



ASHESI UNIVERSITY COLLEGE

**GA DICTIONARY APPLICATION: AN ANDROID AND WEB PLATFORM TO
FACILITATE THE EASY LEARNING OF THE GA TRADITIONAL LANGUAGE**

APPLIED PROJECT

B.Sc. MANAGEMENT INFORMATION SYSTEMS

Emmanuel Nii Tackie

2016

ASHESI UNIVERSITY COLLEGE

**Ga Dictionary Application: An Android and Web Platform to Facilitate the Easy
Learning of the Ga Traditional Language**

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 Management Information Systems

Emmanuel Nii Tackie

April 2016

Declaration

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

Supervisor's Signature:

Supervisor's Name:

Date:/...../.....

Acknowledgement

I would like to express my sincerest gratitude to my Faculty Advisor, Supervisor and mentor all in the Person of Mr. Kwadwo Gyamfi-Osafo Marfo for his tremendous support and contribution. I would also like to thank my family and friends for their encouragement and support throughout this project.

Abstract

Communication by a common language is an effective tool to foster oneness and efficient productivity in a diverse community such as Ghana which has over 50 languages. Matched with this is the ever growing rate of the cell phone patronage in Ghana, with over 65% of the populace being owners of a mobile device, as well as the gradual increment in smart phones ownership, allowing Ghanaians to tap into various resources provided by the internet. Resources may include but are not limited to mobile applications, cloud services and pools of information from various databases. A clear observation made was the absence of efficient online translation tools to aid the learning process of Ghanaian local languages. The project identifies the absence of the online presence of vital translation tools with respect to the ever growing improvements in mobile technologies. It focuses on the Ga Traditional language and attempts to provide translation services to tourists, Ga students (mostly in junior high schools), the Ga diaspora and any and every interested party. It is motivated by the Afrocentric need to bridge the technological gap as well as promote Ghana online.

Contents

Declaration	i
Acknowledgement	ii
Abstract.....	iii
Chapter 1: Introduction	1
1.1 Introduction and Problem.....	1
1.2 Motivation.....	3
1.3 Related Work	5
1.3.1 The Wulomie.bb Website	5
1.3.2 The Kasahorow.org Website.....	6
1.3.4 Google Translate.....	6
1.3.5 Duolingo	6
1.4 Existing Technologies.....	7
1.4.1 Rule-Based Machine Translation.....	7
1.4.2 Statistical Machine Translation.....	8
1.4.3 Hybrid Machine Translation	8
1.5 Goal.....	11
Chapter 2: Requirement Analysis	12
2.1 Requirement Acquisition Procedure	12
2.2 Requirement Analysis.....	14
2.2.1 Audience	14
2.2.2 User Description	14
2.2.3 Scope.....	14
2.2.4 Objectives	15
2.2.5 Current Systems	15
2.2.6 Proposed System.....	16
2.2.7 Functional Requirements	16
2.2.7.8 Output Functionality	19
2.2.8 Non-functional Requirements	19
2.3 System Models.....	20
Use Case Diagram.....	21
Basic Flow of Activities.....	22
Chapter 3: System Architecture	23
3.1 Platform.....	23
Context Diagram.....	24
3.2 Architectural Pattern	24

Model View Controller	25
.....	25
3.3 Software Modules	26
3.4 User Interface	26
4.1 Data Model.....	27
Entity Diagram.....	29
Chapter 4: Implementation	30
4.1 Tools	30
4.1.1 Hardware.....	30
4.1.2 Software	30
4.2 Research and Learning.....	31
4.3 Methodology	31
4.3.1 Database Creation	31
4.3.2 Importing SQLite Database into Android Studios	33
4.3.3 Creating the User Interface	36
4.3.4 Creating the Proverbs Page	38
4.3.5 Creating the Feedback Activity.....	40
4.3.6 Creating the Guessing game activity.....	41
4.3.7 Creating the main search functionality	43
4.3.8 Creating the Suggest a Word Activity	44
Chapter 5: Tests	45
5.1 Hardware tests.....	48
5.2 Screen size and form factor test	48
5.3 Speed Tests	49
5.4 Time Tests.....	49
Chapter 6: Conclusion	51
6.1 Limitations	51
Future Development.....	51
References	52

Chapter 1: Introduction

This introductory chapter highlights the issues associated with language translation, the motivation, related works and technologies, all in association to this dissertation.

1.1 Introduction and Problem

Language is the principal means by which we conduct our social life, and it is bound by the culture of a people (Kramsch, 2003). The absence of online translators and the scarcity of hard copy dictionaries for Ghanaian local languages do not only retard the process of studying Ghanaian languages locally, but inhibits the acquisition of knowledge for foreigners as well. Though it is naturally easier to study a language which is spoken at home, or taught in schools via jaw-jaw utterances of a teacher, there is a need for efficient documentation and accessibility to resources. It is however unfortunate that these scarce hardcopy resources may not withstand the test of time, and may not be portable enough to be carried everywhere. Also, Ghanaian traditions over the years may have experienced a bit of deterioration, as ancestors barely documented customs, but solely relied on propagation through word of mouth, emphasizing the need for efficient documentation.

