



**Team**

**HomeAide**

---

# **Software Design**

**Version 1.1**

**February 12, 2021**

---

**Project Sponsors:** Kelly Roberts, PH. D and Jill Pleasant, MA

**Team Faculty Mentor:** Fabio Santos

**Team members:** Seth Borkovec, Ethan Donnelly, Courtney Richmond, Noah Baxter

# Table of Contents

<b>Introduction</b>	<b>3</b>
<b>Implementation Overview</b>	<b>5</b>
<b>Architectural Overview</b>	<b>6</b>
System Architecture	6
Mobile Application Architecture	8
<b>Module and Interface Descriptions</b>	<b>9</b>
Mobile app	9
App Profile	9
User Profile	9
House	9
Recommendation Algorithm	10
UML diagram for the mobile app and component details	11
Website	20
Responsibilities	20
Recommendation Algorithm	20
UML Diagram for the website and component details	21
Database	23
Tables	24
<b>Implementation Plan</b>	<b>31</b>
Module Breakdown:	32
<b>Conclusion</b>	<b>34</b>

# I. Introduction

In the United States alone, the number of individuals who are age 60 and over is more than 65,000,000 people, and of those - nearly 23,000,000 have at least one identified disability they are living with. Taking this into consideration, there are thousands of Assistive Technology (AT) products that can enable persons with disabilities and those experiencing aging-related limitations to be more productive and self-sufficient in daily activities. These items range from simple to complex, inexpensive to costly and commercially available to customized. Devices can be used at home, work, school, or in the community to reduce barriers, enhance participation and increase or regain independence.

Assistive technology devices help people to **compensate** for lost function. For example, a flashing light doorbell alert can be used by a deaf person to let them know that someone is at their door, or assistive technology can be used to **enhance** and support a declining ability – installing grab bars in the bathroom so that a person with poor balance can use the toilet without falling. High tech wheelchairs with sophisticated controls allow people with paralysis to be independently mobile and even drive vehicles. Small speech generating tablet devices produce speech for people unable to talk as a result of brain injuries, strokes, cerebral palsy and autism.

Market wise the Assistive Technology market is expected to reach \$26 billion by 2024 which is almost doubled in size compared to \$14 billion in 2015. There is a great need for these devices in people's homes as well as a consumer need for learning what's available and useful to the person who needs it.

Our sponsor is Northern Arizona University's Institute for Human Development (IHD). IHD is one of NAU's oldest Institutes and focuses its research, teaching, training/technical assistance initiatives, community service and dissemination efforts on issues that affect persons with disabilities across the age span. As a federally designated University Center for Excellence in Developmental Disabilities, emphasis is placed on advancing positive attitudes, universal access and full inclusion in all aspects of life for people with intellectual and developmental disabilities.

Leading the IHD team of more than 45 staff members is IHD Executive Director Kelly Roberts Ph.D., and Jill Pleasant, MA, OTR/L Associate Director. Dr. Roberts is a researcher, tenured professor and has many years' experience with assistive technology. Ms. Pleasant has an extensive occupational therapy background specializing in assistive technology. IHD is also the home to the Phoenix based Arizona Technology Access Program (AzTAP), Arizona's designated Assistive Technology Act Program. Their staff members are also contributing AT content area expertise to this project centering on matching product knowledge to consumer needs.

A quick Google search for "assistive technology" will yield pages of results, but deciding on a specific product that matches a particular person's needs related to their type of disability or

limitation, what they want to accomplish, where they will use it and their budget is complicated. Individuals with disabilities, older adults and family members are often unfamiliar with the resources available to obtain assistive technology information and support.

Summary of problems:

- There are over 65 million Americans aged 60+ (about 20% of the total population).
- 23 million of them (about 35%) have at least one disability.
- Thousands of Assistive Technologies devices exist.

With this information, it has led our team to work with our sponsors to create a solution that involves building a cross-platform mobile application which will help these individuals find the Assistive Technology that will best suit their needs. This mobile app will pair individuals with devices tailored to their specific difficulties and be able to offer them tips and resources so that they can be best prepared to live their life to the fullest extent.

Following this idea, some of the key user level requirements that we are going to have in this application are a method for determining the best fit technology, accessibility options, and the ability to provide advice to the user. With this in mind we also have to consider the functional requirements that describe what our system can do which include having the user be able to provide their general limitations, allowing them to identify their trouble areas in the house/room, having the user create a user profile, and having the app allow the user to save their favorite products they have viewed. Then our performance requirements which will help our system achieve those functional requirements will be to have the app sync to the database regularly, have the app accommodate color blindness and screen readers, and to have the app be able to perform a query or display a timeout in less than 10 seconds. One more thing to consider is the environmental constraints that we may face which mainly includes being HIPAA compliant for user privacy.

Summary of main features:

- Cross-platform app for Android and iOS.
- Recommending assistive technologies to users.
- Central inventory of assistive technology devices.

With all of this in mind the rest of this document will serve as a guide as to how this software will be implemented and the specifics to the architecture of the application. There are many things to consider when designing software and this will provide the reader a better understanding as to our plan going forward.

## II. Implementation Overview

We will be building a cross-platform mobile application that will help guide users to find assistive technologies and helpful information and to pair them with the devices that they need. This application will be a means for helping walk people through the process of determining various solutions for in the home, and recommend assistive technologies that are catered to their specific needs.

Through the use of Django, a website will be created within a web framework that will connect to the database. This website will act only as a means of administrative access to the database. From here, Django will integrate with SQLite that will be the actual database itself. Through the website, administrators will have access to assistive technologies and user information that is stored in this database.

Flutter is the environment in which the mobile application itself will be developed in. The application itself will allow the user to create a profile (and make changes if necessary), will be able to input any difficulties or limitations that they experience in life, as well as identify any rooms that they may be having difficulties in. The application will be able to take this input into consideration and given any recommendations that will assist the user in navigating their environment. The application will also give recommendations on any AT devices that could also assist them. If the application determines that the user can benefit from AT, it will connect to the database and pull up any relevant devices for the user to browse through. At that moment, the user will be able to save any devices that they may be interested in.

Currently, Microsoft Azure acts as the Cloud server that is hosting the database that both the website and mobile app will be able to connect to. Work is currently in place to try to put the database on the AzTap servers in which AzTap would host the database rather than Azure.

