ENG 4001 Research Project

Talk For Me

Progress Report

Authors:

Addy Dhingra

Keefe Zebastian Dela Cruz

Matthew Fowler

Supervisor:

Derek Abbott

2024s1-EME.EE-DZA-UG-12001

The University of Adelaide

Faculty of Sciences, Engineering and Technology

Semester 1, 2024

NUMBER OF PAGES: 27

Table of Contents

List of Tables

List of Figures

Chapter 1: Executive Summary

With the constant improvements in technology, there has been a clear shift in the ways that non-verbal neurodivergent individuals are able to communicate with the general population. This includes the use of tools, such as phone applications, which using TTS can take the form of an artificial voice for the individual. This project looks to take one such tool and improve it in a way that will create a near seamless artificial voice, which also benefits those who struggle with the process of typing. These tools are more commonly known as Augmentative and Alternative Communication (AAC) tools/applications, communication devices, or "talkers", and assist in construction of sentences/phrases through the use of linking symbols/keywords and predictive keyboards [1]. With numerous options on the market, they look to meet the accessibility needs of neurodivergent people, categorised by cognitive, physical, and speech needs. The main features expected of a AAC application is a simple layout, usually with larger buttons to press to assist those with physical needs who mays struggle to type on a keyboard, as well as predefined options/the ability to categorise and store symbols/keywords for repeated use in communication. With the specific needs of individuals varying highly from person to person, there is often a required trial period to determine if the application/tool is appropriate for the needs of the individual. This can be a slow, and tedious problem, and as a result, a solution that meets all needs is desirable.

Talk For Me, is an application developed by Across the Cloud, which in its current form, presents the user with images/terms to select, then passing these along to a Large Language Model, which creates a sentence to be spoken via TTS. It was created by Dr Matthew Berryman, who suffered a haemorrhagic stroke that left him paralysed for three weeks after and unable to speak. During this time he was frustrated in the limited tools provided by the hospital to allow for him to communicate, sighting paper charts that did not even include things such as the television in his room. As a result, Talk For Me is an application aimed towards predominately the needs of stroke patients, with other disabilities that affect speech also in mind.

The improvements we are looking to make range from UI improvements to better assist the neurodivergent user base, keeping it simple but effective. We also hope to use location and time data, along with the user's previous behaviour to make recommendations on sentences that better fit the context of the user's actions. Such as making recommendations for keywords/phrases based upon the restaurant the user might be at, at the time. Along with this, improvements towards the Large Language Model used will be made to ideally create as seamless of a transition from selection to speaking as possible.

This report goes into greater detail regarding each of the components of the prospective improvements, with a literature review providing greater insight into some of the components of this project, how we will achieve such improvements, and the progress that has been made so far.

Chapter 2: Introduction

2.1. Aims

Talk For Me is a text-to-speech (TTS) iOS application that processes uploaded images to generate contextual text suggestions, which can t hen be spoken aloud by the device. The application is currently in alpha, offering only basic text suggestions. The aim is to enhance this feature and incorporate additional elements to improve the application. Improving these features and subsequently integrating the application into the neurodivergent community ensures future contributions to the advancement of the health caregiving market.

2.2. Background

Nonverbal individuals often face significant challenges with communication. While tools like sign language have provided solutions, they are not universally effective as it is not widely known. Consequently, TTS applications have been developed to enable nonverbal individuals to communicate in real-time with others using a common language.

Artificial Intelligence (AI), particularly through the use of Large Language Models (LLMs), is the key tool driving the improvements for Talk For Me. LLMs can perform language generation and natural language processing, enabling users to input keywords in the form of images and generate complex sentences. With numerous LLM options available, this project will also evaluate their performance to enhance communication speed. Additionally, the project will explore LLMs capable of operating locally on the user's mobile phone.

2.3. Motivation

The Dr Matthew Berryman, of Across the Cloud, inspired the idea for Talk For Me after experiencing a stroke and facing communication struggles. As a results, the application development project is driven by the need to help neurodivergent individuals nonverbally specify their needs to the public. In the past, TTS applications have been slow and cumbersome because users had to manually type and edit every word they wanted to be spoken. However, with the integration of AI, these interactions are expected to become more natural and seamless, eliminating the pauses caused by manual input. AI will achieve this by predicting the most likely sentences based on various factors such as location, keywords, and previous behaviour or chosen sentences. It will the give contextual suggestions to the users allowing them to communicate easier.

2.4. Objectives

To ensure success within the allotted time frame and considering the team's expertise, the project focuses solely on enhancing the application's user interface (UI), functionality, performance, and analytics. By narrowing the scope, the purpose is to develop a high-quality, scalable application. Specifically, the objectives are to make UI improvements, integrate location-based systems, and LLM exploration. UI improvements will enhance both the readability and marketability of the product, particularly for neurodivergent audiences. Integrating location-based systems will create a more personalized user experience, significantly improving the user's quality of life.

Exploring and experimenting with Large Language Models (LLMs) will ensure the generation of consistently appropriate sentences, reducing the frequency of errors.

2.5 Project management

Regular meetings with the team will be held to ensure tasks are complete. Furthermore, regular bi-weekly meetings with Dr Berryman will also ensure that the project remains aligned with the company's needs and values. To adhere to the time constraints, the schedules outlined in Appendix A: Gantt Chart will be strictly followed.

2.6 Report overview

This report provides a review of relevant literature pertaining to the project objectives. It offers a comprehensive overview of the objectives, outcomes, and strategies implemented to achieve them. Additionally, it provides details on the current progress of each approach and the plan to complete them in the future.

Chapter 3: Literature Review

3.1. User Interface

The User Interface (UI) is an essential component of the success of an mobile application, especially for applications that serve specialized purpose such as TTS. The interaction between the user and the functionality of an application must be ensure usability, accessibility, and overall satisfaction of users. Hence, it is important to delve into literature review concerning the importance and components of UI design and their relevance to a TTS application. Previous work and studies on UI design considerations for application will be also explored to identify strategies for future design.

The UI is how the users will interact with application and be delivered it's intended functionalities. The aim of a successful UI design is to enhance the user's ability to navigate and understand the conveyed information [2]. On the contrary, A poorly designed UI would lead to frustration confusion [2]. This is especially important when considering the context of a TTS, where the user base is likely to be impaired and may have difficulties navigating the application. Several studies have highlighted the importance of a well-designed UI no matter the target audience [3]. Application types ranging from mental health to e-commerce platforms, users were found to value and prioritize user friendliness and UI design when evaluating applications [3]. This highlights the importance of designing an interface which is intuitive and ascetically appealing to improve the overall user experience.

3.1.1. Previous Work on UI Design on Applications

The theory of Cognitive Load suggests that reducing the unnecessary mental load required to navigate an application can improve the effective information intake and performance of a task. This can be applied to UI design to increase user friendliness. By reducing the amount of information on a page, user can channel their focus on the intended task.

Studies conducted by Nielsen emphasised that user friendliness increases when interfaces display relevant details and minimise unnecessary information [4]. It was also found that users are more likely to respond well to interfaces with white spaces where information is shown clearly and reduced clutter [4]. Reducing extraneous information was shown to reduce the cognitive load [4].

Kurniawan and Zaphiris conducted studies on assistive technologies for the elderly, defining many benefiting strategies in increasing usability of UI design [5]. It was found that UI with large font size, readable styles, high contrast, and colours were highly effective in aiding the experience of visually or cognitive impaired users [5]. This aligns with the context of a TTS application, harbouring similar type of user base.

The kinetic load of a UI design refers to the physical effort required to perform a task. Studies have discussed the ergonomics of UI, considering the placements of interactive elements [6]. It is key to accommodate the natural human movements in design. Many strategies can be employed to reduce the kinetic load [6]. Buttons and elements should be minimised, team elements closely and use familiar design choices. By optimizing the placement of interactive elements, physical effort required to use an application will be significantly reduced.

When designing a UI many considerations can be adapted from web applications. The Web Content Accessibility Guidelines (WCAG) is a set list of guidelines which web pages adhere to make content more accessible [7]. These guidelines ensure the principles of perceivability, operability, understandable and robustness [7]. These may include increasing the text readability, keyboard accessibility and incorporating alternative text for interactive elements [7]. It was found by Petrie and Kheir found that not only did WCAG improve the user experience for impaired individuals but also for all general users [8]. This is attributed to the accessibility focused design practices such as increasing visibility and navigation [8]. Hence, this strictly aligns with the context of the TTS application, in which considering these guidelines in the design of UI can make an application accessible to a wide range of people, especially with individuals with motor, visual or auditory impairments.