The Ga language is spoken by the Ga people of Ghana who form about 3.4% of the Ghanaian populace (Wulomei.bb, 2016). Though the population ratio is relatively smaller than other ethnic groups such as the Akan, or Ewe ethnic group of Ghana, it is very important to indicate two reasons which form the bases for the selection of the Ga language, thus making it a good starting point.

The first reason is that the Ga language is one of the main languages spoken in Accra, the capital city of Ghana. The Ga people are the true inhabitants of the city of Accra which has an ethnically mixed population of about 2.573 million people according to the Ghana

Demographic Profile 2014, by Index Mundi (Index Mundi, 2015). This makes Ga, among languages such as Twi, Fante and Ewe one of the most popular languages spoken in Ghana (Ghana Embassy, 2016).

The second reason, however is that Accra is usually the first point of entry for many tourists, as the nation has only one international airport; Kotoka International Airport (World Guides, 2016). This makes the Ga language one of the first local languages one must learn in order to get by in the capital city.

Regardless of the somewhat bias selection of the Ga language as the language of choice for this project, it is very important to note that, given the scope of the project, it will be relatively easy to create similar applications for other languages since the primary differentiating factor is the database of words and their translations, as well as other minor changes. Thus, this will form a template of a sort in the creation of other language applications.

The project considers the Ghanaian Education system and attempts to provide a service that would not only serve as a quick and portable source of translation for students, but as a reference for teachers as well. It will facilitate the acquisition of knowledge by foreigners as well as the Ga Diaspora who want to study the Ga language and provide quick solutions to tourists who may want to explore the hearts of Accra without a tour guide.

The project is also motivated by the quest to create an online presence for the Ga Traditional community and Ghana as well, owing to the fact that there are only about two major online resources that attempt to provide such services. The first is the www.wulomei.bb website and the other, kasahoro.org website.

The project will run on the Android Operating System platform as it is the dominating platform in Ghana according to International Business Times (Mahapatra, 2013). This translation tool

would not only provide an English to Ga translation, but provide phonetic features, sound pronunciation for each word and other functionalities to ensure that the content and context of the meanings of words are not entirely lost during direct translation.

In effect, this dissertation highlights and attempts to tackle 5 significant problems. These are;

1. The scarce nature of physical copies of Ga dictionaries or references
2. The limited presence and popularity of online translators for the Ga language
3. The limited interactivity and support for the existing online translators
4. The absence of tourist support features
5. The absence of a standard online reference point

1.2 Motivation

The primary motivation for this project is the low popularity of such translation tools given the massive penetration of about 69% by the mobile phone market in Ghana as of 2014, as well as the ever growing rate of smartphone patronage and internet usage (Pew Research Center: Global Attitudes & Trends, 2015). This thus highlights the availability of technology for a successful implementation.

Other motivations could be identified within the direct and indirect benefits of an efficient implementation of online translation tools. These include but are not limited to the following;

Portability and Availability

Contrary to the very scarce hardcopy dictionaries and reference material for the Ga language as well as their heavy and delicate nature, online and mobile applications prove to be a more portable and readily available option. This is because the Ghanaian smartphone market continues to grow at exponential rates. According to the National Communications Authority of Ghana, the total number of mobile data subscribers increased from 17,728,312 as of the end of November 2015 to 18,031,188 at the end of December, 2015, thus accounting for an overall of 65.74% of mobile data penetration in the country (Ghana National Communications Authority, 2016).

Promote the Ghanaian Culture Globally

An online presence breaks the geographical reach and extends the Ghanaian culture globally. This would make it easier for others—both the diaspora and foreigners, outside the country to experience a critical element of Ghanaian culture—its language, without ever visiting the country. It will expose a great wealth of Ghanaian thoughts and practices through word choices and word combinations as well as wise sayings.

Self-learning Capabilities

Another motivation is to create a platform that offers self-learning capabilities, at flexible rates, with the goal of creating a seamless learning process which does not significantly distort one's everyday activities.

1.3 Related Work

There exists two major projects worthy of discussion as of the time of the creation of this documentation who attempt to provide relevant resources for word translation and language learning online of the Ga language. These are the wulomei.bb website and the kasahoro.org website. However, other language translators and tools are observed in full detail, which serve as guides, and, or a source of ideas for the project as well as potential features references for future works

1.3.1 The Wulomie.bb Website

Wulomie.bb is a free translation and online dictionary for the Ga people of Ghana, West Africa. It is a well-structured forum endowed with functionalities such as wide array of words (most of which are crowdsourced), phrases and sentences reflecting content from the Ga Bible, a brief history of the Ga people, and a somewhat active forum. Other useful features are links to the African Ancestry website, which through DNA analysis may give a fare idea on one's African origin and links to media content on African events.

It however lacks support structures for pronunciation such as efficient phonetics and sound. It also lacks the needed interactivity which according to the research paper; Literacy through Play, by Owocki Grentchen, stimulates Literacy development in children and adults alike (Owocki, 1999).

1.3.2 The Kasahorow.org Website

The Kasahorow Foundation on the other hand provides a collection of services that aid in the assimilation of West African local languages. These service primarily range from reading material such as story books, to plugins or software such as local keyboards for the Android platform and Microsoft Word.

The books target a much younger audience and are broken down into age categories such as 0-7 years, 8- 12 years and 13+ years, while giving guidance to teachers with respect to its usage.