### III. Architectural Overview

This section will detail how the system will be built using the tools we decided on the Technological Feasibility document. Those tools include Flutter for the mobile application, Azure for the Cloud server, Django for the web framework, and SQLite3 for the database. We've included diagrams of the architectural overview along with descriptions of each module, starting with the high-level system overview and then looking more at the structure of the mobile application. In the section following this one, we will describe the modules and interfaces in detail.

#### A. System Architecture

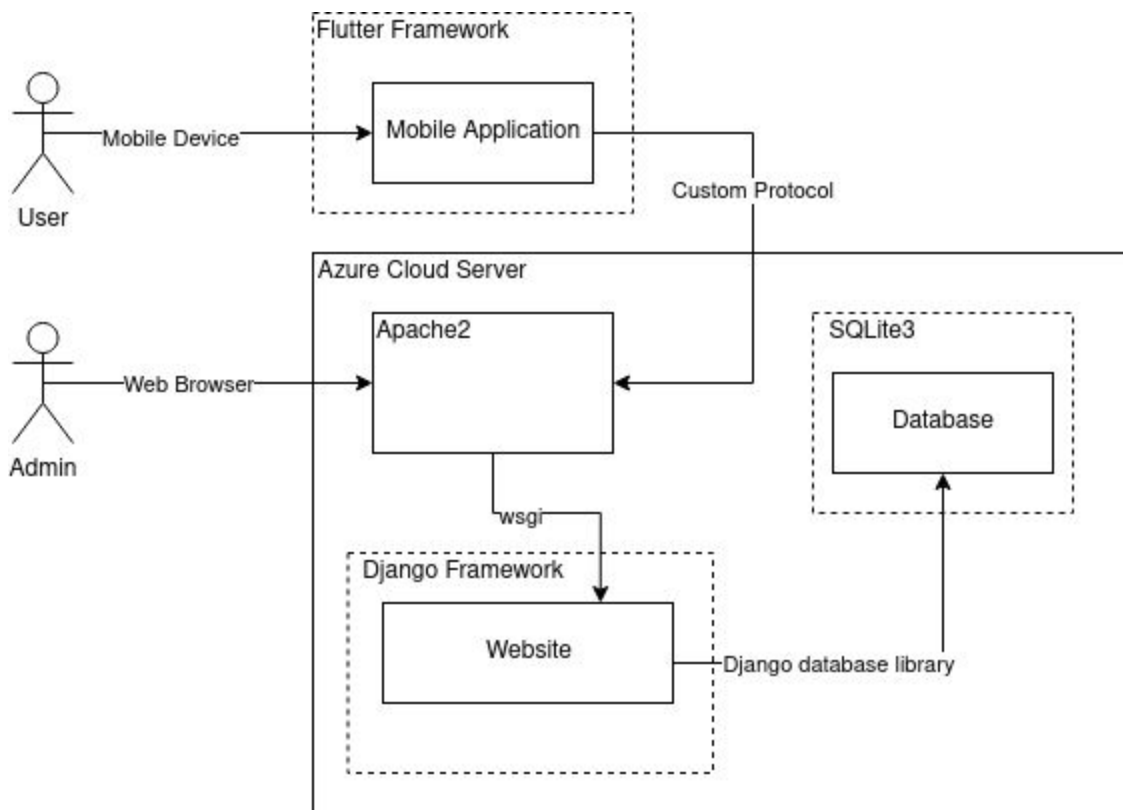


Figure 1: Architectural overview.

The main role of the mobile application is to provide AT device recommendations to the end user. The end user only interacts with the app and not with the website. The app will allow the user to create a profile to save information, but will also have app settings that the user can change to their preferences. Since our end users will have physical difficulties and/or limitations, the app also needs to implement accessibility options. The user can navigate through a virtual representation of a home with different rooms and receive recommendations for AT devices that will help them in those areas. To do this, the app needs to get information from the database but

it won't connect to the database directly. Instead, it will use a custom API we design so that the request is forwarded to the Django framework.

The website provides a front-end for the administrators to manage the inventory of AT devices in the database. Only the administrators will interact with the website, and they will be able to add, remove, and edit AT devices in the inventory; as well as sending messages or surveys to the end users via the database. The website is built in the Django framework which has built-in libraries to interact with the database.

The database only interacts with the website via Django. The database will inventory the AT devices, messages, surveys, user profiles, and administrator profiles. No SQL statements will be used, and instead everything is managed using the website.

The Cloud server provides a home for both the website and the database. We are using an Ubuntu server as the operating system. The software used to serve the website is Apache2 which serves the Django-based website using the module WSGI. Apache2 will not handle any requests itself and will instead forward all requests to the Django-based website.

The communication between the mobile application and the website uses a custom protocol we developed for secure communication which relies on a unique ID and key pair which are given to the app by the website. Initially, the app has no ID or key. When the app is first run, it will ask the server for a unique ID and key pair. The server generates this pair, saves it in the database, and assigns it to the app. After receiving the pair, the app will use it to send a request back to the server for verification that the pair is correct. After receiving an acknowledgement from the server, the app will store the pair into its permanent storage. Every communication between the app and server then requires the unique ID and key pair for authentication.

The overall architecture is a server-client relationship where there is one server and many clients. The custom protocol we developed between the app and server are based on the TCP 3-way handshake protocol for computer networks.

Since the mobile application is the main component of the project, next we will look more into its architecture.

