

COMMUNICATION AND MOBILITY ASSISTANT FOR DISABLED PERSONS

Final Report

Council on Aging & Human Services



Team DAK

Darrel Nitereka

Kyle Parker

Apram Cherra

02/02/24

TABLE OF CONTENTS - (Estimated)

I. Introduction.....	4
II. Team Members & Bios	5
III. Project Requirements Specification.....	5
III.1. Project Stakeholders	5
III.2. Use Cases	6
III.3. Functional Requirements.....	6
Food & Drink Button	6
Bathroom Button.....	6
Body Part Indicator	6
Emotion Logging.....	7
Survey Function.....	7
III.4. Non-Functional Requirements.....	7
Usability	7
Reliability.....	7
Performance	7
Supportability	7
Implementation	8
Interface	8
Operation	8
Packaging	8
Legal	8
IV. Software Design - From Solution Approach	9
IV.1. Architecture Design	9
IV.1.1. Overview	9
IV.1.2. Subsystem Decomposition	9
IV.2. Data design	13
IV.3. User Interface Design.....	14
V. Test Case Specifications and Results	18
V.1. Testing Overview	18
Unit Testing	18
Integration Testing.....	18

System Testing.....	18
Functional testing:	18
Performance testing:	19
User Acceptance Testing:	19
V.2. Environment Requirements	19
V.3. Test Results	19
VI. Projects and Tools used	21
VII. Description of Final Prototype	21
VIII. Product Delivery Status	24
IX. Conclusions and Future Work	24
IX.1. Limitations and Recommendations.....	24
IX.2. Future Work.....	24
X. Acknowledgements	24
XI. Glossary.....	25
XII. References.....	25
XIII. Appendix A – Team Information	27
XIV. Appendix B - Example Testing Strategy Reporting	28
XV. Appendix C - Project Management.....	28

I. Introduction

This document will explain the current prototype and requirements that need to be fulfilled. An overview of the approach and test plans will follow requirements. The alpha prototype will be described in detail with subsystems following suit. This document will conclude with future work to declare and outline project needs.

In this project, we are teaming up with Council on Aging (COA) and Boost Collaborative, two nonprofit organizations that serve as cornerstones in their communities in helping children with special needs. At the start of this project, we were tasked with continuing the application designed in the previous year for the COA clients Dan and Missy and specifically for the child of Dan who required an assistive tracking app and nonverbal communication aide. Now that we have been able to mostly satisfy the COA clients requirements, we are now evolving our project into incorporating the idea of a different nonprofit Boost Collaborative which is looking for a type of survey implementation for the users of the app. We look forward to meeting all the needs of our clients.

Cooperative games for those with special needs have gone through many designs and changes and have been carefully developed to help those with special needs in fun and encouraging ways. Communication, Time Management, Cognitive skills, Decision-making ability, and so much more can be learned through the proper usage of cooperative games (Cooperative Games for Children with Special Needs). All of which are difficult to come by when verbal skills are minimal or non-existent. This app will allow those without a voice to communicate with others in a minimalistic way.

The domain in which we are trying to solve the project problem is that of assistive technology (AT), which encompasses a wide range of devices, software, and equipment. They are designed to enhance the learning, communication, and overall functioning of individuals. AT is present in entities such as mobility devices (i.e., wheelchairs, walkers, etc.), and software (i.e., text to speech, etc.) that assist people with motor or speech impairments.

The pursuit of AT can include high purchase and maintenance costs, issues relating to quality or production, limited resource access, poor training, and the underlying challenge of securing enough government funding that support assistive technology initiatives.

Initially, an application tracked the user's movement, provided loud audio messages, animations, and audio/visual encouragement. DAK pushed forward the app to add enhanced voice features, cleaner pages, and new colors.

In Speech and for every button/tile, a recorded chunk of audio plays. These are recordings of an actual person so there is not always a speech synthesizer utilized. In the case that a recording is not available for the word (i.e., glitch, new feature, etc.), then the speech synthesizer will be used.

Each page is color coded so users can easily associate the page with their respective actions. For example, gray has been assigned to Speech; the homepage icon is gray and each button within Speech has been turned to gray. In Emotions, each icon has a unique tile color associated with the emotion. That is, sad is associated with blue and alright is in the middle with grey. Some tiles have been merged into settings to increase the ease of use so that patient users do not see the tracking sheets whereas nurse users will want to see the statistics.

The project's main goal is to promote and empower non-verbal patients with greater self-advocacy. For people whose main mode of communication is non-verbal, this app can provide them with the ability to express their needs and basic responses (i.e., yes, no, etc.). This goal has been reached to the satisfaction of the clients thus far.

II. Team Members & Bios

Apram Cherra is a Computer Science student. His skills include C#, Java, React, JavaScript, HTML, and SQL. He has worked at BoxWave as a Web Developer. On this project, his responsibilities are to add features to the interface, namely adding emotions.

Darrel Nitereka is a Computer Science student. His skills consist of Java, Python, HTML/CSS, and Mobile Application Development Experience. He took part in a STEM workshop assisting kids understand algorithms and basic components of Computer Science. He is responsible for building the interface further, namely working with food, drink, and bathroom commands.

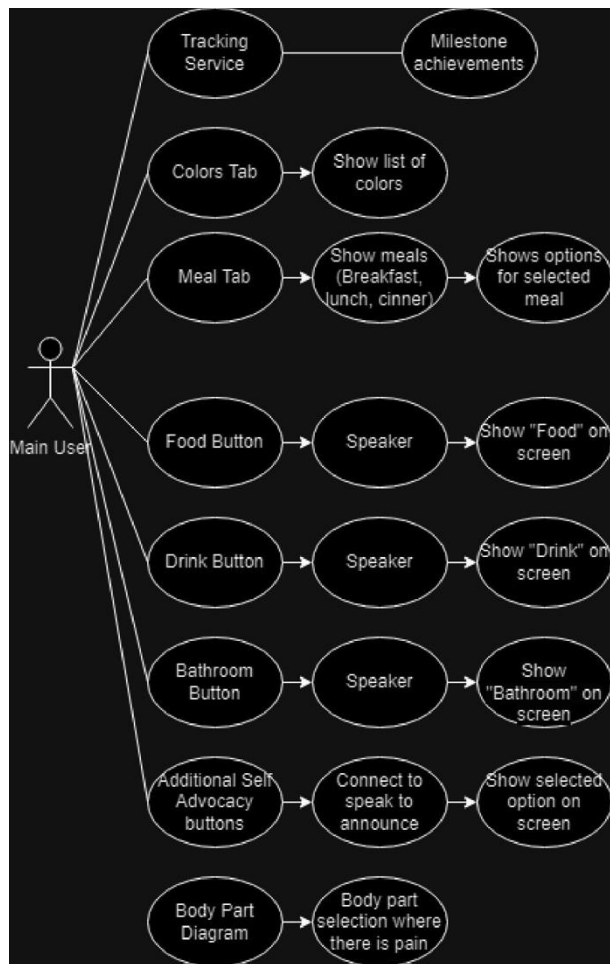
Kyle Parker is a Software Engineering student. His skills consist of Swift, C, C++, PostgreSQL, and iOS development. His experience consists of being a teaching assistant for an introductory C class at WSU. He is responsible for building a framework enabling quicker UI development and enabling human voices for the button presses. He will record male and female voices then integrate them into the application. He will also do the human body indicator parts.