The books are sold as hardcopy material online, and are delivered to the recipient's address

The underlying shortfall of this website—one to which this project seeks to eliminate or minimize, is the usage of hardcopy material which are in most cases scarce, not easily portable and convenient, and are hard to reach by people living outside the country.

1.3.4 Google Translate

Google translate is a highly sophisticated multilingual statistical machine translation application which converts not only words, but sentences, speech, images or even real time video from one language to another. It is provided by Google and offers both a web interface and mobile interfaces for Android and IOS operating systems. It currently supports over 100 languages while offering features such as an offline mode to ensure that one is always equipped with translation services on the go. It employs Statistical Machine Translation which involves consistent studies on large amounts of data in order to identify patterns (Google, 2016)

Its shortfall however is not with respect to its functionality, but the absence of the Ga language.

1.3.5 Duolingo

Duolingo is a free language learning application which teaches one how to read, right, listen and speak new languages within an interactive gaming experience. The activities on this

platform are divided into bit sized skills to support the easy assimilation of the language of study. It provides a multiple choice gaming experience where points are gained upon successful completions and lives (game lives) lost upon failures while providing a statistical update on progress made. It currently has more than 50 million downloads from the google play store (Duolingo, 2016)

With respect to this project, Duolingo lacks the support for the Ga language.

1.3.6 Memrise

Memrise takes a different approach with respect to the applications mentioned above. It utilizes ‘mems’—mnemonic flashcards which help users to remember vocabulary based on pre-existing knowledge. It runs on both Android and IOS mobile operating systems (Merritt, 2013).

1.4 Existing Technologies

1.4.1 Machine Translation Technologies

It is of significant importance to address the impact of machine translation as well as the available technologies in existence. Basic Machine Translation could best be understood as a simple substitution of words in one language for words in another. This however on its own cannot solve the issues or direct-translation gaps or distinct word combinations which arise from different circumstances. The following are advancements made in the area of machine translation;

Rule-Based Machine Translation

This system involves a large amount of rules that are manually developed by human experts, which map structures from the source to targets. This involves a significant amount of manual

labour which can be quite costly and time consuming. With the addition and updating of rules increases the potential of generating ambiguity, thus degrading the translation output. Its pros however could be found within the human skills invested, thus creating a potential of delivering fairly good automated machines and predictable results (Safaba Translation Innovation, 2016).

Statistical Machine Translation

This system utilizes computer algorithms that explore millions of possible ways of putting piece of words together in attempt to produce the best translation. It utilizes a large scale database comprising of documents that have already been translated by professional human translators thus learning automatically from previous translated text. This is the same technology employed in the Google Translate application. It is dramatically faster, more cost efficient and easy to update. Its shortcomings however include challenges in controlling the learning process and identifying the source of potential errors (Safaba Translation Innovation, 2016).

Hybrid Machine Translation

The approach is a combination of both Rule-Based and Statistical machine translation. This harnesses the strengths of both systems, as well as a combination of their flaws as well. (Safaba Translation Innovation, 2016).

1.4.2 Hardware and Operating Systems

The scope of this dissertation limits the discussion on hardware to mobile phones and tablet computers, thus drawing two major categories. The first are voice or text based mobile phones, and the other, smart devices (smartphones and tablet computers).

With respect to voice or text based mobile phones (pre-smartphones), a solution such as the Chinese and Spanish translation service, Muuzii is worth mentioning. This allows users to send text messages to a specific short-codes (to a server) which in turn replies with the translation on the inputted text (Gigaom, 2014). This however cuts across all OEMs. It is important to note that very little attention is shown to such devices in this day and age as was before, due to their limitation in hardware capabilities and little or no developer support as a result of low economic benefits due to low number of users.

With smartphones and tablet computers, however comes the power for much more computation. These can best be grouped in to their various operating systems (platforms) which in some cases could be directly linked to their OEMs, as is the case of IOS for Apple and Blackberry OS for Blackberry devices, whereas the Android and Windows Phone operating Systems could be seen on different OEMs, indicating an indirect relationship. These different platforms require different programming languages and such there exists differences in expertise and implementation.

IOS for example require the use of the Swift programming language which is an attempt to slowly retire the use of Object-C and C programming languages (Developer.Apple, 2015).

Microsoft Windows Phone on the other hand utilizes a combination of XAML, C# or Visual Basic as programming languages (Windows Dev Center, 2016). Whereas Blackberry OS applications are built on C. These however form the native programming languages or

solutions to app development. Thus are advantageous in performance, speed, fluidity, and control.

Further advancement in app development have led to new and improved methods for programming mobile application which utilize—well in most cases, web-based programming languages such as PHP and Javascript or some combination of other languages other than the smartphone's operating system native language that best suit developer needs. The approach is colloquially termed as the hybrid approach, and this involves the use of popular platforms such as Phonegap and IBM Bluemix. Its pros include faster development as well the ease of deploying to various smartphone operating systems. It however compromises of speed, performance, and full control of hardware (Ziflaj, 2014).

Storage Technologies

The choice of storage options for a mobile application could mean life and death in terms of user preference. To put forth a simpler and more concise argument, the documentation generalizes storage options into two categories. The first is internal storage which covers storage on the application itself-- that is to say on the mobile device or on a memory card. This is however advantageous because it fast and does not require internet speeds as a factor of performance. Its disadvantage however may lie within the size of the database which may be too large and as such eat into the user's storage. External storage, within this context caters for cloud or server base storage. This is advantages to minimize the storage burden of users, but is burdened by the reliance of network or internet service provider connection speeds to ensure performance (InfoWorld, 2015).