## B. Mobile Application Architecture

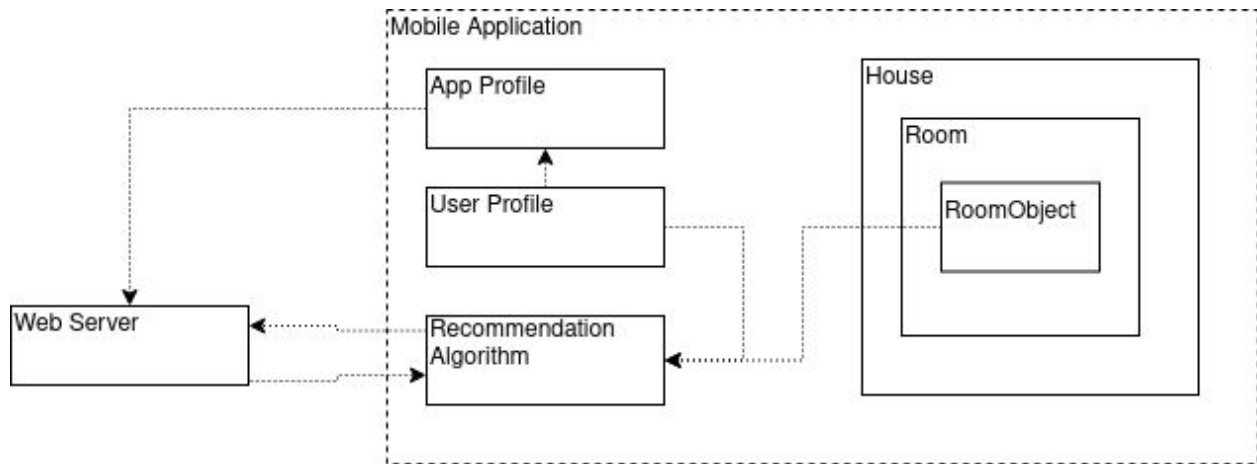


Figure 2: Mobile application overview

The overall architecture of the mobile application is shown above in Figure 2. The app has its own profile to store the settings, configuration, and the user profile. The app profile connects to the website to acquire its unique ID and key pair as described in the previous section using HTML and our custom protocol. The user profile is used to keep track of user preferences and general user limitations used by the recommendation algorithm when finding recommended AT devices. The user navigates a virtual house which is made up of individual rooms. A room has objects or areas inside the room that the user can select. After selecting an object in the room, the app can provide AT recommendations to the user using the information from the user profile and the location in the house, and sending this to the web server. The web server searches the database of AT devices for entries that match the user's limitations and location in the house, and sends them back to the mobile app to show the user.

The next section will provide more details about the components described here.



## IV. Module and Interface Descriptions

In this section, we provide a detailed design description for each module mentioned in the previous section which includes the module's responsibilities, UML diagrams of the classes, and descriptions of the public interface. The next section will discuss the plan for implementing these modules.

### A. Mobile app

The mobile application is being developed using Flutter which provides built-in API's and functionality for both Android and iOS. It serves as the interface for the end user. A diagram of the mobile app framework is shown in Figure 2 of the previous section. Overall, the main purpose of the mobile app is to provide AT recommendations to the end user based on their physical limitations and area of a room in which they could benefit from assistance.

#### 1. App Profile

The App Profile is responsible for storing the app configuration and settings in permanent storage. The configuration includes the unique ID and key pair that the app uses to securely communicate with the server. It also includes a setting for notification preferences for the user to disable notifications from the app. If the user chooses to create a User Profile, the App Profile will be responsible for providing this feature to the user, and to store the information in permanent storage. The app will also provide the option to users to enable the following accessibility features.

- a) Colorblind modes
- b) Text-to-speech
- c) Speech-to-text

#### 2. User Profile

The user will have an option to create a user profile that stores general physical limitations, app preferences, favorite AT devices, and past recommendations. The user can also choose to delete their profile or to create a new one. If provided, the general physical limitations are used by the Recommendation Algorithm when looking for recommended AT devices for the user.

#### 3. House

The "House" does not exist as its own class in the app, but rather as a symbolic grouping of the rooms in a house. The "House" is Flutter's default built-in view that is displayed when the app is run. When this view is loading, it loads the Rooms of the

virtual house for the user to interact with directly. A further division of a Room is the RoomObject which exists as another layer of organization for more specific AT device recommendations. A "House" can have many Rooms and a Room can have many RoomObjects. An ATDevice is associated with at least one Room or RoomObject but can be associated with both. This provides a lot of flexibility for organization of AT devices.

#### 4. Recommendation Algorithm

This algorithm is responsible for providing a list of recommended AT devices that may benefit the end user. When the user has selected a RoomObject to see recommendations, the mobile app compiles a list of criteria to send to the web server. Alternatively, the user can request recommendations at the Room level. The criteria for the algorithm includes any physical limitations that were provided by the user in the User Profile, as well as the selected Room or RoomObject. Since each ATDevice in the database inventory is associated with physical limitations and Rooms and/or RoomObjects, the website can search the database using the criteria to find matches. When these results are returned to the mobile app, this algorithm can display them to the user. Furthermore, the user will be able to view more detailed information about each recommendation and add them to a favorites list which is tracked in the UserProfile.