When specifically looking at the UI design choices for an TTS application, many strategies can be explored. When considering TTS platforms for users with speech impairments, a study employed the use of modifiable interface which allow the users to physically change the speech speed, volume, and pitch [9]. The ability to cater to the specific needs of the user and fine tune different settings can significantly improve the accessibility provided by the application [9]. This addition reported positive feedback from users. Although the adjustments to the speech output is handled by the language model, the UI design will facilitate the user's interaction with the functionality.

Joen et al. explored the usage of iconography in UI design for TTS applications. It was found that increasing the reliance usage of visual cues rather than text for UI, improve users' comprehension of the application's functionality and status [10]. This would be done by using visual elements such as checkboxes, animated icons and progress bars. This would lead to an increase user friendliness and also aid impaired individuals to navigate efficiently within the app [10].

3.1.2. Relevance to project and Future Consideration

The review of literature was able to detail the importance of user interface design in presenting the functionalities of the application and enhancing the overall accessibility and experience. Given the context of TTS and the target user base of individuals with different impairments, the insights gathered from the previous studies are directly applicable and can be integrated with in the current system. For instance, the emphasis on minimizing cognitive and kinetic loads aligns with our goal to create an intuitive and user-friendly interface. Moving forward, it important to continuously research data for different designs styles and evaluating them based on user needs. The de. signs should also be compared against the latest market application and trends to ensure the application is up to market standard. Since UI design involves visuals, it may be subjective when evaluating its success. Hence, it would be suggested to test the design and acquire feedback from users regarding its appeal. Regular usability testing with our target user base will help identify areas for improvement. Once designs are finalised, they can be incorporated with in future updates.

3.2. Location-based Systems

The location-based contextual suggestion system will elevate user experience by reducing the need for extensive user input. Planned solutions include location-aware menu suggestions and a location-based word list.

3.2.1. Comparison of APIs for Location-aware menu suggestions

Location-based systems have been utilized in a wide range of applications. The primary advantage of these systems is their ability to retrieve detailed information about the user's current location and effectively integrate this data into the application. The most efficient way to acquire data without conducting extensive surveys is by accessing large databases. This can be seamlessly integrated into the application using an Application Programming Interface (API). Examples of applications and systems that use APIs to achieve this include: "I'm Feeling LoCo", "Contextual Suggestion Track", "URecipe", and "What's Open".

3.2.1.1 I'm Feeling LoCo

"I'm Feeling LoCo" is a location-based, context-aware recommendation system designed by Savage et al. [11] to provide users with personalized suggestions based on their preferences, mood, and contextual information. This system mines social network profiles and leverages the Foursquare Places API. The Foursquare API enables the system to access check-in data, retrieve venue information, and provide location-based services. It also has additional features that ensure data privacy, and regulatory requirements when it comes to location data, and optimal battery usage [11].

3.2.1.2 Contextual Suggestion Track

"Contextual Suggestion Track" is a system designed by Hua and Alonso [12] to deliver tailored recommendations by analyzing explicit user preferences and contextual relevance. It builds explicit models of both general and specific user interests to evaluate the relevance of candidate suggestions [12]. Leveraging the Yelp API, the system accesses a wealth of data from the Yelp website, including reviews, ratings, and location-based information. This extensive dataset serves as the foundation for generating personalized recommendations aligned with the user's preferences and current context. Although the Yelp API was successfully used to gather user information, it is not effective for acquiring restaurant menu data, according to the Yelp API documentation. This highlights the importance of selecting the appropriate API for a project to ensure it aligns with the project's goals.

3.2.1.3 URecipe

URecipe is an application created by Phromchomcha [13] which serves as a virtual kitchen assistant aimed at improving users' culinary skills. Utilizing Android Studio for development and Firebase Realtime Database for user accounts and recipe storage, URecipe also integrates the Spoonacular API to access a vast collection of recipes [13]. The Spoonacular Nutrition, Recipe, and Food API provides "access to over 380,000 recipes, thousands of ingredients, 800,000 food products, and 100,000 menu items" according to Spoonacular support team. Spoonacular allows users to find recipes through natural language searches, such as "gluten-free brownies without sugar" or "low-fat vegan cupcakes." [14]. This gives the possibility to allow the user to be recommended menu items based on food preferences they have selected.

3.2.1.4 What's Open

"What's Open" is an iPhone application designed by Gaston [15] to help users find nearby restaurants based on their opening hours. The app utilizes Factual's Places API as its primary source of restaurant data, providing access to business hours directly within the initial query results. To complement this, Google's Places API is employed to obtain restaurant photos, as Factual's API does not currently provide access to this information. Harmonizing data between the two APIs involves matching restaurant records obtained from Factual with records in Google's Places database using a combination of restaurant names, street addresses, latitude and longitude coordinates. Despite some challenges with inconsistent data formatting and variations in restaurant names between the two databases, the app successfully harmonizes the data to provide users with comprehensive restaurant information, including both business hours and photos. Additionally, Gaston delves into menu information from various providers, including OpenMenu's API, which holds promise due to its generous usage limits, provision of restaurant menus, and potential integration with Google Places for seamless harmonization. Although OpenMenu is ideal for accessing restaurant menus, a significant limitation is it requires a formal company website, restricting its use to businesses only [15]. Despite this restriction requiring a formal company website, it doesn't affect Talk for Me since it is company-supported. Hence, OpenMenu can still be considered as a potential API for the project and could possibly be integrated with Google Places as achieved in "What's Open".

3.2.2. Relevance to project and Future Considerations:

These systems and applications show the reliance on the API for its location-based attributes. Therefore, selecting a robust API that seamlessly integrates with the system is crucial. When evaluating APIs, it's essential to assess their functionalities and capabilities as outlined in their documentation as well as their compatibility with the project. What will need to be further investigated includes the security of the user data and the interference of third parties. This is important to mitigate potential data breaches of sensitive user information when the user uses the application. Look into the possibility of merging APIs for the application.

3.3. Large Language Models

Large Language Models (LLMs) are a subset of Artificial Intelligence technology, and are fundamental to the project, as they are the way in which the application can make predictions on what sentence to respond with. Language Modelling is the process of modelling the likelihood of word sequences, to make predictions on following or missing phrases [16]. As a result, a LLM is a Language Model that has been pre-trained, and scaled in model size to allow for further improved performance [16] specifically in the area of Natural Language Processing (NLP). NLP being the challenge associated with understanding increasingly larger amounts of data/information, in a restricted amount of time [17].

LLM's can be considered as "Cloud-based" or "Locally ran" depending on how the user is able to interact with them. There are significant benefits and trade-offs in selecting one or the other and has been identified as an important consideration for the project. It has also been decided that that the ethical concerns with using AI and LLM technology must be considered, as we are dealing with potentially sensitive information in the pursuit of AI-Mediated Communication (AI-MC) [18].

3.3.1. Cloud-Based LLMs

There are many benefits for using a LLM that is hosted via the cloud. This includes but is not limited to avoiding the memory and processing requirements for a device capable of running the LLM [16], as well as having easier access to newer, and potentially better performing versions of the LLM. With most popular LLM's having some form of API available for use, take for instance ChatGPT, which using the cloud API allows the user to progress from version 3.5 to 4 without having to download any new technology [19]. It also allows for all users to share the same LLM, allowing for the LLM to further train upon the provided input data, to hopefully provide more accurate results.

Two of the leading options for Cloud-Based LLM options are ChatGPT by Open AI and LLaMa 2 by Meta. With both possessing their similarities, there are some key differences to consider. LLaMa 2 is an open source LLM, whereas ChatGPT is based on proprietary software [20], a fact that is especially important when considering AI for communication, as an open-source solution usually leads to a solution with less ethical concern, as greater scrutiny can be placed upon it. Although, with ChatGPT 4 currently outperforming LLaMA 2 on multiple different performance metrics, such as common-sense reasoning, and better scores when it comes to avoiding hallucinating of results (an issue common to LLMs, where incorrect information is predicted)[21].

Thus, in consideration of the ever-increasing number of options for LLMs that are cloud based, further testing will likely be required, to prioritise some of the more unique requirements of the project, being a fast, accurate, and ethical solution that encourages natural communication.

3.3.2. Locally ran LLMs

LLMs can be ran locally, although this is not a typically chosen solution. This is due to the size of LLMs, and the processing power often required [16]. Although in the specific use case of this project, a local option is still of interest since it would allow disconnected users to still access new phrase. With LLaMa having its own mobile compatible versions, this would allow for the user to access a well-developed model, without the need for an internet connection. There are sacrifices made in choosing solutions that work locally though. Notably, performance is a major trade-off with locally ran solutions. With usually worse performing solutions being required to meet the performance specifications of mobile devices, the user will likely run into issues regarding the quality of their responses.

Thus, with benefits existing for locally implemented solutions, these likely do not outweigh the performance trade-off just yet, and thus continuing to use a Cloud-Based solution is preferable until a solution that performs at a comparable level to Cloud-Based solutions is presented.