1.5 Goal

This project will serve as a one stop powerhouse containing the necessary features and functionalities to smoothing the acquisition and depth of knowledge of the Ga language within a seamless mobile interaction. It will seek to totally eliminate the use of paper while immersing the user within a fun packed and interactive environment.

The product is packed with the necessary bells and whistles, including games to facilitate a smooth learning process.

Chapter 2: Requirement Analysis

This chapter provides in full depth, a requirement analysis for the implementation of the Ga Dictionary application.

2.1 Requirement Acquisition Procedure

This entails information pertaining to the stakeholders of the project, and the implementation procedure in the acquisition the requirement needs.

The primary stakeholders of the project are students, teachers and tourists. However the application is open to anyone who wants to study the Ga language.

Ga Students

Students play a vital role in the utilization of the application, and as such heavy priority was given to their input. The project targets Junior High School students as well as university students studying Ga and in rare cases any other local language. It focused on a sample size of about 5 university students who studied Ga or any other local language while in Junior High School as a source of requirement analysis.

The information gathering process involved an unstructured interview process and contain questions such as, but not be limited to the questions below.

- What are the resources available to make the study of local languages easier?
- Do you appreciate fully the current methods of teaching?
- Would you be happy if you could have a portable local language reference resource on your person?

A period of one week was employed to attain such information, which yielded sustainable insights to the project's requirement while reconciling the lapses of the initially assumed user requirements.

Existing Translation Tools and Dictionary Standards

An in debt analysis over a period of 2 weeks was conducted by identifying and analysing existing translation tools such as google translate and other dictionaries to identify the overall logic, methodology and avenues for improvement. This encompassed rigorous research into the social sciences such as language translation—its significance and barriers, as well as more scientific and technological environment to identify efficient platforms, algorithms and theories. Research was also carried out to identify standards such as dictionary standards and platform standards among others to ensure professionalism and operate within an ethical environment.

Individuals of the Ga Community

An informal interview session consisting of mostly family members was conducted. The goal in this context was to examine from a cultural stand point and obtain both direct and subliminal features that will enhance the performance of the project. This acknowledged cultural do's and don'ts and served as a complementing feature to the sub-goal of this project which is to promote Ghana.

The requirements obtained where however tested within user tests for their relevance as well as limitations. Both of which will be discussed further within subsequent chapters.

2.2 Requirement Analysis

The purpose of the requirement analysis is to document the requirements of the Ga Dictionary project. This documentation completely describes the system in terms of functional requirements and nonfunctional requirements. This targets students, teachers and tourist as primary stakeholders. The platform for the application development is the Android Operating System and it based on native android application development programming methods.

2.2.1 Audience

The audience of this documentation for the purpose of this project will be limited to the supervisors of this project as well as other stakeholders in the development process.

2.2.2 User Description

The application is designed for any and every interested party, but more specifically for;

- Ga Students who primarily need quick and easy access to word translations, and word pronunciations.
- Ga Teachers who need a standard online reference point with accurate information
- Tourists who need easier and more interactive ways of studying Ga.

2.2.3 Scope

The project is limited to one local language; the Ga language, though there are other languages without such resources. This will serve as a preliminary stage, and upon efficient results, the translation of other local languages.

It is also important to note that the project at this stage considers single word translations and as such aims to help in the efficient acquisition of the Ga language. Although it is packed with other secondary features that attempt to teach the Ga language, its main aim is not to do so. That is to say that relying solely on the application may not provide you with 100% efficiency in the language, just as the Oxford Dictionary does not guarantee a master in English unless efficiently practiced in conversation.

2.2.4 Objectives

1. To provide a relevant translation tool to meet user requirements
2. Solve the technological gap due to the limited presence of such a tool
3. Promote Ghana internationally
4. To provide tourists an easier way of self-navigation, exploration and a sense of psychological comfort
5. Provide the Ga diaspora with relevant tool
6. Provide relevant resources to students and teachers of the language
7. Ga keyboard #

2.2.5 Current Systems

- Wulomie.bb website
- Kasahorow.org website
- Glosbe.com

2.2.6 Proposed System

This system takes advantage of the limited features of the current systems, and as such is packed with the adequate bells and whistles to make the project a success.

2.2.7 Functional Requirements

Below is a list of the project features broken down into 3 major categories to give a general summary of the functional requirements. Each category contains sub features which either stand alone or compliment others.

1. Input Functionality

- Search via text
- Search via voice
- # Ga keyboard

2. Activity Function

- Tourist mode
- Proverbs
- Word of the day
- Guessing game
- Suggest a word
- Feedback

3. Output functions

- Phonetics
- Word pronunciation
- Used in a sentence

2.2.7.1 Input Functionality

Search Feature

The purpose of this feature is to enable the user search for the word of choice in English to obtain a Ga translation.

Specific Details

This will involve two primary methods. The first is an input via **text**, whereas the other is an input via **voice**.

Input via Text

This feature enables the user to type in desired words of search in English.

It is important to note here that considering the scope of the project, searching will be limited to single words and some selected phrases.