## 5. UML diagram for the mobile app and component details

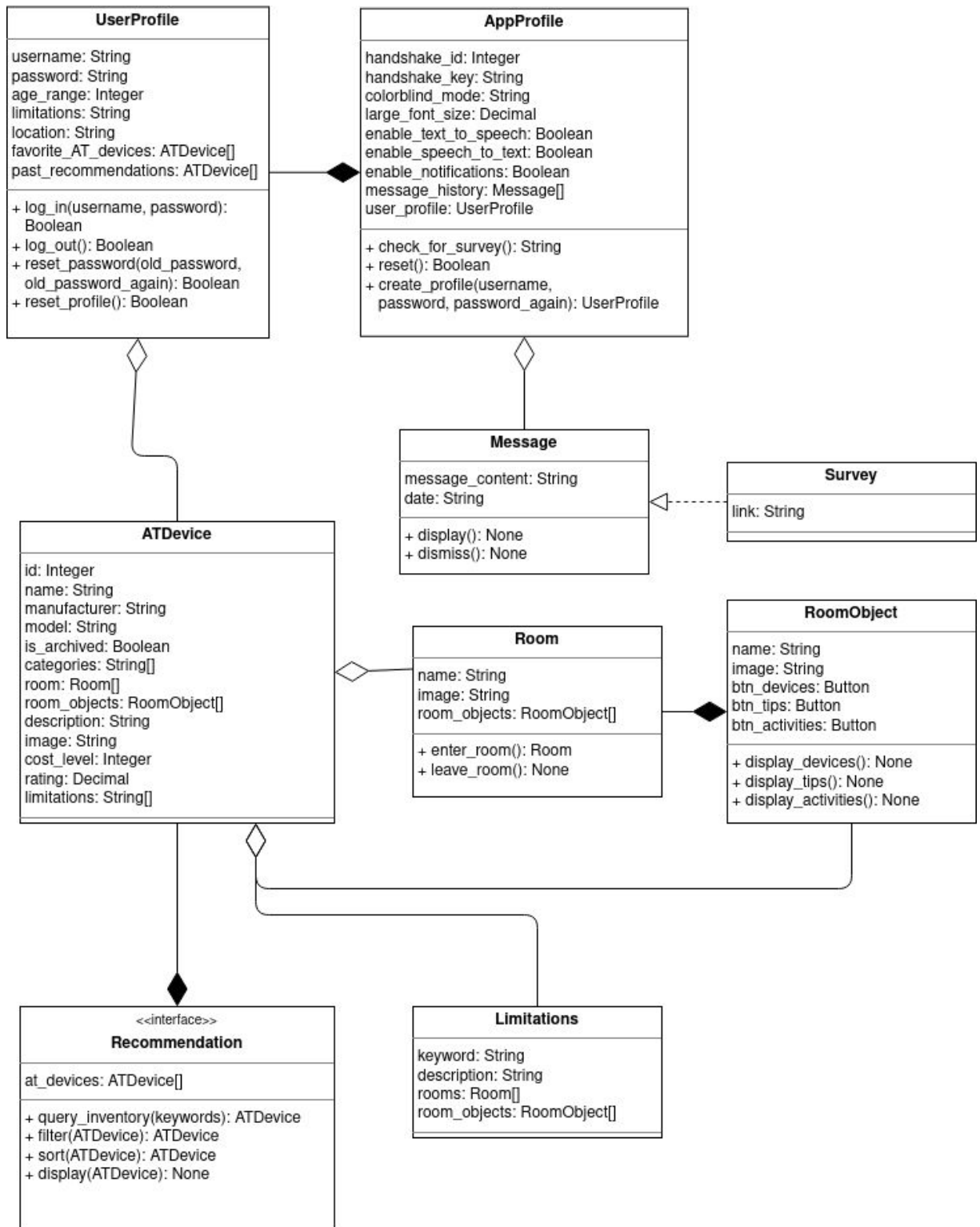


Figure 3: UML diagram for the mobile application

## a) UserProfile

The user profile is needed for better recommendations, to receive communications from the clients, and to give information about local resources.

### (1) Fields

#### (a) username

A String that the user chooses for their profile to log in under.

#### (b) password

A String of at least 8 characters including numbers and special characters.

#### (c) age\_range

An integer representing the start of the age range where the range expands ten years. For example, "60" is used for ages 60 through 69.

#### (d) limitations

The physical limitations that a user has as an array of Strings.

#### (e) location

A String for the user's location at the US state level to comply with HIPAA privacy.

#### (f) favorite\_AT\_devices

An array of ATDevices that the user has indicated to be a favorite.

#### (g) past\_recommendations

An array of ATDevices that the app has recommended to the user already.

### (2) Methods

#### (a) log\_in

Allows the user to log in using their username and password. Returns True when successful, False otherwise.

#### (b) log\_out

Allows the user to log out of their profile. Returns True if successful, False otherwise.

(c) reset\_password

Resets the user's password using their old password and their new password entered twice. Returns True if successful, False otherwise.

(d) reset\_profile

Allows the user to reset their profile to erase the stored information and start new. Returns True if successful, False otherwise.

b) AppProfile

The app profile stores the user's preferences and application settings on the device.

(1) Fields

(a) handshake\_id

An integer that represents the profile's unique ID for secure communication with the server. The default value is "0" before it receives its assignment from the server during the initial handshake.

(b) handshake\_key

A String of a secure key given to the app by the server during the initial handshake. The initial default value is "0" until it is assigned its key by the server.

(c) colorblind\_mode

A String to indicate the user's selected mode for colorblind Accessibility.

(d) large\_font\_size

A decimal value to represent the relative size of fonts in the application as chosen by the user.

(e) enable\_text\_to\_speech

A Boolean to enable or disable the text-to-speech Accessibility option.

(f) enable\_speech\_to\_text

A Boolean to enable or disable the speech-to-text Accessibility option.

(g) enable\_notifications

A Boolean to enable or disable app notifications per user preference.

(h) message\_history

An array of class Message to store the messages received from the server.

(i) user\_profile

An instance of a UserProfile created on the app.

## (2) Methods

(a) check\_for\_survey

Communicates with the server to see if a survey is available. Automatically is scheduled on an interval of once per day.

(b) reset

Resets the app profile back to the initial state and erases any stored information.

(c) create\_profile

Creates the instance of the user profile to be used on the app.

## c) Room

A Room represents a room in a house. This allows us to organize AT recommendations based on the room in a house they are applicable to. A Room consists of multiple RoomObjects which provide an even finer level of organization for AT devices.

### (1) Fields

(a) name

A String to represent the common name of the room in a house.

(b) image

A small and simple representation of the room that is easily identifiable on small screens.

(c) room\_objects

An array of class RoomObject for objects and interactables inside a Room.

## (2) Methods

### (a) enter\_room

Entering a room changes the display in the app to show the RoomObjects for that Room.

### (b) leave\_room

Leaving a room removes the view from the app to return to the previous view which is the representation of the house.

## d) RoomObject

A RoomObject is an interactable object inside a room of a house, for example the kitchen sink. A RoomObject helps to further narrow down an AT device recommendation from the room level.

## (1) Fields

### (a) name

A String representing the name of the interactable object in the room.

### (b) image

A small and simple image to represent the interactable object. It should be easily identifiable on mobile devices.

## (2) Methods

### (a) display\_devices

This is associated with a button or hotspot in the RoomObject view of the app which will send a query to the server and change the view to display the recommended AT devices.

### (b) display\_tips

This is associated with a button or hotspot in the RoomObject view of the app which will send a query to the server and change the view to display tips associated with the RoomObject which are not AT devices.

(c) display\_activities

This is associated with a button in the RoomObject view of the app which will send a query to the server and change the view to display activities associated with the RoomObject which are not AT devices.

e) Recommendation

The class Recommendation is a container for ATDevices that are returned in a query that was sent to the server. These are already filtered by the server according to the associated Room and RoomObject.