III. Project Requirements Specification

III.1. Project Stakeholders

List your stakeholders, summarize their needs.

III.2. Use Cases



III.3. Functional Requirements

Food & Drink Button

- Button will show a new page with a food button and drink button within.

Bathroom Button

- Button to show new page which contains a button that textually, visually, and verbally indicates they need to use the bathroom.

Body Part Indicator

- Buttons for Each Body Part: An image will be presented, and the user will click on a body part. Unlike other pages, this will not consist of a lot of buttons.

Emotion Logging

A Colorful Emotion Page: Various emotions each with a tile whose color reflects such emotion (per popular culture).

Survey Function

Survey feedback Function: Various survey questions posed to the user about their experience using the app and their progress being made overtime with the data tracked on their app usage.

III.4. Non-Functional Requirements

Usability

- System shall provide an easy to user interface, that provides an OOBE.
- The system shall adhere to WCAG Level AA standards to ensure accessibility for users with special needs.

Reliability

- The system will use a service with high server availability. Local use should always be available on- and offline.
- In the event of server downtime, data will be stored locally and enqueued to be pushed when server connectivity resumes.

Performance

- The system shall be able to provide a quick response:
 - Within two seconds for server queries (on high-speed networks); and
 - Within one minute initially download the package (on high-speed networks); and
 - Within one-hundred milliseconds for UI-requests (i.e., screen change); and
 - Load speech audio files within one-hundred milliseconds
 - Shall be asynchronous by default and synchronous in rare scenarios.
- In lose terms, response to clicking a button should appear instant and feel like an extension of the user their self and not frustrating.

Supportability

- The system shall be modular and provide good documentation with examples (as needed) to ensure maintainability.
 - Regular updates to documents should occur with bug fixes and feature additions.
- The system shall be portable to run on different hardware and software, following industry standards.

Implementation

- The system shall account for hardware constraints. Proper documentation should state minimum requirements where present.
- The system shall account for constraints possibly imposed by future maintenance.
- The system shall account for constraints imposed by the testing team. Constraints include specific testing procedures or the testing environment.

Interface

- The system shall be able to interact with the existing system.
- The system shall include data import and export features as needed and adhere to industry-standard formats.
- The system shall support relevant data standards and ensure privacy with applicable laws.

Operation

- The system shall be managed by a designed team with defined roles and responsibilities.
- The System shall include documented procedures for system operation such as:
 - Backups; and
 - Disaster recovery

Packaging

- The system shall be installed OTA or manually by qualified personnel.
- The system shall maintain a count of installation to track company need and scalability.
- The system shall download in a reasonable time and remain a small application.

Legal

- The system shall follow licensing terms for the system and clearly define the user's rights. Local laws apply and should be disclosed.
- The system shall outline all fees and royalties in the licensing agreement.

IV. Software Design - From Solution Approach

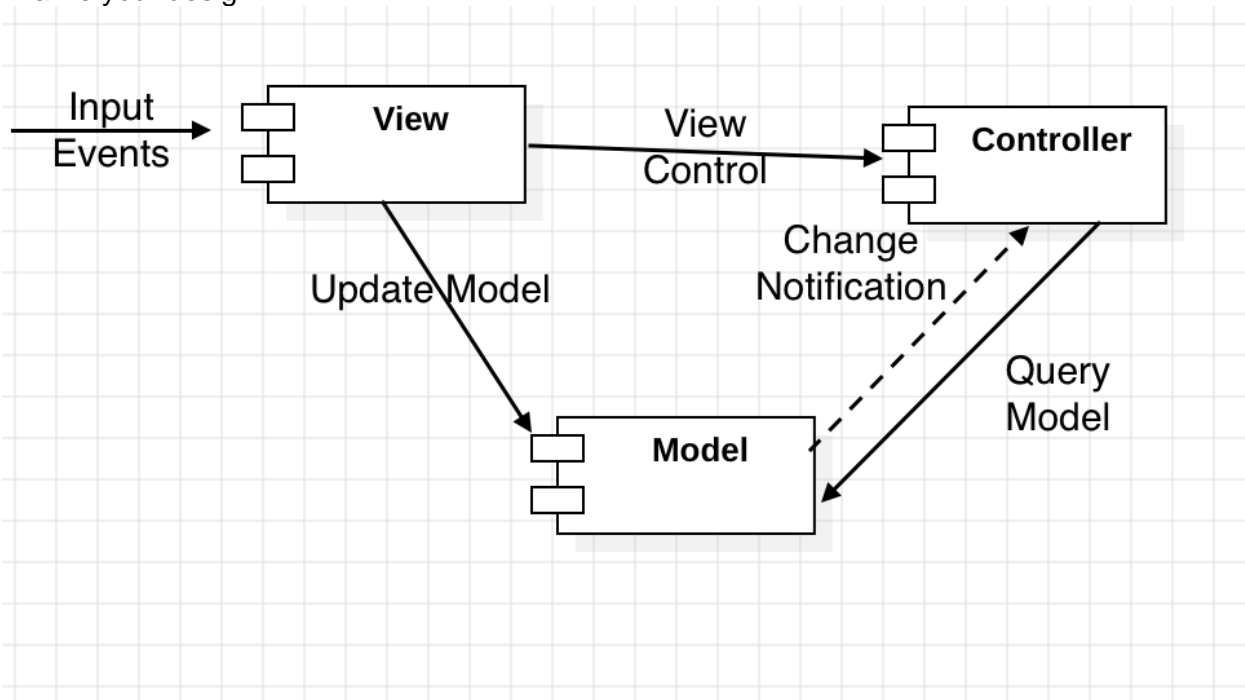
IV.1. Architecture Design

IV.1.1. Overview

Model View-Controller architecture design is the optimal pattern for the user interface (UI) section. This design will allow for user interaction to trigger either speech synthesis or screen changes. The model will be responsible for all UI changes such as changing the current screen. The UI will be responsive to signals received from said model layer. All underlying data, layouts, and other components will be classified under the "model". The controller is responsible for providing data the view represents. In essence, the view visualizes the controller which interacts with the model to enable updates and such to the UI.