Input via Voice

This on the other hand enables the user to search for words by speech.

The Output

- Ga word
- **Phonetics**
- **Used in a sentence**
- **Sound pronunciation of the Ga word**

2.2.7.2 Activity Function

In attempt to smoothen the flow of the learning process, this category provides functions such as a tourist mode, proverbs, word of the day, suggest a word and the ability to give feedback.

Specific Details

Word of the day

The feature will randomly select a word worthy of study each day for the user. On special occasions such as seasons, festivals or other time periods, words will be tailored to suit the occasion.

Tourist Mode

The tourist mode contains short lessons that teach tourist how to go about daily life in Accra.

Proverbs

This feature takes a tour into the culture of the Ga people and provides the user with wise sayings.

Guessing Game

This feature allows the user to play a game by choosing from a set of images or buttons, the one that best matches the word below.

Suggest a Word

This feature allows the user to suggest a word to be added to the dictionary. This feature extends the dictionaries input creates room for crowd sourcing.

Feedback Activity

The feedback feature allows users to give feedback on any feature as well as make suggestions towards new functional requirements,

2.2.7.8 Output Functionality

This functionality is characterized by the output or results

Phonetics

This breaks down each Ga word into phonetic syllables making it easy to pronounce.

Used in a Sentence

This output feature displays the Ga word searched for in a sentence to enable users to know and understand how it could effectively be used.

2.2.8 Non-functional Requirements

The project will be keen to execute the following non-functional requirements

2.2.8.1 Efficient Space Utilization

A minimal space requirement will be adhered to so as not to overburden the user with large files that may cause storage problems. This will also aid in faster downloads. A maximum size of 70 megabytes will be considered.

2.2.8.2 Usability

The project will be easy to use. It will support a simply user interface with clear typography and buttons to ensure user friendliness. It will adhere to affordance as well as a series of test to ensure that it could be easily manipulated by the average user.

2.2.8.4 Dependability

This project will undergo tests to ensure that it is survivable, reliable, maintainable, and all other relevant dependability checks to ensure its efficiency.

2.2.8.5 Speed

Search results as well as transitioning from activities will be displayed in less than a second. It is worth mentioning that the hardware capabilities of the mobile devices as well as internet speed will play a significant role in attaining such optimum performance.

2.3 System Models

The diagrams below represent a basic illustration of the requirement analysis.

Use Case Diagram

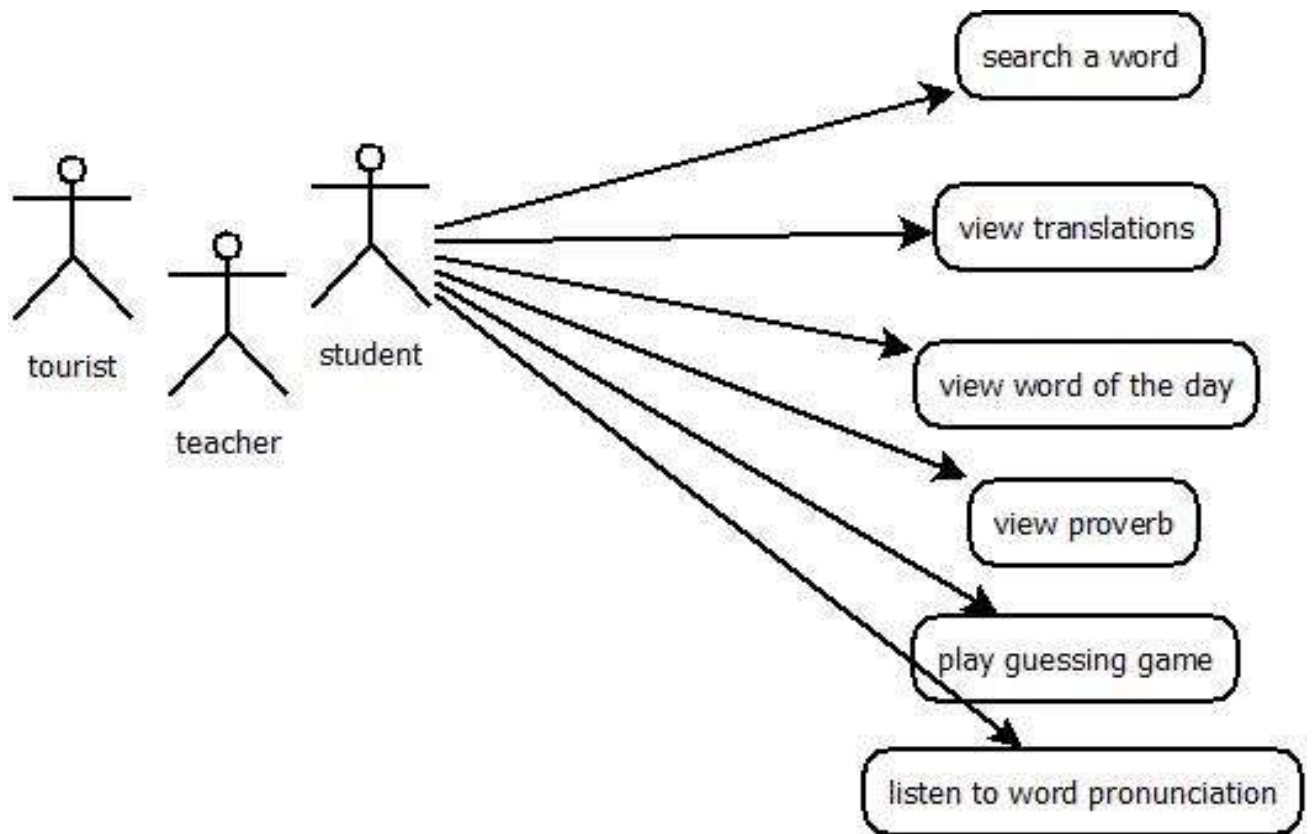


Figure 2.1 Use case diagram

Figure 2.1 above, is a simple illustration of the functions a user can perform while using the application.

