ASHESI UNIVERSITY

TWIGRAD: AN ASR-BASED APPLICATION FOR LEARNING TWI

APPLIED PROJECT

B.Sc. Computer Science

Pamela Quartson

2021

ASHESI UNIVERSITY

TWIGRAD: AN ASR-BASED APPLICATION FOR LEARNING TWI

APPLIED PROJECT

Applied Project submitted to the Department of Computer Science, Ashesi University in partial fulfilment of the requirements for the award of

Bachelor of Science degree in Computer Science.

Pamela Quartson

2021

DECLARATION

I hereby declare that this applied project is the result of my own original work and that no part of it has been presented for another degree in this university or elsewhere.

Candidate's Signature: 

……

Candidate's Name: Pamela Quartson

……

Date: May 14, 2021

……

I hereby declare that preparation and presentation of this applied project were supervised in accordance with the guidelines on supervision of applied project laid down by Ashesi University.

Supervisor's Signature: 

……

Supervisor's Name: David Ebo Adjepon-Yamoah (Ph.D.)

……

Date: May 14, 2021

……

Acknowledgements

To my supervisor Dr David Ebo Adjepon-Yamoah, whose encouragement and academic advice helped me undertake this project, I say thank you. I again thank Mr Dennis Asamoah-Owusu for supporting and guiding me through this project with his profound knowledge in the field of Natural Language Processing. I extend my sincere gratitude also to Elvis Okoh-Asirifi for his timely support and advise that kept me on course throughout this project. Finally, I appreciate all the volunteers and participants who supported the project with their time and effort throughout the development process. I am grateful for your all your generous contributions, feedback and critique.

Abstract

Language is a very important aspect of every people's culture and identity. Language is the basic human form of communication made up of various sounds and symbols that pass information from a person to another. The cultural significance of language makes it important for cultural identification and pride, hence the inability to communicate in one's language is simply a loss of part of their identity. The loss of native or heritage language is a problem most common among people living in a place other than their native lands where the dominant language is different from their native language. This project presents a technology-based solution aimed at the acquisition and maintenance of Twi (a Ghanaian language) among people who wish to learn the language or increase their speaking proficiency in it. The product, an ASR-based mobile application was well received, with majority of the stakeholders expressing an 80% acceptance of the functionality and usability of the application.

Table of Content

List of Tables

List of Figures

Chapter 1: Introduction

1.1 Background

Language in humans, is the fundamental medium of communication made up of a sea of distinct sounds and symbols used in a structured and conventional way and conveyed mainly by speech, writing, and gestures. Language is important for communication but even more, for identity, more specifically cultural identity. Language is intrinsic to the expression of culture because it is how we convey our innermost self across generations, transmitting and expressing our culture and its values; hence a fundamental aspect of cultural identity [13]. Given the close association between language and culture, the loss of language will inevitably lead to loss of cultural identity, which has many dangerous implications [13]. The possibility of losing one's cultural ties due to loss of language is particularly higher among people born in or living in places where their native language is different from the language in predominant use. Immigrant parents usually fail to teach their heritage language to their children because they want to ease their assimilation into the new environment [13]. In the context of African immigrants, some fail to pass their language to younger generations because of the negative attitudes they have towards their native languages; deeming that of their ex-colonial masters as more prestigious [1]. In homes where children are taught their native or heritage language, there are still problems that impede their maintenance. Findings suggest that gains made in the acquisition of heritage languages before the preschool years, at home, were lost as soon as children began attending school [9]. Another factor for failed maintenance is the internal force fuelled by the children's desire for social acceptance and conformity to the dominant group in their host country because of peer influence and the media of their country of birth [12].

1.2 Problem Statement

Transferring native or heritage language in the African Diaspora can be quite challenging as it is visualised against a backdrop of 'linguistic competition' between the immigrant languages and dominant language(s) among immigrant families [9]. This, when left unchecked, can lead to the loss of the language in subsequent generations. Given the importance of language to cultural identity, the loss of native or heritage language by Africans in the Diaspora adversely affects the development of Africa in the long run, amidst other effects.

1.3 Motivation

A sense of cultural identity among Africans in the Diaspora promotes and facilitates their engagement with the continent, paving ways for increased human, social, and financial capital for political and economic development [2]. In the age of globalisation, one of the great instruments for the African renaissance is language [8]. Hence, significant effort must be taken in its preservation within the continent and beyond. This is what inspires this project, the preservation of African culture among Africans in the Diaspora through language learning.

1.4 Related Works

Research into second language acquisition has shown that naturalistic, implicit learning is not always sufficient to achieve high-quality L2 proficiency (Common European Framework) and that explicit instruction helps overcome some of the problems [10][3]. Attaining oral proficiency in a language with a physical tutor can also be time-consuming as sufficient instruction and feedback are essential [17], but this impedes scale. Thus, interest in the application of Automatic Speech Recognition (ASR) to oral proficiency language learning is becoming more sought after [4]. ASR involves deriving the word sequence (transcription) of an utterance given a speech waveform [14]. This technology can

be applied to Computer-Assisted Language Learning (CALL) systems to enable pronunciation practising by mapping some given acoustic signal to the appropriate string of words. Presented below are some of the several papers and works on ASR-based CALL systems intended for practising grammar for oral proficiency.

(1) **Utterance Verification in Language Learning Applications**

In [17], a system aimed at providing oral L2 proficiency for second language learning students using ASR is designed to provide intelligent feedback on important aspects of L2 speaking such as pronunciation, morphology, and syntax. This is achieved by generating a predefined list of possible (correct and incorrect) responses for an exercise, for which a two-step procedure is used to determine what the student spoke (speech recognition) and how the student spoke it (error detection). The first step of the procedure is determining what was said (content) and subsequently how it was said (form). In the first phase (where user input recognition is done), the system is allowed to tolerate deviations in the way utterances are spoken, while in the second phase (error detection), strictness is required for utterance verification. The designed system uses SPRAAK, an open-source Hidden Markov Model (HMM) ASR package for its speech recogniser. The SPRAAK package setup includes Acoustic Pre-processing, Language Model and Pronunciation Lexicon, and Acoustic Models. The speech recogniser has a constrained language model and small vocabulary used for selecting the spoken utterance from a list. After this process is the utterance verification. Utterance verification is done by using a confidence predictor combination approach. It uses two types of predictors: acoustic likelihood ratio and duration-related features to train a logistic regression model that combines the predictors into one confidence measure to determine utterance accuracy.

(2) **EduSpeak®: A speech recognition and pronunciation scoring toolkit for computer-aided language learning applications**

[6] focuses on the EduSpeak® system, a software development toolkit for Computer-Assisted Language Learning (CALL) application developers. EduSpeak®is a high-performance real-time recognition system that is specifically adapted for CALL applications, particularly acoustic models tailored for non-native speech recognition, pronunciation scoring algorithms, and several system features developed especially for the CALL domain. The speech recognition implementation for the system uses Hidden Markov Models (HMMs) as the underlying statistical model. Models are trained on several thousand sentences of non-native data, balanced over gender and language proficiency. For pronunciation scoring, the system uses Machine scores and features of non-native speech indicating pronunciation quality. The features used are Spectral match, Phone duration, Word duration, and Speech rate. Mispronunciation detection methods are applied after the phonetic segmentation of the utterance has been obtained using the speech recognition engine and the known transcription for phone level mispronunciation detection.

(3) **Evaluating automatic speech recognition-based language learning systems: a case study**

This paper [16] evaluates a prototype of an automatic speech recognition (ASR)-based language learning system that provides feedback on different aspects of speaking performance (pronunciation, morphology and syntax) to students of Dutch as a second language. The project –Development and Integration of Speech technology into Courseware for language learning (DISCO)– was started with the aim of developing the ASR-based system. This system will automatically detect pronunciation errors (mispronunciations of speech sounds), and grammar

(morphology and syntax) errors in Dutch L2 speaking and generate appropriate, detailed feedback on the errors identified. The learning process in the program starts with conversation simulation (a dialogue). Based on the type of errors the students make, they are then offered remedial exercises. The exercises focus on specific speech sounds or syntactic and morphological structures without a conversational context. The feedback strategy is immediate corrective feedback visually implemented through highlighting, which puts the conversation on hold and focuses on the errors.

