



ASHESI UNIVERSITY

**THE USE OF VOICE INTERFACE SYSTEMS TO AUGMENT
SELLING AND BUYING ON UNIVERSITY CAMPUSES**

APPLIED PROJECT

B.Sc. Computer Science

**Emmanuel Jojoe Ainoo
2019**

ASHESI UNIVERSITY

THE USE OF VOICE INTERFACE SYSTEMS TO AUGMENT SELLING AND BUYING ON UNIVERSITY CAMPUSES

APPLIED PROJECT

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

Emmanuel Jojoe Ainoo
2019

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:

.....

Date:

.....

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 College.

Supervisor's Signature:

.....

Supervisor's Name:

.....

Date:

.....

ACKNOWLEDGEMENT

I would firstly like to honor God, my family, my supportive friends and lecturers who stood by me and assisted me in building this project and achieving my goals.

I would like to express my intense gratitude to my project supervisor Mr. Dennis Owusu, for his involvement in the project from the very start till the end, instrumental guidance, suggestions and criticisms through the successful completion of this project.

I also wish to thank my parents for their support in all areas. They provided resources especially with regards to payment of subscriptions for particular application programming interfaces (API's) used in building the project.

Lastly, this project would not have been a success if the feedback from the various faculty during progress presentations were not present. The feedback guided me and drew insights so concerns I may never have thought of, and for that I thank them all.

ABSTRACT

Our everyday shopping lives have been significantly augmented by rapid advances in Commerce Technologies. Buying and Selling of items as well as payments are currently mostly done online using advanced technologies, which have sped up shopping activities and made lives more comfortable for consumers.

In spite of the rise in e-commerce, e-business, internet communication, and payment systems, physical cash is still popularly used in buying and selling of items on the University Campus in Ghana. There is no issue with that. However, a lot more problems surface when the seller has to give change to the buyer. The inconvenience of getting change for buyers especially when the change amount is quite small, such as GHC 20 pesewas is becoming menacing. This project thus seeks to reduce the inconveniency associated with change collection during buying and selling by allowing students and staff to accumulate their change amounts electronically through voice interface systems.

This paper presents a comprehensive implementation of the OkNsesa system which comprises of speaker recognition and speech recognition components to allow users update their electronic accounts using voice commands. The key advantage of using voice interfaces is the ability to automatically log users into the system by recognizing who the user is from his voice.

TABLE OF CONTENT

DECLARATION	I
ACKNOWLEDGEMENT	II
ABSTRACT.....	III
LIST OF FIGURES	VI
LIST OF TABLES.....	VI
CHAPTER 1: INTRODUCTION	1
1.1 Aim	1
1.2 Background and Motivation	1
1.3 Problem Statement	3
1.4 Proposed Project	3
1.5 Contributions	4
1.6 Related Works.....	4
CHAPTER 2: REQUIREMENTS	8
2.1 Requirements Gathering	8
2.2 Functional Requirements	8
2.3 Interface Requirements	10
2.4 Non-Functional Requirements	11
CHAPTER 3: ARCHITECTURE AND DESIGN	13
3.1 Design and Analysis	13
3.2 System Modelling	17
CHAPTER 4: IMPLEMENTATION	21
4.1 System Components	21
4.2 Speaker Recognition	21
4.3 Speech Recognition	25

4.4	Database.....	29
4.5	Interface	29
4.6	Implementation Technologies.....	31
4.7	Implementation Issues	33
CHAPTER 5: TESTING AND RESULTS		35
5.1	Overview.....	35
5.2	Unit Testing	35
5.3	Case Testing.....	38
5.4	End-To-End Testing.....	40
5.5	Usability Testing.....	42
CHAPTER 6: CONCLUSIONS AND FUTURE WORKS		44
6.1	Conclusion	44
6.2	Recommendations and Future Works.....	44
REFERENCES		47
APPENDICES		49
Appendix A – Requirements Gathering.....		49
Appendix B – Error Handling.....		50

LIST OF FIGURES

Figure 3.1 High Level Architecture	14
Figure 3.2 System Architecture	16
Figure 3.3 Use-Case Diagram.....	17
Figure 3.4 Account Update Activity	18
Figure 3.5 Seller Confirm Update.....	18
Figure 3.6 Sequence Diagram Account Update.....	19
Figure 3.7 Inserting into Database	19
Figure 3.8 ER Diagram	20
Figure 4.1 Identification Enrollment Period	22
Figure 4.2 Enrolled User.....	22
Figure 4.3 Identification.....	23
Figure 4.4 Verification Enrollment Period	24
Figure 4.5 Enrolled Verification User	24
Figure 4.6 Verification.....	25
Figure 4.7 Request Format [17]	27
Figure 4.8 Response Format [17].....	27
Figure 4.9 Recognizing Speech	28
Figure 4.10 Command Extraction.....	29
Figure 4.11 Database Updates	29
Figure 4.12 Voice Input Interface.....	30
Figure 4.13 Display Page Interface.....	31

LIST OF TABLES

Table 5.1 Speaker Identification Group Cases	36
Table 5.2 Speaker Verification Group Cases.....	36
Table 5.3 Microphone Input Group Cases	36
Table 5.4 Speech Recognition Group Cases.....	37
Table 5.5 Command Extraction Group Cases.....	37
Table 5.6 Database Queries Group Cases.....	37
Table 5.7 Case Testing.....	38
Table 5.8 End to End Testing	41
Table 5.9 Usability Testing.....	43

Chapter 1: Introduction

This Chapter introduces background of the problem associated with change collection and the proposed solution to solving the problem.

1.1 Aim

The project seeks to explore the use of Voice User Interfaces (VUIs) to augment buying and selling in Ashesi University. The aim of the project is to build a Natural Language Processing (NLP) System called OkNsesa, that uses voice features and commands to allow users accumulate change amount electronically. Users are recognized and verified to access their account based on their voice features. From there, they can now use voice commands such as “*Update my account with GHC 2*”, to update their respective change accounts and use the accumulated money for other purchases.

1.2 Background and Motivation

The population of Ashesi University continues to increase yearly [19]. Consequently, buying and selling on campus increases. Out of 30 students who use cash in purchasing items, 30 of them can recall an instance where they have had to leave some amount of change with the seller. The issue aggravates overtime as the institution expands violently. There have been a number of complaints from students especially, to authorities but nothing seems to be done about it. It has swollen to an extent where people have accepted the problem and live with its consequences as a norm.

An approach to solve this problem, proposed by the shop owners themselves, is a paper-receipt containing the change amount and date of purchase to buyers, with the idea

that buyers can use the chit later for another purchase or to directly receive the cash equivalent. These papers however get destroyed or lost most of the time and are rendered useless.

Certain times, sellers either keep the change or merely tell buyers to come for their change later. After weeks, both buyers and sellers forget this day ever been mentioned or the exact amount to be recollected as change.

In more frustrating situations, sellers do not collect the full amount needed to buy a product to avoid needing to give out change. For instance, a seller may collect GHC 2.00 for an item that costs GHC 2.40. Other occasions, buyers are forced to purchase extra items or purchase items in a manner, such that change collection, is avoided. The current approaches such as the paper-receipt to handling these problems are not satisfactory and tend to create auxiliary problems.

Both buyers and sellers complain of losing change. This negatively impacts customer experience, and in effect, cripple the business. Also, there is a considerable sum of money that are just lost, mainly when the lost receipts are not accounted for.

Another inept way in which shops try to tackle the problem is by keeping a record of change to give to buyers using a book. A lot more drawbacks with this approach stem from the issues with the traditional file approach of record keeping such as the inefficiency involved in searching for a particular record.

A few shops make use of Graphical User Interfaces (GUI's) generally for buying and selling. The drawback of these systems is that they do not directly account for providing change for buyers and in other situations tend to slow transaction processes as demonstrated in [4].

1.3 Problem Statement

Though there may be no issues with regards to purchasing with physical cash, a lot more problems surface when the seller has to give change to the buyer. The inconvenience of getting change for buyers especially when the change amount is quite small, such as GHC 1.50 pesewas, 20 pesewas, etc. is nerve-wracking.

1.4 Proposed Project

The OkNsesa project proposes a system allow sellers to give buyers electronic change through a voice interface and thus enhance the buying and selling of items on the Ashesi campus.

A voice interface may be a quick and efficient way for sellers to provide change [7]. The goal is for buyers' change to be converted to electronic currency which can be used for other purchases to avoid change loss. This process of converting change to electronic money during a purchase must be fast enough to be worth it. Purchasing food from the campus cafeterias already frustrate buyers with the length of the queues. Both sellers and buyers may be reluctant to waste precious time on 20 or 50 pesewas change. In the situation where the seller may not have the exact change amount to give to the buyer, the buyer could say, for instance, *"Update my account with 20 pesewas change"* and their electronic account gets credited with the said amount. All the seller has to do is confirm the transaction of the change, at the particular instance, to approve the crediting.

I assume it will be faster and more convenient than having to navigate a graphical user interface to perform the same task [12]. Part of the speed and convenience of voice interfaces lies in the fact that voice instructions can simultaneously be used for user identification and authorization and thus skipping the process of logging in with a GUI.

1.5 Contributions

The contributions of the project include:

1. A voice interface system for buying and selling food on the Ashesi Campus
2. Electronic change that can be integrated with other electronic currencies
3. An interactive voice system to allow buyers query transaction details
4. An electronic piggy bank for change
5. Prim research on the problem of change collection

1.6 Related Works

General research and reporting on the issue of difficulty in getting change for buyers have not been formally and adequately contributed to academic knowledge. Thus, a good number of systems have not been developed as an approach to confronting the issue. However, concerning the principal technologies, i.e. Spoken Language Understanding (SLU), Speaker Identification and Verification and Speech Recognition involved in developing a such a system, there has been a great deal works contributed.

An early study presented by [5], demonstrate a speaker identification system that can be embedded in a large range of online web applications for conversations. The aim of the paper was to present a system that answers the question “*who is speaking now?*” [5].

In implementing the speaker detection system, the authors organized the system into training and testing modes. In the training mode, the goal was to enroll users together with audios of them speaking, as inputs to train a Gaussian Mixture Model (GMM). They add a noise-model that takes audio of speaker speaking with noisy background as part of the training model. The testing mode, which is also the recognition mode, aimed to follow the same processing steps as the training, and recognize a speaker from the processed audio amongst enrolled users from the database [5].

They evaluated the system using the Diarization Error Rate (DER) and concluded that the system works well and maintains a good accuracy with users up to about twenty (20).