(1) Fields

(a) at\_devices

An array of ATDevices as the result of a query to the server.

(2) Methods

(a) query\_inventory

The RoomObject class uses this method to query the server for recommendations. It passes along keywords as Strings which are associated with the query so that the server can filter the results for that Room, RoomObject, and user's limitations.

(b) filter

The user may want to apply an additional filter to limit the ATDevices shown based on a specific keyword. The keyword can be from any field of an ATDevice. This method does not modify the results of query\_inventory, but does limit what is displayed.

(c) sort

Allows the user to sort the recommendations based on attributes of the ATDevices such as by price or manufacturer.

(d) display

After the query is completed and filtered or sorted, the RoomObject class can use this method to display the recommended ATDevices to the user in the app.



f) ATDevice

An ATDevice is an assistive technology that helps people with difficulties or limitations live more comfortably in their own home. An ATDevice can be a commercial product or a common object found around a normal household which can assist the user in an uncommon way.

(1) Fields

(a) id

An integer representing the ID assigned to the device by the database management system.

(b) name

A String representing the name of the AT device.

(c) manufacturer

A String representing the company that made the device.

(d) model

A String representing the model number of the device if it has one.

(e) is\_archived

A Boolean to allow for archiving a device without deleting it from the inventory.

(f) categories

A String array of keywords associated with the AT device to help determine its type.

(g) room

An array of Room objects that the ATDevice can be used in for devices that are generally helpful for a room but do not really fit in with a RoomObject.

(h) room\_objects

An array of RoomObject objects to further organize AT devices for better recommendations. A RoomObject is an object or area of a room that an ATDevice can be associated with.

(i) description

A String for the detailed description of the ATDevice.

(j) image

An image to show what the device looks like if an image is available.

(k) cost\_level

An Integer to represent that relative cost. It works in the same manner as restaurant cost ratings where "\$" represents low cost and "\$\$\$" is higher cost. The integer can be used to determine the level and is independent of what is displayed to the user.

(l) rating

A decimal to represent the average rating of the ATDevice by the users.

(m) limitations

An array of Limitations that the ATDevice can help with.

g) Message

A Message is an object that allows the administrators to send messages to the app users. The message is displayed as a notification on the mobile device.

(1) Fields

(a) message\_content

The text to represent the content of the message to display to the user.

(b) date

The timestamp associated with when this message was received. This allows for viewing multiple unread Messages in order.

(2) Methods

(a) display

Displays the content of the Message in a notification on the app.

(b) dismiss

When the user dismisses the message, this method removes it from the device's storage.

h) Limitations

A Limitation is a keyword or phrase that describes a physical user limitation which an ATDevice can help with. These are used in the recommendation algorithm to provide recommended AT devices to the user.

(1) Fields

(a) keyword

A String word or phrase to use when searching for AT devices in the recommendation algorithm.

(b) description

A user-friendly description of what the limitation is so that the user can understand if it describes their situation.

(c) rooms

An array of Room objects to which this limitation can be associated with.

(d) room\_objects

An array of RoomObject objects to which this limitation can be associated with.

i) Survey

A Survey is a subclass of a Message that has an additional field for the link to the third-party Qualtrics survey.

(1) Fields

(a) link

A String containing the Qualtrics URL for the survey. The user will complete the survey in a browser on their mobile device.

## B. Website

The website will be developed through Django that will provide front-end access to the database.

### 1. Responsibilities

- a) Provides administrative access to the database

Acts as a front-end website that will allow administrators to access information stored in the database.

- b) Allows administrators to add, remove, or edit information stored in the database

The website will provide administrators the ability to add, remove, or edit information about the AT devices that are stored in the database. Will also provide administrators with the ability to add additional information to AT devices in the method of adding or removing categories attached to them.

- c) Allows administrators a tool to send and view surveys that have been sent to users

Surveys will be sent through a third party application Qualtrics from the administrators to users. Once users have sent in their responses, they can be viewed through the website.

- d) Administrator management

The website will give administrators the ability to create, delete, or disable administrative accounts.

### 2. Recommendation Algorithm

The website's responsibility for the recommendation algorithm is to use the criteria as provided by the mobile application to search the database for ATDevices that match. Either a Room or RoomObject will be specified as the location to search. If Room is specified, then the search includes all ATDevices which are associated with the same Room, and all ATDevices which are associated with a RoomObject that is associated with that Room. Otherwise, it looks for ATDevices that are associated with the provided RoomObject only. If general user limitations are also provided, the website can narrow the results further by ATDevices with matching limitations. After gathering a list of specific recommendations, it creates a JSON object to return to the mobile application.