Input events consist of the user tapping the screen to navigate to a desired speech or game. All data will be updated from the server at launch. Data transfers will be reliant on a client-server architecture since the application reaches out to a server and waits for a response. The client will reach out for regular updates on launch. Until the application is closed (not background), updates will be received in real time and will take effect the next time the data is loaded. The server will be managed centrally and push updates to clients observing.

This section should describe the overall architecture of your software. The architecture provides a top-level design view of a system and provides a basis for more detailed design work. This will be the initial draft of your software architecture. Next semester you will revise this draft and finalize your design.



IV.1.2. Subsystem Decomposition

2.1. [UI Handler]

2.1.1. Description

The subsystem manages the user interactions with the UI elements of the app, including buttons for bathroom needs indicators, meal selection, drink options, custom text input, as well as emotion logging.

2.1.2. Concepts and Algorithms Generated

Several sub-classes make up the UI Handler, each corresponds to a specific UI element within the application. These sub-classes account for the maintenance of user interactions within their respective UI elements. Algorithms are employed to capture user input events, process button clicks, and trigger the appropriate actions within the application. Furthermore, the UI Handler assures the smooth integration of these UI elements, supplying an accessible and cohesive experience.

2.1.3. Interface Description

Services Provided:

Service Name	Service Provided To	Description
Food/DrinkSelectionButtonAction	User Interface Elements	Handles button actions for food and drink selection.
BathroomNeedsButtonAction	User Interface Elements	Handles button actions for bathroom and urgency of bathroom needs.
EmotionLoggingButtonAction	User Interface Elements	Handles button actions for user to display their emotion.
BodyPartButtonAction	User Interface Elements	Handles button actions for users to display body parts they want to indicate.

Services Required: None (internal subsystem).

2.2. [Communication Handler]

2.2.1. Description

The communication subsystem manages communication between the user interactions with the external elements of the app which include audio playback in response to interaction with the app's UI elements.

2.2.2. Concepts and Algorithms Generated

The communication subsystem manages the sophisticated process of playing back user interaction into spoken language, ensuring effective communication. Further, it oversees the storage and retrieval of user data, conserving crucial information for any future reference and analysis.

2.2.3. Interface Description

Services Provided:

Service Name	Service Provided To	Description
AudioLabelPlayback	Pre-recorded audio labels	Handles the audio played back from clicking on a specific button.
DataStorageAndRetrieval	Database	Handles storage and retrieval of the user's data, specifically for the logging of emotion and audio labels, to ensure that the data is stored securely and accessibly.

Services Required: None (internal subsystem).

2.3. [Input Handler]

2.3.1. Description

The system core subsystem manages the core of the app, overseeing the system's overall functionality and user data.

2.3.2. Concepts and Algorithms Generated

The system core subsystem generates various algorithms and concepts that advance the core functionality and concepts of the app. The System logic, User profiles, and database management are all key areas of this subsystem.

2.3.3. Interface Description

Services Provided:

Service Name	Service Provided To	Description
SanatizedFood/DrinkSelection	System Core	Handles sanitizing user inputs for food and drink selection meal selection, ensuring choices are communicated accurately to the core engine.
SanatizedBathroomNeeds	System Core	Handles sanitizing user inputs for bathroom necessities, ensuring accurate communication to the core engine.
SanatizedEmotionLogging	System Core	Handles sanitizing user inputs for emotion logging, ensuring accurate communication to the core engine.
SanatizedBodyPart	System Core	Handles sanitizing user inputs for body parts, ensuring accurate communication to the core engine.
SanatizedAudioPlayback	System Core	Handles sanitizing audio playback when user interaction takes place with a button, ensuring accurate communication to the core engine.

Services Required: Core engine(System Core).

2.4. [System Core]

2.4.1. Description

The system core or the core engine subsystem manages the core of the app, overseeing the system's overall functionality and user data.

2.4.2. Concepts and Algorithms Generated

The system core subsystem generates various algorithms and concepts that advance the core functionality and concepts of the app. The System logic, User profiles, and database management are all key areas of this subsystem.

2.4.3. Interface Description

Services Provided:

Service Name	Service Provided To	Description
UserProfileManagement	Input Handler	Handles the audio played back from clicking on a specific button.
DatabaseActions	Communication Handler	Handles action within the database for the user data storage, retrieval, and management to maintain data security within the app.

Services Required: Input data from the Input Processing Subsystem.

2.5. [Feedback Handler] [OBJ]

2.5.1. Description

The feedback subsystem manages the real-time audio played back to the user when they interact with certain buttons within the application.

2.5.2. Concepts and Algorithms Generated

The feedback handler generates concepts and algorithms that enhance the user's interaction and overall experience with the application. This includes the audio mechanism that plays audio when the user engages with certain buttons in the app.

2.5.3. Interface Description

Services Provided:

Service Name	Service Provided To	Description
VerbalizedSelection	User Interface Elements	Handles the audio playback functionality is working effectively.

