strong communication skills and presentation experience, which will assist in the various presentations throughout the project.
* Project management experience gained from GNG1103.

Missing skills

* No web development experience or skills (Can be improved by following both the direction of more experienced groupmates along with the use of online resources).
* Limited experience with AI (Can be improved by following both the direction of more experienced groupmates along with the use of online resources)

Jonathan:

* Strong problem-solving skills and a methodical approach to tackling challenges, which can contribute significantly to debugging and designing logical structures for the software.
* Ability to understand complex systems and break them down into manageable parts, useful for software architecture and understanding project requirements.
* Experience in CAD software and visualization tools, which can be helpful in prototyping user interfaces or visual aspects of the software.
* Practical project management experience from working on mechanical projects, which can assist in planning, task delegation, and ensuring project deadlines are met.

Areas for Improvement:

* Limited experience in software development languages and frameworks used in the project (e.g., JavaScript, React, Node.js).
* Unfamiliar with the process of integrating APIs, such as OpenAI API, into software applications.
* Needs more exposure to version control systems like Git, commonly used in collaborative software development.

William:

* Experience in front-end development
* Knowledge of commonly used software to develop websites as well as aiding team members getting familiar
* Experience in UX design
  + Missing skills:
    - Less experience with web frameworks such as react and node.js
      * Will watch tutorials and learn as I progress through the project as well as advice from Andrius
    - Unfamiliar with developing back-end
      * SEG 2105 class has us developing an android application including a back-end system, most of my knowledge will come from this class, otherwise I will watch YouTube tutorials

**Realistic Time Assessment**

Below is our realistic time assessment of how our project implementation project will go. This estimate is based on our available times and other responsibilities we have such as schoolwork.

* Building the website front-end
  + 2-2.5 weeks
* Styling the website and creating animations
  + 2.5-3 weeks
* Implementing and creating the back end
  + 2 weeks
* Implementing OpenAI’s API
  + 2 weeks
* Final touches/ fixing bugs
  + 2 weeks

It is important to note that different members of the group will be working asynchronously on different tasks. As such, these time assessment tasks can be developed simultaneously and thus brings our total estimation of time for the entire project to be finished to be: 2 MONTHS.

**Critical Product Assumptions**

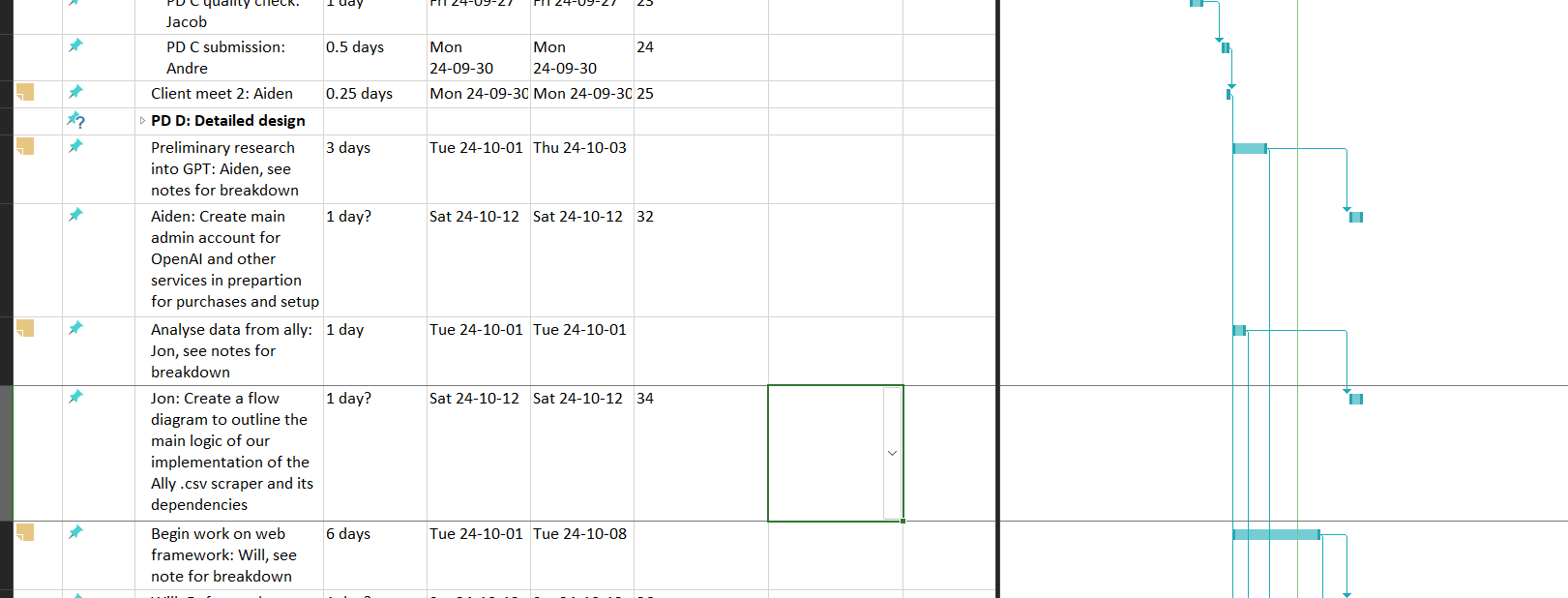
* OpenAI’s model will be available for purchase when we start developing the website. Our development plan is dependent on its services. If we are unable to implement it onto our website, we will have to find an alternative large language model to use such as CLAUDE or Gemini.
* The chatbot will be able to be trained to perform a specific task of analyzing a .CVS file and giving step by step instructions to the professors. None of us have experience training chatbots so we will have to do our research well to accomplish this task.
* Assumes that the AI will be able to handle errors encountered such as a file not being in .CVS file, the file being empty or not in the format expected.
* Assumes that the response times from the API are fast enough to provide real-time or near real-time analysis without frustrating users.
* Assumes that OpenAI's API is secure, and user code submissions are handled in a way that ensures confidentiality.