Basic Flow of Activities

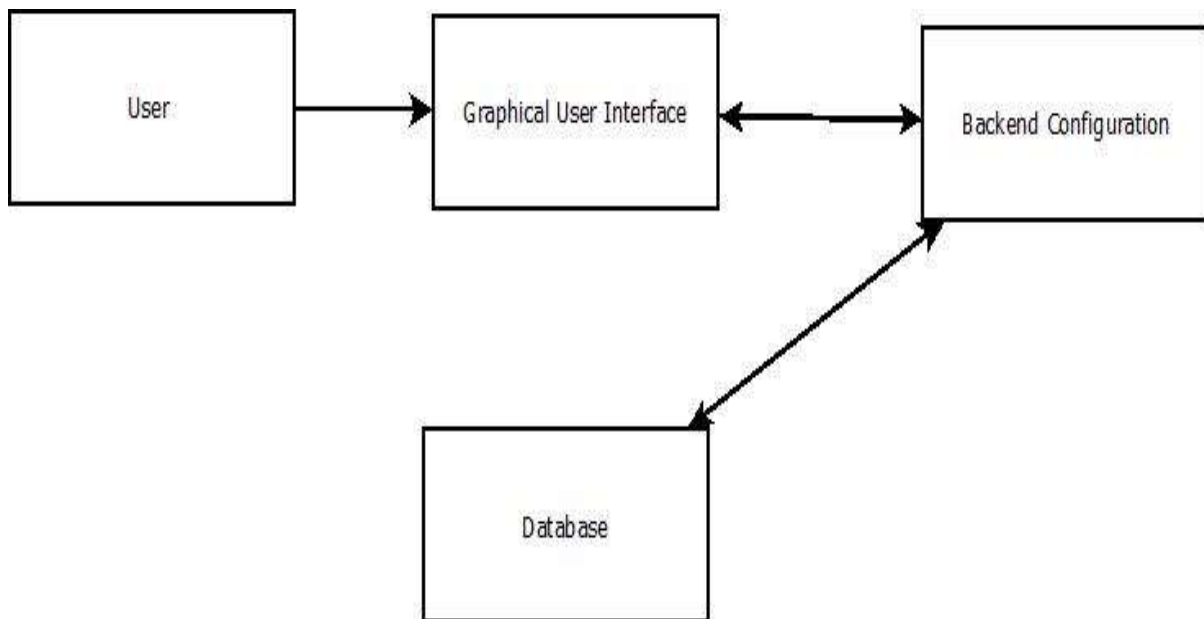


Figure 2. 1 Basic Flow Diagram

Figure 2.1 above illustrates the basic flow of activities when the user interacts with the application, whereas more details will be given within the next chapter. The user interacts with the graphical user interface and based on inputs (for example, a search via text), information is then sent to the back end configuration which in turn communicates with the database. Upon success, data acquired from the database is further processed within the backend configuration and then displayed for the user's consumption.

Chapter 3: System Architecture

This documentation contains relevant architectural information which served as a guide in the overall implementation process. It provides a detailed breakdown of how the application works to both the developer and third parties interested in understanding or building upon the application. This documentation segments the projects architecture in to 6 broad categories which encompasses the architectural need of the project. These are the **platform, architecture pattern, modules, interface, data models and context models**.

3.1 Platform

The project will run on the Android platform, and as such will be implemented on Android smartphones and tablets. It will be based on the **XML, Java and SQLite** frameworks and will deploy the necessary framework libraries (Java and XML libraries) to see the project's success. Other tools imbibed in Android Studios (the IDE of choice) such as Grandle will also be utilized.

Though the application will be stored primarily on the smartphone, it will interface with the Google Play Services (Store) to provide updates to the application as well as database server that will serve as a reception point for suggested words (functional requirement) for the application

Below is a context diagram which illustrates the application as one component, and its interaction with other external components

Context Diagram



Figure 3.1 Context Diagram

Figure 3.1 above illustrates the interaction of the application with a database server and the google play store.

3.2 Architectural Pattern

The primary architectural pattern for the project is the Model View Controller model. Though the Android operating system or java for that matter is built on a layered architecture, emphasis here is laid on the coherent view where the project components are broken into three core interconnected components. This thus separates the internal representation of the information from the view the end user attains. The Ga translation tool will consist of a user interface (View), a logic or back-end component (Controller) and a database component (Model). This will facilitate easy update of each component, easy debugging and faster development.