[5] presents the necessary steps, organization and questions needed to be answered in understanding and undertaking the process of Speaker Identification as a very important component of the OkNsesa project [5].

The concept of using more intuitive commands for speech-based systems was advanced by [6] with a study using a voice mail system. The aim was to point out that speech interfaces for control are much more efficient designs than touch-tone control. The system used the concepts of more intuitive commands such as help, previous, replay, yes and no to allow users interact the system more easily [6].

The overall thesis of their research was to prove that voice control has more benefits compared to DTMF signals for Telephone answering machines and network-based voice mail systems. They generally present the concept of speech recognition as field of study, and the underlying technology can overcome the issues of DTMF signals which include the inability of every phone to generate DTMF signals and the fact that users need memory cards because of numerous number codes [6].

The develop their prototypes after three iterations, where in the first iteration, they sought to introduce more intuitive commands. After some research and analysis, they realized certain commands are more likely to be used by users than some others. The goal of the second iteration was to improve efficiency, by investigating the length and content of voice menus, and the speaking rate of system announcements. The speaking rate of announcements were improved by allowing a professional to re-record the announcements and reduced the length of voice menus boosting the overall efficiency of the system. In the third iteration, the authors extended the time to speak, as a means of preventing users from barging to many commands within the 1 second time frame [6].

With acceptance testing using focus groups of participants from different countries, the authors were able to verify their design goal of how voice control overcomes the limitations of DMTF signals [6].

A typical example how voice systems can be applied in commerce was designed in [7]. They used a couple of Speech Recognition Systems (SRS) including IBM's Watson speech-to-text to demonstrate purchasing of items from an ecommerce platform. The architecture of their project was the integration of the SRS with the e-commerce web application and exhibiting how voice commands can be used to select categories and products [7].

To allow more accessibility and flexibility to websites the authors of the paper decided to use Speech Recognition for an ecommerce website control such that visually impaired can use voice features to access website features [7].

The paper uses IBM Watson's speech-to-text technology which uses intelligent and advanced machine learning concepts was integrated with an e-commerce website and works as follows: Users input a voice command, the IBM watson's speech-to-text converts the speech to text, it then extracts the intention of the user from the text, then allows the intention to perform a variety of actions ranging from searching to writing text.

After development, embedding SRS in web applications to promote multi-tasking and easier interaction, users could now use natural language to shop for items [7].

The idea of speaker recognition was extended by [15] such that it can be used for real time applications. The project presents the general overview and architecture of speaker recognition and its requirements when it has to be live. The study uses a Gaussian Mixture Model algorithm for modelling of speech and MPEG 3-layer compression on mel frequency cepstral coefficients for feature extraction. They experiment the system

live with real speakers already added to their database, experiment with background noises and uses of thresholding to avoid invalid speaker detection [15].

After the study, the authors concluded that speaker can improve speaking time and can be done in real time [15].

[2] from IBM Corporation present an approach of using pattern recognition for speaker verification. It uses the concept of pattern recognition using phrases that carry speaker-dependent information to verify that a particular speaker said a specific phrase twice. The authors concluded from the study that pattern recognition for speaker verification has proven to be successful with better accuracy than other conventional methods such as the Adeline Procedure [2].

Chapter 2: Requirements

This Chapter discusses the requirement specifications of building the system and how these requirements were gathered and analyzed to specify how the system should be built.

2.1 Requirements Gathering

OkNsesa is a Voice Interface System that augments buying and selling by providing an avenue to handle the difficulty in change collection. The idea is that the system would allow buyers to update an electronic account with the change amount given that the seller confirms the transaction.

The process involves buyers automatically logging onto the system based on features derived from their voice. The buyer says for instance *“Add 1 cedi to my account”*. A confirmation message is immediately sent to the seller, to confirm the crediting. The seller confirms, and the buyer’s account gets credited. The buyer can then use the credited amount for other purchases. Such a system that involves direct interactions with real people needs to meet certain requirements. Requirements specify how the project should be designed and built to meet user expectations and thus a very essential part of the project.

The stage of specifying requirements involved interactions with students, staff, and faculty as well as workers of Cafeterias and Shops both on and off campus, as they all form part of the user base. Requirements for the OkNsesa Project were gathered through informal interviews and observations. These requirements were categorized into functional, non-functional and interface requirements.

2.2 Functional Requirements

These requirements underline the functionalities of the system that is required to solve the problem discussed above. The functional requirements of the system could be further categorized into User and System requirements.

A. User Requirements

This user requirements summarizes the high-level needs of the user that have been transformed into achievable requirements based on direct interactions with the users of the system. These requirements are:

1. A buyer should be able to log in onto the system with his/her voice
2. A buyer should be able to add electronic change amount to his/her account with voice commands
3. A buyer should be able to remove electronic change amount to his/her account with voice commands.
4. A buyer should be able to check his/her past transactions using voice commands
5. A buyer should be able to update his account details
6. A buyer should be able to log out from the system
7. A buyer should be able to undo a transaction
8. A buyer should be able to be enrolled for Speaker Identification
9. A buyer should be able to be enrolled for Speaker Verification
10. A seller should be able to confirm a buyer's change transaction
11. A seller should be able to request past transactions and records

B. System Requirements

These requirements are the specifications of the system, including what it should do and what it should not. These requirements are:

1. The system should have a voice interface that allows buyers to record their voice commands
2. The system should have a speaker recognition system that automatically logs buyers into the system
3. The system should have an electronic change account for each buyer
4. The system shall update an electronic change account based on a change amount figure given in a speech by a buyer
5. The system should be able to identify buyers from voice features
6. The system should be able to verify buyers based on voice features
7. The system should be able to recognize the buyer's speech to perform an action
8. The system should be able to extract the intent of the buyer from his/her speech
9. The system would not update an account until the seller has confirmed the update
10. The system would not allow a transaction to be undone if not approved by seller
11. The system shall perform rigorous security checks before allowing updates
12. The system should reject any user that does not pass speaker verification

2.3 Interface Requirements

These requirements specify the system's interface of which the user interacts with.

They include:

1. The system shall provide a user-friendly interface that allows both buyers and sellers to record their voice commands
2. The system shall provide an interface for buyers to credit their account with change
3. The system shall provide an interface for sellers to confirm the buyer's crediting of the account
4. The system shall provide an interface for buyers to check past transactions
5. The system shall provide an interface for sellers to review recent transactions.
6. The system shall provide an interface for buyers to reuse credited account for other purchases
7. The system shall provide an interface to confirm users' login process when there are mismatches.

2.4 Non-Functional Requirements

These requirements emphasize the quality attributes of the system to make the system usable. They include:

1. Security: The system would employ features to ensure that the information that is stored in the system is safeguarded from both internal and external attacks.
2. Available: The system would be available to use 24 hours a day, seven days a week.
3. Confidentiality: All user information shall only be accessed by authorized personnel.
4. Usability: System shall be easy to use, and consistency with regards to user actions would be enforced.
5. Reliability: System shall consistently perform specified functions without failure.

6. Safety: System shall bring no harm to users.
7. Integrity: System shall ensure that user data is maintained without any corruption.
8. Maintainability: Problems with the system shall easily be fixed.
9. Portability: The system shall deploy on both desktop and mobile platforms.

Chapter 3: Architecture and Design

This Chapter presents the structure and design of the OkNsesa System. It contains all the different components of the system and how they communicate in a single architecture as well diagrams of the system design.

3.1 Design and Analysis

The OkNsesa System is an application that allows buyers to store change amounts and reuse for later purchases. The system is designed to provide a platform where buyers can record their commands as voice input through a microphone.

A. System Overview

The application allows buyers to login automatically when they speak, through a large number of voice features. An important phase at this part of the application is the speaker verification phase. We do not want a buyer logging in to another buyer's account because of confused voice feature recognition or accents, as such the speaker verification phase is very crucial to re-confirm that an identified user is the actual user from his voice and features and thus can access can make changes to his/her account.

Given that the user has successfully logged on to the system, they can then use voice commands to make updates to their electronic change accounts. Buyers say a specific voice command *"Please add 20 pesewas to my account"*. Their account should be credited with the said amount by the system. This process involves the system recognizing the speech and processing it as text. Extracting what the speaker intends to do from the collected speech, whether adding or removing from his/her account and determining exact amount.

Assuming the buyer has accumulated change enough to make other purchases, he/she can then use the money on the account in any of the shops on campus for other items.

Though the voice commands passed to the system may be syntax or structure-specific, the system is flexible enough to allow certain variations to the commands, for example “*subtract 20 pesewas from my account*” instead of “*remove 20 pesewas to my account*”.

Sellers receive a confirmation popup, to confirm whether, the buyer’s request to update his/her change account with a particular amount is accurate. Thus, the updating process must be done in the presence of both buyers and sellers, in order to solve issues of over or under-estimate of change amounts. This functionality is however not implemented yet and presented as one of the future works elements of the project.

B. High Level Architecture of the System

The system is organized into four main components: Speaker Recognition System, Speech Recognition System, Change Account System and an Interactive Interface. These four components constitute the process of a buyer, automatically been logged onto the system with voice features and updating his electronic account with a said change amount as described above. The components communicate as a single unit in rendering this functionality to the buyers and sellers. Below is a diagram of the high-level architecture of the system.

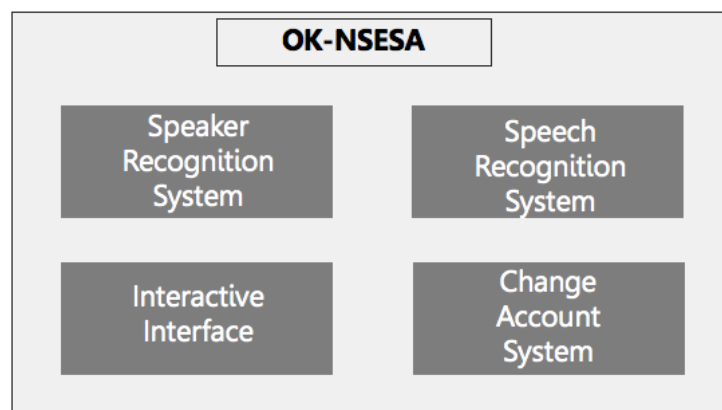


Figure 3.1 High Level Architecture

a. Speaker Recognition

This component is made up of two phases: Identification and Verification. The goal of the component is to identify which buyer spoke, so as to map on to the right buyer account. Simultaneously, from the buyer's speech, he/she would be logged into the system automatically, after identifying and verifying it is the right mapping on to a buyer

b. Speech Recognition

The Speech Recognition component of the system is responsible for understanding what exactly a buyer said, to trigger an action, for instance update account or display transactions. The component, using appropriate optimization algorithms, would process buyers' speech input such that wrong pronunciation and wrong input would be handled to avoid the buyer from repeating comma multiple times.

c. Change Account System

This component is responsible for the managing of buyers' change account with regards to creating, updating and deleting accounts. Buyers can accumulate change amounts and use for later purposes, thus they need to be able to constantly check how much accumulated change they have so far, to know their purchasing power.

d. Interactive Interface

The user needs to communicate with an interface that allows him/her to record speech and issue commands. The interface would consist of microphone for taking voice input from the user.