3.3.3. Ethical Considerations of AI/LLM Technology

When considering the use of AI and LLM technology in the mediation of communication, it is necessary to consider some of the ethical consequences, and risks, associated. These can include, the bias and fairness of information, transparency in the use of AI-MC, as well as misrepresentation and manipulation [18], and all pose a risk to the integrity and privacy of the conversation [22]. There is also the environmental impact of running the sort of hardware needed for this type of Artificial Intelligence, that has to be considered.

Bias in LLMs can occur in many ways, such as issues in the training data, issues in the defining of the goal of the LLM, etc [23]. This can lead to sometimes unintentional problems where not only can the product sometimes perform sub optimally, but it can also provide incorrect information to the user as a result. In the case of AI-MC, this is of particular concern, as bias in language can take sometimes even change the meaning of some sentences, by undermining certain styles of communication [18].

There are multiple ways to consider the ethical problem of transparency. There is that of the transparency of using AI-MC, and how AI tools are consistently attempting to hide the fact they are tools. A key example of this was in Google's Duplex system, which when first shown in examples did not identify itself as AI, for which there was critical outcry, with Google later having to backtrack and confirm that going forward it will begin to identify itself, as to avoid any problems with transparency [24]. With a further desire for transparency coming from the end-user's desire to be able to further scrutinise the information provided by AI-MC, with transparency required to know when to do so [18].

Misrepresentation and Manipulation are the concern of intentionally false information being provided, to induce "false beliefs" [18] with there being a monetary benefit for some companies to manipulate users [22], AI has also been used to spread fake news [25] . This links with bias, where it is in the best interest of the reputation of the creator, as well as for the user to use an unbiased, non-manipulated LLM, such as to maintain fair conversation.

The environmental impact of running the hardware required for LLM and Natural Language Processing (NLP) technology cannot be understated. In Table 3-1 it was found that the training of one model can lead to emissions comparable to 5 times the emissions of the lifetime of a car. As a result, it means that the choice of when to train, and the type of energy used in this type of training is significant, so as not to have a negative impact on the environment.

Table 3-1 Estimated CO_2 emissions from training common NLP models, compared to familiar consumption. [26]

Consumption	CO_2e (lbs)
Air travel, 1 passenger, NY↔SF	1984
Human life, avg, 1 year	11,023
American life, avg, 1 year	36,156
Car, avg incl. fuel, 1 lifetime	126,000
Training one model (GPU)	
NLP pipeline (parsing, SRL)	39
w/ tuning & experimentation	78,468
Transformer (big)	192
w/ neural architecture search	626,155

Ethically the goal is to maintain an unbiased, transparent, and non-manipulated solution for the user, such that there is no concern in false information, allowing for the user to build as close to an authentic voice as possible. As well as a solution that is not have a directly negative impact on the environment.

Chapter 4: Approach & Methods

4.1. User Interface

4.1.1. Steps Required to Achieve Project Aims for Location-based Systems

The approach to improving the user interface of an IOS app involves a lengthy and systematic approach. The following are the steps undertaken.

Table 4-1 - Steps for UI Approach

Approach	Details
Design research and analysis	Conduct adequate research on contemporary UI designs, gathering insights from academic papers, industry reports, and conduct case studies of IOS market apps. Analyse how different solutions increase accessibility for impaired users, ascetics or improve the kinetic and cognitive load. To limit the scope of the task, any addition that enhances the user experience shall be considered as possible additions.
Learn Swift Language and iOS Environment	Since the application is based on swift, it is important to familiarise and gain aptitude with the development environment. This involves undertaking online tutorials and documentations. IOS design guidelines shall also be practiced.

Initial designs and mock-ups	Before implementing the solution into the application and code, it is important to develop conceptual mock-up designs. These can be designed using tool such as paint, photoshop or Adobe XD. This will allow for the UI design to be visualised to evaluate its utility against the user needs and best practices detailed in the research phase. It will also allow for the layout and aesthetics to be planned before coding. Before these plans are finalised and implemented, feedback from the stokeholds and team members shall be squired.
Implementation	The mock-up designs shall be translated into UI components in swift. The implementation shall be modularised, where small components such as buttons and sliders are coded separately. This stage is iterative, involving back-and-forth adjustments based of testing and feedback.
Testing, debugging and refining	After implementation, many bugs and errors are likely to arise. Hence, thorough testing shall be undertaken, ensuring the solution is consistent across all devices.

4.1.2. Discussion of foreseeable Issues regarding UI

Designing and integrating new UI elements into an application poses many foreseeable challenges needing careful consideration.

Table 4-2 - Considerations for UI Improvements

User adaption	The addition of new UI elements and a shift in the aesthetics of the application may require users to adapt to the new UI design, especially if it significantly differs from the previous version. To mitigate this, additions can be rolled out gradually and releasing tutorials. This would allow users to ease into the new UI.
Device Compatibility	When developing for IOS, it is important to consider range all the different devices. These devices may have different screen sizes, resolution, or software. The additions need to be robust and scale to all the platforms. To mitigate this, testing needs to be conducted on multiple devices.
Accessibility Compliance	When moving towards a more accessible app, there’s a chance not every user need, and guidelines are considered. Hence, it is important to garner feedback and

	continuously engage with users to improve the design process and try achieving the most accessible UI possible.
Performance decrease	With the addition of new UI elements, there is a high possibility of decreased performance. This may be due to poor coding or lack of storage. To mitigate this, the implementation shall be optimised during the testing phase, ensuring redundant code is minimised.
Introduction of bugs	The addition of new UI elements may tamper with existing code and introduce unknown bugs. During the testing phase, it is important to test the application as a whole rather than just the new additions.

4.1.3. UI Outcomes

Based on the approach, the following outcomes are projected:

Table 4-3 - Expected Outcomes of UI Approach

Expected outcome	Achievement Strategy
Enhanced usability	The main outcome is an improved user interface that more usable. A new user shall be able to pick up the application intuitively and find it easy to navigate. The kinetic and cognitive loads are improved upon. The adjustable functions of the applications have been renovated into visual icons easy to access. This shall lead to reduced time to complete common tasks, and positive user feedback highlighting ease of use.
Increased accessibility	The UI shall be more catered to users with impairments. Principles such as the WCAG is incorporated. Features such as large fonts, high contrast, and alternative texts are provided. The application shall aim be some what compliant with accessibility standards and receive positive feedback from users with impairments.
Robust Performance across devices	The UI improvements are scalable across all different IOS phone types, ensuring a smooth and responsive experience regardless of screen size or resolution. It is expected that

	minimal bugs are reported related to the UI issues.
Visual design and appeal	The conceptual designs of the UI translate successful, and the visual design of the UI is aesthetically pleasing. Incorporate feedback from stakeholders and research modern apps to refine the visual elements

4.2. Location-based Systems

Successfully achieving the project goals involves executing four key steps: conducting thorough API research and analysis, honing Swift proficiency, strategizing integration and implementation, and rigorously testing and debugging the system.

4.2.1. Steps Required to Achieve Project Aims for Location-based Systems

1. API research and analysis

Conduct comprehensive research to identify the suitable location-based API for the project. Analyze the functionalities, documentation, prices, and limitation of each API. For this project, the focus will be on studying APIs that enable the retrieval of essential photos for user display and those that provide access to restaurant menu items. To limit the scope, the API data will be focused on fast food restaurants. However, this can be expanded later for scalability purposes.

2. Swift Language Proficiency development

Enhance proficiency in Swift through working on smaller priorities in the project. By doing this step it ensures a solid foundation to be able to effectively implement the chosen API into the IOS application.

3. Integration and implementation planning

Develop a detailed integration plan outlining the steps required to seamlessly incorporate the location-based API into the iPhone application. This includes defining endpoints, handling authentication, managing data retrieval and manipulation, and ensuring compatibility with existing app features.

4. Testing and debugging

Conduct thorough testing of the integrated API functionality to identify and resolve any potential issues or bugs. Utilize testing frameworks and tools to ensure the reliability, accuracy, and performance of the location-based features within the application.

4.2.2. Discussion of Foreseeable Issues for Location-based Systems

Integrating a location-based API into an iPhone application brings forth several foreseeable challenges that demand careful consideration.

Table 4-4 - Considerations/Challenges for Location-Based Systems

Issues	**Details**
Privacy concerns	Privacy concerns arise primarily due to the use of APIs that access the user's location. To mitigate this, the application must adhere to strict security protocols and provide transparent user agreements, so users are fully informed about the data being accessed.
Battery drain	Additionally, the constant GPS access required by the API raises concerns about battery drain. This highlights the need for optimization measures to lower this impact and ensure acceptable battery drainage for the users.
Data usage	Managing data usage effectively is another challenge, as the need for the internet to access location causes the reliance on mobile data consumption in areas of no internet connectivity. This can lead to increased costs for users and potential usability issues. Although this is the case, it is important to investigate how the API manages data usage and consider the implementation of an offline location finder as potential solutions.
API limitations	Anticipating and addressing potential API limitations, such as usage limitations or feature restrictions, is essential to prevent compatibility issues and ensure seamless integration with the application.
Updates	Staying informed about updates to both the API and iOS platform is crucial in addressing emerging issues or vulnerabilities, ensuring the long-term functionality and reliability of the integrated features. By acknowledging and consistently updating the application and its features ensures a robust and user-friendly experience for all.