Given the Android platform as a means of executing the project, the user interface (view) will consist of XML implementations, the logic or back-end (controller); Java implementations, and the database (model); SQLite. In effect the java bit of the source code will act as the controller between the user interface and the database. It is important here to note the functionality of components such as the Gradle in android application development which coordinates all other activities ranging from file structures right down to ensuring connectivity of the view, controller and model of the application

Below is a diagram illustrating the Model View Controller Architecture for an Android Application;

Model View Controller

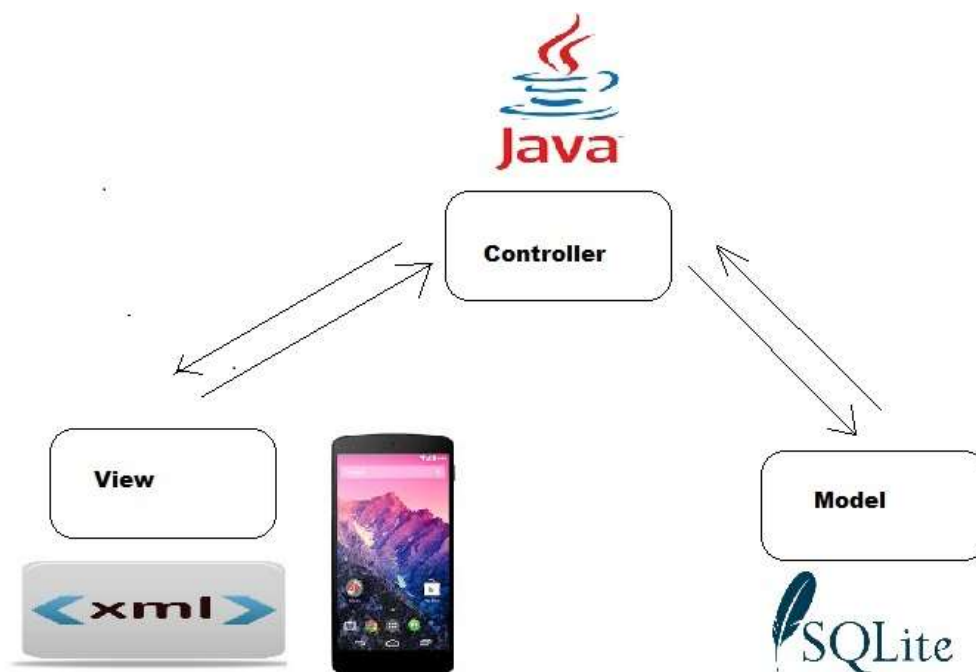


Figure 3.2 Model View Controller

3.3 Software Modules

The functional requirement will be broken down into three main modules based on their similarities. These models are;

- Output module: This highlights all output activities relevant to the main dictionary output. This includes the output structure and layout and the sound functionalities
- Input module: This module tackles all inputs into the application. This will range from search via text or speech to user suggestion where the user gets to suggest words which may have been omitted.
- Activity module. The module encompasses activities such as word suggestions, word of the day, proverbs and any other activity to facilitate the learning process.

3.4 User Interface

The diagrams below illustrate the initial interface designs of the application. It highlights the relevant views and served as an instrumental component of the actual design implementation.

Welcome screen



Figure 3.3 Welcome screen

Figure 3.3 above illustrates the welcome screen, which allows and teaches the user how to download and install a Ga Keyboard.

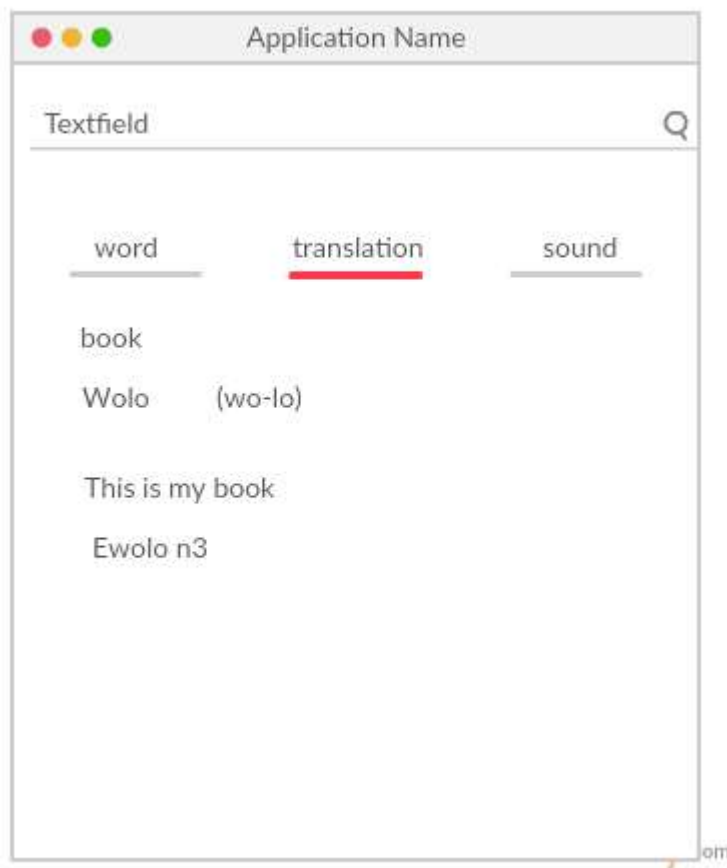


Figure 3.4 Translation view

Figure 3.4 above illustrates the primary view of the application.