C. System Architecture

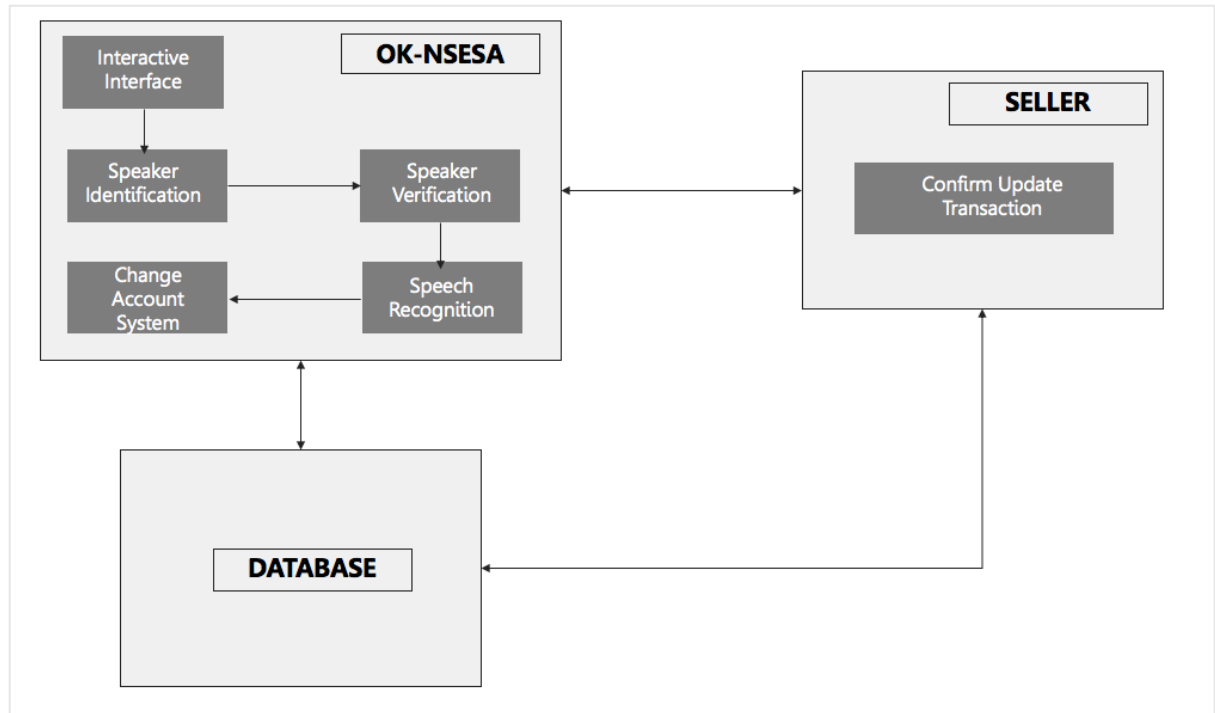


Figure 3.2 System Architecture

D. The Actors of the System

The actors of the system represent the primary users involved directly with the OkNsesa system. There are two principal actors of the OkNsesa System namely:

- a. Buyers (purchase items from shops and cafeterias)
- b. Sellers (sell items from shops and cafeterias)

E. Scenarios

The following represent typical scenarios demonstrating certain functionalities of the system as it interacts with the actors. Three scenarios would be discussed:

- a. After a lot of adding of change to Esi's OkNsesa account, Esi realizes she has GHC 70 in her account. Esi has been wanting to buy a journal for so long but her Stanbic Visa Card has been having issues. Esi then uses her OkNsesa account to purchase the Journal which cost GHC 60 and has never been this excited.
- b. Arthur has a class in 2 minutes but wants to get new notebook before the class.

Arthur plans to not waste time at the Essentials shop. Arthur buys his item, but change has to be given. On a normal day, Arthur would just let the change go because of how much in a hurry he was. Arthur knew that he could say a simple sentence to access his account and update it with the change, which would not less than 5 seconds. Arthur simply adds the change amount to his account and did not have to go through a log in process and thus could make it to class on time.