(4) **Phone-level pronunciation scoring and assessment for interactive language learning**

[18] investigates a method of automatic pronunciation scoring for use in CALL systems. The research is focused on designing a system that measures the pronunciation quality of non-native speech at the phone level to locate pronunciation errors, assess the closeness of the pronunciation to that of a native speaker, and identify systematic differences when compared to a pronunciation dictionary. It uses a likelihood-based 'Goodness of Pronunciation' (GOP) measure, which includes an individual threshold for evaluating each phone in a given utterance. Given a speech waveform, frontend feature extraction converts it into a sequence of Mel-frequency Cepstral Coefficients (MFCC) followed by two recognition passes, the forced alignment pass and the phone recognition pass. Based on the output, the individual GOP scores are calculated for each phone, and finally, a threshold is applied to each GOP score to reject badly pronounced phones. The quality of the GOP scoring procedure, however, is dependent on the quality of the acoustic models used.

1.5 Final Product

Targeted mostly at Ghanaians and other Africans in the Diaspora, the proposed final product for this project is an ASR-based mobile application that helps people learn the Twi language through interactive lessons and exercises for oral proficiency.

Chapter 2: Requirement Analysis

This section presents the functional and non-functional requirements for the proposed system. It describes in detail the specific ways in which the application is to behave. The requirements for the application were inspired by data gathered through an online survey designed for stakeholders, specifically prospective users. This survey was completed by seventy people, mostly Ghanaians living abroad and home.

2.1 Project Perspective

The computer-assisted language learning tool will be a mobile application for use on mobile devices, including phones and tablets. The application would be supported by both Android and IOS platforms and will require internet connection to function. It uses backend services from Google's Firebase and only caches some of the user's data on their device for quick access.

2.2 Functional Requirements

To use the application, a user would have to register or sign up to create a free account. After this, they can log in and start their learning journey. The product will have one learning level only, that is, the beginner level. This level will teach alphabets, numbers, and simple words in Twi. The user, after logging in, can start learning, knowing that their progress through the various modules in the level will be tracked and saved. Also, the knowledge gained from each module will be tested and score with a simple interactive test. **Table 1** below states and describes all the functional requirements of the application, and **Figure 1** illustrates a use case of these functionalities.

Requirement Number	**Description**

R01	The registration subsystem of the application should allow users to register or sign up for a free account
R02	The authentication subsystem of the application should allow all users to log into their accounts using their email and passwords to use the application
R03	The authentication subsystem of the application should allow a user to reset their forgotten passwords
R04	The translation subsystem of the application should allow users to select their preferred choice of language (English/French) for the application
R05	The lesson content subsystem of the application should allow users to select a module to learn
R06	The lesson assessment subsystem of the application should allow users to take an assessment for a completed module
R07	The lesson content subsystem of the application should allow users to listen to audio pronunciations of text in Twi
R08	The lesson assessment subsystem of the application should allow users to input speech in Twi or English, or French for assessment
R09	The lesson assessment subsystem of the application should be able to determine the validity of users' speech input in Twi and give feedback on it
R10	The lesson content subsystem of the application should be able to save a user's progress through the various learning stages
R11	The application should allow users to chat with a customer support agent
R12	The application should allow users to edit their profile
R13	The application should allow users to change the appearance of the user interface

Table 1: Functional requirements

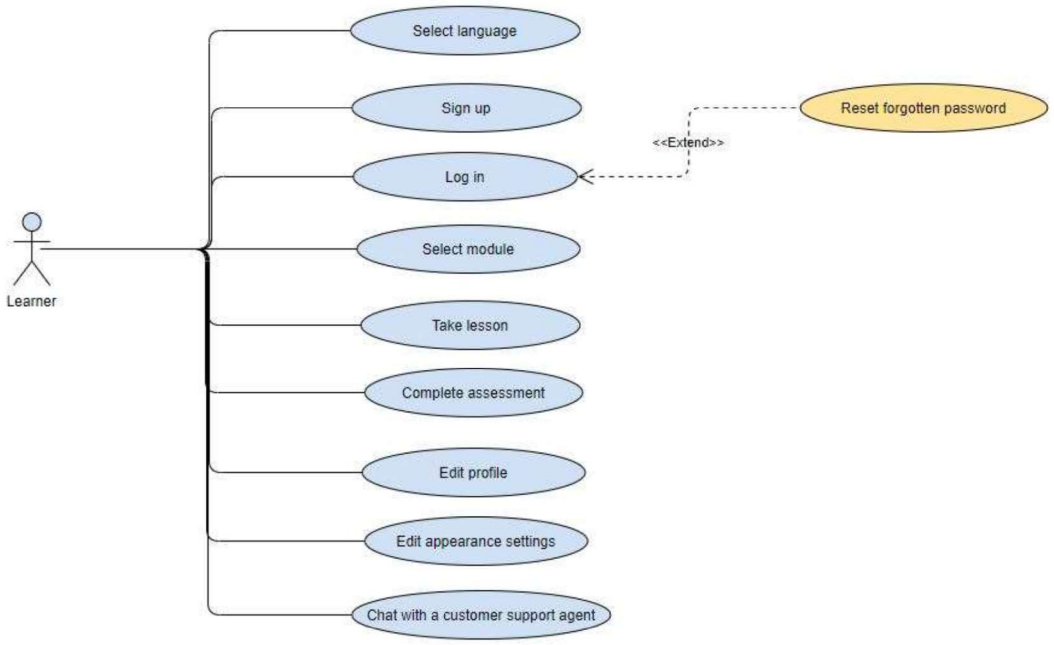


Figure 1: Use case diagram for functional requirements

2.2 Non-functional Requirements

(1) Availability: The application must be available for use at any point in time.

(2) Security: Since the system holds the user's personal data, it should be secure enough to deny access to the data to unauthorised people.

(3) Maintainability: It should be easy for the application to be maintained and improved upon.

(4) Dependability: The application should be trusted to produce meaningful and accurate results that can be relied upon to generate an accurate score.

(5) Scalability: The application should support an increasing number of users without compromising performance or low latency.

(6) Usability and learnability: Users should not find it difficult to use the application or navigate through it to employ its functionalities.

Chapter 3: Architectural Design

This section provides information on the architecture used for the product and discusses its advantages and disadvantages with regards to the application. According to [15], the architecture of a system has a strong relationship with the non-functional requirements of the system. He mentions again that the architectural design of any system should be influenced by the non-functional requirements of that system. With this in mind, suitable architectures have been selected to describe the structure and design of this product based on its non-functional requirements. The list of non-functional requirements considered before finding the suitable architecture for this system are:

(1) Availability: The application must be available for use at any point in time. Hence the architecture must support the continual running of the system.

(2) Security: Since the system deals with the health records and other valuable data of people, it should be secured in such a way that unauthorised people cannot get access to this data.

(3) Maintainability: It should be easy for the application to be maintained and improved upon; the architectural design should therefore make room for that.

(4) Dependability: The application should be trusted to produce meaningful and accurate results that can be relied upon to generate accurate pronunciation scores. The chosen architecture should be able to provide support for this function.

(5) Scalability: The application should be able to support an increasing number of users without compromising performance and low latency. The architecture should, hence, support this property.

(6) Usability and learnability: Users should not find it difficult to use the application or navigate through it to employ its functionalities. The architecture should, therefore,

support and concise representation of the system that makes it easy for users to interact with it or learn how to do so.

The architectural designs selected for this product is the Model-View-Controller (MVC).

3.1 Model-View-Controller (MVC)

Commonly used for web-based applications, this design abstracts a system into three separate logical units called the model, the View and the Controller to separate the presentation and interaction of system data [15]. The Model component handles the system data and all related operations that can be performed on it. The View determines and controls how data is presented to the user or how the user 'views' the data. Finally, Control is the component in charge of managing user interactions. User interactions come in the form of mouse clicks, keyboard and voice inputs, etc. These interactions are passed on to the View and Model by the Controller for processing. This architecture was chosen for this application also because the framework which the application will be built with (Ionic) builds with this architecture. For this application, the Model is the central data system that keeps the information of users like their authentication data, progress, profile data, etc. The View also, in this case, is the set of web pages that the users will be interacting with to carry out their tasks. The Controller is the algorithm that will respond to a user's interaction with the View. Users' interaction with the View will mainly include mouse clicks/taps and voice and keyboard inputs. The major advantage of MVC is its modularity. This allows for a level of independence among the various components; hence data can be changed without any effect on its representation and vice versa [15]. This feature of the model will be very advantageous to the design of the application on the basis of its non-functional requirements. The modularity of the design, ensuring the independence of the components, makes it difficult for a failure in one component to affect the entire system. This will ensure a good level of availability, dependability, scalability and security of the system. It will also help

with maintainability, as the separation of components will make it easy to determine where a failure is coming from, and the independence among the components will be useful when fixing this failure because a change in one component does not have much effect on the others. The usability of the system is ensured with this design because of the abstraction and separation of the View module. With this, users only need to focus on providing the right inputs to the system through the user-friendly interface and not worry about anything that goes on beyond that. The disadvantage of this system, however, is that even simple data model and interactions can require additional and quite complex code to implement [15]. This, however, is not a problem for this application because it is a medium-scale application.

