# Ayvu Project Finalization Document

## Objectives:

The main objective of this document is to inform the final status of the so-called Ayvu application for those who are interested in continuing our work. The work extended from 10/15/2024 to 02/20/2024, with a ~14 weeks of labor.

## Authors:

- Dr. Edson Mitsu – Coordinator;
- Felipe XXX – Dev Team Member;
- Gustavo YYY – Dev Team Member;
- José Henrique Zeferino – Dev Team Member;
- Pedro Daniel Bomtempo Medeiros – Project Manager, Dev Team Member.

## Contact:

The link to our GitHub is: https://github.com/Pedro-Daniel/Ayvu_app it was made public on 04/24/2025.

# Table of Contents:

# How to read this document:

First of all, we need to analyze how the application is being built, before implementing any changes. If you already have access to the apk, open it! Viewing the tabs will be important for reading the document. If you don't, follow the steps: (the following text is aimed at people who understand the subject)

1. Access our Github;
2. Open VsCode on a personal computer;
3. Clone the remote repository;
4. In VsCode, access the "main" file in the "lib" folder;
5. On a personal cell phone, activate the developer options;
6. In the developer options panel, activate the USB debugging option;
7. Connect the cell phone to your personal computer;
8. When a pop-up appears on the cell phone, press "File Transfer";
9. Back in VsCode, in the bottom right corner, if it is not there, select your cell phone in "Select a device to use";
10. Click the run button in the top right corner;
11. After some time, the application will be built on the cell phone;
12. Lastly, click "Install" on your cell phone to finish the installation;
13. Open the app.

Finally, with the app in hand, explore it, try to see what it does and what it doesn't do. Now, this document will explicitly point out everything that was not implemented in the application that prevents it from becoming an effectively functional and commercial product.

# What is the Project?:

The Ayvu project consists of an Open Source mobile application that offers users a set of tools aimed at collecting language data. It allows users to: Record audio from speakers, georeference that recording, and issue metadata about speakers of a particular dialect. In addition, any user can search, view, and access any recording contained in a global database. The app allows people who are not tech-savvy to use these utilities. Ayvu means "language" in Tupi Guarani.

But what did the project actually become? This will be explained in more detail in the next captions of this document, but, in short, the final state of the project was a recorder of .aac files that generates a local database that associates this recording with a georeference and other metadata of a speaker.

# What should have been done?:

Previously at the [Project Initiation Document](#) was established that the team main focus would be to develop:

- Menu page with access to other functions;*
- Recording page;*
- Upload page to upload an existing recording from the mobile device;

- Playback (Ability to listen to the audio within the application);
- User Registration page in the application (Google Login);
- Recording Sharing page with a public server;
- Navigation Buttons to move between different sections;*
- Pop-up notifications with instructions and information;
- Speaker Data Collection Form:*
  - Recording Language;*
  - Region of origin of the recording with georeferencing if possible; otherwise, manual input (e.g., Maceió, Alagoas, Northeast Brazil);*
  - General notes on the background of the conversation;*
  - Personal Information of the Speaker(s):
    - Name;*
    - Gender;*
    - Date of Birth;*
    - Native Language;*
    - Second Language;
  - [Ability to add more spoken languages];
  - [Possibility to register the user's own profile];
- Contact page containing GitHub information;
- Shared Recording Access from Another Computer;

Where * means the functionality is mandatory. And [] indicates secondary focus.

# What have we done:

Quickly going through each of the features.

## Menu:

The menu page was changed, and now is called Database Page. The menu functionality is now handled by a Bottom Navigation Bar. This can be seen in Figure 1, the page layout.
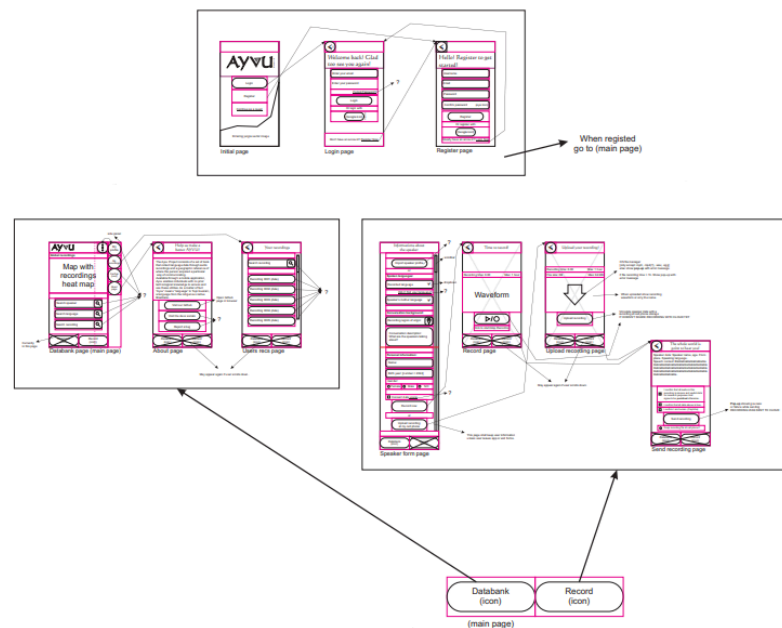
Figure 1: Page layout and navigation logic. Made with Corel Draw.

