GNG 2101
Design Project User and Product Manual

Getting Started with Our TTS Application

Submitted by :
Accessible Speech Group C14

Sirjan Singh Rekhi	300019180
Karim Shekh-Yusef	300160453
Dragi Plakalovic	300109973

4/11/2021

University of Ottawa

Table of Contents

List of Figures

List of Tables

List of Acronyms and Glossary

Table 1 : Acronyms

Acronym	Definition
AAC	Augmentative and Alternative Communication
GUI / UI	(Graphical) User Interface
TTS	Text To Speech

If a word is bolded, it appears in this table.

Table 2 : Glossary

Term	**Acronym**	**Definition**
Augmentative and Alternative Communication	AAC	An easy way to talk without actually talking, typically making use of computer software.
(Graphical) User Interface	GUI / UI	Refers to the easy to use graphics the user interacts with.
Text To Speech	TTS	Refers to software that interprets text and recites it by speaking it out loud.

Introduction

This document's purpose is to be used as a user manual, or documentation of our project, and is assumed to be used as such. It may prove useful for those who desire to learn how to use our application, or to gain some insight as to where to start their own projects / endeavours, as it will outline the various workings of the application, and the development process.

Overview

The problem we are attempting to alleviate is an anxiety disorder called selective mutism. Selective mutism causes the person who suffers from it random instances of mutism. Our client suffers from selective mutism, and their current solution for communicating with others entails either using sign language or writing out their message on paper. Therefore, our client wanted a solution that would speed up the process, and is easily accessible, leading to the choice of an **AAC** application.

While **AAC** applications are an existing product, we found that many of these products were far too cluttered, and far too expensive, rendering a needed solution behind a steep paywall and many complex functions. Overall our product aimed to achieve a relatively simple layout while also being able to maintain the desired functionality.

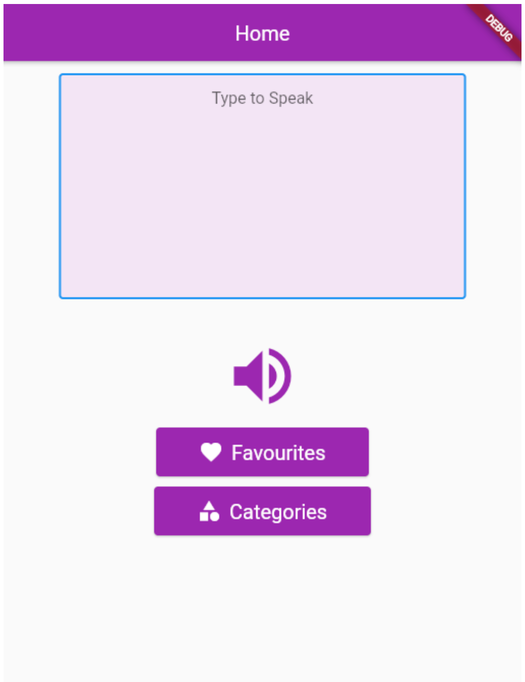


Figure 1 : Application home screen

This solution is an app based solution that is able to recite text through text to speech, allowing the user to make edits as they go along. The user is able to access all these functions through our **GUI** and various buttons which aim to simplify the process.

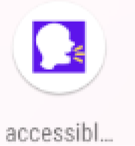
Getting Started

Set-up Considerations

- The application was designed for Android 9.0, as per the Client's device specifications. The app is supported on future android versions, if the client were to update.
- The app will run on Android 9.0 and up.
- The app may be downloaded from the Play Store or installed through an APK
  - For the App to be installed through and APK, following instructions should be followed: (Primarily, third party installation should be enabled)
  - https://www.javatpoint.com/how-to-install-apk-on-android#:~:text=Copy%20the%20downloaded%20APK%20file,tap%20on%20it%20to%20install.
- The client will be given access to the Play Store Link or APK when available

User Access Considerations

- Once installed, app can be accessed through the Application icon on the Client's Mobile Phone