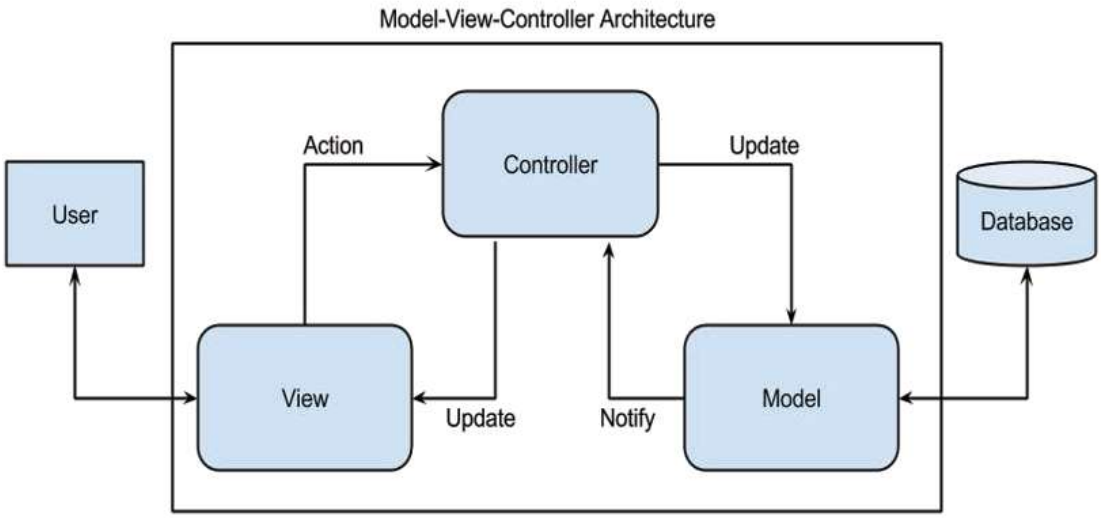


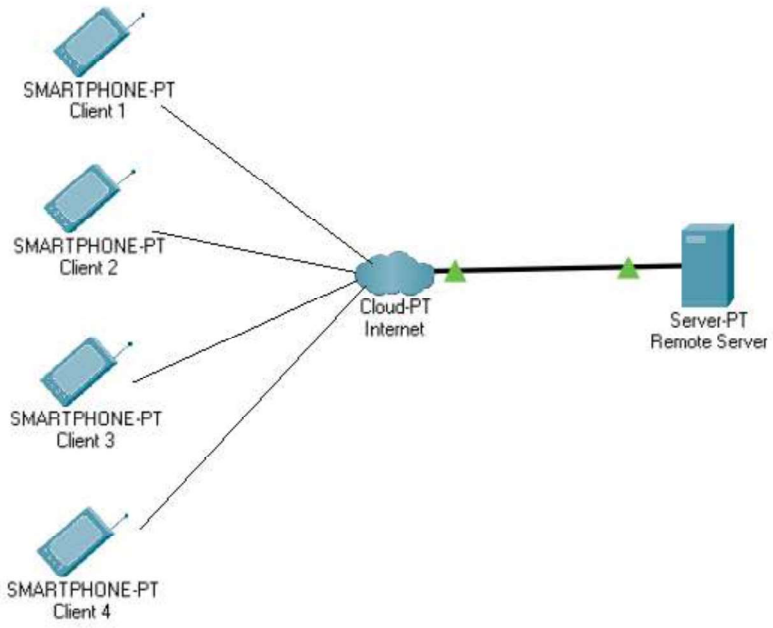
Figure 2: MVC Architecture

3.2 Client-Server Architecture

The client-server architecture system, according to Sommerville [15], is organised as a relationship between some servers providing specific services and some clients who make requests to these servers to access those services. Although this architecture is usually regarded as one involving multiple servers, it is also possible to implement it with independent services provided by a single server [15], as is the case with TwiGrad. With TwiGrad, the set of services include user registration and authentication, password reset,

profile update, progress viewing, etc. All these services are hosted on a remote server and access by the clients (users) via the mobile application on their devices. For example, a learner who wants to reset his/her password would go to the 'change password' option in the settings page, click on it, provide the needed data for the new password, and then click Save. This action will send an HTTP request to the server hosted on the Internet. On receipt, the server will process this request, effect the changes in response. This response, when it arrives on the client's device, will be displayed as a simple feedback message informing them of the success or failure of the action. **Figure 3** below shows a client-server architecture model, where clients (smartphones) are accessing services from a remote server through the Internet.

Figure 3: Client-Server Model



Chapter 4: Implementation

This chapter focuses on the various technologies and tools that have been used in the development of the application. It discusses the software, tools, methods and processes that underline the functioning of the application.

4.1 Technology Stack

Given that the application will be a mobile application, there was the need to make platform considerations. This involved choosing to build either a native application for Android and IOS separately, or a cross-platform application, or a hybrid application that supports both platforms. A summary of the differences between the above-listed technology stacks is given in **Table 2**. This application was developed as a hybrid mobile application. The choice was informed by the time constraint for the completion of the project, limited budget, learning curve for technologies used, and platform portability.

Table 2: Technology Stacks Summary

	Native	Native Cross-platform	Hybrid
Description	Application is built for a specific platform, and it is developed with a programming language supported by that platform. This gives it access to all device features	Application has a single code base that can be compiled for execution on multiple mobile platforms like Android, IOS, etc. A few changes are required to compile executable files for different platforms	Application is built using web technologies (HTML, CSS, JavaScript) that are wrapped in a WebView object that renders the contents as a website would when the application is in use. This takes advantage of the advanced web components of native Android and IOS SDKs and can be used on any platform without any tweaking
Programming languages	Objective-C, Swift for IOS Java, Kotlin for Android	Xamarin, React Native, Titanium	HTML, CSS, and JavaScript

Device Integration	Provides full device integration	Provides limited device feature support	Provides limited access to device features and platform-specific features
Performance	High performance	Performance delays with CPU and GPU-intensive tasks	Provides limited access to device features and platform-specific features
Flexibility	Unlimited access to all features of the platform	Feature implementation flexibility is limited to those provided by the used framework	Limited access to full features of native platform
Cost	Most costly to develop because an app has to develop for each platform separately	Less costly to develop for different platforms	Least costly to develop for different platform
Code reusability	Code cannot be reused to build for a different operating system	Code can be reused to develop for a different operating system with some tweaking	Code can be used for all platforms without any tweaks
User experience	Provides more convenient user experience	Provides less convenient user experience compared to native apps	Provides less convenient user experience compared to native apps

4.2 Frameworks

The next decision to make after choosing the technology stack was the framework for development. A software framework is a set of code libraries, compilers, and other tools and programmes designed for software development. There are many frameworks available for developing hybrid mobile applications, but this application was developed with the Ionic 5 framework. This framework was chosen because it provides adaptive styling for a better user experience, it is easy to build, and it also provides simple and functional yet beautiful designs.

4.3 Tools and Technologies

4.3.1 Ionic Framework

Ionic framework is an open-source UI toolkit for building high performance, high-quality mobile and desktop applications using web technologies — HTML, CSS, and JavaScript — with integrations for popular frameworks like Angular, React, and Vue [7]. Ionic uses a single codebase that runs on multiple platforms. It focuses on the frontend User Experience (UX) and User Interface (UI) interactions of an app, i.e., UI controls, interactions, gestures, and animations, with clean, simple and functional designs [7]. Emulating native app UI guidelines and using native SDKs brings the UI standards and device features of native applications together with the power and flexibility of the open web, optimising it for both native environments and the web [7]. To ensure beautiful UI designs, Ionic provides pre-designed components and beautiful and extensible base themes to get developers started.

4.3.2 Google Firebase

Firebase is a cloud computing and development platform developed by Google for creating mobile and web applications. The platform provides several products split into three groups: Build, Release Monitor, and Engage. Build is a serverless backend solution that enables developers to deploy their applications to Firebase's cloud servers. The products under this service are Firebase databases, machine learning infrastructure, hosting and storage solutions, and Cloud Functions. The Release & Monitor solution is aimed at helping developers improve their apps' quality by carefully rolling out features and monitoring their adoption [5]. Engage is aimed at engaging customers by providing a platform to test new ideas and uncover insights from a customer base and for designing customised products for them. The TwiGrad app makes use of Firebase's Build solution. It uses the Firebase database, authentication, and hosting and storage products from the serverless hosting service.