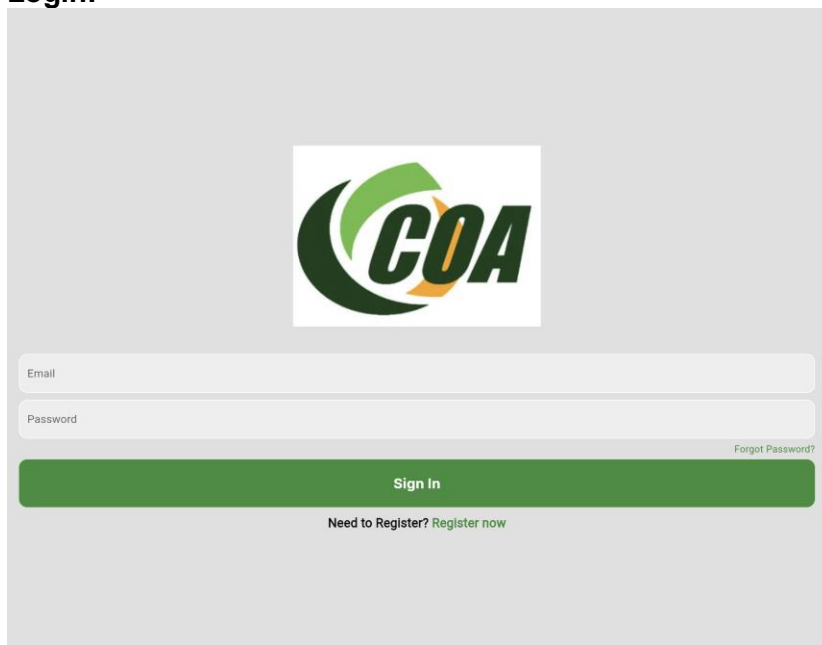
Services Required: None (internal subsystem)

IV.2. User Interface Design

Current UI: This UI mockup will be a basic overview of the application as it stands and just show the most fundamental aspects of our app while giving key insight into our progress from a UI perspective. This UI mockup will not show all the pages in our app or all the features as more detail will be left for section VI (Description of Final Prototype).

The first image shows what the user should see when they open the app, which is a login screen prompting them to either login or register for a new profile if they have not yet created one.

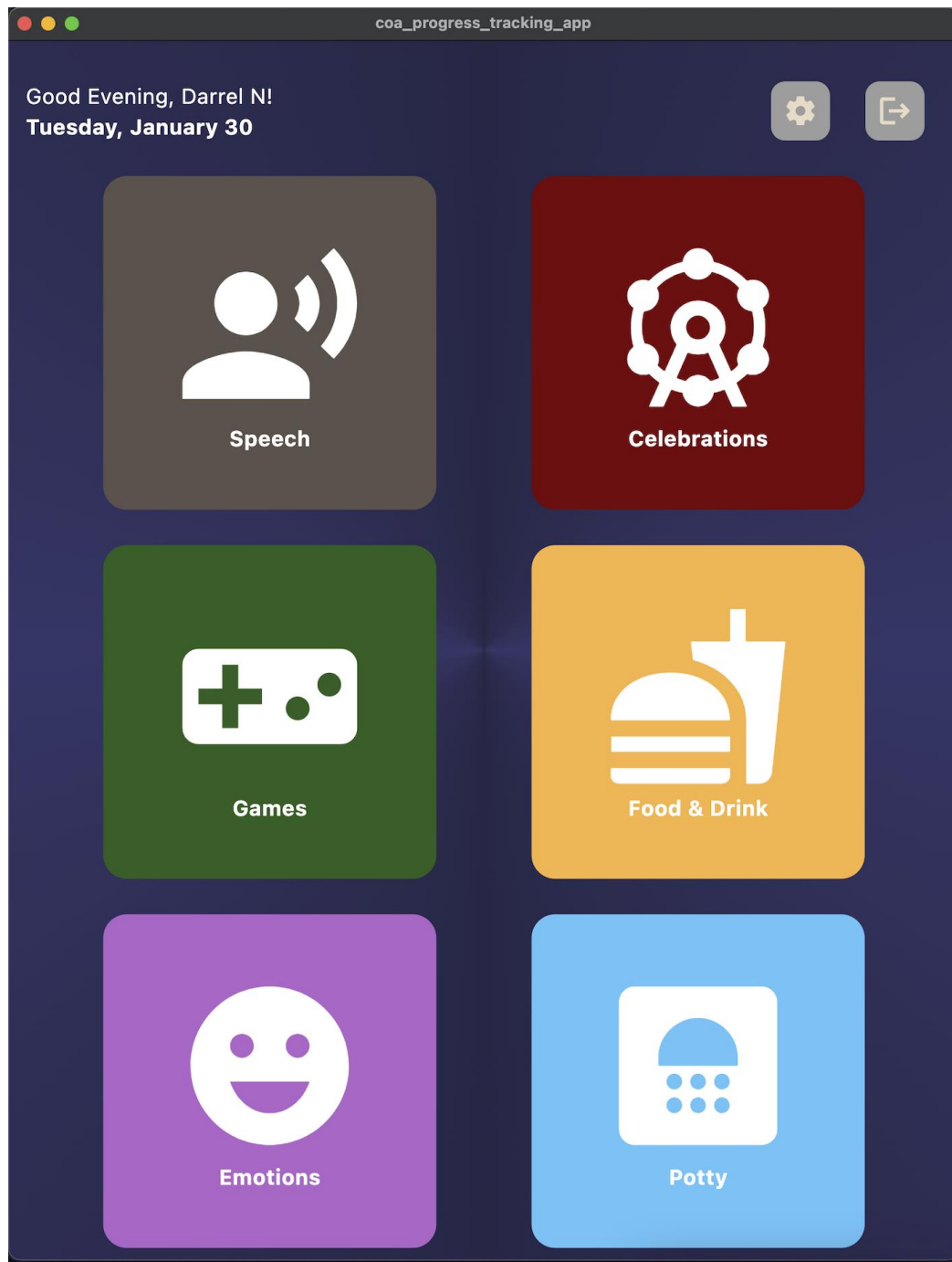
Login:



The login screen features a light gray background. At the top center is the COA logo, which consists of a green and orange circular emblem next to the letters 'COA' in a bold, dark green font. Below the logo are two white input fields with gray borders; the first is labeled 'Email' and the second is labeled 'Password'. To the right of the password field is a small, faint link that says 'Forgot Password?'. Below these fields is a wide, solid green button with the text 'Sign In' in white. At the bottom center, there is a line of text that reads 'Need to Register? Register now', where 'Register now' is in a green color.