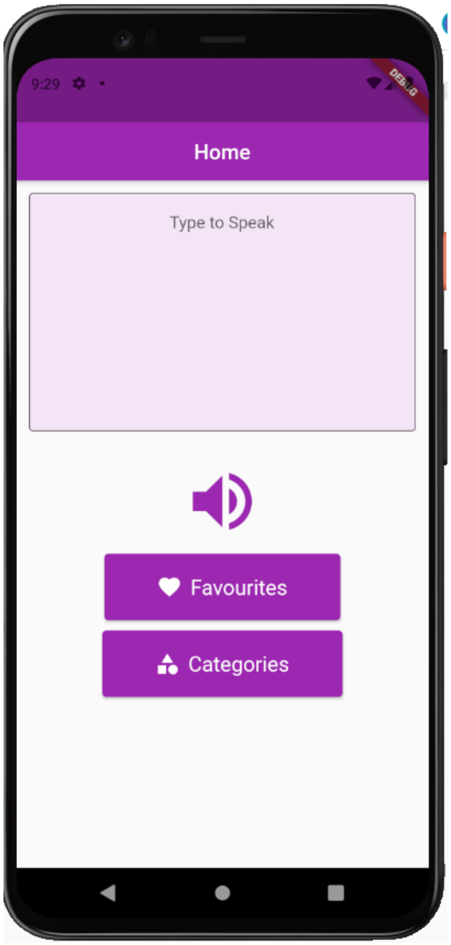
- Icon: 

- This icon may be placed anywhere as per client’s preferences.
- It is however recommended to place this icon on the client’s most visited home screen, and in a spot that is easily accessible (for example, bottom of the screen as opposed to the top, and not surrounded by other applications to prevent miss-clicks/confusion in case of disorientation)

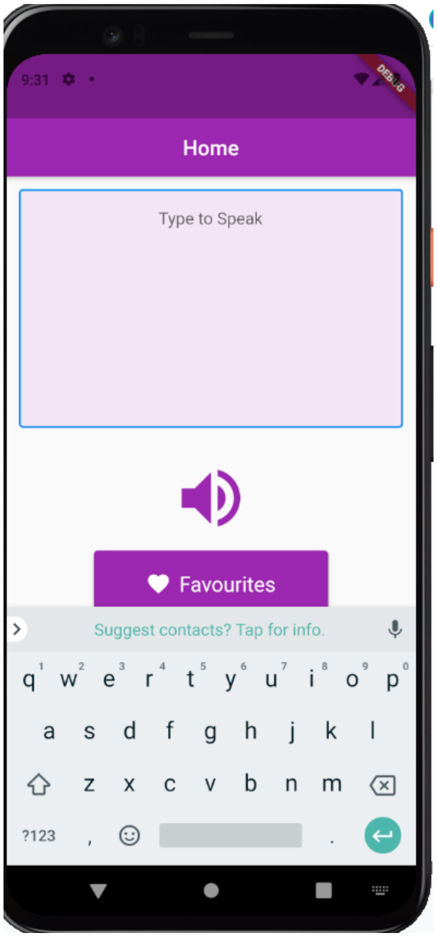
Accessing the System

- Application system must first be installed through the means mentioned under “Set-up Considerations”
- After installation, client may access the application by tapping on the application’s icon

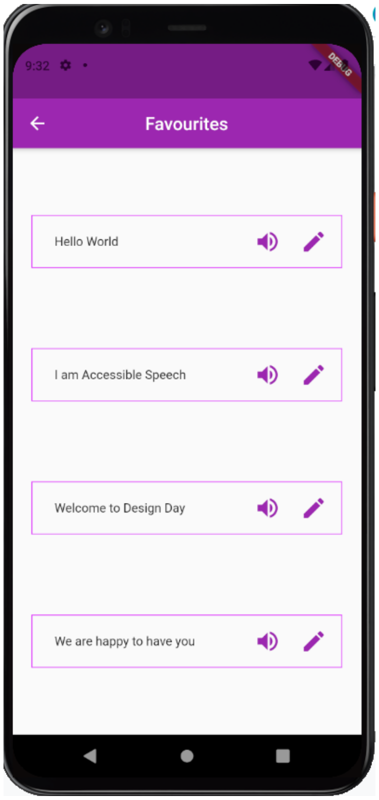
System Organization & Navigation

- System is organized into 4 main sections (Specific features found under “USing the System”:
    - **Home**
    - 