4.2.3. Location-based Systems Outcomes

From following this approach, the following outcomes are a location-aware menu suggestions and a location-based word list.

Table 4-5 - Expected Outcomes & Achievement Strategy

Expected Outcomes	**Achievement Strategy**
Location-aware menu suggestions	When the user is in a restaurant, café, or similar establishment, display the menu items. This will involve modularizing the app to include a contextual data version. Implement location services within the app, so that if the user is near a restaurant, the app will automatically display its menu, allowing users to add items to their suggestions list for that restaurant.
Location-based word list	Incorporate images and words commonly used by other users at the current location into the images/words list of a user visiting that space. This could streamline the process, eliminating the need for landmarks to be individually added to each user's list. Additionally, consider implementing a "save" feature to allow users to add these presented words to their ongoing word bank. To accomplish this, a module needs to be developed to display suggested words, which will vary depending on the location. These suggested words will be obtained from the most searched words based on the restaurant, utilizing data acquired from an API.

4.3. Large Language Models

4.3.1. Steps Required to Achieve Project Aims for LLMs

To enhance our application's performance, we will systematically test various Large Language Models (LLMs) and utilize prompt engineering. By evaluating multiple LLMs, we aim to identify the models that best meet our specific needs. Concurrently, prompt engineering will be employed to fine-tune input prompts, optimizing model outputs for accuracy and relevance. This combined strategy of model selection and prompt refinement is designed to maximize the capabilities of LLMs, ensuring superior functionality and user experience for our application.

This is further summarized in Table 4-6.

Table 4-6 – Approach/Method of LLM Outcomes

Approach	**Method**
Selection of Most appropriate LLM Model	To begin, the group will collect all available, and suitable, LLM models to test. This is ever changing, and thus is hard to list currently. To then test these, the group will setup a testing harness, in which speeds will be measured for the time taken to receive a response. This will also be considering the response, if it is formatted correctly, and if it makes sense given the provided keywords/symbols. The best performing of this will be selected to use going forward.
Testing of Different Prompts	Prompt engineering will involve writing of multiple varying prompts, with different ways of asking for the same thing essentially. The prompt that gets the closest to the desired output (given a large list of test cases) will be considered to use going forward.
Testing of Prompts requesting Multiple Outputs	The idea of receiving multiple outputs is ideal, as it will further limit the amount of typing required by the user, a major requirement of the application, and thus functionality for receiving multiple outputs from the LLM will require some prompt engineering. It will also then require integration into the main application
Testing of Prompts, given previous accepted outputs.	Using the applications already existing database functionality (to store saved keywords/symbols), the group intends on saving the selected (or edited phrase) to pass the user as either a direct suggestion, or via prompt engineering, to the LLM to better educate it on what phrases/sentences to provide. This is seen as another thing to improve performance, as the LLM will be able to learn more about the user, and their preferences.

4.3.2. Discussion of Foreseeable Issues for Large Language Models

With the use of externally written models, there are some foreseeable issues that may occur, and that can be planned to reduce and issues for, seen in

Table 4-7 - Foreseeable Issues regarding LLMs, with Details & Mitigation

Issues	**Details & Mitigation**
Old models, or code written to communicate with no longer supported models becoming deprecated.	With the use of models that come from several different companies, with different coding standards, etc. there is the real likelihood of technology, or integration with technology, becoming deprecated at some point. To combat this, all added code will be done so keeping extensibility in mind. With the idea that it should be able to fit all future models, this should be somewhat achievable for the intended outcomes.
Ethical/Security risks due to externally written models	There are legitimate concerns to do ethical standpoint of using AI for communication, as mentioned in 3.3.3. As a result, effort will be placed to not influence, or add bias to the output of the LLM, when prompt engineering, as has been done by the engineers writing many of the LLMs on the market. The group will also avoid passing any identifying information, without given permission, to the LLM, when regarding the features added.

Chapter 5: Progress

5.1. User Interface

The progress of the UI at the current stage of the project consists of literature research, case studies and swift environment exploration.

5.1.1. Research of Similar UI Examples

After literature review was undertaken, various popular IOS applications, including Apple Music, and Specify, were analysed. A screen capture of the application UI was taken and its features were dissected. These analysed features will form the basis of the conceptual design undertaken in the future.

As detailed by the notes in Figure 5-1, the Apple Music has considered reducing cognitive and kinetic load. The app incorporates white space to reduce clutter. The icons are large, consistent, and easy to view. Similar elements are teamed together, reducing the mental effort required to switch between icons. Hence, proving to be an effective UI design.

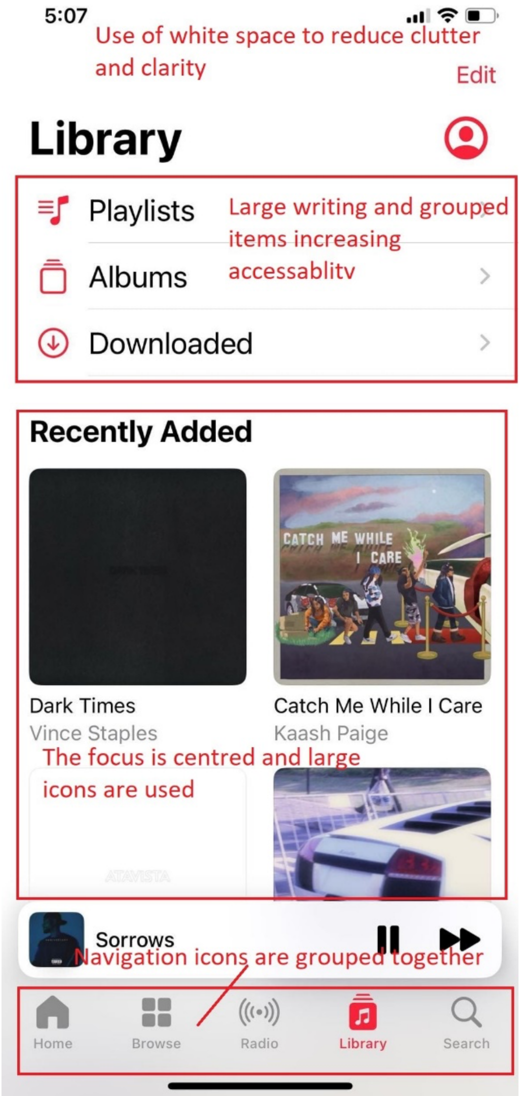


Figure 5-1 - Breakdown of Features of Apple Music

Specify is a TTS application, specify, and it has an effective UI, as seen in Figure 5-2. The labels dissect the various features of the application. Although the application has many features, it reduces the clutter by using white space. The ions are quite large and teamed by caatgory, allowing users to easily identify and navigate the different features. Hence, reducing the overall kinetic and cognitive load. These merits of this app can be used in the UI design of the project.

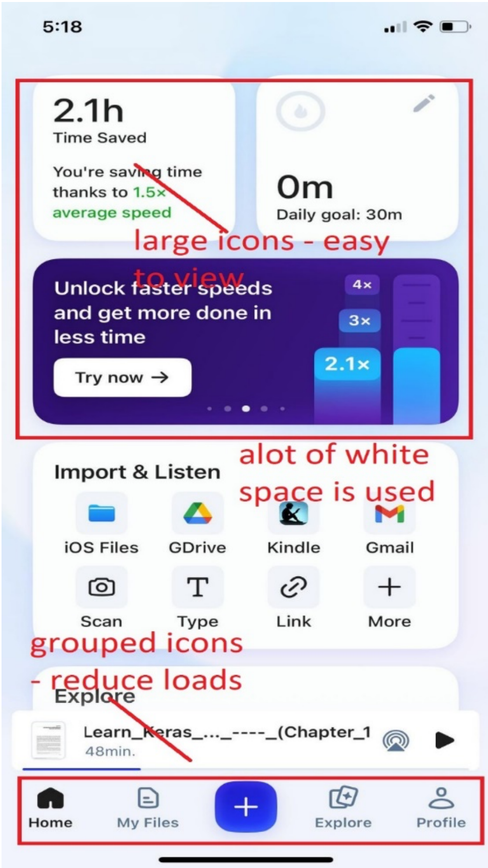


Figure 5-2 - Breakdown of Specify Application

5.1.2. Minor Bug in Existing Codebase

A bug in the application was also identified by the stakeholders in the early stages of the project, seen in Figure 5-3. It was indicated on some versions of IOS, and certain iPhone models, that there were problems with some of the keywords/symbols overlapping when they should not.