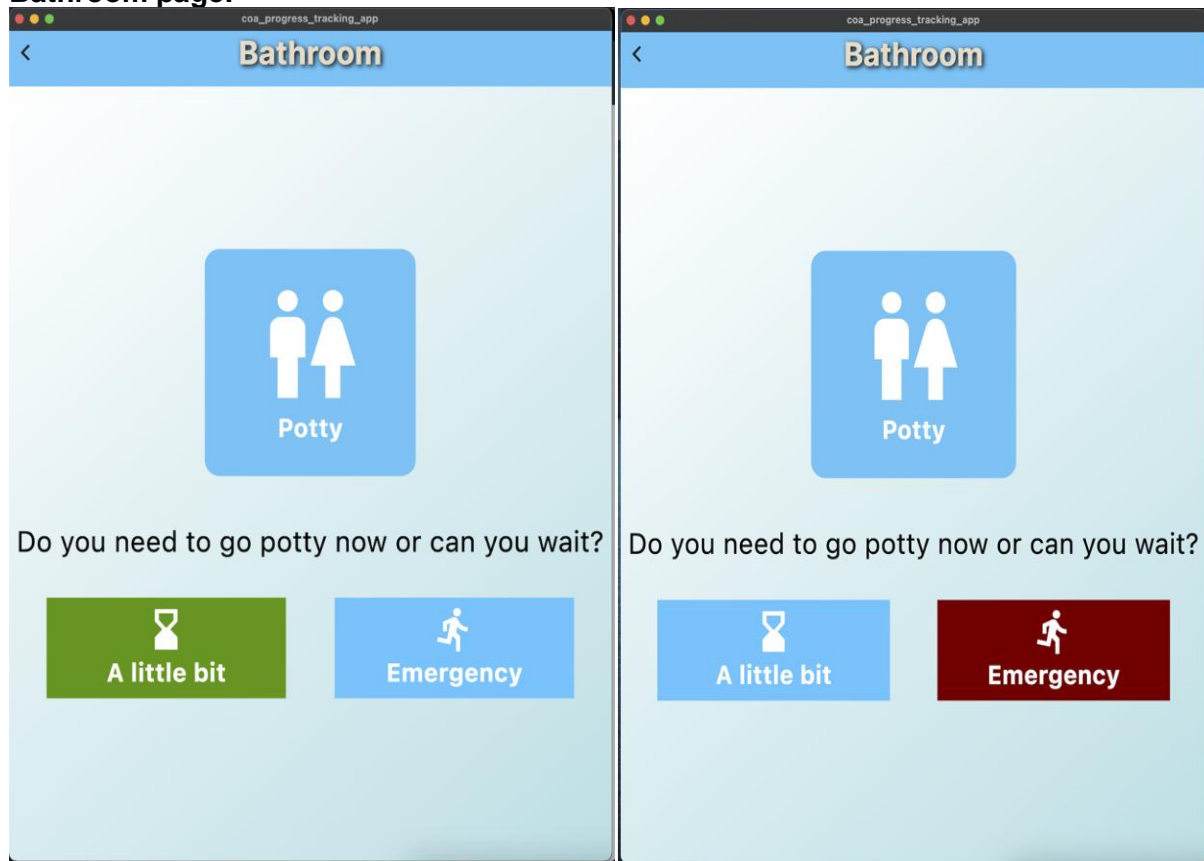
The next image showcases the screen the user sees once they are logged into the app. The home page shows the various pages we have that the user can choose to interact with from the speech page to the Food & Drink page etc.

Home Page:



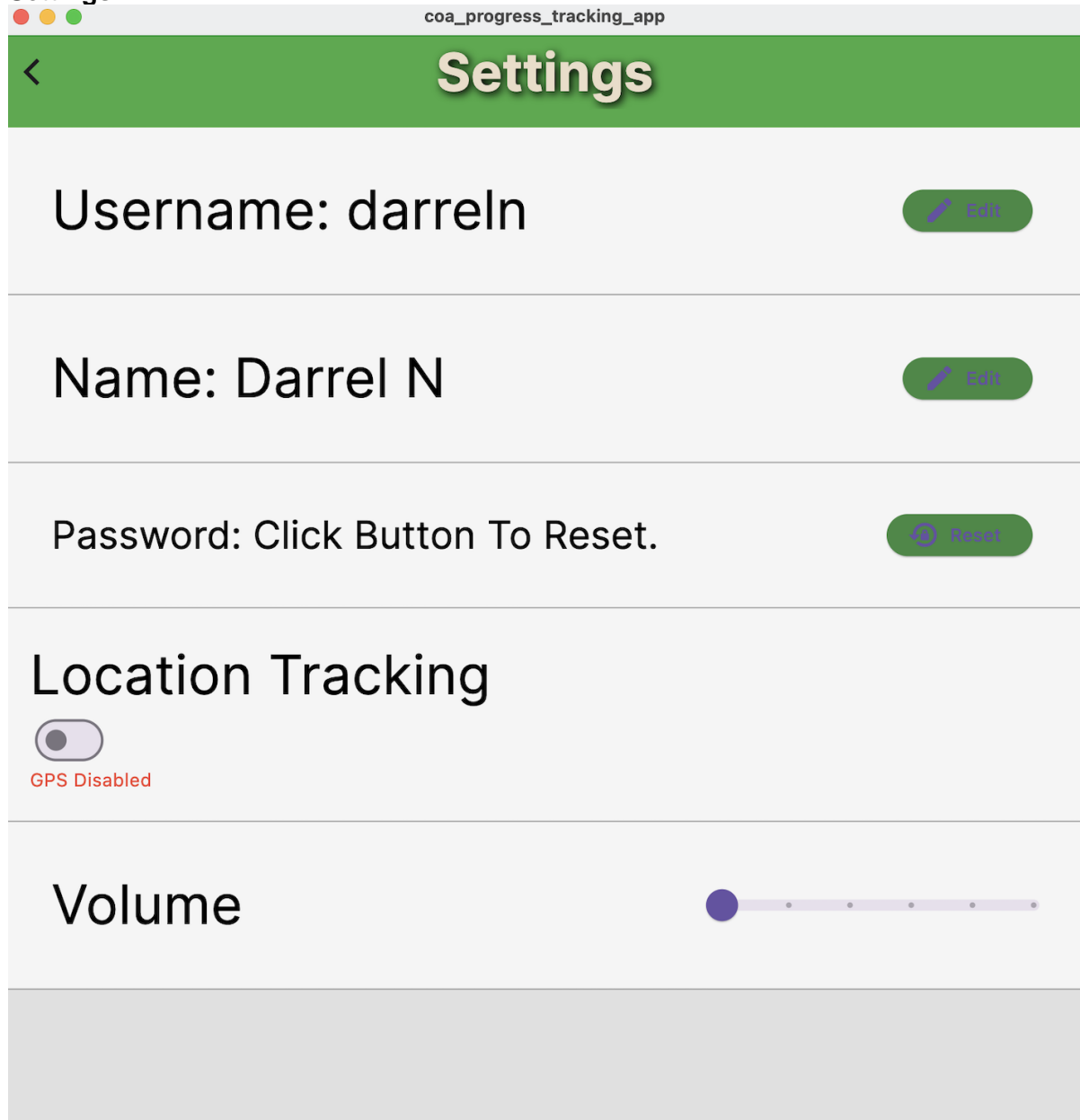
These images show the bathroom page which gives the user the option to indicate if they need to use the bathroom and gives them the ability to demonstrate whether they need to use the bathroom immediately or can wait.