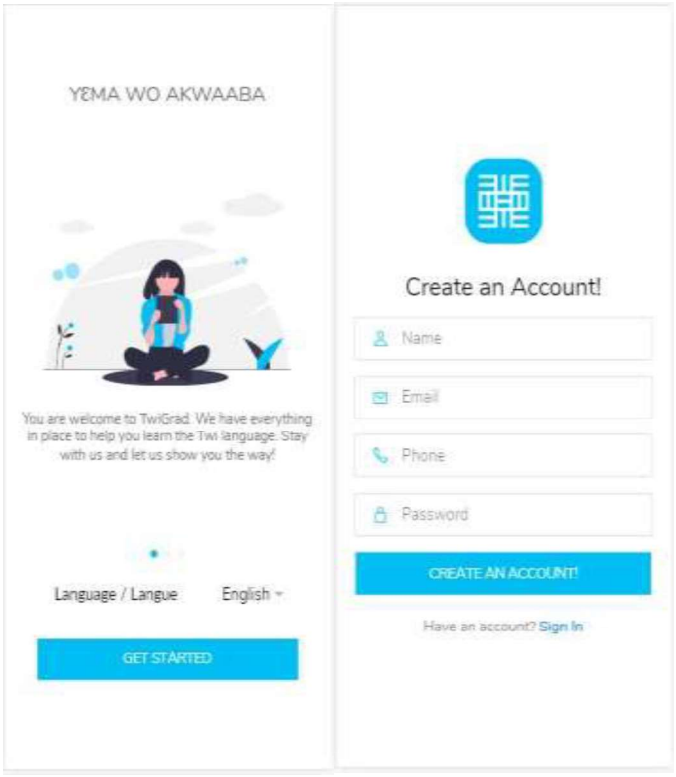
4.3.3 Kaldi ASR

Kaldi is a toolkit for speech recognition written in C++ and licensed under the Apache License v2.0 [11]. It provides access to its library functionalities through command-line tools written in C++. The tools are then called from a scripting language like Python for building and running speech recognisers. Each tool delivers very specific functionality with a small set of command-line arguments [11]. Notable features of the Kaldi toolkit are its integration with Finite State Transducers (FST), its extensive linear algebra support, its extensible algorithmic design, and its non-restrictive license [11]. The feature extraction, acoustic modelling, and decoding tools from Kaldi were used to implement the ASR system for the TwiGrad application.

4.4 TwiGrad App

This section presents the TwiGrad application with relevant screenshots showing how the application functions. **Figure 4** and **Figure 5** below shows the welcome page (first page for unregistered user), log in, forgot password, and sign-up pages for the application. On the welcome page, the user can choose the language of their preference between English and French. This is to allow users of the app to learn Twi from the language in which they are most fluent.

Figure 4: Welcome and SignUp pages



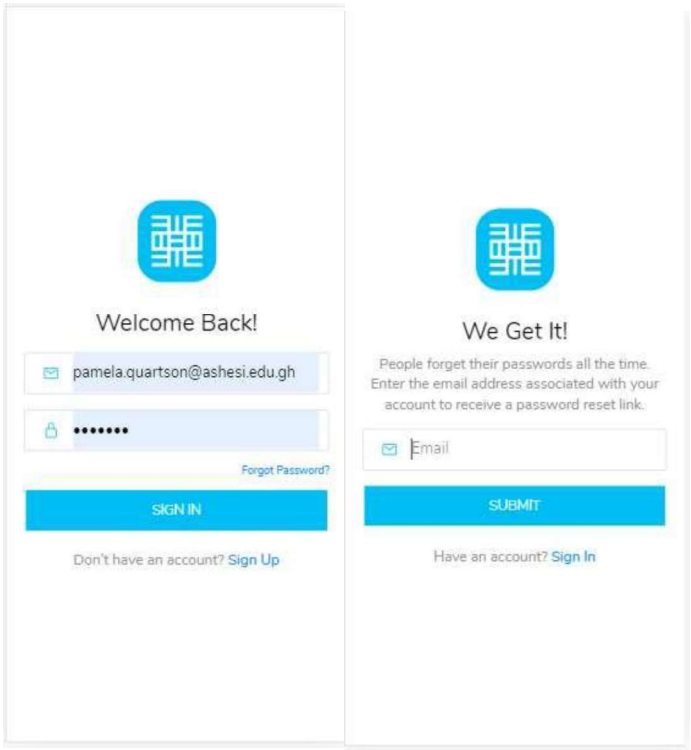


Figure 5: Login and Forgot Password pages

Figure 6 shows the home page, the page a user is directed to after a successful login. This page contains all the available modules to be learnt. The application, for now, focuses only on beginner level training; hence all available modules are for beginner level learning only.

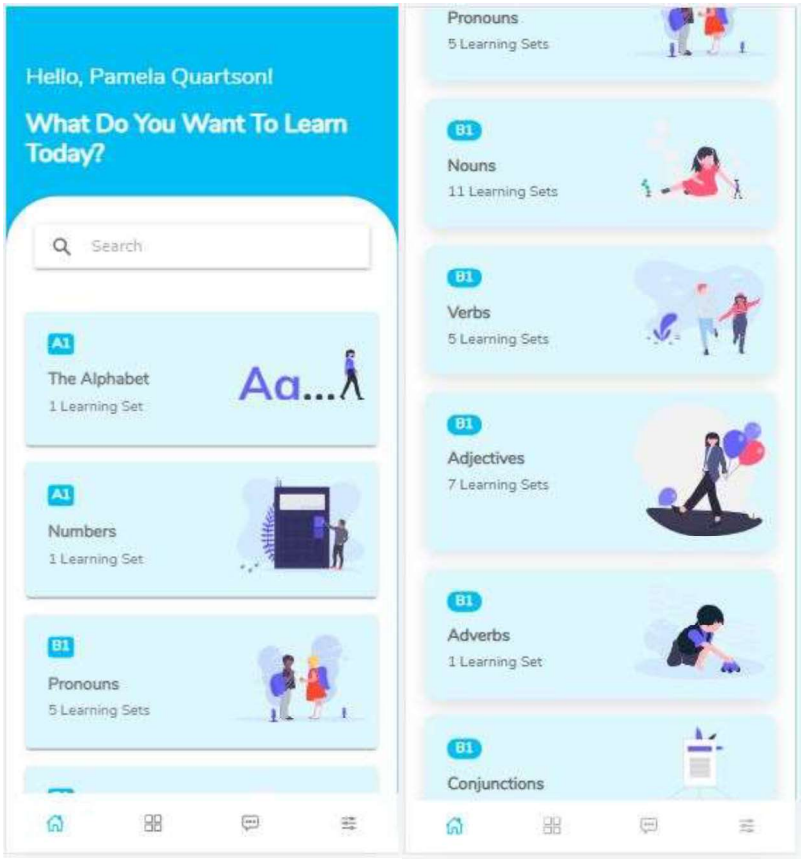


Figure 6: Home Page

In **Figure 7** below, the 'Alphabets' lesson is opened, and the user is shown all the Twi alphabets. The inner play forward button below the lesson content moves the user to the next slide, where they can see individual alphabets and play their sounds. The inner play backwards button sends them to the previous slide. The outer play forward button takes the user to the end of the lesson, and the outer play backwards button takes them to the beginning of the lesson. The square-shaped button takes them out of the module completely and back to the home page.