- “Type to Speak” is a Text field where client may tap to open keyboard:



- **A favourites** Section with Client’s Favourite Phrases:

- 

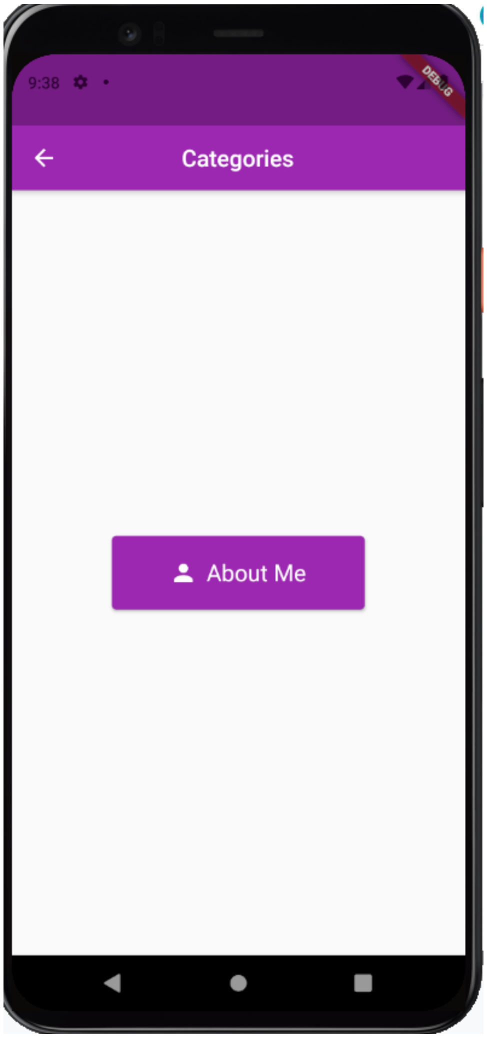
Hello World

I am Accessible Speech

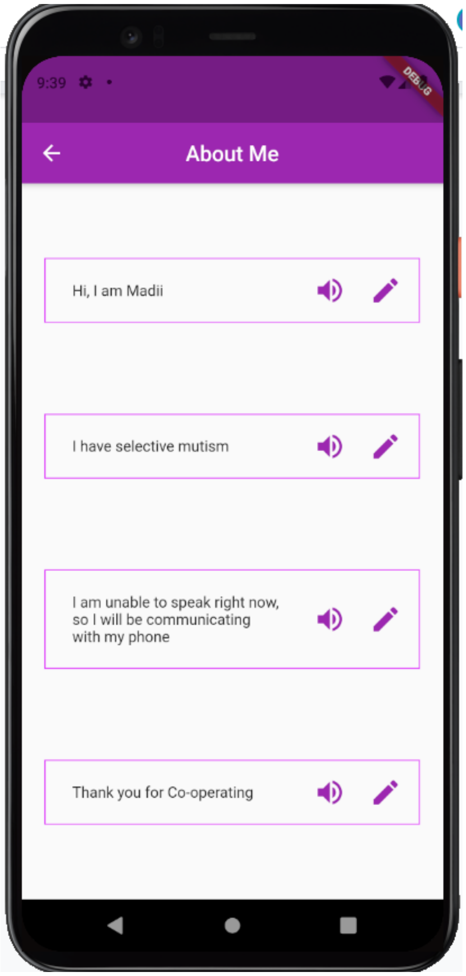
Welcome to Design Day

We are happy to have you

- A Category Page Reachable through Categories Button:

- 

- The Specific page For Chosen Category:



About Me

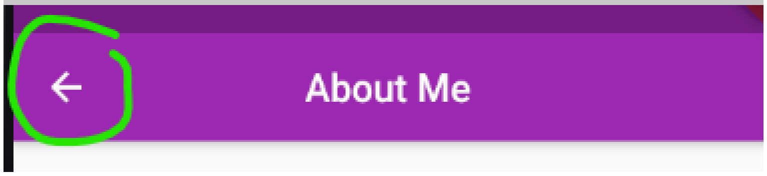
Hi, I am Madii

I have selective mutism

I am unable to speak right now, so I will be communicating with my phone

Thank you for Co-operating

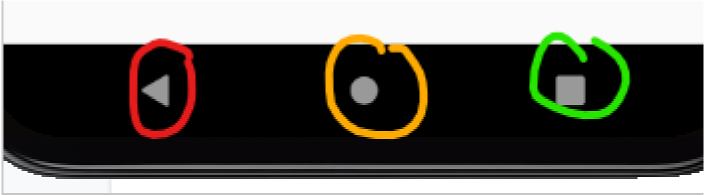
- ○
- ○ Client may navigate to the Previous page (from any page) by using this button:



- ○

Exiting The System

- To exit, client may use the following buttons:



  - ○ Red to go back to a previous page or home screen, depending on the client's current page in the application
  - ○ Yellow button can be pressed anytime to get to the home screen

- Green button an be used to return to application after exiting using the above methods, or to close the application through the standard Android application closure procedure

Using the System

Use of our application is fairly straightforward. The application hosts a few key features as of the latest revision, which include **TTS** playback, favourites category, and a general category button.

Home Screen

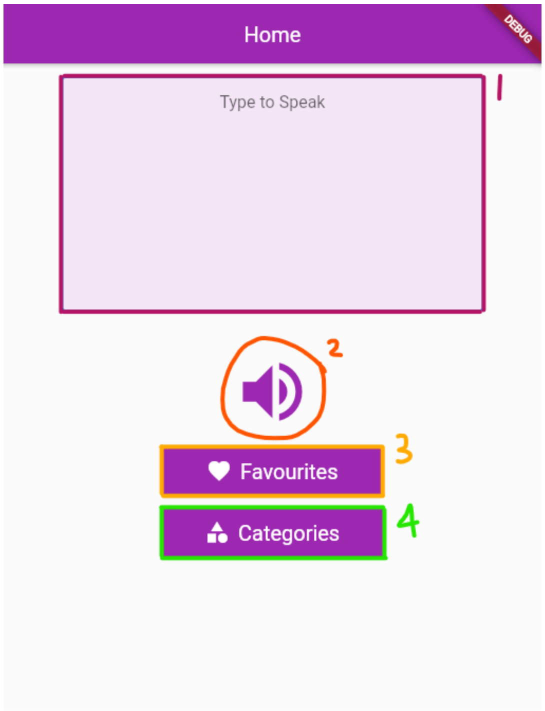


Figure 2 : Home screen and labelled buttons

There are various functions that can be performed from the home screen. #1 in figure 2 is the typespace, an area to enter text. #2 on the home screen is the playback button, which speaks out the text within the typespace. #3 is the favourites button, which enters the favourites screen. Similarly, # 4 enters the categories screen, from which the user can select from a variety of categories.

Playback

Text to speech playback can be achieved in a variety of ways. The best way of performing text to speech is by entering the desired phrase in the typespace at the top of the home screen as is shown in figure 2 in the area highlighted as #1. Once the desired phrase is entered, playback can be achieved by pressing #2 (also depicted in figure 2). Further playback methods are covered in the categories and favourites section following this section.

Categories

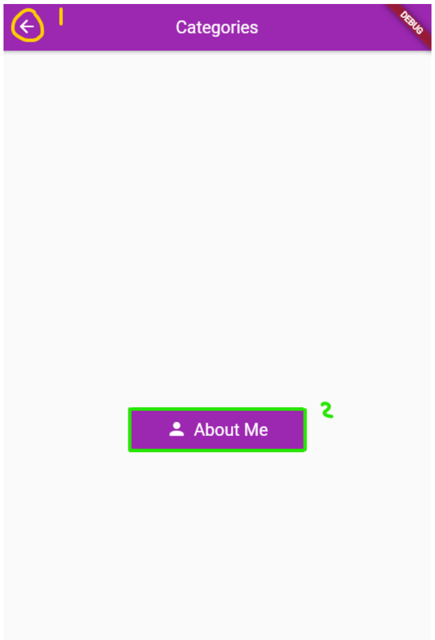
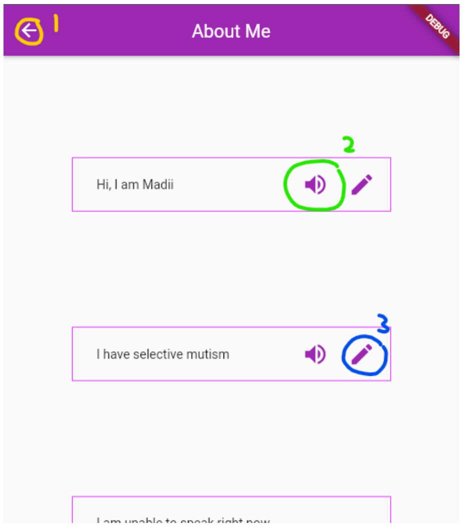