3.2 System Modelling

A. Use Case Diagram

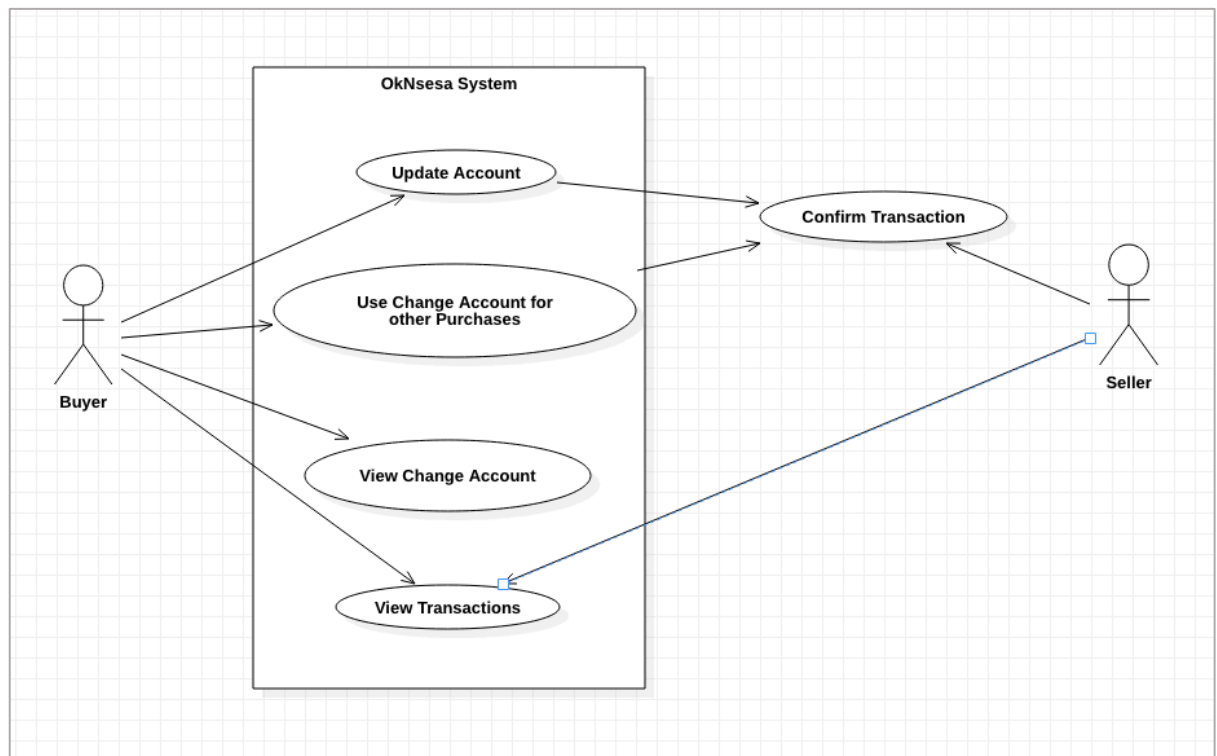


Figure 3.3 Use-Case Diagram

B. Activity Diagram

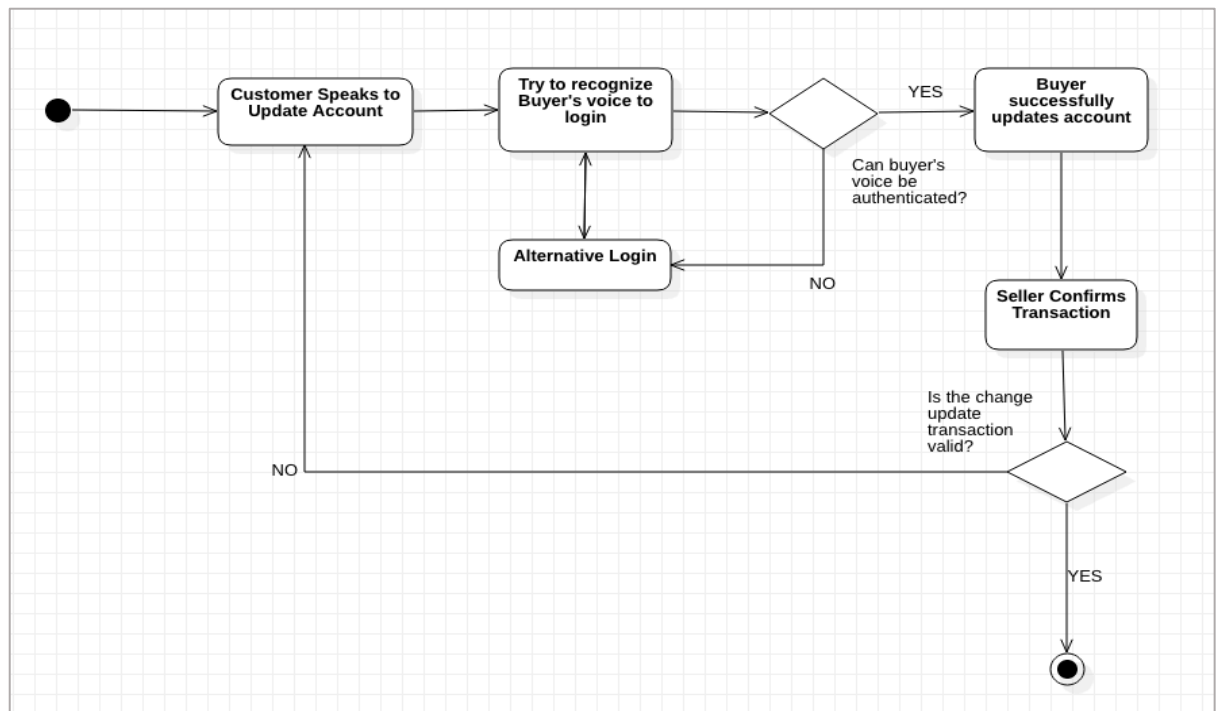


Figure 3.4 Account Update Activity

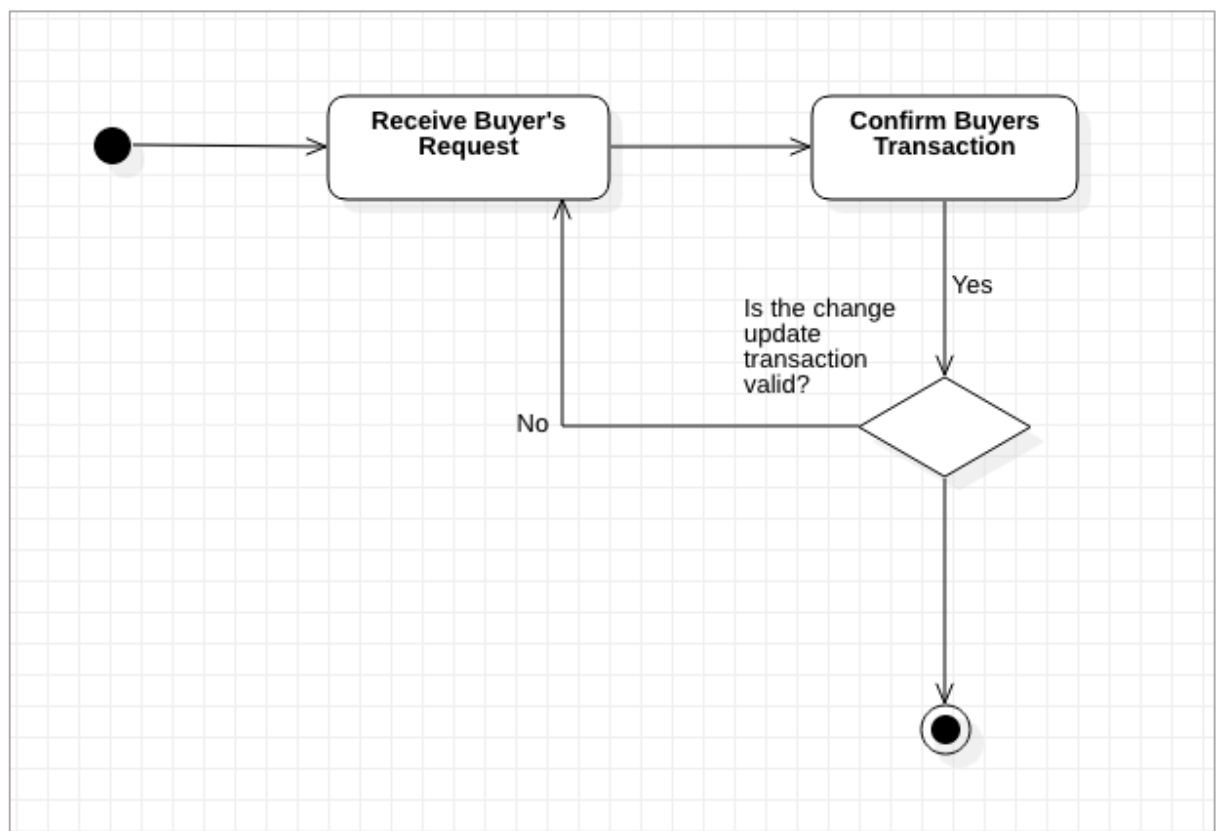


Figure 3.5 Seller Confirm Update

C. Sequence Diagram

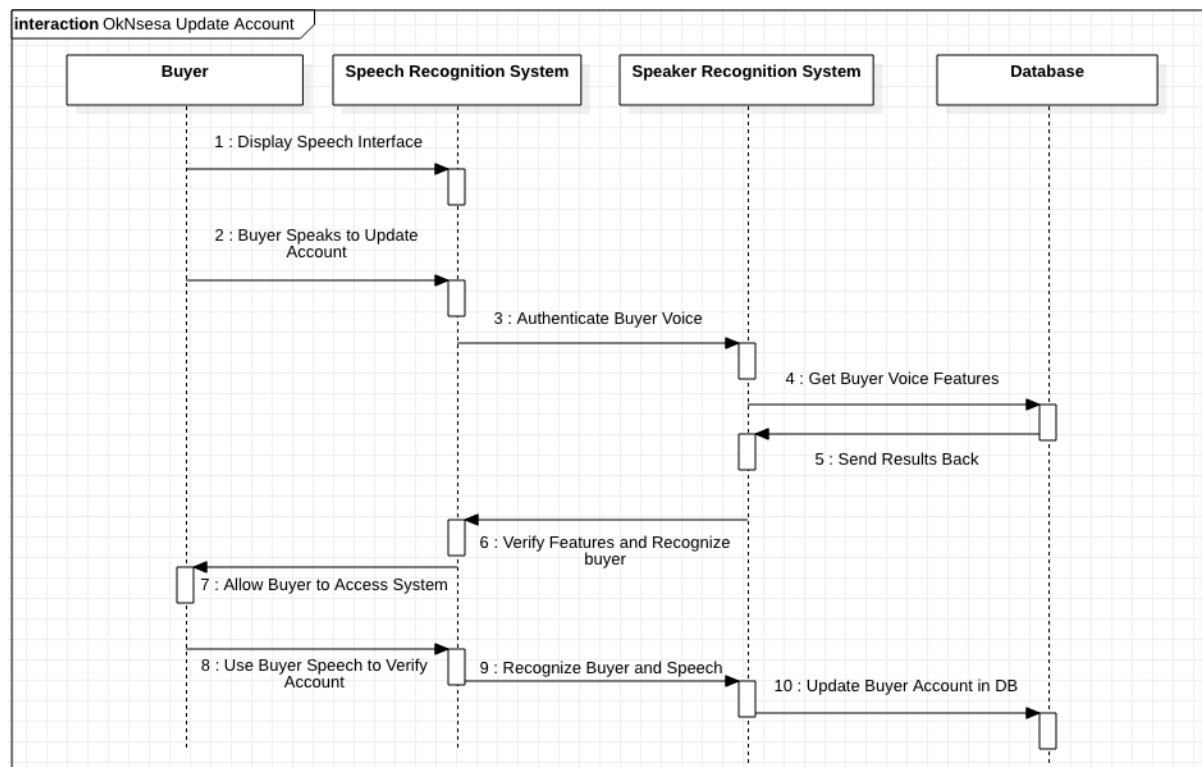


Figure 3.6 Sequence Diagram Account Update

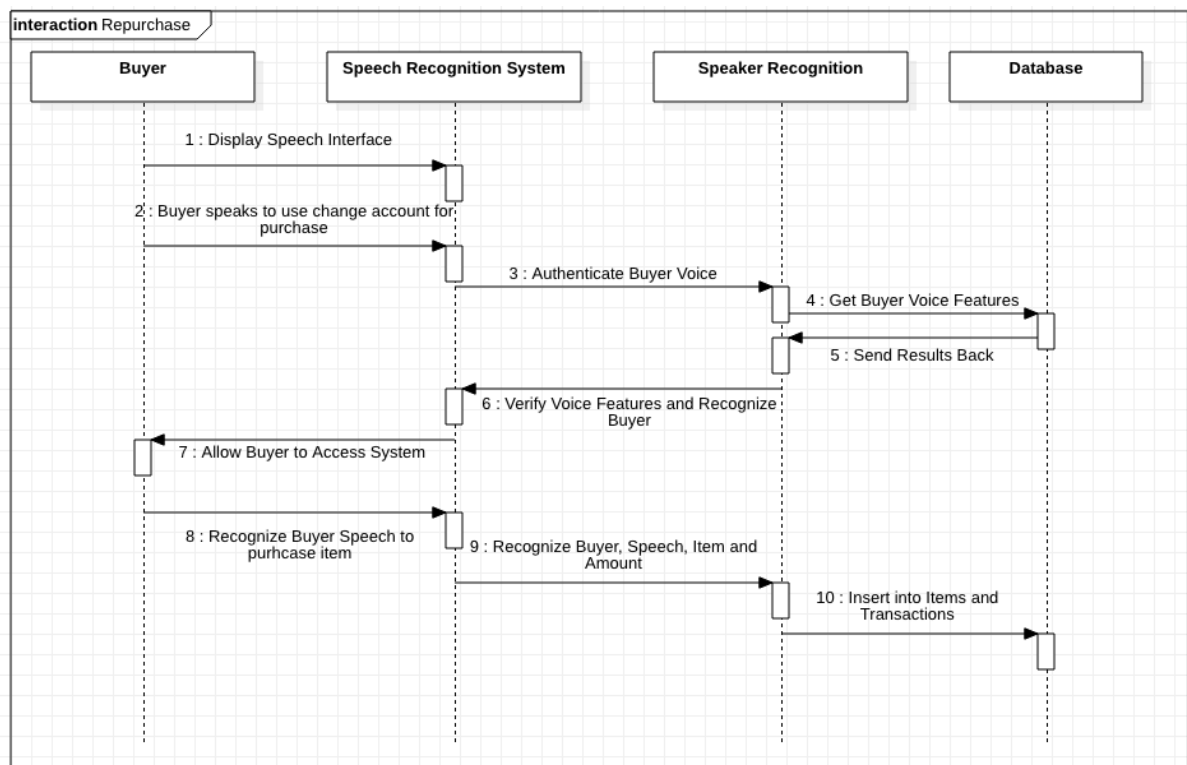


Figure 3.7 Inserting into Database

D. Database Architecture

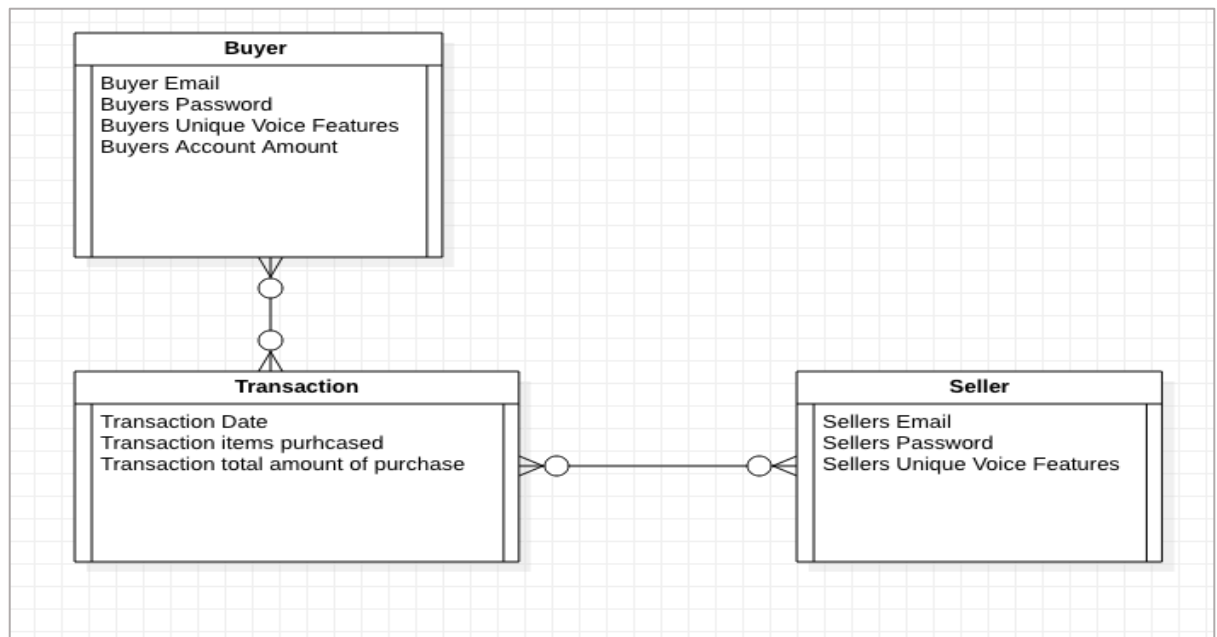


Figure 3.8 ER Diagram

Chapter 4: Implementation

This chapter discusses the details of the implementation of the OkNsesa system and how different components used to build the system interact. The chapter is divided into sections discussing the four major components of the system and the technologies implemented to allow these components function.

4.1 System Components

Since the project is heavily designed with machine cognitive concepts i.e. Speaker Identification, Speaker Verification, Speech Recognition, machine learning libraries prebuilt were used to perform these individual complex tasks were used. The system as explained in the beginning chapter is a platform to allow users to easily accumulate change amount that could have been lost. The core of the system is organized as a process of Speaker Identification, Speaker Verification, Speech Recognition then Account Update.

4.2 Speaker Recognition

Speaker Recognition involves two major tasks Speaker Identification which seeks to find a match of a voice pattern among a set of voice patterns and Speaker verification which seeks to confirm a voice pattern matches a predefined voice pattern [16]. Both Speaker recognition and Identification tasks were completed using Microsoft's Cognitive Services API for speaker recognition [3]. These two separate tasks are executed sequentially, however communicate in a seamless means which would be explained below.

A. Speaker Identification

Speaker Identification is the process recognizing the person speaking in an audio file, given a group of prospective speakers [3]. The input audio is paired against the provided

group of speakers who are pre-enrolled in the system. The speaker identity of the speaker whose voice features match the voice features of the input audio is returned.

There are two basic tasks involved in the Speaker Identification process: Enrollment and Identification.

a. Enrollment

In this task, the speaker's voice is recorded for a time period, in this case 4 minutes, and a number of features are extracted to form a unique voice signature for the specific user. Enrollment for speaker identification is text-independent, which means that there are no restrictions on what the speaker says in the audio [5]. These speakers are enrolled into the system with their recorded voices and unique IDs. At least 45 seconds are required for recognition to work properly [5]. The enrollment times were up to a least 60 seconds

```
Jojoes-MacBook:Cognitive-SpeakerRecognition-Python Jojoe$ python3 Identification
/EnrollProfile.py 3b1ec42c30884c2eadaec6f1975ec0f6 32761929-3a47-4038-9d30-58942
3646f65 "/Users/admin/Desktop/Courses/CS Applied Capstone Project/Project Folder
/capstone-final/Speaker Recognition/Cognitive-SpeakerRecognition-Python/Data/Tra
ined Voices/GladysTrain1.wav" False
Total Enrollment Speech Time = 11.34
Remaining Enrollment Time = 18.66
Speech Time = 11.34
Enrollment Status = Enrolling
```

Figure 4.1 Identification Enrollment Period

```
Jojoes-MacBook:Cognitive-SpeakerRecognition-Python Jojoe$ python3 Identification
/EnrollProfile.py 3b1ec42c30884c2eadaec6f1975ec0f6 32761929-3a47-4038-9d30-58942
3646f65 "/Users/admin/Desktop/Courses/CS Applied Capstone Project/Project Folder
/capstone-final/Speaker Recognition/Cognitive-SpeakerRecognition-Python/Data/Tra
ined Voices/GladysTrain3.wav" False
Total Enrollment Speech Time = 36.11
Remaining Enrollment Time = 0.0
Speech Time = 11.9
Enrollment Status = Enrolled
```

Figure 4.2 Enrolled User

b. Identification

The identification task is the actual cognitive process of identifying the user by

determining a match of voice features from an input voice and voice features of the enrolled speakers. The audio of the unknown speaker, together with the prospective group of speakers, is provided during recognition. The input voice is compared against all enrolled speakers in order to determine whose voice it is, and if there is a match found, the identity of the speaker is returned [13]. If there is no match found, the speaker is asked to re-record the audio.

```
Jojoes-MacBook:Cognitive-SpeakerRecognition-Python Jojoe$ python3 Identification
/IdentifyFile.py 3b1ec42c30884c2eadaec6f1975ec0f6 "/Users/admin/Desktop/Courses/
CS Applied Capstone Project/Project Folder/capstone-final/Speaker Recognition/Co
gnitive-SpeakerRecognition-Python/Data/Trained Verify/GladysVerify.wav" True "06