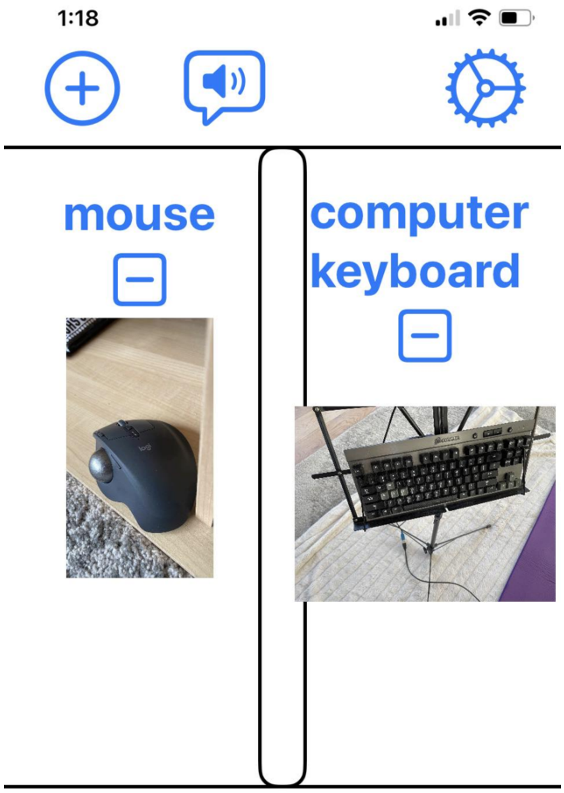


Figure 5-3 - Border issue present on limited versions of IOS

This was corrected by simulating the compromised screen size and changing the dimensions of the icon. With the resulting output appearing cohesive with the rest of the design of the application.

5.1.3. Modifying "±" feature

After setting up swift and successfully building, it was time to work on the priority. In this case the priority issue was to modify "±" feature under the pictures to show to change into a check mark instead. The original "±" feature is shown in Figure 5-4.

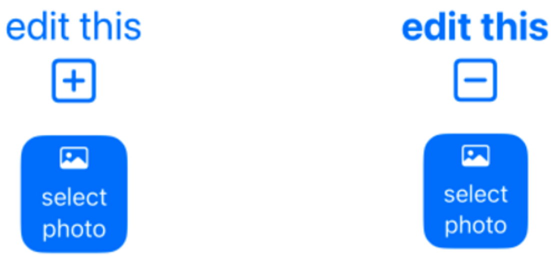


Figure 5-4 - Original +/- box feature

As can be seen in Figure 5-5, the plus and minus symbols are just pictures which change on click (indicated by the red "-" lines). Hence, to change it into a checkbox feature, a picture of an unfilled box and a checkmark box were needed to replace the ±. To find the appropriate name for these symbols, the SF symbol library was downloaded. In this library, the name of these symbols is "square" and "checkmark.square". These symbol names were then implemented and the "±" symbols were changed to checkboxes as seen in Figure 5-6.

```
        if item.isSelectedForChatGPT != nil && item.isSelectedForChatGPT == true {
-           Image(systemName: "minus.square").font(Font.system(.largeTitle))
+           Image(systemName: "checkmark.square").font(Font.system(.largeTitle))
        } else {
-           Image(systemName: "plus.square").font(Font.system(.largeTitle))
+           Image(systemName: "square").font(Font.system(.largeTitle))
```

Figure 5-5 - Code for +/- to Checkbox Changes

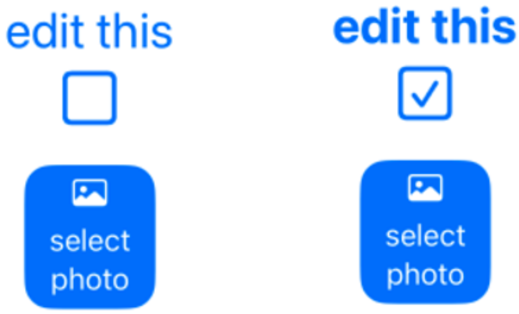


Figure 5-6 - Checkbox functionality implemented

5.2. Location-based Systems

The progress done so far are API research and analysis, swift proficiency development, and initial planning of integration and implementation.

5.2.1. API research and analysis:

Table 5-1 - Comparison of Different Location APIs

API	Functionality	Cost (AUD)	Data Access
Open Menu	menu, menu items, trends, analysis, heat maps, gap analysis, location, deals, restaurant, ingredients	Monthly payment $30.17 / month	100 countries, 550K+ menus, 25+ million menu items.
Google Places	find place, nearby search, text search, place details (no menu), place photo, autocomplete	Per request $0.0043 per API request ($4.30 per 1000)	Google dataset.
TripAdvisor	Location details, read reviews, rating, price level	Requests / month 50 / month: $0 per month 5500 / month: $12.06 per month 70000 / month: $75.47 per month Unlimited: $526.79 / month	TripAdvisor dataset.
Yelp Fusion API	Business details, transactions, reviews, events, autocomplete	Requests / month 10 / day: $0 / month 10000 / month: $43.77 / month 30000 / month $89.06 / month 300000 / month: $451.32 / month	6,990,280 reviews, 150,346 businesses, 200,100 pictures, 908,915 tips by 1,987,897 users, Over 1.2 million business attributes.
Spoonacular	Nutrition data, price, cooking tips, health information, substitutions, conversion,	Requests / day 50 / day:	5000+ recipe, 600+ products,

	mapping to products, nutrition analysis, cost breakdown, recipes, ingredient analysis, descriptions	$0 / month 500 / day: $43.77 per month 2500 / day: $149.43 / month 30000 / day: $1507.92 / month	115k+ menu items.

5.2.2. Location and Time Module

To begin in developing functionality that includes time and location information, it was important to add a module to the application that can get the user's location and time data (to later be processed for functionality reasons). As a result, a modular and extensible solution is required to allow for any changes that may be required in the future. In Figure 5-7 the code for this module can be seen, with two separate manager classes, the TimeManager which returns the current Date and time using the systems information. And the LocationManager, which after receiving the user's permission to use Location data, an example of which is provided in Figure 5-8, is able to provide the application with constantly updating location information. With the goal of developing a dynamic solution that can constantly update and provide the best solution, it was important for the LocationManager to provide continually updating location information, and as a result was considered when implementing it. In terms of coding structure, it was kept to the standards set by the stakeholder, with all formatting and testing completed to ensure it fits into the system without conflict.

The plan is for this information to then be applied and passed along to APIs to access menu information, or to make suggestions based on the user's previous behavior at specific times/places.

```swift
import Foundation
import CoreLocation

public class TimeManager {
    public static func getCurrentTime() -> Date {
        return Date()
    }
}

public class LocationManager: NSObject, CLLocationManagerDelegate {
    private var locationManager = CLLocationManager()
    public weak var delegate: LocationManagerDelegate?
    override init() {
        super.init()
        locationManager.delegate = self
        locationManager.requestWhenInUseAuthorization()
        locationManager.desiredAccuracy = kCLLocationAccuracyBest
        locationManager.startUpdatingLocation()
    }

    public func locationManager(_ manager: CLLocationManager, didUpdateLocations locations: [CLLocation]) {
        guard let location = locations.last else {return}
        delegate?.didUpdateLocation(location)
    }

    public func locationManager(_ manager: CLLocationManager, didFailWithError error: Error) {
        locationManager.stopUpdatingLocation()
        delegate?.didFailWithError(error)
    }
}

public protocol LocationManagerDelegate: AnyObject {
    func didUpdateLocation(_ location: CLLocation)
    func didFailWithError(_ error: Error)
}
```

Figure 5-7 - Code for Location Time Module

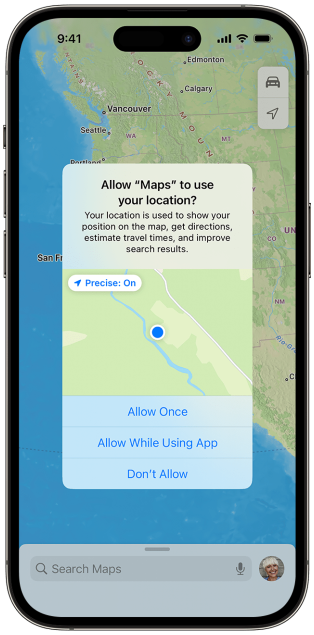


Figure 5-8 - Example of acquiring permissions for Location Data [27]

5.2.3. Planning of integration and implementation

The current plan for integrating the API into the application involves creating a function required to call the API database. This function will call the database when the location-time-module detects the user in a specific area. For instance, if permitted by the user, the location-time-module could recognize when the user is at a fast-food restaurant such as KFC. In response, it could display the KFC menu along with images of the items, facilitating easier selection for the user. A visual idea of this feature can be seen in Figure 5-9.