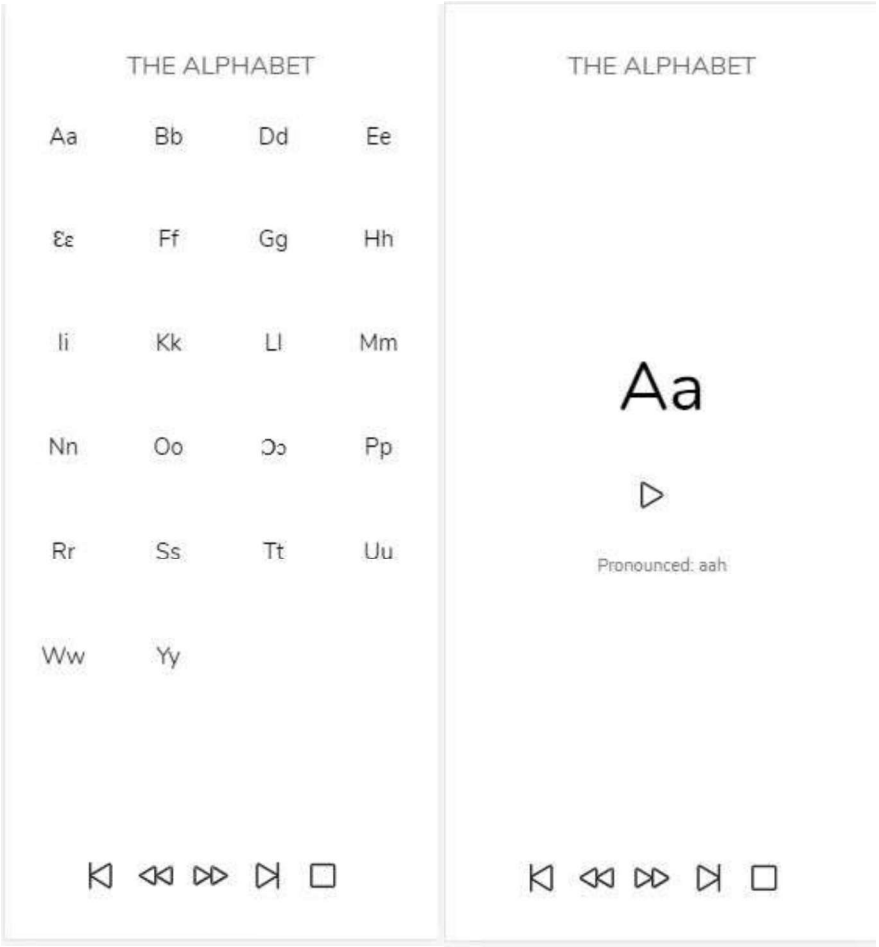


Figure 7: Alphabets Lesson Pages

Figure 8 and **Figure 9** below presents the lessons and assessment available in the 'Pronouns' module. These pages also have the same buttons underneath the lesson content like the 'Alphabets' module, and they perform the same functions as described above. The pictures shown for the Alphabets and Pronouns modules are representative of all the other modules in terms of UI design and the sequence of the design.

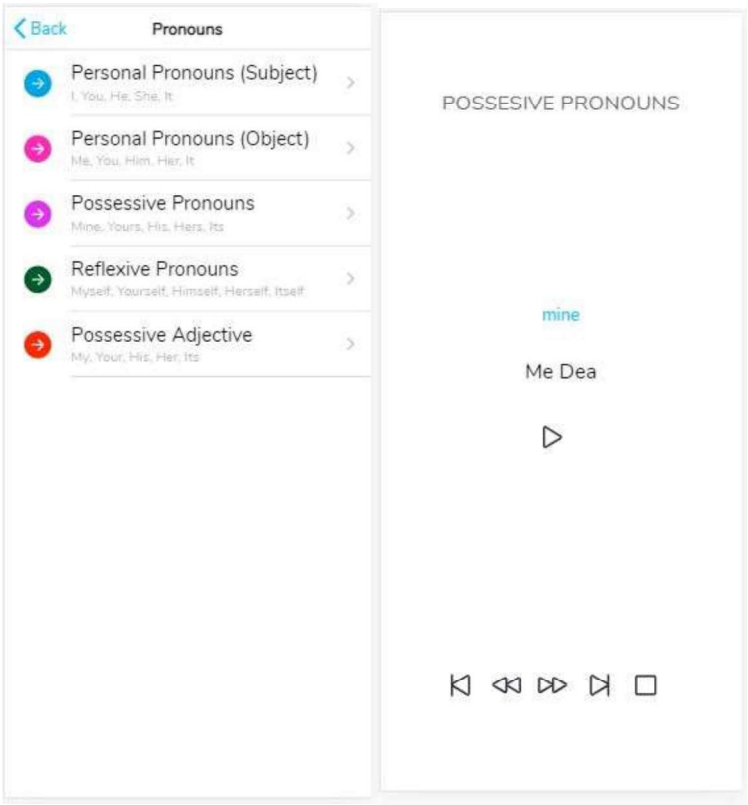


Figure 8: Pronouns Lesson Pages

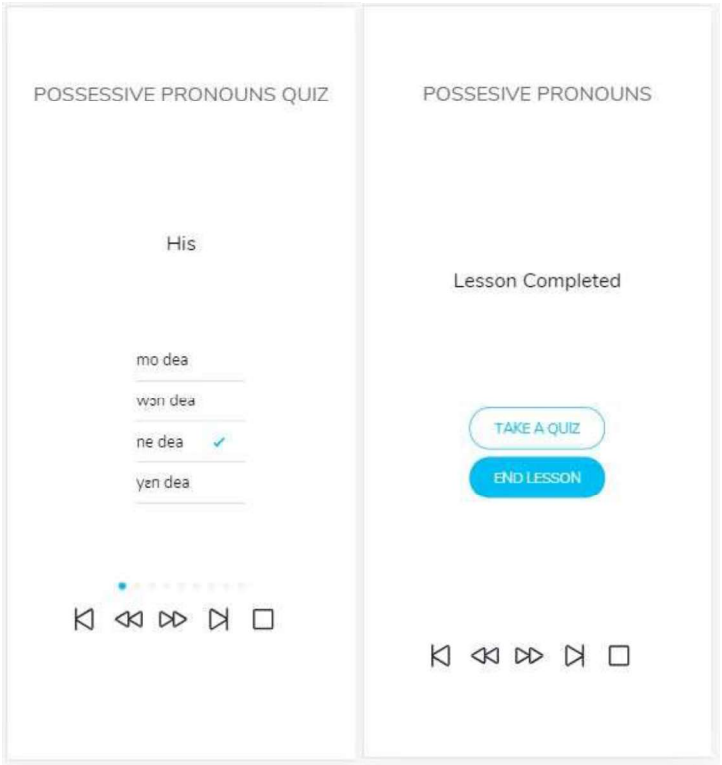


Figure 9: Pronouns Assessment Pages

Chapter 5: Testing and Evaluation

Testing and evaluation are necessary for the development life cycle of a piece of software to ensure it performs as expected, conforming to a user's requirements with no errors [15]. The development tests carried out for the application during the development process are detailed in the following sections of this paper.

5.1 Unit Testing

This type of test involves testing individual functions in the application to check whether or not they work as expected. All functions in the application were tested as they were built during the development of the app. These included mostly the backend functions that control authentication, audio playing, page navigation, and results verification for lesson assessments. These functions were tested individually with both valid and invalid input to prove their accuracy. For instance, the answer validation and scoring functions of the assessment feature of the application are supposed to mark a student's answer choice and award a mark only if the answer is correct. These functions were tested against both right and wrong answers on different devices to ensure they always worked accurately.

5.2 Component Testing

A component of an application refers to groups of individual functions integrated together to complete a common task. Component testing hence refers to testing components of an application to verify if their purpose is executed as expected. An example of a component in the TwiGrad app is the authentication service which includes the login, password recovery, and registration functions. To test such a component, the credentials of an already registered user were passed through the login page to check whether such a person would be granted access to the next page. The same was done for an unregistered user, as well as a registered user with an updated password. The login page from **Figure 4** and the home page from **Figure 5** show the flow from the authentication component to the

learning component. Component testing was performed on all components in the app accordingly to ensure they all functioned properly and linked to the right interfaces after them.

5.3 System Testing

System testing is where the various components of the application are tested together as one complete system serving some desired needs. This test also is to ensure compatibility among all the components of the application and interact correctly to transfer the right data at the right time across the various interfaces [15]. For the system testing procedure, the application was deployed on a Samsung Galaxy A10s Android mobile device, an iPhone 6s device by different users, and then on a web browser of a desktop computer to evaluate the complete application. During the process, bugs were caught and fixed and the whole process iterated until it produced satisfactory results.