Figure 3 : Categories screen, with labelled buttons

Once entering the categories screen, the user is prompted to either select a category (figure 3, #2) or return to the home screen (figure 3, #1).



Hi, I am Madii

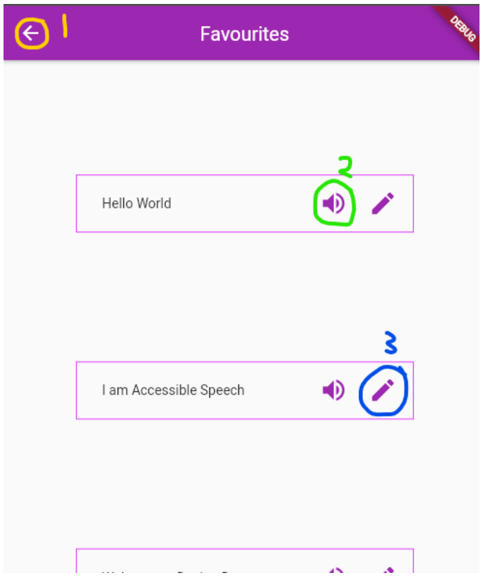
I have selective mutism

I am unable to speak right now

Figure 4 : Selected category, with labelled buttons

Upon selecting a category from figure 3, the category screen will open along with some saved phrases within the category. Currently, the only working functionality includes button 1 in figure 4, which returns to the previous screen and button 2, which plays the saved phrase. Had the desired functionality been achieved, button 3 would allow the user to add the phrase to the typespace in figure 2, allowing for quick edits / save.

Favourites



Hello World

I am Accessible Speech

Figure 5 : Favourites screen with labelled buttons

The favourites screen is not too dissimilar from the categories section, however, while the categories requires the user to select the categories button followed by the desired category, the favourites button on figure 2 sends the user directly to the **UI** depicted in figure 5 , which incorporates the same **UI** features as the categories.

Currently, the only working functionality includes button 1 in figure 5, which returns to the home screen and button 2, which plays the saved phrase. Had the desired functionality been achieved, button 3 would allow the user to add the phrase to the typespace in figure 2, allowing for quick edits / save.

Troubleshooting & Support

Known Error Messages or Behaviours

- The voice that speaks out the sentences may change in the following way without client's explicit input:
  - Pitch may change
  - Speed of diction may change

    - Volume may change
- Language is always English - US, but Flutter does have other supported languages. Even though a change in language has never been observed, it is possible due to the presence of other languages.

Special Considerations

- The editing saved phrases feature by adding to the typing space feature is not developed yet, so typing is needed.
- If some core phrases are needed by the client, they can be added before the adding phrases by client functionality is implemented

Support

- Voice based bugs can be mitigated by adding voice setting controls.
- Client may request support/phrases through emailing at srekh085@uottawa.ca

Product Documentation

TTS Application

The final prototype was built with a few key factors in mind. Those being, simplicity, functionality, and time. As we only had one developer, a lot of the final decisions were made based on time and simplicity, leaving some things to be desired with regards to functionality. Throughout designing the application, we relied heavily on software tools such as android studio and flutter to build up the framework of our application. Overall we decided to first prioritize the text to speech functionality, and then work towards the implementation of categories, typing in text, **UI** changes, and eventually editing and saving. These decisions were made mainly with time constraints in mind.