10dd36-3fe7-44a1-83ab-9327b499cd72,380b5a3f-8a2b-4099-aede-b664e1a5c19c,3ab419dc
-d1f1-44d5-9bb9-9707f2f7240e,726dd798-3991-49c8-ba37-b47dcb7303cf,32761929-3a47-
4038-9d30-589423646f65"
Identified Speaker = 32761929-3a47-4038-9d30-589423646f65
Confidence = High
```

Figure 4.3 Identification

B. Speaker Verification

Speaker Verification is one of the most important phases of the system as it ensures that the right user accesses the right account. Voice has unique features that can be used to recognize a person, just like other biometric features. Using voice as a signal for access control and authentication scenarios has emerged as a new innovative tool for control [3]. The verification process seeks to match a user's voice patterns of a specific said phrase against the user's recoded voice pattern of the same phrase, with the idea that the system would most likely have a higher confidence of the voice match.

The process also involves two tasks: Enrollment and Verification.

a. Enrollment

This enrollment process is a different enrollment process from the Identification enrollment process and are two separate tasks. Unlike enrollment for Identification, enrollment for speaker verification is text-dependent, which means speakers need to choose a specific pass phrase to use during both the

enrollment and verification tasks [2]. In enrollment, the speaker's voice is recorded saying a specific phrase, for example “*My voice is my password, verify me*” then a number of features are extracted, and the chosen phrase is identified. Both extracted features and the chosen phrase form a unique voice signature.

```
[Jojoes-MacBook:Cognitive-SpeakerRecognition-Python Jojoe$ python3 Verification/EnrollProfile.py 3b1ec42c30884c2eadaec6f1975ec0f6 eea3741f-d836-4def-bf61-e76262ab62c3 "/Users/admin/Desktop/Courses/CS Applied Capstone Project/Project Folder/capstone-final/Speaker Recognition/Cognitive-SpeakerRecognition-Python/Data/Trained Verify/GladysVerify.wav"
Enrollments Completed = 1
Remaining Enrollments = 2
Enrollment Status = Enrolling
Enrollment Phrase = my voice is my passport verify me
```

Figure 4.4 Verification Enrollment Period

```
[Jojoes-MacBook:Cognitive-SpeakerRecognition-Python Jojoe$ python3 Verification/EnrollProfile.py 3b1ec42c30884c2eadaec6f1975ec0f6 eea3741f-d836-4def-bf61-e76262ab62c3 "/Users/admin/Desktop/Courses/CS Applied Capstone Project/Project Folder/capstone-final/Speaker Recognition/Cognitive-SpeakerRecognition-Python/Data/Trained Verify/GladysVerify.wav"
Enrollments Completed = 3
Remaining Enrollments = 0
Enrollment Status = Enrolled
Enrollment Phrase = my voice is my passport verify me
```

Figure 4.5 Enrolled Verification User

b. Verification

During the verification, an input voice saying a particular phrase are compared against the enrollment's voice signature and phrase—in order to verify whether or not they are from the same person, and whether correct phrase is being said [2]. This task is a simple accept or reject process, where if the voice features and phrase match the same person, then it is accepted otherwise rejected.

```
Jojoes-MacBook:Cognitive-SpeakerRecognition-Python Jojoe$ python3 Verification/VerifyFile.py 3b1ec42c30884c2eadaec6f1975ec0f6 "/Users/admin/Desktop/Courses/CS Applied Capstone Project/Project Folder/capstone-final/Speaker Recognition/Cognitive-SpeakerRecognition-Python/Data/Trained Verify/GladysVerify.wav" eea3741f-d836-4def-bf61-e76262ab62c3
Verification Result = Accept
Confidence = High
```

Figure 4.6 Verification

The Identification process happens before the Verification phase as shown in the architecture in chapter 3. The identified ID from the identification phase is passed onto the verification phase. And the Verification task is run on that ID as another input voice is taken to verify if the ID is the correct speaker. Once the verification process fails, the speaker is asked to restart from the identification phase. No matter how confidently accurate the system identifies the right speaker, the verification phase still takes place and for various reasons. Some of these reasons include:

- i. Speaker's voice features may change daily as humans grow daily [14]. Thus, speaker's voice features may be very similar to another speaker's over some time. This means that the system may identify a wrong speaker or ID for a given input voice. The verification stage uses a much more confident voice signature as the same phrase pattern is being compared and thus, the wrongly identified speaker would not get through the verification phase.
- ii. The system deals with people's actual money. And thus, one would expect thieves to try and use someone's account for fraudulent purchases. Thus, for security reasons to make sure one cannot falsely access another user's account by tweaking his/her voice to pass the identification phase, verification is needed.

4.3 Speech Recognition

Speech Recognition is the task recognizing what a speaker said in the form of text [8].