### 3. UML Diagram for the website and component details

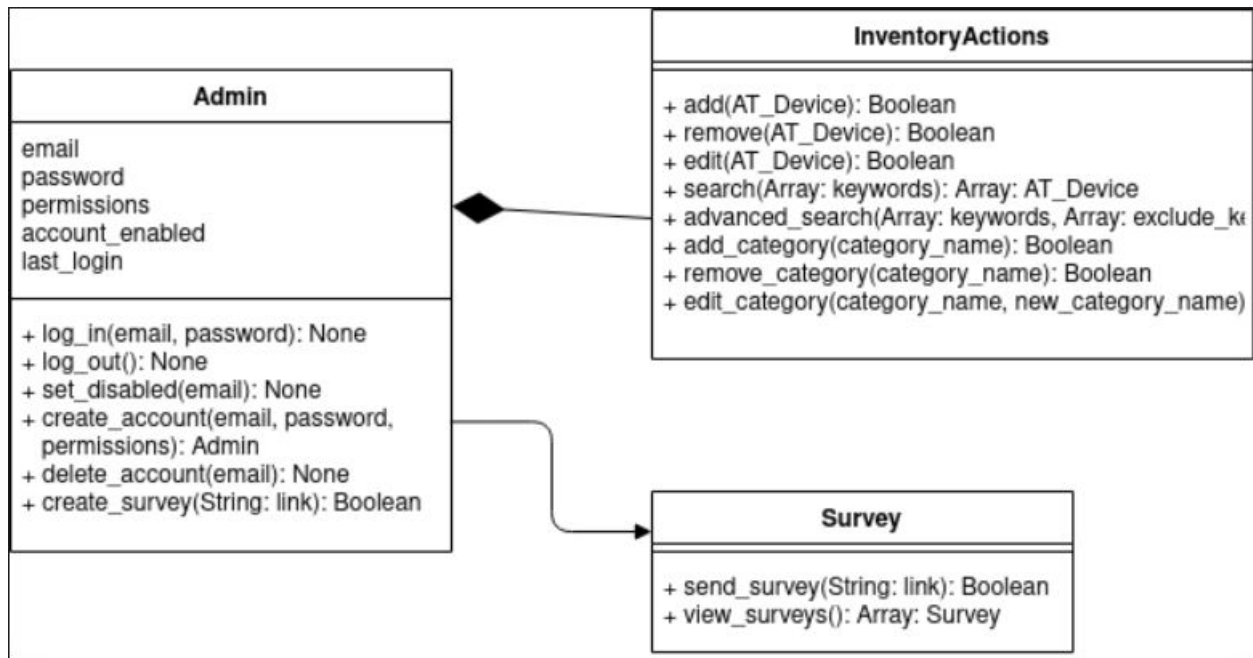


Figure 4: UML diagram for the website

#### a) Administrators

##### (1) Username

A string that the administrator will choose to log in with

##### (2) Password

A string of at least 8 characters including numbers and special characters

##### (3) Permissions

An array or list of boolean values that give administrators permissions on what they can or cannot do within the database.

##### (4) Active/Inactive

A single boolean value that represents whether or not the account is active.

##### (5) Email

A String up to 20 or 30 characters that will store the administrators email address

(6) Logging of activity

An external file that will contain administrative activity within the database

(a) Last Login

A String that will hold the date and time in which the administrator last logged into the website.

(b) Actions

Contained within a text file that will log all of the actions the administrator has taken.

b) Inventory Actions

(1) Add

Being able to insert AT devices into the database

(2) Remove

Being able to remove AT devices from the database

(3) Edit

Being able to edit information about AT devices stored in the database

(4) Search

Provides a variety of ways in which administrators can search for AT devices in the database.

(a) Sort By

Will sort all devices in the database by certain keywords

(b) Advanced Search

Allows administrators to provide specific information and the database will return select devices that meet the criteria

(c) Search by Keyword

Allows administrators to search the database by an inputted keyword

(d) Display Results

Displays the results of all AT devices found in the searching parameters

c) Surveys

The surveys will be sent from administrators via Qualtrics to users. Upon user response, the administrators will be able to view the results.

(1) Send Survey

Allows administrators to push a link to the survey to the user through a notification that the user can view.

(2) View Survey

Once a user has sent in their response, the administrator will be able to view these results via Qualtrics

## C. Database

The database has several responsibilities. It is responsible for storing all of the AT devices and the relevant information to each of them, as well as all of the admin and user profiles. It will have several tables that are able to handle different data to allow the client and the end user to have access to the information that they need.

Figure 5 below shows the relational diagram for the architecture of our database. It displays the various tables that our database contains as well as listing how they will be connected to each other. Following the visual display of our database architecture is an in depth description of each table and the fields that each table contains.

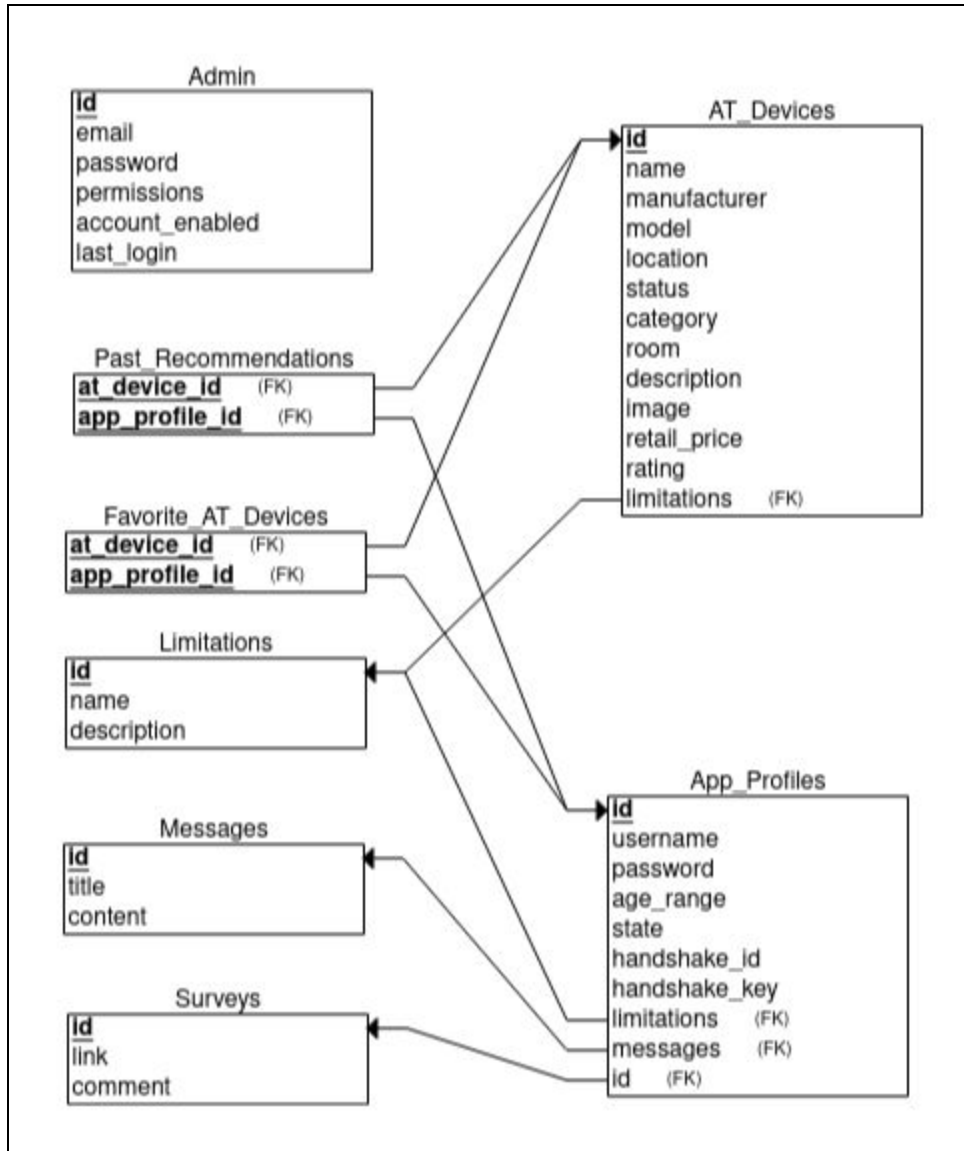


Figure 5: Relational diagram for the database

## 1. Tables

### a) Administrators

The admins table will in part be handled by the web framework and Django. On the database side the admins table will be responsible for storing all of the data that is relevant to each admin account. Admins will have the ability to manage the database and make changes to its contents.