## Recording:

The recording functionality has been implemented, but the page itself could be much better. There were attempts to implement a playback, which is now commented out (I think) in the current codes. At this version, it is already possible to record in .aac. It is not possible to pause the recording. The recording file can only be accessed outside the application, in the file explorer. The recording can only be listened to through a player from another application, such as Spotify.

It would be interesting to have some kind of visual confirmation that the recording is actually happening, such as a spectrogram or wave viewer, besides the simple message that shows that the recording is happening. The recording time is not currently shown on screen. The recording will last as long as the application remains open or the phone's storage can handle receiving data. Even though there is a message saying "Max: 1 hour", there is nothing that actually limits the recording, and there is no logic for how the application should handle or inform the user that this limit has been reached or is approaching.

## Upload page:

The upload functionality has not really been implemented. There is no routine in the app that opens the phone's file explorer and selects a recording for use. The frontend of the page has been developed, but the backend is missing.

## Playback:

As previously stated, there were attempts to implement a playback, but it was not possible within the project delivery deadline.

## Registration and Login with Google:

The frontend of each page was created and is available in the Github files (on the "tests" page). These pages do not appear at any time during the current main execution of the application. In the project, it was not possible to establish external connections with Firebase. It would be a key part of the effective structuring of login with Google and all the user creation logic. This part was also not developed in the base application.

## Sharing with a public Server:

This is another case where the frontend was developed but the soul of the application is non-existent. Due to the lack of connection to Firebase, it is not possible to send the generated files to any type of cloud, and therefore this recording cannot be retrieved on another cell phone.

## Navigation:

All navigation in the application works beautifully, the only pages that do not have a specific path are the Registration and Login pages, which were omitted because the functionality related to this was not implemented. All navigation occurs as shown in Figure 1, the page layout.

## Pop Ups:

The functionality of showing different Pop Ups has been implemented and can be called by importing the popup_helper file.

## Speaker data Forms:

The speaker information form already collects all pertinent information from the application, including: Geolocation, date, speaker name, speaker gender, year of birth, language spoken in the recording, speaker's native language, region of recording (in written form). The application makes this data available in a local database and this information can be viewed in the "My Recordings" page.

However, several aspects can still be improved, such as: The current form does not validate if the submission data is empty (this was implemented, but later changes throughout the page made it no longer work). The user should not actually need to click on the geolocation button to register this information, this was a poor design choice. No type of consentment form is generated so that the user can read and validate the use of their data. The user must consent with the use of data to allow the process to be finalized.

There is a limited number of languages shown in the dropdown. The logic should be changed so that the user writes the name of the spoken language and chooses it if it already exists in the global database, otherwise a new language must be created. (What if the name of the language cannot be written, such as those with click sounds? This is a topic for even more future versions! We need to continue with small steps.) The current form only allows one to enter a single language. But what if the speaker speaks more

than one language, or has more than one native language? It would be interesting to add more spoken languages.

An important point is that currently the user always needs to fill out the form again and again, and often with the same information from previous captures. It would be interesting to be able to save the speaker's data so that it can be accessed again in the future. It would also be useful to have the option of configuring the user profile itself as a speaker profile.

## Others points of attention:

At the project's beginning, other important points were devised for the app development and marketing, but they did not become a focus during the execution of the project. Some of them are:

- Recording Editing page:
- Settings page:
  - Change app idiom;
  - Enable dark mode;
  - Contrast adjustment;
  - Font size adjustment;
  - Other accessibility options such as text-to-audio;
- Spoken Languages Information page, also called Databank Page:
  - Map showing the approximate locations of data collection;
  - Statistics:
    - Show the number of collections in the last month;
    - Show the number of recorded audio files;
    - Number of recorded speakers;
    - Number of recorded languages;
  - Possibility to search recordings by filtering by metadata, to find specific recordings or users.
- A better state management throughout the app;
- Automatic Audio Transcription;
- Effective upload to AppStore;
- Start to think about other operating systems, like iOS, and consider how complex it would be to switch between them.

# Conclusion:

With this context in mind, the authors believe that this document specifies the key parameters that can still be improved in the Ayvu application. This document formally concludes the development of the Ayvu Project by the authors in the universitary subject of Technology Integration Project. The authors hope that this work can serve as a basis for other works, whether direct or indirect continuations, in the presence or absence of the authors. We offer our compliments to the stakeholders and to Professor Dr. Edson Mintsu, our coordinator. And may the magic of Open Source begin!