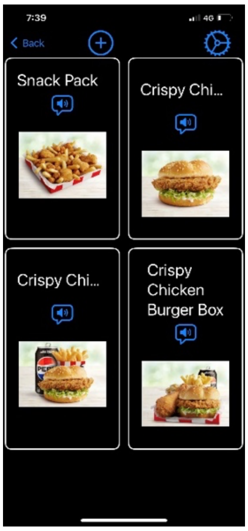


Figure 5-9 - Visual of user arriving at KFC

While this feature will initially be limited to fast food chains, the goal for the future is to expand its coverage to include a broader range of establishments such as fine dining restaurants or cafes. Another possibility is to allow users to create their own custom menu of where they are at, and when the location-time-module locates them at that establishment it will show their custom suggestions.

5.3. Large Language Models

As of the time of this report, there has not been much done in the way of progress for Large Language Models. With most of the focus put on User Interface, and Location-based Systems related tasks, as they were predominately pre-requisite tasks towards further LLM tasks, and provided a The team intends on working on the LLM tasks in conjunction with the tasks related to 5.1 and 5.2 as the project goes on, but as there is no task that requires these for a pre-requisite, it is yet to be done.

Most of the progress for LLMs can be summarised as research. As much of the team was not too familiar with how to interact with LLMs outside of the available “chat bots” online, there was a learning curve for understanding the pre-existing code, and how the application currently operates. This understanding will be important for all future tasks, as they will revolve predominately around writing testing harnesses to run the existing code with different models/prompts. The team has made plans on how we intend to implement these testing harnesses, with the intentions of measuring performance of different LLMs by their speed in response, as well as accuracy in intended response.

Some modifications will also be made to the current functionality of the program, in how it handles LLM requests, that will be discussed further in section 6.3.2 of this report.

Chapter 6: Completion Plan

The project completion plan will cover Objective 4, the Mid-Year break, and Objective 4.5 as outlined in Appendix A: Gantt chart. The completion plan for each objective outcome is detailed as follows:

6.1. User Interface

1. Consult team and stake holders and finalise visual design for the UI design. This will be an iterative design.

2. Implement improvements to UI which increase the aesthetics, accessibility and user experience of the application. This addition will be seen when opening the application and navigating throughout.

3. Test and scale the additions to various iOS devices with different screen sizes and resolutions, increasing the robustness of the system. Use tools like Xcode’s simulator to cover a wide range of device types.

6.2. Location-based systems.

The completion plan for location-based systems involves progressing on the following outcomes:

6.2.1. Location-aware menus suggestions

1. Create function to access the API database.

2. Design a logic system to activate a function whenever the location-time-module detects the user within a designated area. Once triggered, the function retrieves menu names and corresponding photos by invoking a separate function to access the database.

3. Display a user-friendly interface on the screen showcasing menu names alongside photos of each item. Users can simply click on the desired item, prompting the device to verbally announce their selection for confirmation.

6.2.2. Location-based word list

1. Create function to call most used words at a specific location.

2. Use API call to determine most used words at a specific location determined by the location-time-module.

3. Incorporate system into contextual text created by LLM.

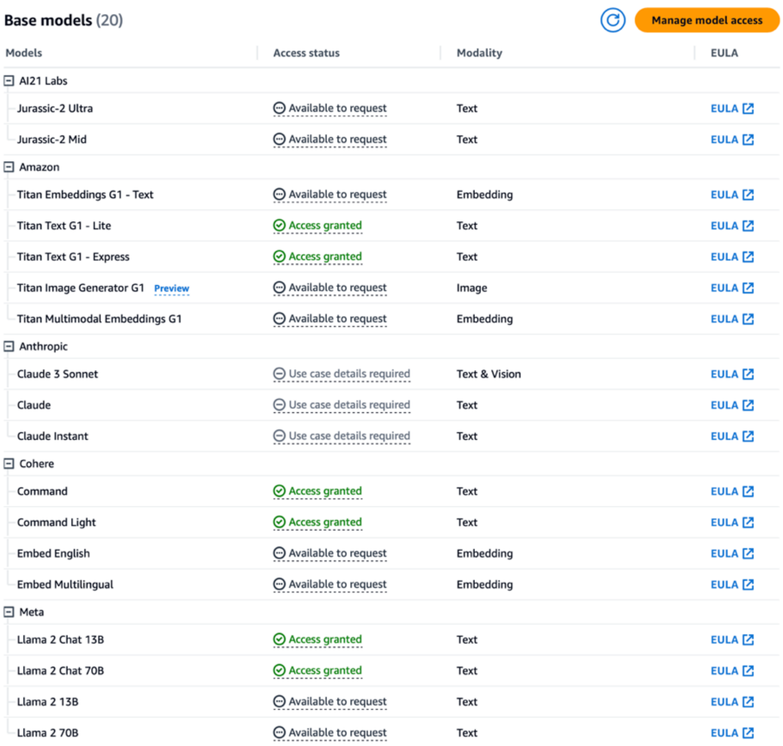
6.3. Large Language Models

Despite the limited progress made thus far in the development of our LLM outcomes, the foundational framework is now firmly established. This initial phase has laid the groundwork for the upcoming stages of advancement. Moving forward, our primary focus will be on the two key areas of testing LLM options and modifying/testing prompts to be used by the LLM.

6.3.1. Testing LLM Options

To achieve this the focus will be placed on the development of testing harnesses, which are able to test multiple LLMs on the same hardware/framework that is already in place, to decide upon whether there are better performing options available for this specific application.

To do this, it will require first compiling options relating to LLMs to test. The team was provided such a list at the beginning of the project by the stakeholders, see Figure 6-1, although as the year has progressed many of these solutions have since had new versions released, and as a result this list may be outdated. There is also the interest of Locally ran LLMs, as previously mentioned in section 3.3.2, which are not mentioned in Figure 6-1 which will be further investigated. With Bloomberg suggesting it is likely we will see a native LLM option provided in the newest version of IOS [28], likely to be announced in June of this year, this will also be of interest to the team to test, for previously mentioned reasons.


Base models (20)

Manage model access

Models	Access status	Modality	EULA
AI21 Labs			
Jurassic-2 Ultra	Available to request	Text	EULA
Jurassic-2 Mid	Available to request	Text	EULA
Amazon			
Titan Embeddings G1 - Text	Available to request	Embedding	EULA
Titan Text G1 - Lite	Access granted	Text	EULA
Titan Text G1 - Express	Access granted	Text	EULA
Titan Image Generator G1 Preview	Available to request	Image	EULA
Titan Multimodal Embeddings G1	Available to request	Embedding	EULA
Anthropic			
Claude 3 Sonnet	Use case details required	Text & Vision	EULA
Claude	Use case details required	Text	EULA
Claude Instant	Use case details required	Text	EULA
Cohere			
Command	Access granted	Text	EULA
Command Light	Access granted	Text	EULA
Embed English	Available to request	Embedding	EULA
Embed Multilingual	Available to request	Embedding	EULA
Meta			
Llama 2 Chat 13B	Access granted	Text	EULA
Llama 2 Chat 70B	Access granted	Text	EULA
Llama 2 13B	Available to request	Text	EULA
Llama 2 70B	Available to request	Text	EULA

Figure 6-1 - List of LLMs to Test provided by Stakeholders

In terms of the implementation of the testing harness, the team intends on following the currently in place structures set by the Stakeholders, writing it in python, and leaving it as modular and extensible (to account for any new and improved models that are released once testing may be complete) as possible.

6.3.2. Prompt Engineering

A smaller part of the outcomes for increasing productivity with LLMs is to do what is known as prompt engineering. The only factor impacting the output of a model, that is not related directly to the model itself (and its training/processes) is the prompt provided to it. Prompt Engineering is thus the process of testing minor, or sometimes major, changes to a prompt to get better performance from a model.

Like 6.3.1, a testing harness will be constructed to test several prompts for their performance, this performance will be measured based on how appropriate the response is. This is difficult to measure in an objective way, as given a few keywords/symbols, there are numerous things that could be hoped to be implied. To combat this, the team hopes to modify the current implementation to provide more than one output. To give the user more options, but also potentially passing to the LLM previously selected options (using the same keyword/symbols) to better prompt the LLM on what the intended sentence/ phrase is.

Chapter 7: Summary

Significant progress has been made by the team in terms of preparation research, and setup for upcoming major tasks. The workload and outcomes have been divided into the three main objectives: UI enhancements, development of location-based systems, and optimization of LLM performance.

7.1. User Interface

Upon conducting literature review, many strategies for improving application UI were defined. The application can be modified to increase its accessibility and usability. These findings will aid in developing initial designs of UI. The team has also increased in swift proficiency through fixing the over lapping issue and changing the "±" feature on the application to be checkboxes.