5.4 Test Validation

Test ID	Feature description	Test decision
T01	The user sign-up functionality takes a valid full name (text), valid email (text), valid password (text), and a valid phone number (numbers) as input. Upon successful registration, it should notify the user and direct them to the login page.	Pass
T02	The login feature takes a valid user email (text) and a corresponding password (text). On successful login, the user is sent to the home page where the lesson modules are. If not, they are notified that either their email or password may be incorrect.	Pass
T03	The lesson learning feature requires a user to be logged in and to tap on any of the available modules. A user would be sent to whatever module in the list they tap on.	Pass
T04	The word pronunciation feature requires a user to tap on the play icon underneath any word in any module to hear the Twi pronunciation of the word.	Pass
T05	The lesson assessment feature includes a list of multiple-choice questions with their answers. The user only has to tap on the correct answer to answer all questions	Pass

	available. Then they can see their score at the end of the test or opt-out of the test altogether—users who opt-out will not receive a score for the assessment.	
T06	The 'edit profile' functionality takes a valid full name (text), and a valid phone number (number) as input from a user. It then processes and alerts the user whether or not the operation was successful. If successful, the user's data will be updated in the database.	Pass
T07	The 'forgot password' and 'change password' features allow a user to reset their password. This takes the user's email address and sends a reset link to the address if it exists in the database. On receipt, the user can reset their password by providing a new valid password (text). The user will be alerted of the success or failure of the process. A successful update will change the user's old person to the new one provided.	Pass
T08	The change theme functionality allows a user to change the app's theme between light and dark by toggling the 'change theme' button. This should make the theme dark when it is toggled on or light when it is put off.	Pass
T09	The language changing feature requires the user to select between English and French by tapping on the preferred language. All the texts in the application (except Twi) should be translated to the chosen language.	Pass
T10	The 'contact customer support' feature should allow a user to send a message (as an email) to a support agent. It takes the user's message in and sends it to the support agent.	Pass

Table 3: Requirements Test and Validation

Figure 10: Login and home pages

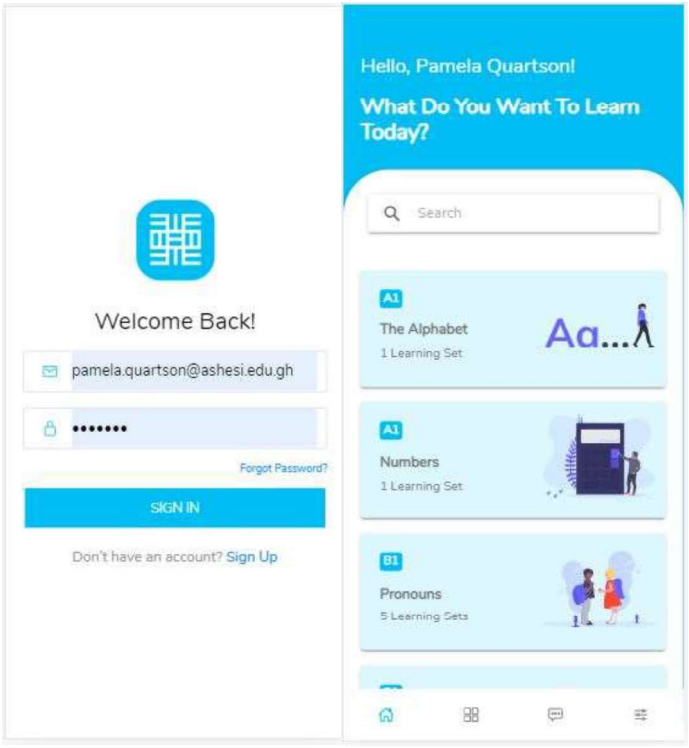


Figure 11: Verbs Lesson Module

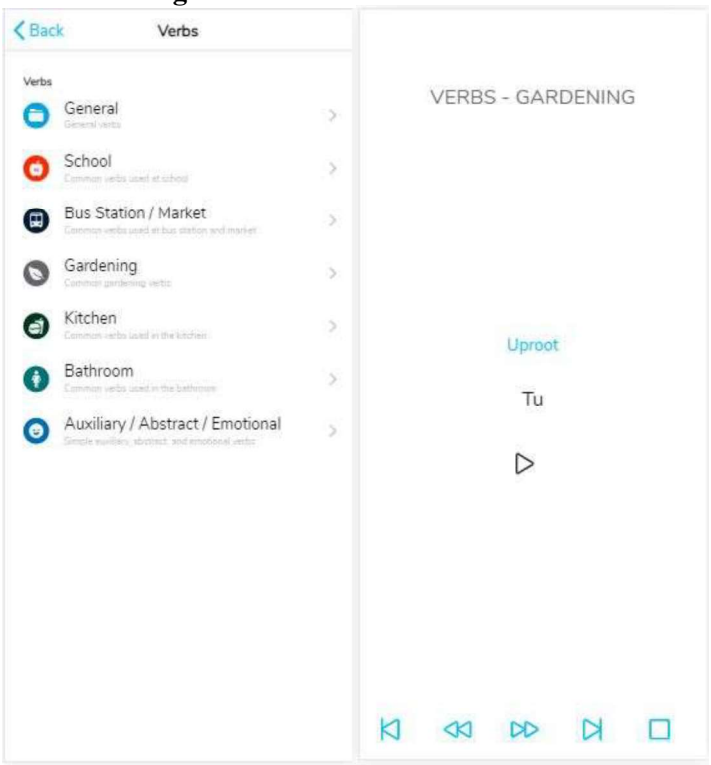
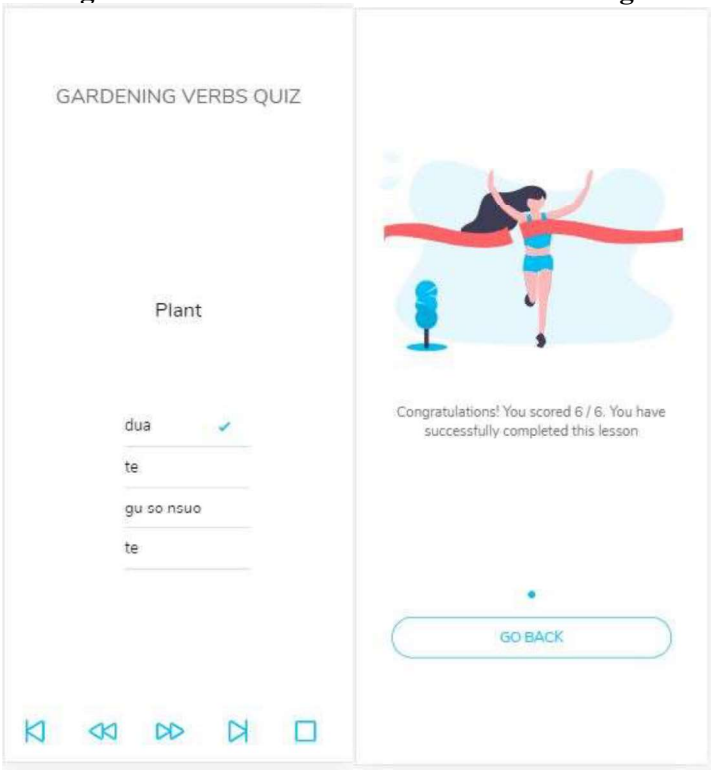