(1) ID

All tables will have an automatically assigned ID that will serve as the primary key for each table. As the ID attribute in each table serves the same purpose this will be the only table it is listed under.

(2) Email

The email will serve as each admin's username. It will be unique to each person and will be required with their password in order to be able to log into the web framework.

(3) Password

The password field will store the password for each admin account. It will be the second half to ensure that admin accounts are secure and will be required along with email in order for the admin to be able to log into their account.

(4) Permissions

Not every admin account will have the same permissions. Some admins will be able to perform actions that not all admins are able to make. This field will store the permissions that are allowed to each admin so that only the correct people are allowed to do certain actions.

(5) Account Enabled

The purpose of this field will be to state whether or not a certain admin account is currently active. It will be a Boolean value and can be changed between the enabled and disabled state as opposed to removing an admin account entirely.

(6) Last Login

The last login field will store the information as to when each admin account last logged in. It will mainly be used for administrative purposes to give a log of what admins are active and when.

b) App Profiles

The app profiles tables will be used to store important information pertaining to each end users account. It will be used to keep limited information on each user that is important to having relevant recommendations that address each user's difficulty areas.

(1) Username

The username field will be the name that each user chooses to use when they log into the mobile application. It will be used along with the password field to ensure a secure login.

(2) Password

The password field is the second part of each user's secure login. When initially creating an account, each user will set their own unique password.

(3) Age Range

This field will be used to store the various age ranges of the end user. In order to protect the user's privacy their actual age will not be stored in the database rather each user will be grouped into a predetermined age range category.

(4) State

The state field will serve as a location for the user. This field will be used to help point users to local agencies where they can get more information on specific difficulties they are having. Like in the case with age range, the user's location will not be stored beyond the state level to help protect their privacy.

(5) Handshake ID

Originally when the user creates an account, the handshake\_id field will be used to give each user a unique id that will be used to ensure a secure initial connection to the database.

(6) Handshake Key

The handshake\_key field will be used along with the handshake\_id field as part of our systems secure handshake protocol when a new user creates an account.

(7) Limitations

The limitations field will be a foreign key pointing to the limitations table which will store information on general categories of limitations that could apply to many users.

#### (8) Messages

The messages field will be a foreign key pointing to the messages table. This table will store messages that are sent to the users from the admins.

#### (9) ID

This specific ID will be a foreign key pointing to the surveys table.

### c) AT Devices

The AT devices table will be used to store all of the information that is important and relevant to each AT device that is listed in the database. It will contain information that can help the user know what devices might be right for them as well as information that will be used in the app's recommendation algorithm when providing recommendations to the user.

#### (1) Name

The name field will store the name of each AT device in the database.

#### (2) Manufacturer

The manufacturer field will list what company makes each of the AT devices

#### (3) Model

The model field will store the model of each specific AT device.

#### (4) Location

The location field will store the physical location at which each AT device is currently stored at. Should a user wish to get one of the AT devices, this field will tell them where they can do so.

#### (5) Status

The status field will be used as a means of archiving any devices that were once in the database, but may not be available or recommended. This is to prevent any deletion of information.

(6) Category

The category field will state what category the AT device falls under. For example, some devices are used to help with mobility difficulties and others are for more general purposes. This field will keep track of that information.

(7) Room

The room field will store the room(s) that the AT device can be used in. Some AT devices have a general use and can help with various difficulties while others are more specialized and are meant for a specific room in the house.

(8) Description

The description field will store a description of what each AT device does, and what it can be used for along with any other relevant information of the device that might be useful for the end user.

(9) Image

The image field will display a graphical image of what each AT device looks like.

(10) Retail Price

The retail\_price field will display the current price for each AT device.

(11) Rating

The rating field will display the rating of each AT device has received from previous people who have used it.

(12) Limitations

The limitations field will be a foreign key that will point to the limitations table and will list what limitations each AT device is designed to help with.

d) Limitations

The limitations table serves to list the various limitations a user might have and will be used in the app profiles and AT Device table to keep the limitations in a centralized place.

(1) Name

The name field will list the name of each limitation.

(2) Description

The description field will describe in more detail what each limitation means and provide characteristics that each limitation often has.

e) Messages

The messages table will be used to store the messages that will be created by admins and sent out to users, notifying them of various important information such as the addition of a new AT device in the database.

(1) Title

The title field will be used as the title of the message when it is sent out to users.

(2) Content

The content field will contain the main information of each message.

f) Surveys

The surveys table will serve to store links to the external survey site Qualtrics that the admins are using to give to the users as a means of feedback on the mobile application as well as any other information that admins wish to collect.

(1) Link

The link field will be used to store the link to the external survey that will be created in Qualtrics

(2) Comment

The comment field will store a brief description of the purpose of each survey.

g) Security

The security of the databases will not have a specific field or data stored in the database but is more heavily related to external functionality.

## (1) Encryption

The database and all information contained within it will be encrypted using built-in encryption functionality of SQLite to protect private information about users and administrators.

## (2) Backups

The database will utilize a backup system in the event that data is lost or corrupted.

## V. Implementation Plan

Development on the mobile application, website, and database began on January 24th, 2021 and is estimated to have a working version by March 5th, 2021. The March deadline will be the estimated date in which a working version of the application is up and running and each of the modules listed below will be in working order. The remainder of March and April will be spent on UI, design, and adding finishing touches to the application.