7.2. Location-based systems

Upon reviewing past work, it becomes evident that harmonizing various APIs presents the most optimal solution for achieving the desired outcomes of the location-based system. These APIs will grant access to databases housing essential menu items and photos required for the application. The team has also created a location and time module which can be used to locate the user's specific position. This module is a necessity to achieve future outcomes. The planning and integration of this module and APIs have also been conducted. The main plan being to create functions to call the APIS.

7.3. Large Language Models

For LLMs, the team has progressed in term of their understanding beyond how to interact with LLMs beyond online chatbots and how the current application operates. This knowledge is crucial for future tasks, which will primarily involve creating testing harnesses to run the existing code with various models and prompts. The team has also planned to implement these testing harnesses to measure the performance of different LLMs based on response speed and accuracy.

7.4. Objective Outcomes

With the listed progress, the outcomes for each objective are:

7.4.1. User Interface Objective Outcomes

The main outcome is a more intuitive and user-friendly interface, allowing new users to easily navigate the application. Improved kinetic and cognitive loads, along with renovated visual icons for adjustable functions, reduce the time to complete common tasks and result in positive user feedback. The UI is designed to accommodate users with impairments by incorporating WCAG principles, such as large fonts, high contrast, and alternative text, aiming for compliance with accessibility standards and positive feedback from these users. Additionally, UI improvements are scalable across all iOS phone types, ensuring a smooth and responsive experience regardless of screen size or resolution, with minimal UI-related bugs expected. The conceptual designs are successfully translated into an aesthetically pleasing UI, refined through feedback from stakeholders and research on modern apps.

7.4.2. Location-based system Objective Outcomes

When the user is in a restaurant, café, or similar establishment, the app will automatically display the menu items. This requires modularizing the app to incorporate a contextual data version. By implementing location services, the app can detect when the user is near a restaurant and promptly display its menu, allowing users to easily add items to their suggestions list for that location.

Incorporate images and words commonly used by other users at the current location into the user's image/word list. This will streamline the process, eliminating the need for each user to individually add landmarks to their list. Additionally, implement a "save" feature that allows users to add these suggested words to their ongoing word bank. To achieve this, develop a module that displays suggested words, which will vary depending on the location. These suggested words will be sourced from the most searched terms associated with the restaurant, using data acquired from an API.

7.4.3. Large Language Model Objective Outcomes

The outcomes we intend achieving of the LLMs surround the improved performance of the group. With the goal of having a real-time artificial voice, performance and speed of responses is paramount, and should only be dependent on the user's ability to use the application (assisted by the previous sections) and not on waiting for a LLMs results.

To do this, the group will test various options regarding LLMs (including those that can run locally on a IOS device). As well as, testing options regarding different prompts for the LLM, as well as, requesting multiple prompts, and constructing prompts using previously accepted sentences/phrases.

References

,[1] "Augmentative and Alternative Communication (AAC) Applications," disabilityinformation. Accessed: May 26, 2024. [Online]. Available: https://accessibletelecoms.org.au/telecom-home/device-search/apps/augmentative-and-alternative-communication-applications.html

[2] D. A. Norman, "The Design of Everyday Things,", Cambridge, MA: *The MIT Press*, 2013.

[3] F. Alqahtani and R. Orji, "Insights from user reviews to improve mental health apps," *Health Informatics J.*, vol. 26, no. 3, pp. 2042–2066, Sep. 2020, doi: 10.1177/1460458219896492.

[4] J. Nielsen, *Usability engineering*. Boston: Academic Press, 1993.

[5] M. Gomez-Hernandez, X. Ferre, C. Moral, and E. Villalba-Mora, "Design Guidelines of Mobile Apps for Older Adults: Systematic Review and Thematic Analysis," *JMIR MHealth UHealth*, vol. 11, p. e43186, Sep. 2023, doi: 10.2196/43186.

[6] A. Kodzhoshalieva, "Reducing Cognitive Load for a Data-Intensive Web Application User." Accessed: May 26, 2024. [Online]. Available: https://www.doria.fi/handle/10024/177793

[7] J. Thatcher *et al.*, "Constructing Accessible Web Sites," 2003, p. 415.

[8] H. Petrie and O. Kheir, "The relationship between accessibility and usability of websites," Apr. 2007, pp. 397–406. doi: 10.1145/1240624.1240688.

[9] J. O. Wobbrock, S. K. Kane, K. Z. Gajos, S. Harada, and J. Froehlich, "Ability-Based Design: Concept, Principles and Examples," *ACM Trans. Access. Comput.*, vol. 3, no. 3, p. 9:1-9:27, Apr. 2011, doi: 10.1145/1952383.1952384.

[10] M. Jeon, B. Davison, M. Nees, J. Wilson, and B. Walker, "Enhanced Auditory Menu Cues Improve Dual Task Performance and are Preferred with In-vehicle Technologies," Sep. 2009, pp. 91–98. doi: 10.1145/1620509.1620528.

[11] N. Saiph Savage, M. Baranski, N. Elva Chavez, and T. Höllerer, "I'm feeling LoCo: A Location Based Context Aware Recommendation System," in *Advances in Location-Based Services: 8th International Symposium on Location-Based Services, Vienna 2011*, G. Gartner and F. Ortag, Eds., Berlin, Heidelberg: Springer, 2012, pp. 37–54. doi: 10.1007/978-3-642-24198-7_3.

[12] Hua, L. and Alonso, R. (2014). *Defense Technical Information Center*. [online] Available at: https://apps.dtic.mil/sti/citations/tr/ADA618623 [Accessed 29 March 2024].

[13] Phromchomcha, S. (2023) *Urecipe Technical Report*. Available at: https://norma.ncirl.ie/6832/1/sutthitaphromchomcha.pdf [Accessed 29 March 2024].

[14] "spoonacular API | spoonacular API | Postman API Network." Accessed: May 26, 2024. [Online]. Available: https://www.postman.com/spoonacular-api/workspace/spoonacular-api/collection/7431899-ef0368a7-643c-4c87-975c-68399d4f0c12

[15] Gaston, B. (2013). *What's Open: a native iPhone app for finding nearby restaurants organized by open hours*. Available at: https://doi.org/10.17615/nd5p-vb85 [Accessed 29 March 2024].

[16] W. X. Zhao *et al.*, "A Survey of Large Language Models." arXiv, Nov. 24, 2023. Accessed: May 12, 2024. [Online]. Available: http://arxiv.org/abs/2303.18223

[17] K. R. Chowdhary, "Natural Language Processing," in *Fundamentals of Artificial Intelligence*, K. R. Chowdhary, Ed., New Delhi: Springer India, 2020, pp. 603–649. doi: 10.1007/978-81-322-3972-7_19.

[18] J. T. Hancock, M. Naaman, and K. Levy, “AI-Mediated Communication: Definition, Research Agenda, and Ethical Considerations,” *J. Comput.-Mediat. Commun.*, vol. 25, no. 1, pp. 89–100, Mar. 2020, doi: 10.1093/jcmc/zmz022.

[19] “A Brief Overview of ChatGPT: The History, Status Quo and Potential Future Development | IEEE Journals & Magazine | IEEE Xplore.” Accessed: Apr. 04, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10113601?casa_token=07Rd6DlsdUcAAAAA:JttYjwPqj9sS-Oo2XWo7ZantzECynpxHzKnvk77u8MJSjxxqA4YgUqNs3SfrQCjcX7tKPvTZbDQ

[20] V. Arnold, “Battle of the bots: Llama 2 vs ChatGPT - Which chatbot dominates?,” neuroflash. Accessed: May 13, 2024. [Online]. Available: https://neuroflash.com/blog/battle-of-the-bots-llama-2-vs-chatgpt-which-chatbot-dominates/

[21] S. Minaee *et al.*, “Large Language Models: A Survey.” arXiv, Feb. 20, 2024. Accessed: May 13, 2024. [Online]. Available: http://arxiv.org/abs/2402.06196

[22] T. Hagendorff, “The Ethics of AI Ethics: An Evaluation of Guidelines,” *Minds Mach.*, vol. 30, no. 1, pp. 99–120, Mar. 2020, doi: 10.1007/s11023-020-09517-8.

[23] D. Roselli, J. Matthews, and N. Talagala, “Managing Bias in AI,” in *Companion Proceedings of The 2019 World Wide Web Conference*, San Francisco USA: ACM, May 2019, pp. 539–544. doi: 10.1145/3308560.3317590.

[24] N. Statt, “Google now says controversial AI voice calling system will identify itself to humans,” The Verge. Accessed: May 12, 2024. [Online]. Available: https://www.theverge.com/2018/5/10/17342414/google-duplex-ai-assistant-voice-calling-identify-itself-update

[25] S. Vosoughi, D. Roy, and S. Aral, “The spread of true and false news online,” *Science*, vol. 359, no. 6380, pp. 1146–1151, Mar. 2018, doi: 10.1126/science.aap9559.