## BOM

Table #. Preliminary Bill of Materials

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Item Name | Description | Units | Qty | Unit cost | Extended cost | Link |
| GPT-4o API Key | OpenAI’s latest large language model implementation. The API key will allow us to make requests to the OpenAI’s model. The pricing is based on volume of input tokens | API Tokens | 1 | $2.50 / 1M input tokens $1.25 / 1M cached\*\* input tokens $10.00 / 1M output tokens | For one month of development testing, and assuming 50 requests a day (15 000 input and output tokens) is roughly $0.1875 per day or $5.65 for one development month | <https://openai.com/api/pricing/> |
| Render Hosting | A hosting platform to host our website after development is finished. | n/a | 1 | 0$ per month for small scale projects | 0$ during our development process for the semester. Prices will increase when in deployment for client | <https://render.com/pricing> |
| MongoDB Atlas | A cloud database used to store our information collected by website such as user accounts etc. | n/a | 1 | 0$ for small scale projects | 0$ during our development process. Costs will increase when real users begin to create accounts during deployment | <https://www.mongodb.com/pricing> |
| Developer Environment | Our detailed developer environment is used to implement the software as shown in the detailed design. All technologies are free. | n/a | 1 | 0$ | 0$ |  |
| Total product cost (without taxes or shipping) | | | | | $5.65 USD |  |
| Total product cost (including taxes and shipping) | | | | | $6.48 + $0 Shipping USD |  |

## Project plan update



# Conclusions

In conclusion, the development of the accessibility tool proposed our team demonstrates our commitment to improving course accessibility for professors and students using the Ally system. By leveraging OpenAI technology, the tool simplifies complex accessibility reports, providing professors with clear, actionable steps to enhance the accessibility of their Canvas courses. The project highlights sustainability through its social, environmental, and economic benefits, while also addressing concerns about user privacy and data security.

The team's design for accessibility, usability, and energy efficiency ensures that the software not only meets current accessibility guidelines but also remains scalable and adaptable to future needs. Overall, the project is on track to provide a valuable solution that aligns with modern web standards and accessibility requirements

# Bibliography

Accessibility Standards and Guidelines

World Wide Web Consortium (W3C). (2018). Web Content Accessibility Guidelines (WCAG) 2.1. Retrieved from <https://www.w3.org/TR/WCAG21/>

US Department of Justice. (2024). New Rules on Web and Mobile Accessibility. Retrieved from <https://www.justice.gov/>

Life Cycle Assessment (LCA) Framework

Curran, M. A. (2017). Life Cycle Assessment: Principles and Practice. National Risk Management Research Laboratory, U.S. Environmental Protection Agency.

Client Needs and Problem Definition

Cagan, J., & Vogel, C. M. (2012). Creating Breakthrough Products: Revealing the Secrets that Drive Global Innovation. FT Press.

Ulrich, K. T., & Eppinger, S. D. (2020). Product Design and Development (7th ed.). McGraw-Hill Education.

* **Design for X (DFX) and Concept Development**

Dieter, G. E., & Schmidt, L. C. (2013). Engineering Design: A Materials and Processing Approach (5th ed.). McGraw-Hill Education.

Cross, N. (2008). Engineering Design Methods: Strategies for Product Design (4th ed.). John Wiley & Sons.

* **Usability in User Interface Design**

Nielsen, J. (1994). Usability Engineering. Academic Press.

Shneiderman, B., & Plaisant, C. (2017). Designing the User Interface: Strategies for Effective Human-Computer Interaction (6th ed.). Pearson.

* **Sustainability and Social Impact of Educational Tools**

Elkington, J. (1997). Cannibals with Forks: The Triple Bottom Line of 21st Century Business. Capstone Publishing.

United Nations. (2015). Transforming our world: The 2030 Agenda for Sustainable Development. Retrieved from <https://sdgs.un.org/2030agenda>