Equipment List

- Android Studio - https://developer.android.com/studio
- Flutter development kit - https://flutter.dev/
- Any PC
- Android phone (optional)

Instructions:

1. The app was developed using Android Studio through the Flutter Plugin.
2. A simulated Environment for Android 9.0 was used to develop and test the application
3. The main Programming language used for the application is Dart

4. The framework used for developing the application is Flutter
5. Flutter provides native libraries and widget based development tools to develop features of the application, adn has support for routing between widgets (pages) natively.
6. The API used for text to speech and speaking out saved phrases is called Flutter TTS (text to speech)
7. The Flutter TTS api provides different tone, pitch, accent and language options, as it is backed by Google.
8. Flutter is also developed by Google, adn the presence of Flutter TTS was our main reason for going with Flutter as the development tool.
9. Android Studio can be used to generate an "App Bundle" for the created application, which will help us upload it to the play store for the client top download.
10. The App Bundle is associated with a key which is needed by the developers to add further updates for the application to the Android Play Store

Testing & Validation

Table 3 : Core assumptions tested

#	**Core Assumption**
1	Clicks for sentence output (Non-Favourites, +1 to open App) is <number of words in sentence.
2	Clicks for sentence output (Favourites, +1 to open App) is Minimum 2, and Average 4(accounting for scrolling through options).
3	Time taken for sentence output (Favourites, after app is opened) is <= 10s
4	Time taken for sentence output (Non-Favourites, after app is opened) is <= time taken to type sentence at 35 words per minute
5	The ranks at least a 4 on a 0 - 5 scale of usability. (5 being incredibly easy to use, and 0 being incredibly difficult) (measures user friendliness, ease of navigation). (Client statement from meeting 1 and 2)
6	Client would like the voice to be similar to the voice sample in the prototype (based on client statement from meeting 2)

Testing of our prototype centered around these assumptions. The first two entries in table 2 measure the clicks required to a sentence output , one favourite, the other non favourite. In the case of favourite phrases, we ended up with 2 clicks, one to open the favourites category, and one to play the given message.

Non favourite messages are a different story, as playback can be achieved in a few ways. One way is by navigating the categories section, requiring a minimum of 3 clicks, one to open categories, one to open a selected category, and a final one for playback. However, non favourite messages can be played via the typespace, outlined in figure 2 #1. This would require the user to, at maximum, click the same amount of letters in the phrase they desire + 1 to play that message, far exceeding the ideal outcome, that being less than the words in a given phrase.

Assumption 3 and 4 test the time in seconds taken for audio playback of a given phrase in either favourites or non favourites. Similarly, favourite phrases will exceed the ideal of 10s to play, while non favourites can vary quite a bit, especially when typing out a word in the typespace, though through testing it was found that for the most part, it exceeded the ideal time too.

Assumption 5 was subjective, and in our interviews the client ranked our prototype **UI** a 4, and we promptly made adjustments to exceed the ideal outcome.

Conclusions and Recommendations for Future Work

Overall lots of new lessons were learned with respect to group work, how difficult it is to develop an app, and how to develop an app that is competitive in the proverbial sea of applications on the market. With respect to future work, the best place to start is to finish more key features, most notably the save / delete / edit features which have yet to be implemented, and then to follow up with more **UI** changes which would be more accommodating and pleasant to look at for the general user.

While the development team was small, we had high hopes for this project, with the potential to include translation features, extra voices, user settings to change **UI**, and much more. With a few more months of time to exclusively work on this project, it wouldn't be unreasonable to expect that the core features would be completed, and the **UI** changes to be implemented.

Bibliography

Appendix I : Design Files

MakerRepo : https://makerepo.com/KarimSY/828.codersgonehome-accessible-speech-group-6

Table 4 : Referenced Documents

Document Name	Document Location / URL	Issuance Date
Android Studio	https://developer.android.com/studio	Aug 2020

Flutter	https://flutter.dev/	Mar 2021
Github	https://github.com/srekh085/Accessible_Speech	Apr 2021
JavaTPoint	https://www.javatpoint.com/how-to-install-apk-on-android#:~:text=Copy%20the%20downloaded%20APK%20file,tap%20on%20it%20to%20install	N/A

Appendix II : Other Appendices