Figure 12: Verbs Submodule Assessment Pages



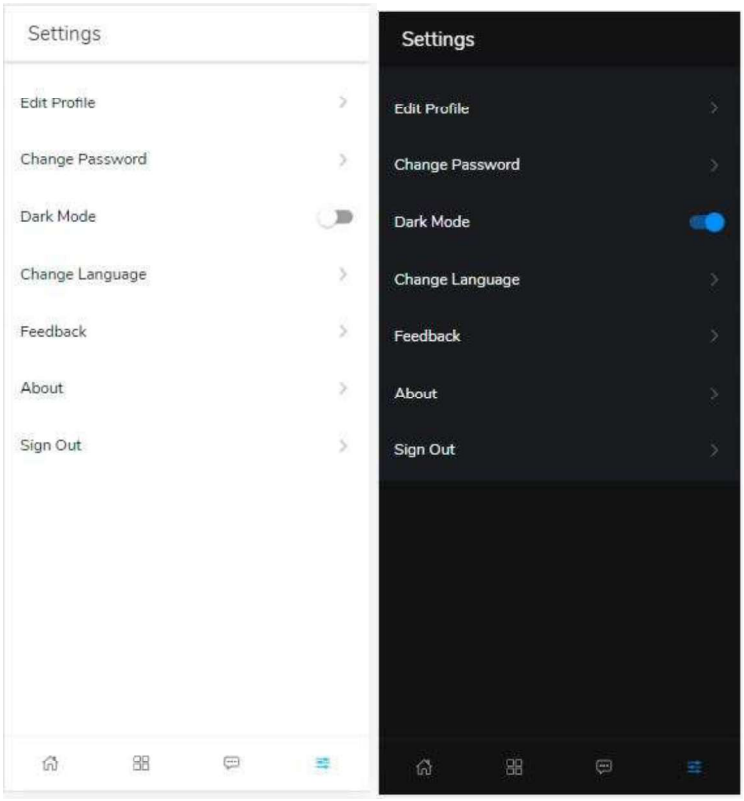


Figure 13: Theme Changing Feature

Figure 14: Customer Support Feature

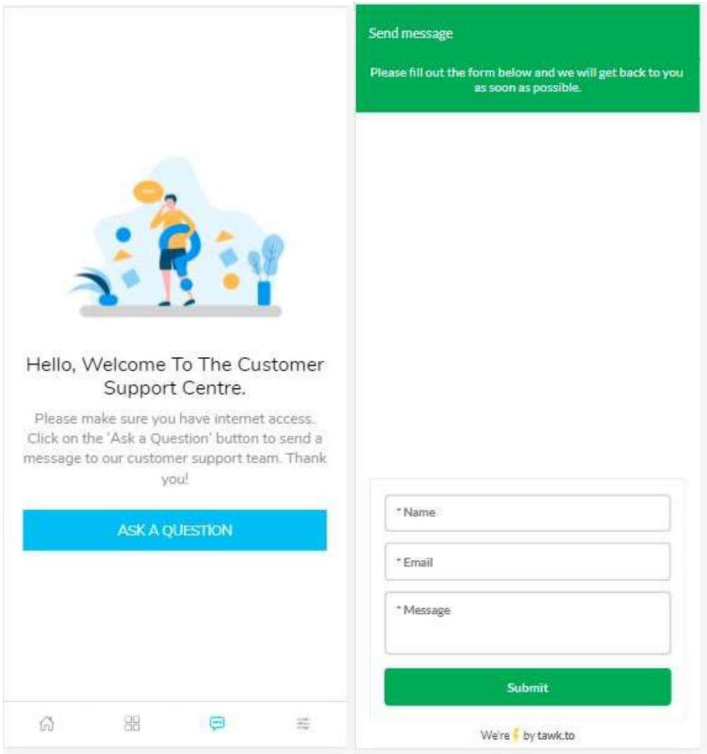


Figure 15: Language Changing Feature

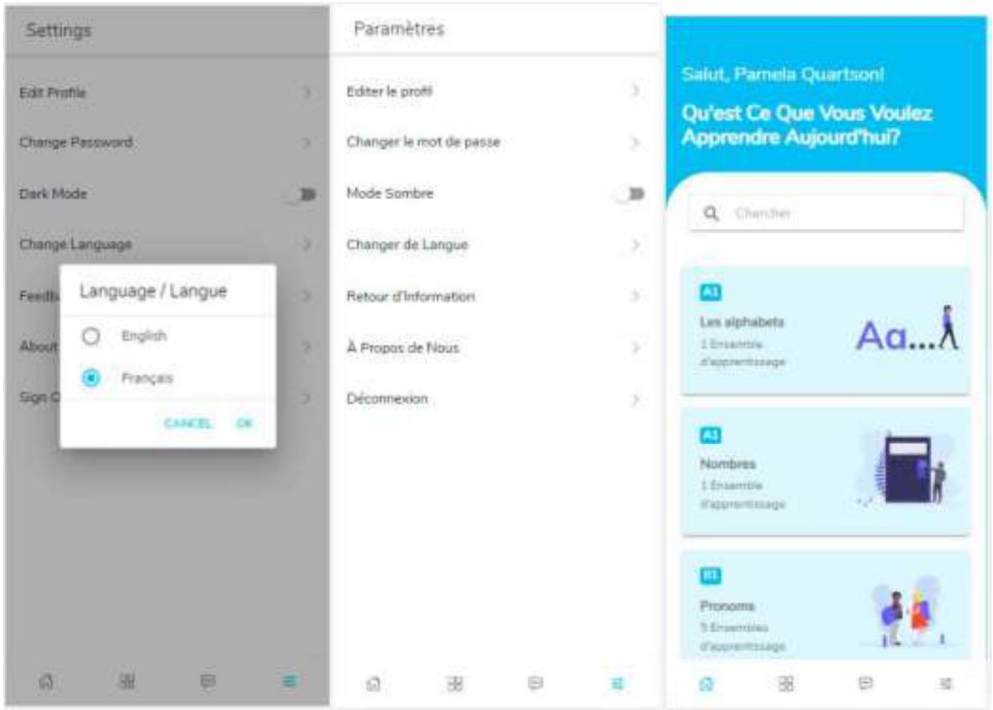


Figure 16: Password Reset Feature

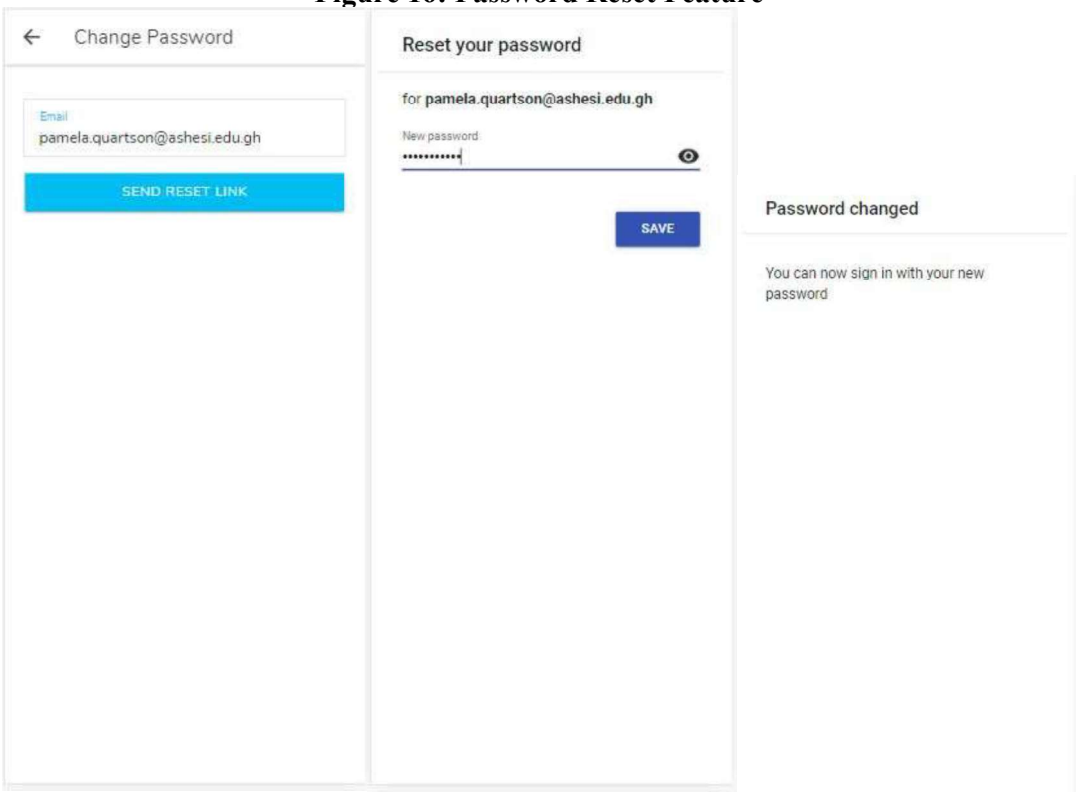
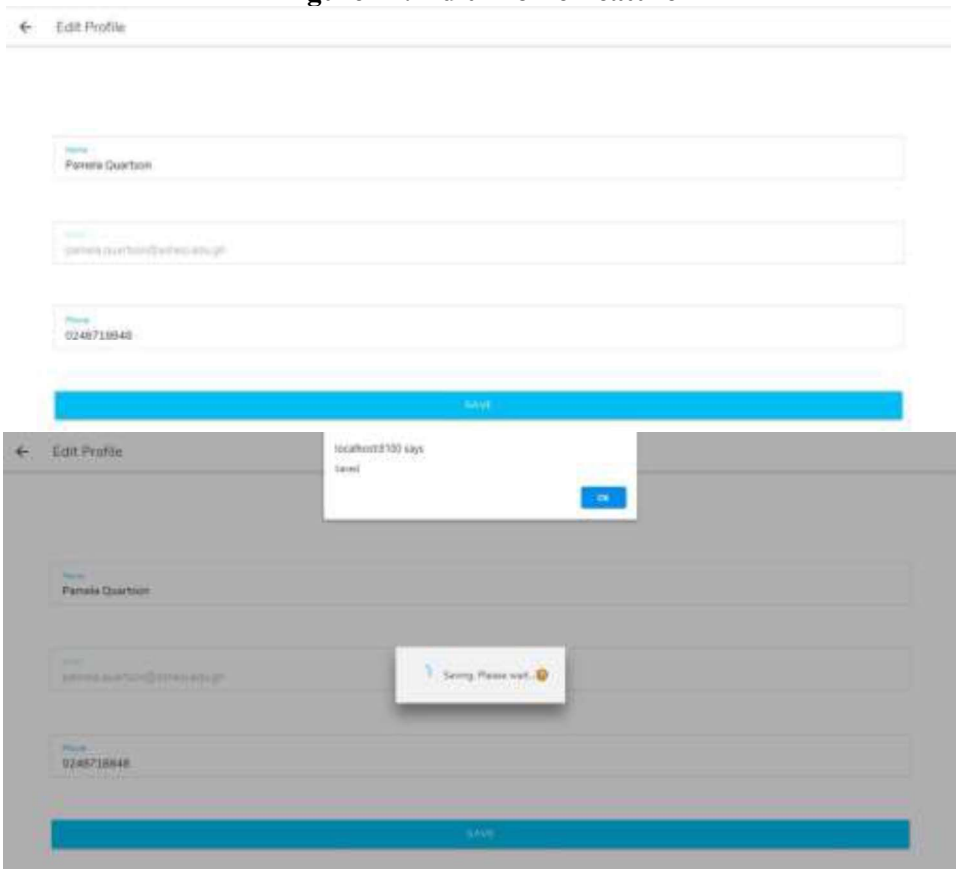