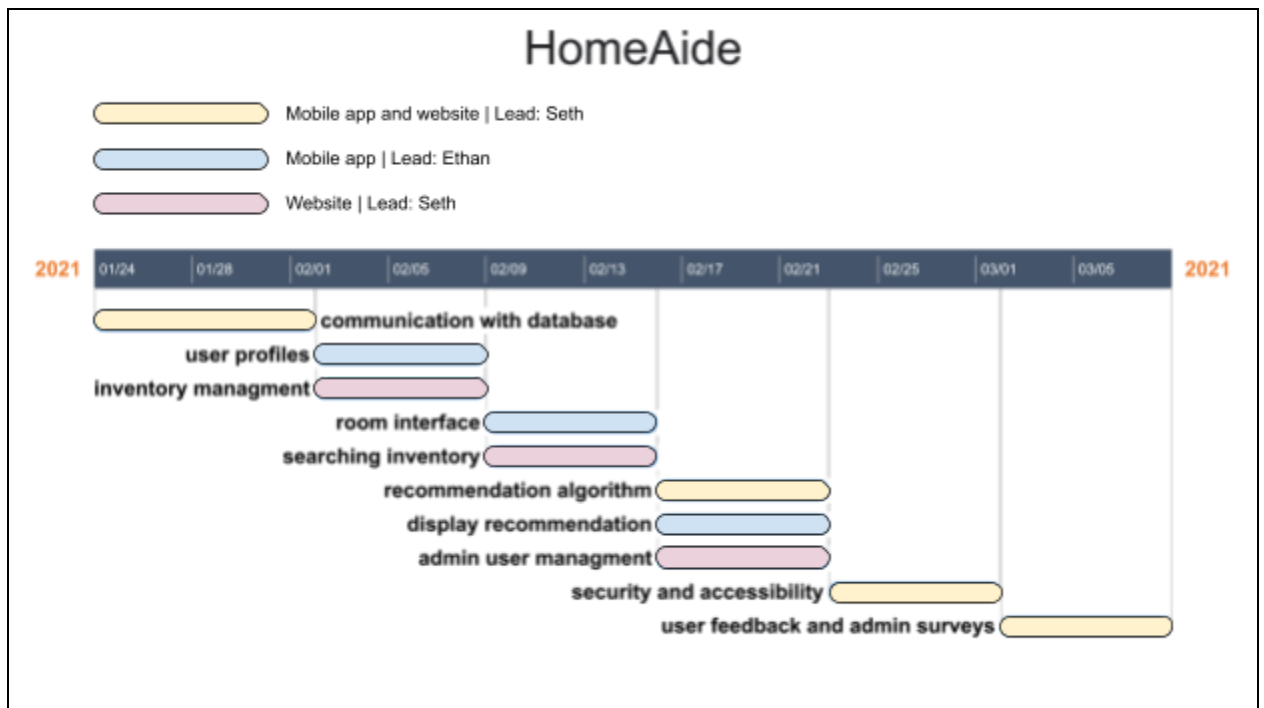


Figure 6: Gantt chart for Spring 2021 semester

The Gantt chart in Figure 6 shows the overview of our implementation plan. We organized the major modules so that we can work on both a mobile app task along with a website task in the same week for maximum efficiency. While everyone in the team is contributing, Ethan is leading the tasks related to the mobile app and Seth is leading the tasks involving either the website or that span both components.

## Module Breakdown:

- Communication with database
  - The initial focus at the beginning of development will be making sure the website and the mobile application can access the database and be able to pull data from it.
- User profiles
  - This step will focus on allowing users to be able to make profiles. These profiles will include their username, passwords, and a list of any limitations they may be experiencing. This data will also have to be able to be stored in the database.
- Inventory management
  - Focus here will be spent allowing administrators to manage all of the AT devices stored in the database. They will be able to add, remove, or edit any devices contained in the database.
- Room interface
  - This will be the core functionality of the application that will allow a user to navigate around a virtual home and select rooms. From here, they can be given any advice or recommendations if the user may be having difficulties in that room.
- Searching inventory
  - Additionally functionality will be added to the inventory in which the administrators will be able to search for information contained within the database. Searching parameters can look by name of the device, the type, or even what limitations they could assist with.
- Recommendation algorithm
  - This is the algorithm that will consider user limitations and room difficulties to provide any AT devices or general advice to assist the user. It is crucial that the algorithm will be able to access the database and pull information about the AT devices out of the database and display to the user. The mobile application provides criteria to the website for aggregating recommendations to return to the mobile app. This algorithm spans the mobile application and the website since the app provides the criteria and the website can search the database.
- Display recommendation
  - This is more of a sub-module of the above in which advice or any recommended AT devices will be displayed. This part must be adaptive in how it is displayed depending on how much information the user wants to see at a time.



- Admin user management
  - Will provide the administrators with the ability to be able to manage other administrators. They will be able to determine the level of access each administrator will have to the database.
- Security and accessibility
  - Encryption functionality and even administrative activity will be developed to help secure the database.
- User feedback and admin surveys
  - The application will have integration with Qualtrics which will act as a third party application that will provide surveys to users about their experience. These responses will be sent back to the clients where they can evaluate the data from these surveys.

## VI. Conclusion

Aging is inevitable, but the goal is to age well by having access to resources, services, and support including a continually expanding number of assistive technology (AT) products designed to promote safe independence. The problem is that although there are thousands of AT devices on the market, many older adults and their family members do not have the knowledge or resources to connect with these devices or even realize they exist. This is where our solution comes in; we aim to create an easy to use mobile application that will guide individuals towards information and AT targeted to their specific needs to help maintain or achieve more independence especially for routine tasks that they do at home on a daily basis.

This document was created in order to introduce and outline the problem at hand and to provide insight towards the software designs that will implement this solution. This document also helped us explain specifically what steps are being taken to ensure our solution is being met through our implementation plan. We showcased the big picture that we are after with our implementation overview and the technologies that will be used to reach these goals. Our architectural overview outlined the specific components of our software so that we can see how each feature is implemented. Most importantly our module and interface descriptions broke down each component on a deeper level so that we left no stone unturned when it comes to understanding every aspect of our system. After everything we've considered and discussed in this document, we feel confident about the state of our project and know we are on track to delivering a promising product to our client.