The core of OkNsesa has to do with voice commands in allowing students/staff to add

change amounts to their respective accounts, thus being able to extract what they say from a microphone is key. The Google Speech Recognition API wrapped in a python speech recognition library was used for the task of speech recognition in this project [17]

A. Speech Recognition Systems (SRS)

Over the years of advance progress in Natural Language Processing, SRS have become rather common and highly optimized to handle various forms of voice inputs. Some of these SRS systems include Google's Speech-to-Text API, IBM Watson Speech-to-Text, Microsoft's Bing Voice Recognition and a lot of others. SRS's have been embedded in to applications such as call routing, voice dialing, searching etc [12]. Speech recognition systems are speaker-independent systems that do not require any form of enrollment but simply process voice input and deduce the most accurate text transcription of the voice input [4].

The task of transcribing what the user spoke is an essential part of the OkNsesa system, where the system needs to be able to recognize the buyer's speech to know the exact actions intended by the user [9].

B. Google Speech Recognition

For this project Google's Speech Recognition API was used for the Speech Recognition component of the system. Google's API was chosen because firstly it is free and can be used without any paid subscriptions, it is very fast and returns responses within a second, it has a high accuracy for transcribing audio to text according to popular reviews and it has a wider community of discussion forums where issues can be posted, and they would be resolved [17].

The OkNsesa system can process up to a minute of audio input sent to Google's Speech-to-Text API in a synchronous request. Once the API finishes processing and recognizing the audio within a second, it returns a response.

```
{
  "config": {
    "encoding": "LINEAR16",
    "sampleRateHertz": 16000,
    "languageCode": "en-US",
  },
  "audio": {
    "uri": "gs://bucket-name/path_to_audio_file"
  }
}
```

Figure 4.7 Request Format [17]

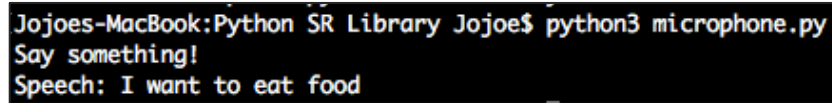
This is a sample of the request made with the exact Uri of the audio file and required parameters for processing the audio sent to Google's API [17]

```
{
  "name": "6212202767953098955",
  "metadata": {
    "@type": "type.googleapis.com/google.cloud.speech.v1.LongRunningRecognizeMetadata",
    "progressPercent": 100,
    "startTime": "2017-07-24T10:21:22.013650Z",
    "lastUpdateTime": "2017-07-24T10:21:45.278630Z"
  },
  "done": true,
  "response": {
    "@type": "type.googleapis.com/google.cloud.speech.v1.LongRunningRecognizeResponse",
    "results": [
      {
        "alternatives": [
          {
            "transcript": "Four score and twenty...(etc)...",
            "confidence": 0.97186122,
            "words": [
              {
                "startTime": "1.300s",
                "endTime": "1.400s",
                "word": "Four"
              },
              {
                "startTime": "1.400s",
                "endTime": "1.600s",
                "word": "score"
              },
              {
                "startTime": "1.600s",
                "endTime": "1.600s",
                "word": "and"
              }
            ]
          }
        ]
      }
    ]
  }
}
```

Figure 4.8 Response Format [17]

This is an example of the response sent by the API containing the transcribed word and timestamp within the audio input. It returns the transcript and confidence level indication

how accurate the particular recognition was, and these are the needed values to extract the command from the transcript.

A terminal window with a black background and white text. The text shows a command prompt, a command to run a Python script, a prompt for the user to say something, and the resulting speech recognition output.

```
Jojoes-MacBook:Python SR Library Jojoe$ python3 microphone.py
Say something!
Speech: I want to eat food
```

Figure 4.9 Recognizing Speech

C. Command Extraction

There are two actions the user could perform as at this stage of the OkNsesa System and these are:

1. Add to existing account
2. Remove from existing account

These actions need to be extracted from the user's transcript derived from the audio input.

A simple algorithm was written to perform this extraction. Since the actions were only two, it did not require a complex machine learning algorithm for processing the Intent of the text, as it would slow the system at this point. However, as the application expands and a lot more users are enrolled, a more intelligent algorithm needs to be implemented.

An intelligent searching algorithm that looks for the specific “*add*” or “*remove*” commands and synonyms of these words as well handling duplicate words was implemented. Supposing a user says “*Please, add GHC 20 to my account*”, the action “*add*” is extracted and the amount GHC 20 is extracted too. These are the most important elements of every transcript as this stage of the OkNsesa system.

```
===== Extracting Commands from Speech =====  
Speech: please add 20 cedis to my account please  
['add', 20.0]
```

Figure 4.10 Command Extraction

4.4 Database

All enrollment data together with user profiles for Speaker recognition are stored on the Microsoft Cognitive Services API online and could be managed easily from a web interface. Python's SQLite was used to create a database to manage buyer's accumulated change and other buyer's details. These two databases are connected in the main script where ID's are matched to indicate that ID "1" in the user account Database, is the same as ID "0610dd36-3fe7-44a1-83ab-9327b499cd72" for the enrolled speaker.

Therefore, for every transaction or account update, the accumulated account on the database of the specific user gets updated.

```
===== Database Account Updates =====  
Extracted New Amount: 20160  
Current New Amount: 20180.0  
Successfully Updated Database  
Jojoes-MacBook:Cognitive-SpeakerRecognition-Python Jojoe$
```

Figure 4.11 Database Updates

4.5 Interface

The interface describes the platform to allow users record their audio input and make changes to their accumulated change amounts. The interface uses a microphone as the input device to extract audio form users for a specific number of seconds. It uses statements to instruct the user on what to do and when to record as shown below.

```

Jojoes-MacBook:Cognitive-SpeakerRecognition-Python Jojoe$ python3 main.py
Opened database successfully
Start Recording Speech (Say Something):
Done Recording Speech
===== Recognizing Speaker =====
Identified Speaker = 0610dd36-3fe7-44a1-83ab-9327b499cd72
Confidence = Normal
===== Recognizing Speech =====
Speaker: Jojoe

Please say the Phrase below as a Verification Step:
My Voice Is My Passport Verify Me

Start Recording Speech (Say Something):
Done Recording Speech
===== Verifying Jojoe =====
Verification Result = Accept
Confidence = High

System has Verified the Speaker: Jojoe
===== Extracting Commands from Speech =====
Speech: please add 20 cedis to my account please
['add', 20.0]

===== Database Account Updates =====
Extracted New Amount: 20160
Current New Amount: 20180.0
Successfully Updated Database
Jojoes-MacBook:Cognitive-SpeakerRecognition-Python Jojoe$ █

```

Figure 4.12 Voice Input Interface

There is another interface that intends to display user's accounts from the database and demonstrate changes to these accounts after said commands. This is a web platform built using flask is used to display the updates on the users' accounts after voice commands. The page pulls information from the SQLite database where the users' updates are reflected. This is not essentially part of the core of the system, as it only serves as a proof of concept that system works.

OK Nsesa			
Speaker and Speech Recognition System for Change Collection			
Users	Old Amount (GHC)	New Amount (GHC)	Location
Jojo	20160	20180	Akornor
Edwin	20000	15000	Akornor
Kwame	20000	20020	Akornor
David	20000	65000	Akornor
Julianne	65000	57300	Akornor
Joseph	57340	57320	Akornor
Gladys	30000	20000	Akornor

Figure 4.13 Display Page Interface

4.6 Implementation Technologies

A. Python

Python is a well-known interpreted high-level language for general purpose programming ranging from Graphical User Interfaces to Machine Learning. In this project, Python was used significantly to develop most of the system's functions. The choice for Python lies in its simplicity in allowing for easy readability and the implementation of functions without the need for typing a lot of code. Also, python has been greatly optimized for machine learning with a lot of pre-built libraries that make it easier to work with the concepts of speaker and speech recognition [1,11].

B. Port Audio

Port Audio is a cross-platform, open-source and very simple library to handle audio. It allows functions of playing audio and recording audio, accessing audio devices as well as processing audio. In this project, the technology that allows users to record their voice

inputs through a microphone is Port Audio. In addition to the fact that Port Audio is easy to use and cross platform, it can be easily integrated with python, as it is a python library to perform various audio processing functions [18].

C. Wave

The Microsoft Speaker Recognition enrollment stages have specific wave type audio requirements and as such the Wave technology that is inbuilt with python was used for this processing. Wave is used firstly for reprocessing the recorded audio as port audio returns to match the requirements for the Microsoft's Speaker Recognition including a 16000-sampling rate and mono channel. Then Wave is also used to save the recorded and processed audio as file to be sent to both the speaker and speech recognition API's [11].

D. File System

In this project a whole lot of file reading and writing were done to allow the system to operate. File processing was used both in processing the audios and processing the responses of the distinct API's. It was serving as an interface between the API's to allow them to communicate as no structured interface has been built to allow this capability yet. There is a "*profile.txt*" file that writes and stores the returned identified user ID from the Speaker Identification phase during a specific transaction. There is a "*verify.txt*" file that writes and stores the verified user ID from the Speaker Verification phase during the current transaction. There is also a "*test.wav*" that stores the recorded audio from the recognition phases at every point in time.

E. SQLite

SQLite is a small, simple, fast, cross-platform and high-reliability SQL database engine for storage. In this project, the SQLite was used to create and process the database for managing the accumulated change amounts of buyers. SQLite was chosen because it could be easily integrated with python and because of its speed in querying.

F. Flask

Flask is a simple web framework written in Python that does not require particular tools or libraries. It was used for building the simple web page that demonstrates the changes in the user's account after voice updates. Flask was used because of its integration with Python and SQLite, and because of the fact that the webpage is not an integral part of the system and does not require the technologies heavy web applications would require [10].

4.7 Implementation Issues

A. Reuse

In building the OkNsesa system, failing to admit the system was constructed by reusing existing components would be thievery. Existing components at different levels as mentioned above were used. Google's Speech Recognition and Microsoft's Speaker Recognition were integrated to build the system. Other prebuilt libraries such as Port Audio and Wave mentioned above, were used for easy processing of audio. The web interface was also implemented by reusing Flask design components with an SQLite backend. Thus, a number of components were reused to fully develop the system.

B. Configuration Management

The project work was broken down into various sections of development and as such the changes needed to be managed. Thus, version management specifically git, was used

to track changes with individual systems on a master repository and later all the various branches were merged and integrated to get the complete OkNsesa system.