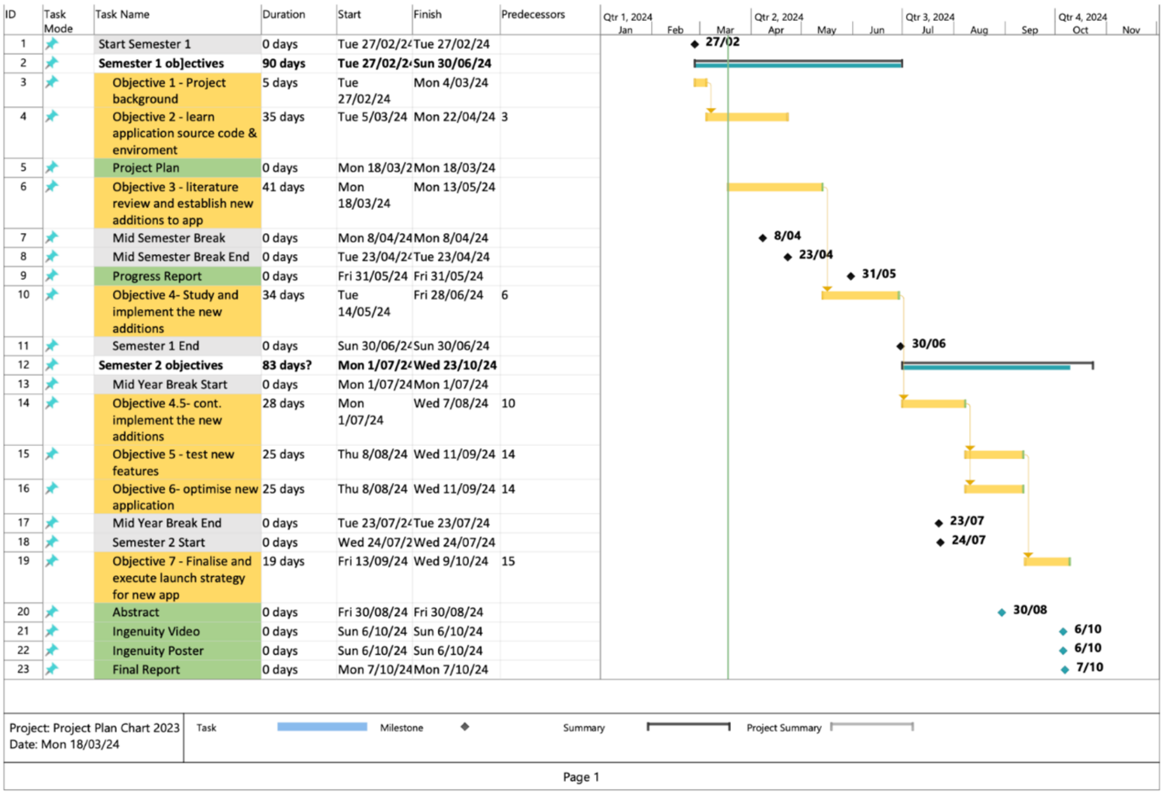
[26] E. Strubell, A. Ganesh, and A. McCallum, “Energy and Policy Considerations for Deep Learning in NLP.” arXiv, Jun. 05, 2019. Accessed: May 27, 2024. [Online]. Available: http://arxiv.org/abs/1906.02243

[27] Apple, *iPhone 14 Pro - Allow maps to access location while using app*. 2024. Accessed: May 27, 2024. [Online]. Available: https://cdsassets.apple.com/live/7WUAS350/images/ios/ios-16-iphone-14-pro-maps-allow-maps-to-access-location-while-using-app.png

[28] “Apple Needs a True Low-End iPhone to Help Revive Growth,” *Bloomberg.com*, Apr. 21, 2024. Accessed: May 27, 2024. [Online]. Available: https://www.bloomberg.com/news/newsletters/2024-04-21/apple-aapl-growth-opportunities-southeast-asia-and-africa-lower-end-iphone-lv9itkna

Appendices

Appendix A: Gantt chart



ID	Task Mode	Task Name	Duration	Start	Finish	Predecessors
1		Start Semester 1	0 days	Tue 27/02/2	Tue 27/02/24	
2		**Semester 1 ob]ectives**	**90 days**	**Tue 27/02/2**	**Sun 30/06/24**	
3		Objective 1 - Project background	5 days	Tue 27/02/24	Mon 4/03/24	
4		Objective 2 - learn application source code & enviroment	35 days	Tue 5/03/24	Mon 22/04/24	3
5		Project Plan	0 days	Mon 18/03/2	Mon 18/03/24	
6		Objective 3 - literature review and establish new additions to app	41 days	Mon 18/03/24	Mon 13/05/24	
7		Mid Semester Break	0 days	Mon 8/04/24	Mon 8/04/24	
8		Mid Semester Break End	0 days	Tue 23/04/2	Tue 23/04/24	
9		Progress Report	0 days	Fri 31/05/24	Fri 31/05/24	
10		Objective 4- Study and implement the new additions	34 days	Tue 14/05/24	Fri 28/06/24	6
11		Semester 1 End	0 days	Sun 30/06/2	Sun 30/06/24	
12		**Semester 2 objectives**	**83 days?**	**Mon 1/07/2**	**Wed 23/10/24**	
13		Mid Year Break Start	0 days	Mon 1/07/24	Mon 1/07/24	
14		Objective 4.5- cont. implement the new additions	28 days	Mon 1/07/24	Wed 7/08/24	10
15		Objective 5 - test new features	25 days	Thu 8/08/24	Wed 11/09/24	14
16		Objective 6- optimise new application	25 days	Thu 8/08/24	Wed 11/09/24	14
17		Mid Year Break End	0 days	Tue 23/07/2	Tue 23/07/24	
18		Semester 2 Start	0 days	Wed 24/07/2	Wed 24/07/24	
19		Objective 7 - Finalise and execute launch strategy for new app	19 days	Fri 13/09/24	Wed 9/10/24	15
20		Abstract	0 days	Fri 30/08/24	Fri 30/08/24	
21		Ingenuity Video	0 days	Sun 6/10/24	Sun 6/10/24	
22		Ingenuity Poster	0 days	Sun 6/10/24	Sun 6/10/24	
23		Final Report	0 days	Mon 7/10/24	Mon 7/10/24	

Contribution Declaration

Group member name	Contribution			
	Approach	Technical output	Writing	Project management
Matthew Fowler	Conceptualisation, Methodology	Software, Investigation	Visualization, Writing (Draft and Review)	Project Administration
Keefe Zebastian Dela Cruz	Methodology	Software, Investigation	Writing (Draft and Review)	Project Administration
Addy Dhingra	Methodology	Software, Investigation	Writing (Draft and Review)	Project Administration

 # Note: The group did not assign roles at any point due to the small size of the group

Group member name	Comments (provide details about each contribution)
Matthew Fowler	• Conceptualisation – Responsible for the ideas in 6.3.2 relating to multiple responses and reusing previous responses. Also responsible for ideas relating to storing previous responses. And recommending generally repeated sentences/statements provided location/time data. • Methodology – Responsible for the methodology behind the module in 5.2.2, as well as the planning for the test harnesses for the LLM and Prompt Engineering • Software – Contributed the module in 5.2.2 • Visualization – Responsible for the overall formatting of the document, as well as the formatting of the reference list, and some consistency in writing and terminology • Project Administration – The group performed much of the organisation without the need of nominating a group leader, due to the small size of the group. Organisation was completed through the established methods of communication and regularly held meetings. • Writing (Review) – The group performed a full review of the document, with each member reading the document and approving content prior to submission.

	• Writing (Draft) – Responsible for the writing of sections - Executive Summary, 3.3, 5.2.2, 4.3, 5.3, 6.3, 7 (predominately LLM)
Keefe Zebastian Dela Cruz	• Responsible for all content done in Chapter 2, 3.2, 4.2, 5.2, 6.2, and 7. • Drafted work. • Reviewed work. • Did research and planning and integration of APIs. • Changed plus and minus feature to checkboxes.
Addy Dhingra	• Responsible for all content done in Chapter 3.1, 4.1, 5.1, 6.1, and 7. • Drafted work. • Reviewed work. • GANNT Chart.

Acknowledgement

By signing below students are acknowledging that the contributions recorded above are a true representation of the contribution made by everyone.

Name: Matthew Fowler	**Student ID:** a1803076	**Signature:**
Name: Keefe Zebastian	**Student ID:** a1793854	**Signature:**
Name: Addy Dhingra	**Student ID:** a1803893	**Signature:**