Bathroom page:



The last image shows the settings page which now incorporates the option to change the username and name of the user, the location tracking mechanism, and ability to increase or decrease the volume of the app audio.

Settings:



coa_progress_tracking_app

Settings

Username: darreln [Edit](#)

Name: Darrel N [Edit](#)

Password: Click Button To Reset. [Reset](#)

Location Tracking

☐
GPS Disabled

Volume

Volume slider control

The pictures and features are the current application UI. However, in the coming weeks, there will be many changes to the UI. After a recent meeting with our clients, we will now begin incorporating a survey mechanism feedback function to give users the ability to indicate their satisfaction with the app and how they think their own progress is going.

V. Test Case Specifications and Results

V.1. Testing Overview

The plan to test our project is to test each component individually, and then the entire project. The testing can occur during the development of functions, or at the end of the project's phases. Testing will also require physical testing of the GPS, as well as testing other technologies such as a speaker.

Unit Testing

Test each function separately. This includes testing each button, testing whether that button displays the correct text, and also outputting the correct vocalized sounds from the speakers at the same time. Similar unit testing can occur in other functions, for example, allowing changes to the settings of the application, would require us to test each setting and ensure the change occurs. Each function would need to be tested by creating unit tests which test for particular cases. For example, one the setting is changed from light to dark mode, we can call a unit test called "CheckForDark" which will check if the theme of the app is actually dark mode or not, and will pass a failed or successful test output depending on what happens. We would do this for every little button and function, to ensure that each small and individual component is working first before testing them together.

Integration Testing

When we integrate the speaker, we would need integration testing. Before we move on with the many features of the speaker, we need to ensure that the speaker is working correctly and is integrated correctly with the rest of the system. For this, we would need to test during our integration of the speaker, allowing us to fully integrate the speakers and ensure they work the way we need them to, before moving on the adding additional functions and testing those functions. Let's say we want the text on the screen to be different colors, and we have created a "colorText" class which is a child class to the "Text" class. Once we are done implementing the colors, we would need to integrate the colors to the actual text. During this phase we would integrate test whether the colors and text are interacting properly. As we integrate the colors and text, we create tests, and see if we have properly integrated all the necessary technologies, libraries, and tools.

System Testing

Functional testing:

For this we want to test the functions that have a high probability of uncovering a failure. For example, with this application, the speakers perhaps play the sound before the text is displayed. While we have already tested the individual, speaker, and text functions, we haven't tested them together, and it is in this functional testing that we are testing for the functionality of the overall application, where we would fine whether individual components can come together to perform operations correctly. Ensuring that the correct services are called, or that the GPS is displaying the correct location is all function testing as these features have a high probability of failure so we must test them vigorously to ensure we have covered all loss ends.

Performance testing:

In performance testing, we want to ensure that our project is performing in. This would require us to test many different performance criteria. For example, we would need to test to ensure that our application isn't draining the battery of the tablet. Secondly, we would also need to test and ensure that our application can run on a fully charged tablet. This will ensure that the overall quality of the project is better. We can also test the stability of the application and ensure the application doesn't crash randomly. Stress testing will be very important, to ensure that the speaker doesn't break, or that the application is able to record all the necessary data. By testing performance criteria such as battery efficiency, GPS accuracy, etc, we can perform performance testing.

User Acceptance Testing:

During user acceptance testing, we would test for whether we are able to download and install the application on a new device, to ensure that the application is able to be installed on any device/tablet. Furthermore, we would need to test and ensure that the user is able to sign up and create an account, and if that account is being stored in our database. This type of testing ensures that our application is ready to be released and that all features are running correctly with no to very little error. During this phase of testing, the stakeholders will also take a look around the application to ensure that we have met all the project agreements.

V.2. Environment Requirements

Our project's testing environment revolves primarily around the framework IntelliJ, which is being used to develop our app. The IntelliJ environment will carry out the testing process, by leveraging its built-in testing tools and features. Both unit and integration testing will be conducted using the IntelliJ testing framework, therefore ensuring a cohesive approach to testing across the project.

There are no specific demands in terms of hardware requirements, as the IntelliJ development environment is platform-agnostic, which will allow us to perform testing on numerous devices and operating systems. This will ensure our app is flexible enough to be thoroughly tested across diverse set of environments, enhancing the strength of the app.

V.3. Test Results

Main Levels

Description:

This is a series of manual test cases for selecting a main level.

Assumptions:

- The app has been opened with a new user

Test ID	Expected Inputs	Expected Outputs	Actual Outputs	Pass/Fail
01	Click on Register now button	The registration page appears to the user.	The registration page appears to the user.	P
02	Click on Sign up Button	The login page appears to the signed-up user.	The login page appears to the signed-up user.	P
03	Click on Sign in Button	The home page appears after user is logged in.	The home page appears after user is logged in.	P
04	Click on Speech Page	The speech page with multiple speech options opens for the user.	The speech page with multiple speech options opens for the user.	P
05	Click on Food & Drink Page	The food/drink page opens for the user.	The food/drink page opens for the user.	P
06	Click on Emotions Page	The emotions page opens for the user.	The emotions page opens for the user.	P
07	Click on Potty Page	The potty page opens for the user.	The potty page opens for the user.	P
08	Click on Settings Button	The settings button opens for the user.	The settings button opens for the user.	P