C. Host-Target Development

Deployment falls within the range of future works for various reasons such as rigorous testing explained in the last chapter. The project was however developed on a local machine (the host) with a couple of software development platforms such as Atom, Sublime Text Editors, and other calls to API's online. The project is to be deployed on the Ashesi Cafeterias coupled with a microphone. It would still make the API calls to the various Natural Language Processing (NLP) systems involved provided there is access to Internet connectivity.

Chapter 5: Testing and Results

This chapter outlines the various test mechanisms used for determining the effectiveness of the system and how well the system meets its requirements.

5.1 Overview

Testing is a vital aspect of any system development process as it confirms that the developed system meets the needs of its users. Various levels of testing were conducted for the OkNsesa project. These include Unit Testing, Case Testing, End-to-End Testing, and Usability Testing. Unit testing was conducted to analyze the functionality of specific components of the system and results indicated whether the components passed the unit test cases or not. Case Testing was used to test extreme cases of interactions with the system and the results also indicated whether the test case was passed or not. End-to-end testing involved testing the system with actual enrolled users of the system as they go through the entire processes involved. Usability testing was used to understand how easy the system was to use and whether generally the user requirements of the system were met. The end-to-end test and usability involved testing with ten (10) actual enrolled users whereas all other tests involved testing with a wide range of users. OkNsesa was developed and tested incrementally, however all of the tests would be displayed in single tables.

5.2 Unit Testing

In this phase of testing, the most important system requirements were tested to prove the system works or not at individual components levels. The idea was to test individual units of the components of the system with the aim of reducing defects in newly added features.

In undergoing these tests, unit test cases were developed and grouped, and they include testing enrollment for Identification, testing formatting of enrollment audio for identification, testing enrollment for verification, testing microphone recording with Port Audio, testing reading and writing of audio as wav files, testing inserting and updating database etc. These test cases and their results are represented in the table below.

A. Group 1 – Speaker Identification

Table 5.1 Speaker Identification Group Cases

Unit Number	Unit Cases	Value
1.1	Recording training audio data	Pass
1.2	Formatting training audio data	Pass
1.3	Enrollment for user	Pass
1.4	Identification for enrolled user	Pass

B. Group 2 – Speaker Verification

Table 5.2 Speaker Verification Group Cases

Unit Number	Unit Cases	Value
2.1	Recording training audio data plus phrase	Pass
2.2	Formatting training audio data	Pass
2.3	Enrollment for user	Pass
2.4	Verification for enrolled user	Pass

C. Group 3 – Microphone Input

Table 5.3 Microphone Input Group Cases

Unit Number	Unit Cases	Value
3.1	Get audio from microphone input	Pass

3.2	Save audio as wave file	Pass
3.3	Read audio file and play	Pass

D. Group 4 – Speech Recognition

Table 5.4 Speech Recognition Group Cases

Unit Number	Unit Cases	Value
4.1	Recognize Speech from Audio	Pass
4.2	Recognize Speech from Microphone	Pass

E. Group 5 – Command Extraction

Table 5.5 Command Extraction Group Cases

Unit Number	Unit Cases	Value
5.1	Run algorithm for command extraction	Pass
5.2	Extract add command from speech	Pass
5.3	Extract remove command from speech	Pass
5.4	Extract similar commands to add	Pass
5.5	Extract similar commands to remove	Pass

F. Group 6 – Database Queries

Table 5.6 Database Queries Group Cases

Unit Number	Unit Cases	Value
6.1	Inserting into Database	Pass
6.2	Updating Database	Pass
6.3	Database feed to webpage	Pass

6.4	Updating after Command Extraction	Pass
-----	-----------------------------------	------

G. Discussion of Results

We realize from the various tables that all the unit cases have a value of “*pass*”, indicating that the particular test case was passed. For instance, if the case was “*inserting into a database*”, once the system successfully inserts into the database, its value for the test case becomes a pass, indicating a success. The unit cases were designed to match the most relevant requirements of the system that need to work before one can claim the entire OkNsesa system works. This explains why all of the test cases passed successfully providing room to undergo more extreme cases of tests.

5.3 Case Testing

Case testing was not much different from unit testing only that it was used to test rare cases, error cases and since the system would interact with humans, certain typical and odd environments were designed as cases for test. Some of these cases include noise cancellation, sound error, no sound/speech, wrong identification and allowed or disallowed verification, wrong command extraction, testing at the Big Ben cafeteria etc

The cases for developed for the case testing task and their results are displayed in the table below:

A. Case Test

Table 5.7 Case Testing

Case Number	Cases	Value
1	Identification with User with a cold	0
2	Identification with Noise Background	0

3	Noise Cancellation	1
4	Identification with User voice tweak	0
5	Verification with User voice tweak	1
6	Verification with Noise Background	1
7	Verification with User with a cold	1
8	Sound Error	1
9	Out of Context Command	1
10	Wrong User Identification, Allowed Verification	0
11	Wrong User Identification, Disallowed Verification	1
12	No Sound	1
13	No Add or Remove Commands	1
14	No Amount to Add or Remove	0
15	Gibberish Sound Record	1
16	No User Identified	1

B. Discussion of Results

The test cases developed in this phase were a step beyond the requirements and unit test cases in the previous phase of unit testing. Each case was developed and run with a specific value to indicate whether the case passed or failed. A value of 1, indicates the case passed and a value of 0 indicates the case failed.

We realize from the resulted values that Speaker Identification execution in rare cases mostly fails as Cases 1, 2 and 4 have values of 0. The insight from these case failures emphasizes the need for the Speaker verification component of the system, as they prove

that Identification alone may not be good enough to handle all kinds of voice inputs and in all kinds of environment.

However, the cases that passed were mostly Verification cases, as one would not expect the system to allow a user to access an account, with the slightest change in voice features or uncertainty. Issues regarding taking the voice input also passed as error handling techniques were applied to cases for no sound inputs, out of context commands etc.

5.4 End-To-End Testing

End-to-end testing was conducted with ten (10) actual users that wholly test the system from end to end eight (8) times each with settings of no noise or with noise. Each user is test four (4) times with the ADD command, where two (2) of them are with noise and other two are not. The other four (4) tests of the same user are for the REMOVE command, similarly with noise and without noise settings. The “*with-noise*” setting represents a typical live scenario where the cafeteria or shop is noisy with a lot of buyers conversing, making requests and huge amounts noise in the environment. The “*without noise*” setting represents a much quieter and serene environment with less activities. The end -to-end tests are the most important since they are tested by actual users of the system and prove that the system actually works. The tests and results of each are shown in the table below:

A. End to end Tests

Table 5.8 End to End Testing

END-TO-END TESTING								
With Noise					Without Noise			
Users	ADD	ADD	REMOVE	REMOVE	ADD	ADD	REMOVE	REMOVE
Jojobe	1	1	1	0	1	1	1	1
Juliane	1	0	1	1	1	1	1	1
Kwame	0	0	1	1	1	0	1	1
Joseph	1	0	0	1	0	1	1	1
Edwin	1	1	1	0	1	1	1	0
Gladys	1	0	1	1	1	1	1	1
Ben	1	0	0	0	1	1	0	0
Sasu	0	1	1	1	1	0	1	1
Thomas	1	1	1	0	1	1	1	1
Monica	1	1	1	1	1	1	1	1
27/40					34/40			

B. Discussion of Results

Averagely, we realize that the system performs better in a without noise setting as expected since a lot of noises and other sounds would not temper with recognition. The values represent whether the full end-to-end test was successful, with the user finally updating his account or not. In between system running errors and cuts do not constitute to a successful end-to-end success. A value of 1 indicates a success and a value of 0, indicates a failure.

Without noise the system works fully from end-to-end 34 out of 40 times that is 17/20 times. This is a very a good average as it indicates the system may fail less often times in actual use. With noise settings, the system works 27 out of 40 times, which is slightly above average. There could be a number of reasons for these results which would be discussed below. However, the system generally works fairly well with noise as well.

From the table we realize users like Monica, were successful in all tests with and without noise, and on the other hand users, such as Ben had 0's in most tests. The following factors could have influenced these results:

1. Training data. The more accurate and clean the training data the more accurate the recognition. The more the training data for a particular user, the more accurate the recognition. Unlike Ben, users like Monica had good audios for training during enrollment indicating why the system works best in their tests.
2. Test data. Most of the time, the test data may not be as clean as we may expect. In some cases, during failed tests, I would play test audios and I would only hear sounds. This could be because of how close the user's lips are to the microphone or how they position the microphone.
3. Changes in Voice features. Also, users may have had completely different voice features during enrollment and recognition causing the system to deny user access. Users may have either tweaked their voices during training or testing, or used different tones and pitches or might have had colds etc. A million number of reasons could be developed for the occurrence of changes in voice features.

5.5 Usability Testing

The aim of the Usability test was to understand the interaction between the system and potential user. The ten (10) users after undertaking the end-to-end test were asked to answer some usability study including questions such as rate how easy it was to learn to use the system or easy it was to complete tasks. The test was conducted in the form of rating questions, were for each statement, the user gives a rating score out of 5, where 5 is the highest This is represented in the table below.

A. Usability test

Table 5.9 Usability Testing

No	Questions	Rating Score					Average (mode)
		1	2	3	4	5	
1	Easy to use and straightforward	-	-	2	1	7	5
2	Instructions are easy to understand	-	-	-	1	9	5
3	Easy to learn to use	1	3	-	3	3	2,3,5
4	You can operate the system on your own	3	1	-	4	2	4
5	The system can identify you clearly	-	-	2	-	8	5
6	The system can verify you clearly	-	-	-	1	9	5
7	The system can recognize what you said clearly	1	-	5	-	4	3
8	The system adds the exact amount you mentioned	-	-	-	-	10	5
9	The process was fast and seamless	-	-	-	1	9	5
10	I felt comfortable using the system	-	-	8	1	1	3
11	The interface was pleasant	-	-	1	9	-	4
12	Valuable for Cafeterias on campus	-	1	-	1	8	5
13	Valuable for Students	-	-	2	1	7	5
14	Overall I am satisfied with the system	-	-	2	2	6	5

B. Discussion of Results

The results of the usability tests were analyzed by using the mode average on the rating scores. For example, out of the 10 users, we realize that 7 users gave a score of 5, 1 user gave a score of 4 and 2 users gave a score of 3. The average would then be the score of 5, since most users rated it 5.

Looking at the averages, since most of them are the score of 5 especially with easy to use, overall satisfied with the system, and easy to understand, we realize the system is generally user-friendly.