Figure 17: Edit Profile Feature



Chapter 6: Conclusion and Recommendation

This section presents a summary of the proposed project, outlining its intended purpose, the completion status of functional requirements, project limitations, and recommendation for future works.

6.1 Project Summary

To strengthen cultural ties between Africans / Ghanaians in the Diaspora and their African roots for development in Africa, this project proposed TwiGrad. TwiGrad is an ASR-based mobile application for learning Twi (a Ghanaian language) for oral proficiency. The early version of the application presents a beginner level introduction to the Twi language. The hybrid mobile application is available in two languages: English and French. This is to make it accessible to both French and English-speaking people who wish to use the application. The application introduces learners to lessons on the Twi numbering and alphabet system, as well as parts of speech like nouns, verbs, pronouns, and adjectives. These are presented through an interactive user interface where users can listen to the pronunciation of the words they come across and also complete assessments after their lessons. Using ASR technology, a learner is able to give voice input in Twi during an assessment to test their speaking and pronunciation abilities. The application aside, this core task also allows a user to edit their profile and UI appearance to suit their preference. Finally, the application provides a chat feature where a user can interact with a customer representative agent to answer any questions they may have about the application.

6.2 Functional Requirements Validation

For all functional requirements presented for development, **Table 4** provides information on those that were completed and those that could not be completed.

Table 4: Functional Requirements Validation

Requirement Number	Description	Status
R01	The registration subsystem of the application should allow users to register or sign up for a free account	Completed
R02	The authentication subsystem of the application should allow all users to log into their accounts using their email and passwords to use the application	Completed
R03	The authentication subsystem of the application should allow a user to reset their forgotten passwords	Completed
R04	The translation subsystem of the application should allow users to select their preferred choice of language (English/French) for the application	Completed
R05	The lesson content subsystem of the application should allow users to select a module to learn	Completed
R06	The lesson assessment subsystem of the application should allow users to take an assessment for a completed module	Completed
R07	The lesson content subsystem of the application should allow users to listen to audio pronunciations of text in Twi	Completed
R08	The lesson assessment subsystem of the application should allow users to input speech in Twi or English, or French for assessment	Not Completed
R09	The lesson assessment subsystem of the application should be able to determine the validity of users' speech input in Twi and give feedback on it	Not Completed
R10	The lesson content subsystem of the application should be able to save a user's progress through the various learning stages	Completed
R11	The application should allow users to chat with a customer support agent	Completed
R12	The application should allow users to edit their profile	Completed
R13	The application should allow users to change the appearance of the user interface	Completed

All functional requirements except R08 and R09 were fully completed. The Project limitations sub-section of this paper gives more information on the cause for non-completion.

6.3 Project Limitations

Given the low resource nature of the Twi language, there are no readily available data set for the training of the ASR model, and therefore speech data had to be manually collected. Finding volunteer speakers to record speech data for the system voluntarily was very difficult. In the end, the number of speakers available was not enough (less than ten) to fully implement an accurate ASR system. The recorded accuracy for the trained model was quite low (below 50%); thus, this feature could not be implemented in the main application. However, the application functions correctly, and even without this feature, learners can effectively learn Twi to a decent degree of oral proficiency.

Another challenge faced during the development process was time. The time constraints faced caused some proposed features of the application to be left uncompleted. For example, only one level (beginner) could be completed for learning in this version of the application, whereas the application was intended to have multiple levels to increase learners' proficiency.

6.4 Recommendation for Future Works

Moving forward, future versions could improve the current application's performance and provide better service to learners by adding the below-stated features.

(1) More accurate pronunciation scoring ASR system.

(2) Development of native applications for IOS and Android for better performance improved user experience and unlimited access to device features.

(3) Introduction of intermediate and professional levels.

(4) Visual representation of words to aid learner absorption.

(5) Classroom feature where learners can interact among themselves and create a support community.

References

[1] Ayo Bamgbose. 1999. African language development and language planning. Social Dynamics 25, 1 (1999), 13–30.

[2] Masud Chand. 2016. Leveraging the Diaspora for Africa's economic development. Journal of African Business 17, 3 (2016), 273–290.

[3] Nick C Ellis and Pamela SH Bogart. 2007. Speech and Language Technology in Education: the perspective from SLA research and practice. In Workshop on Speech and Language Technology in Education.

[4] Maxine Eskenazi. 2009. An overview of spoken language technology for education. Speech Communication 51, 10 (2009), 832–844.

[5] Google Firebase. [n.d.]. Documentation nbsp;|nbsp; Firebase. https://firebase.google.com/docs/

[6] Horacio Franco, Harry Bratt, Romain Rossier, Venkata Rao Gadde, Elizabeth Shriberg, Victor Abrash, and Kristin Precoda. 2010. EduSpeak®: A speech recognition and pronunciation scoring toolkit for computer-aided language learning applications. Language Testing 27, 3 (2010), 401–418.

[7] Ionic. [n.d.]. Open-Source UI Toolkit to Create Your Own Mobile or Desktop Apps. https://ionicframework.com/docs

[8] Ikechukwu Anthony Kanu. 2013. African identity and the emergence of globalisation. American International Journal of Contemporary Research 3, 6 (2013), 34–42.

[9] James Chamwada Kigamwa. 2016. SALIENT FACTORS IN THE ACQUISITION AND MAINTENANCE OF HERITAGE LANGUAGES AMONG US-BORN CHILDREN OF AFRICAN IMMIGRANTS. Linguistica Online 17 (2016).

[10] John M Norris and Lourdes Ortega. 2000. Effectiveness of L2 instruction: A research synthesis and quantitative meta-analysis. Language learning 50, 3

(2000), 417–528.

[11] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely. 2011. The Kaldi Speech Recognition Toolkit. IEEE Signal Processing Society. IEEE Catalog No.: CFP11SRW-USB.

[12] Antje Röder, Mark Ward, Carmen Frese, and Elizabeth Sánchez. 2015. New Irish families: Successes and challenges in heritage language acquisition for second generation migrant children. Dublin: Trinity College Dublin (2015).

[13] Lourdes C Rovira. 2008. The relationship between language and identity. The use of the home language as a human right of the immigrant. REMHU-Revista Interdisciplinar da Mobilidade Humana 16, 31 (2008), 63–81.

[14] K Samudravijaya. 2004. Automatic Speech Recognition. Tata Institute of Fundamental Research Archives (2004).

[15] Ian Sommerville. 2011. Software engineering 9th Edition. ISBN-10 137035152 (2011), 18.

[16] Joost Van Doremalen, Lou Boves, Jozef Colpaert, Catia Cucchiarini, and Helmer Strik. 2016. Evaluating automatic speech recognition-based language learning systems: A case study. Computer Assisted Language Learning 29, 4 (2016), 833–851.

[17] JJHC van Doremalen, Helmer Strik, and Catia Cucchiarini. 2009. Utterance verification in language learning applications. (2009).

[18] Silke M Witt and Steve J Young. 2000. Phone-level pronunciation scoring and assessment for interactive language learning. Speech communication 30, 2-3 (2000), 95–108.