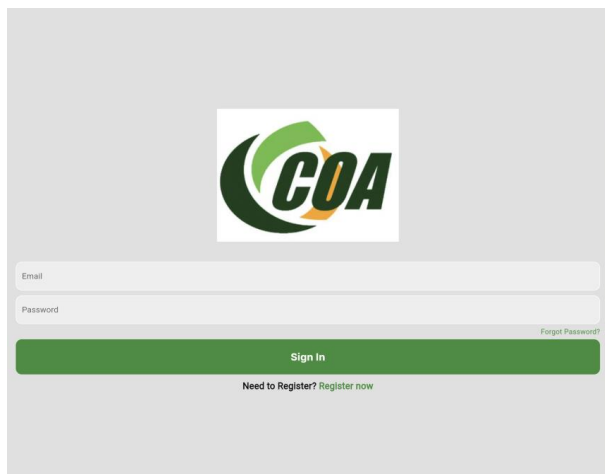
VI. Projects and Tools used

Tool/library/framework	Purpose
Flutter	Framework we're using to design our application natively.
IntelliJ IDEA (with Flutter Plugin)	IDE we are using to develop our application.

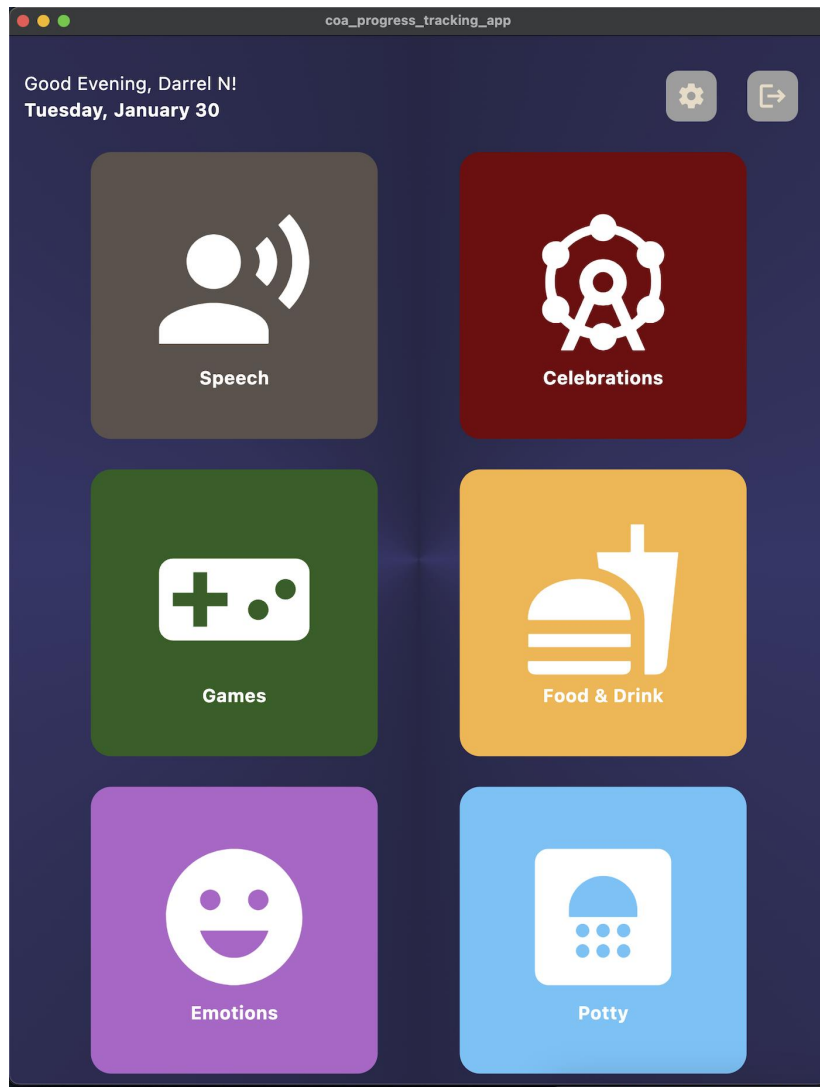
Languages Used in Project			
Dart	Ruby	Swift	

VII. Description of Final Prototype

Users will be able to download the application via the Google Play Store, or the App Store, depending on if we want the application on the store. If not, the application will be installed on desired devices.



Users will be prompted with this log-in/sign-in page, where users will be able to login, or new users will be able to register.



Users will then be prompted with this homepage, where they will be able to select a number of options. Each button will send the user to the next page, which has more buttons, which users can select from. For example, selecting the food and drink button, will send users to a page with other buttons where they can select their meal. Most of the main applications use case scenarios come from the homepage. From here the user can select any button.

- a. The speech button will allow text-to-speech conversion to help with communication.
- b. The games button will display tic-tac-toe for users to be able to play.
- c. The Emotions button will display many emotions which the user can select any emotions from, and that emotion will be recorded.
- d. Potty button to allow the user to express their need to use the restroom.
- e. Food and drink button so the user can select which meal they want.
- f. Celebrations button to allow users to celebrate their achievements.
- g. We also have a settings button, with many features such as user preferences, account information, location tracking, and more.

In our final prototype, we will include the usage of the GPS tracking service. Due to some issue with the database, the team was unable to get the GPS tracking services to function correctly. However, our final prototype will include a somewhat working implementation of the tracking service.

Many changes still may occur to the buttons, such as which exact meals are in the als are in the food&drinks tab. However, these changes are all dependent on the individual.

Recently, Eric, from Boost Collaborative, has joined Dan and Missy, in helping us create an application to better not only students' lives, but the lives of people at work, or at the office. Eric has proposed the idea that we create an application where we can give different types of surveys, these surveys will ask questions such as, "are you happy at work" or "Do you think Boost Collaborative satisfied your needs?". From these surveys, we collect the responses and store them in a database, where we can find averages and other relationships within the data..

With this additional application, there will be an admin account, and the ability for any user to be able to take the survey. Since Eric was sick the first week, and we were not able to meet with him until the week after, we do not have a prototype screenshot to show here for this additional application.

VIII. Product Delivery Status

The Final project will be delivered either as an app that can be downloaded through the Google Play Store or as an executable that the clients can install to their personal tablet. Our team was given the tablet and assistive equipment the main user uses to incorporate our app itself into it. Once we finish the final enhancements to the app, we will update the app in the tablet and meet with clients to hand them back their equipment with the upgraded application.