Chapter 6: Conclusions and Future Works

6.1 Conclusion

This paper presents OkNsesa, a system that allows buyers to accumulate change amounts electronically using voice commands. With the OkNsesa system being implemented, buyers would not lose change amounts from purchases with cash that require change collection and buyers can accumulate change amount enough for other purchases. The process of accumulating the change needs to be fast and seamless as such the traditional idea of graphical logins were replaced with the idea of speaker recognition as it can automatically identify a speaker from his/voice features.

The process starts from speaker recognition, where the buyer who wishes to accumulate change is automatically logged onto the system based on features extracted from his voice command to accumulate change. Then, speech recognition comes in, where what the buyer said is processed to extract the buyer's intention of whether to add to his account or remove from his account. The buyer after saying the command sees the reflected change in his account in a simple web page.

The goal of the project and requirements established in Chapter 3 have been successfully met as users can accumulate and manage it however the please.

6.2 Recommendations and Future Works

This project could be greatly improved by the recommendations made below:

A. Rigorous Testing and Experiments

The OkNsesa system implements the idea of Speaker Recognition as a means of a faster process of identification of users and logging them in, as supposed to traditional graphical user interfaces for the same tasks. The idea is based on the thesis that the users

can be automatically logged in with their voice features as it when they are issuing voice commands.

For this concept to be proven, a whole of lot of rigorous experiments of the OkNsesa system needs to be undertaken with respect to time, functionality, speed, latency and other parameters. These experiments would identify most bugs that were unforeseen at this stage of the system and allow the system aid in making a conclusion concerning the idea of speech systems being quicker than graphical methods.

B. Connecting the Change Account to other Credit Accounts

A lot of monetary accounts these days are connected to allow users make more flexible purchases. In Ghana, most payments are now being done through mobile money and visa cards as they connect to a lot more applications.

As a means to reach more users and make the OkNsesa system more flexible, the account component of the system can be integrated with mobile money systems and other payment systems that users are already connected to. This would enable users transfer accumulated change amount to these other payment systems and thus broaden the scope of repurchases they can make with the accumulated change.

C. More flexible Voice Commands and Interactions

Currently, the OkNsesa allows two basic actions from commands from the user, which are to Add some amount to the user's account or Remove some amount from the user's account.

The system can be allowed to take in more commands to allow the system to be flexible enough such as commands for viewing transactions, for making changes to account details like names and other personal information, etc.

D. Further research on the Change Amount Collection problem

As mentioned in the related works section of this paper, the general problem of difficulty in getting change especially when the change amounts are small have not been formally investigated and researched on.

The OkNsesa has provided some awareness of the problem as well some documented research around the problem. This creates a platform for future researchers to dive deeper into the problem and present a lot more ways of handling the problem as a more structured research problem.

References

- [1] A. Boryssenko and N. Herscovici. 2018. Machine Learning for Multiobjective Evolutionary Optimization in Python for EM Problems. In *2018 IEEE International Symposium on Antennas and Propagation USNC/URSI National Radio Science Meeting*, 541–542. DOI:<https://doi.org/10.1109/APUSNCURSINRSM.2018.8609394>
- [2] S. K. Das and W. S. Mohn. 1969. Pattern Recognition in Speaker Verification. In *Proceedings of the November 18-20, 1969, Fall Joint Computer Conference (AFIPS '69 (Fall))*, 721–732. DOI:<https://doi.org/10.1145/1478559.1478646>
- [3] dwlin. What is the Speaker Recognition API? - Azure Cognitive Services. Retrieved April 18, 2019 from <https://docs.microsoft.com/en-us/azure/cognitive-services/speaker-recognition/home>
- [4] Christine Flounders. 2001. “Are You There Margaret? It’s Me, Margaret”: Speech Recognition As a Mirror. In *CHI '01 Extended Abstracts on Human Factors in Computing Systems (CHI EA '01)*, 459–460. DOI:<https://doi.org/10.1145/634067.634332>
- [5] Gerald Friedland and Oriol Vinyals. 2008. Live Speaker Identification in Conversations. In *Proceedings of the 16th ACM International Conference on Multimedia (MM '08)*, 1017–1018. DOI:<https://doi.org/10.1145/1459359.1459558>
- [6] S. Gamm, R. Haeb-Umbach, and D. Langmann. 1996. Findings with the design of a command-based speech interface for a voice mail system. In *Proceedings of IVTTA '96. Workshop on Interactive Voice Technology for Telecommunications Applications*, 93–96. DOI:<https://doi.org/10.1109/IVTTA.1996.552769>
- [7] M. S. Kandhari, F. Zulkemine, and H. Isah. 2018. A Voice Controlled E-Commerce Web Application. In *2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, 118–124. DOI:<https://doi.org/10.1109/IEMCON.2018.8614771>
- [8] S. S. Ketkar and M. Mukherjee. 2011. Speech Recognition System. In *Proceedings of the International Conference & Workshop on Emerging Trends in Technology (ICWET '11)*, 1234–1237. DOI:<https://doi.org/10.1145/1980022.1980292>
- [9] R. King. 1997. New challenges in automatic speech recognition and speech understanding. In *TENCON '97 Brisbane - Australia. Proceedings of IEEE TENCON '97. IEEE Region 10 Annual Conference. Speech and Image Technologies for*

Computing and Telecommunications (Cat. No.97CH36162), 287 vols.1-.

DOI:<https://doi.org/10.1109/TENCON.1997.647313>

- [10] Geoffrey Mainland, Greg Morrisett, Matt Welsh, and Ryan Newton. 2007. Sensor Network Programming with Flask. In *Proceedings of the 5th International Conference on Embedded Networked Sensor Systems (SenSys '07)*, 385–386. DOI:<https://doi.org/10.1145/1322263.1322307>
- [11] B. A. Malloy and J. F. Power. 2017. Quantifying the Transition from Python 2 to 3: An Empirical Study of Python Applications. In *2017 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, 314–323. DOI:<https://doi.org/10.1109/ESEM.2017.45>
- [12] Hy Murveit and Mitch Weintraub. 1990. Real-time Speech Recognition Systems. In *Proceedings of the Workshop on Speech and Natural Language (HLT '90)*, 425–425. DOI:<https://doi.org/10.3115/116580.1138609>
- [13] S. Ozaydin. 2017. Design of a text independent speaker recognition system. In *2017 International Conference on Electrical and Computing Technologies and Applications (ICECTA)*, 1–5. DOI:<https://doi.org/10.1109/ICECTA.2017.8251942>
- [14] I. Shahin. 2008. Speaker Recognition Systems in the Emotional Environment. In *2008 3rd International Conference on Information and Communication Technologies: From Theory to Applications*, 1–5. DOI:<https://doi.org/10.1109/ICTTA.2008.4530022>
- [15] R. Weychan, T. Marciniak, A. Stankiewicz, and A. Dabrowski. 2014. Real time speaker recognition from Internet radio. In *2014 Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA)*, 128–132.
- [16] Xiaojia Zhao, Yuxuan Wang, and DeLiang Wang. 2014. Robust Speaker Identification in Noisy and Reverberant Conditions. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.* 22, 4 (April 2014), 836–845. DOI:<https://doi.org/10.1109/TASLP.2014.2308398>
- [17] Cloud Speech-to-Text basics | Cloud Speech API Documentation. *Google Cloud*. Retrieved April 18, 2019 from <https://cloud.google.com/speech-to-text/docs/basics>
- [18] PyAudio Documentation — PyAudio 0.2.11 documentation. Retrieved April 18, 2019 from <https://people.csail.mit.edu/hubert/pyaudio/docs/>
- [19] Quick Facts - Ashesi University. Retrieved April 23, 2019 from <https://www.ashesi.edu.gh/about/at-a-glance/quick-facts>

Appendices

Appendix A – Requirements Gathering

Name of interviewer:

Place of interview:

Date of interview:

Duration Interview:

Questions:

1. Do you normally purchase food or items from the cafeterias on campus?
2. If yes, do you use cash or card?
3. If no, where do you normally purchase food or items, and do you use card or cash?
4. If you use cash, how much do you normally spend a day on food or items?
5. Have you ever been a situation where the seller did not have change for your purchase of items with cash?
6. Describe the situation?
7. Did you get your change back on the same day?
8. If yes, how long did it take?
9. Was it frustrating?
10. If no, did you eventually get your change back?
11. If you got your change back eventually, how long did it take you to finally retrieve your change?
12. Can you describe the processes involved before retrieving the change?
13. Was it frustrating?
14. How often do you experience this situation of not getting your change back?
15. Would you describe this as a problem?

16. What exactly would be the problem?

Appendix B – Error Handling

This section presents screenshots of the ways in which errors were handled in code.

A. Error with Identification

```
if(speaker == "error"):
    print("System was not able to Recognize who the Speaker was, Please Try Again")
else:
    print("Speaker: " + speaker)
    print(" ")

    print("Please say the Phrase below as a Verification Step: ")
    print("My Voice Is My Passport Verify Me ")
    print(" ")

    # Record Again
    getAudio("test.wav",FORMAT = pyaudio.paInt16, CHANNELS = 1, RATE = 16000,
            CHUNK = 1024, RECORD_SECONDS=8)

    # Use Recorded file to Run command Line Command for Speaker Verification
    profileVer = verifyProfile(speaker,profileVerify)
    verifySpeaker('test.wav',str(profileVer),speaker)
    # Do Comparision if command is accept
```

B. Error with Command Extraction

```
response = verifyResponse('verify.txt')
if(response == "Accept"):
    print(" ")
    print("System has Verified the Speaker: " + speaker)

    print("===== Extracting Commands from Speech =====")

    print("Speech: " + speech)
    print(" ")

    command = extractCommand(speech)
    print(command)
    print(" ")

    if not command:
        print("Commands Were not Clearly Extracted, Please Try Again")
    elif(len(command) < 2):
        print("Commands Were not Clearly Extracted, Please Try Again")
    else:
        print("===== Database Account Updates =====")
        databaseHandler(command,speaker)
        print("Successfully Updated Database")
else:
    print("Sorry, The System Could Not Verify " + speaker)
    print("Please Try Again")
```

C. Error with Enrollment

```
ion/VerificationServiceHttpClientHelper.py", line 199, in enroll_profile
    raise Exception('Error enrolling profile: ' + reason)
Exception: Error enrolling profile: {
  "error": {
    "code": "BadRequest",
    "message": "TooNoisy"
  }
}
```