IX. Conclusions and Future Work

IX.1. Limitations and Recommendations

Current Limitation is for the Survey Feedback Function as we do not have all the necessary information from our client that we need to fully implement it yet.

IX.2. Future Work

Major tasks include the following,

- Enhance the visual appeal of the application by incorporating improved icons.
- Incorporate Survey Feedback Function for users.
- Improvements on the “emotion” display menu.
- Refined use of colors throughout the application.
- Subsystem name for prototype final submission.
- Add Body Parts Page for broader communication.
- Incorporate voice into game page.

X. Acknowledgements

Thank you very much to Dan, Missy, and Eric for the great opportunity we were given to be able to create such an innovative and helpful application for those in need. Also thank you to the Council on Aging and Boost Collaborative nonprofits for partnering with us for our senior capstone project.

XI. Glossary

Assistive Technology (AT): Any device, software, or equipment that helps people with disabilities learn, communicate, or function better. AT can be low-tech, such as communication boards made of cardboard or fuzzy felt. It can also be high-tech, such as specialized computer software and hardware.

Cloud Services: Applications and infrastructure resources that exist on the internet.

Cognitive Health Assessment: The evaluation of an individual's cognitive abilities, including memory, problem-solving, and decision-making skills, to determine their mental well-being.

Cognitive Skills: The ability of an individual to perform various mental activities most closely associated with problem solving

Cross-platform development: An approach to developing software applications that are compatible with multiple operating systems.

Interface: A screen the user directly interacts with.

OUBE: Out-of-box Experience. An experience that does not require minimal training prior to first use.

OTA: Over the air; a method of delivery for applications requiring access to the internet and a portal such as Google Play Store.

Cloud Solution: A location out of the user's direct management, where user data is stored and managed by the application. It also provides authentication for login/logout operations.

State-of-the-Art: The most current and advanced developments or technologies in a particular field at a given time.

WCAG: Web Content Accessibility Guidelines is a set of guidelines that ensure fair and easy use for people with disabilities.

XII. References

M. A. Hameed, M. Hassaballah, M. E. Hosney and A. Alqahtani, "An AI-Enabled Internet of Things Based Autism Care System for Improving Cognitive Ability of Children with Autism Spectrum Disorders," Deep Neural Networks for Cognitive Health Assessment, vol. 2022, 2022.

L. J. Wali and F. Sanfilippo, "A Review of the State-of-the-Art of Assistive Technology for People with ASD in the Workplace and in Everyday Life," 2019.

"Cooperative Games for Children with Special Needs," Disability Grants, [Online]. Available: www.disability-grants.org/cooperative-games-for-children-with-special-needs.html. [Accessed 16 September 2023].

A. Tlili, M. Denden, A. Duan, N. Padilla-Zea, R. Huang, T. Sun and D. Burgos, "Game-Based Learning for Learners with Disabilities-What Is next? A Systematic Literature Review from the Activity Theory Perspective.," Front. Psychol., 2022.

H. v. B. – C. Psychologist, "Color Psychology," Color Psychology, 2023. [Online]. Available: <https://www.colorpsychology.org/>. [Accessed 02 December 2023].

"Child Voice, AAC, and Safeguarding in the SEND Classroom," Special Education and Inclusive Learning, Dec. 13, 2020. <https://inclusiveteach.com/2020/12/13/child-voice-aac-and-safeguarding/>

M. K. Staff, "Tools to Build Kids' Self-Advocacy Skills," Metrokids, Feb. 17, 2017. <https://www.metrokids.com/tools-to-build-kids-self-advocacy-skills> (accessed Dec. 10, 2023)

A. S.-L.-H. Association, "Augmentative and Alternative Communication (AAC)," American Speech-Language-Hearing Association, 2023. [Online]. Available: <https://www.asha.org/public/speech/disorders/AAC/>. [Accessed 20 Nov 2023].

N. Zeliadt, "New technology gives voice to nonverbal people with autism," Spectrum | Autism Research News, Jun. 22, 2015. <https://www.spectrumnews.org/news/new-technology-gives-voice-to-nonverbal-people-with-autism/>

A. Speaks, "Autism and speech devices: Helping kids advance skills as they mature," Autism Speaks. <https://www.autismspeaks.org/expert-opinion/autism-and-speech-devices-helping-kids-advance-skills-they-mature>

National Autistic Society, "What is Augmentative and Alternative Communication and how can it benefit autistic people?," 27 January 2023. [Online]. Available: <https://www.autism.org.uk/advice-and-guidance/professional-practice/aug-alt-comm>. [Accessed 02 December 2023].

Understood. "Dictation (Speech-to-Text) Technology: What It Is and How It Works." Reading Rockets, www.readingrockets.org/topics/assistive-technology/articles/dictation-speech-text-technology-what-it-and-how-it-works.

Jonathan. "Why Educational Apps for Kids with Special Needs Help." Mussila, 3 June 2022, mussila.com/why-educational-apps-for-kids-with-special-needs-help/.

"Geolocation Harnessing Location-Based Insights for Learning." Utilities One, utilitiesone.com/geolocation-harnessing-location-based-insights-for-learning.

"Working Together: People with Disabilities and Computer Technology." Working Together: People with Disabilities and Computer Technology | DO-IT, www.washington.edu/doit/working-together-people-disabilities-and-computer-technology-0.

XIII. Appendix A – Team Information

List your team members here and provide a team photo.

XIV. Appendix B - Example Testing Strategy Reporting

- 1) Identify the requirements being tested
- 2) Either link to available online test results and/or take screenshots of the various system testing results
- 3) User testing can be demonstrated via survey results or quotes and a discussion of the feedback received

XV. Appendix C - Project Management

Describe your team's weekly schedule, i.e.,

- weekly meetings with the instructor/ mentor
- weekly meetings only with the team members
- other meetings and project related team activities.

Explain the purpose of each of the above activities and describe the routine agenda for each.

Please comment on which team activities/meetings were the most beneficial to your team.

Please include any planning documents you may have used. Examples could include:

- Gantt charts
- GitHub projects - Screenshots & a figure description
- GitHub issues - Screenshots & a figure description
- Notes on team tools used:
 - Email
 - SMS/IM
 - Slack
 - appear.in or other video conference tools
 - Trello
 - etc.