4.1 Data Model

The database will contain 3 primary tables. These tables are listed below.

1. Dictionary Table

This contains attributes; **_id** which is the dictionary id for the English word, as its primary key, **English**; the word in English, **Ga**; the Ga translation, **EnglishSentence**;

the used in an English sentence, **GaSentence**; the Ga word used in a sentence, **Sound**; the sound pronunciation of the Ga word, **Phonetics**—the phonetic structure of the word and **Certified**, a yes or no to indicate whether a user suggested word has been approved.

2. Proverb Table

This table contains **_id**, which its primary key and a **Proverb** attribute, which contains the proverb

3. Phonetics Table

The table contains **Ga**; an alphabet in Ga, **Phonetics**; the alphabet's pronunciation, **English**; the utilization Ga alphabet within an English word and **_id**; its primary key

It is important to note that '**_id**' is a standard syntax recognized by android studios as the id and primary key of the table.

Below is an Entity Diagram illustrating the database entities

Entity Diagram

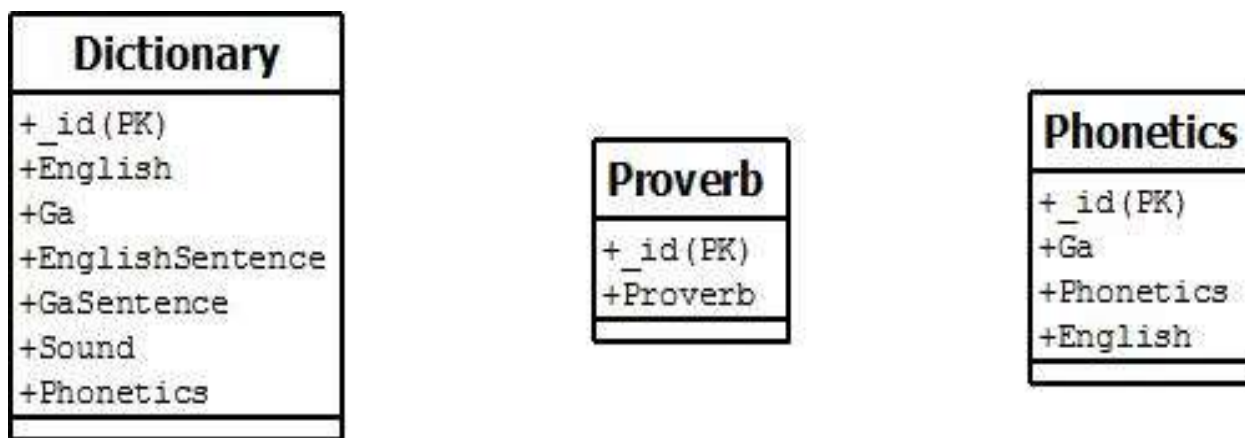


Figure 4. 5 Entity Diagram

Chapter 4: Implementation

This chapter provides a detailed account of the implementation process. It highlights concepts such as the necessary tools, APIs, frameworks and libraries utilized, the methodology as well as all other necessary development choices that led to a successful implementation of the Dictionary. In order to put forth a coherent and well-structured implementation documentation, the following subheadings are deployed;

4.1 Tools

This can be broken down into the relevant hardware and software used towards development of the application.

4.1.1 Hardware

- Samsung NP-QX411 laptop (Primary hardware device used in programming)
- Dell venue 8 android tablet (To test application on an 8 inch screen)
- Samsung galaxy s6 edge smartphone (To test the application on an Android Lollipop device)
- Samsung galaxy pocket smartphone (To test application on a gingerbread device)

4.1.2 Software

- SQLite Manager (This was used for the creation of the database)
- Microsoft Office (This was used for the creation of the project documentation)
- Android Studios (This was used as the primary IDE)
- Adobe Illustrator (For graphic designs)
- Canva.com (for graphic designs)
- Creately.com (for graphic designs)

4.2 Research and Learning

Research and learning was a continual process throughout the development of the application. It begun with initially trying to fully understand android application development as a whole. This stretched from as far as the history of its development through its architecture, components, the activity life cycle, right down to the first dummy application. The preliminary research however was not enough (based on the assumption of an already existing knowledge of java) given that each functional requirement exposed a new learning experiences.

Another avenue for research was with respect to design. This involved observing other standard dictionaries such as the Merriam Webster Dictionary android application and the French-English Dictionary android application for design cues. However, majority of the design decisions implemented were in strict accordance with the material design principles.

It is thus very important that the project acknowledges and gives credit to the Android Developer website, Slidenerd Youtube channel and the Stackoverflow website as tremendous sources of knowledge in the learning and research process.

4.3 Methodology

The steps below indicate the implementation process carried out. Though they are well explained and detailed, they only account for core functionalities of the application.

4.3.1 Database Creation

A fundamental component of the dictionary is its database. However a critical problem of manually making inputs was a serious challenge. Another challenge, however, was that

Android Studios does not provide a graphical user interface for the easy creation and input into a database. This then led to the installation and utilization of the SQLite manager—an open source SQL database management application that support both SQLite syntax as well as a graphical user interface.

Below are snapshots of the database content created within the SQLite manager.

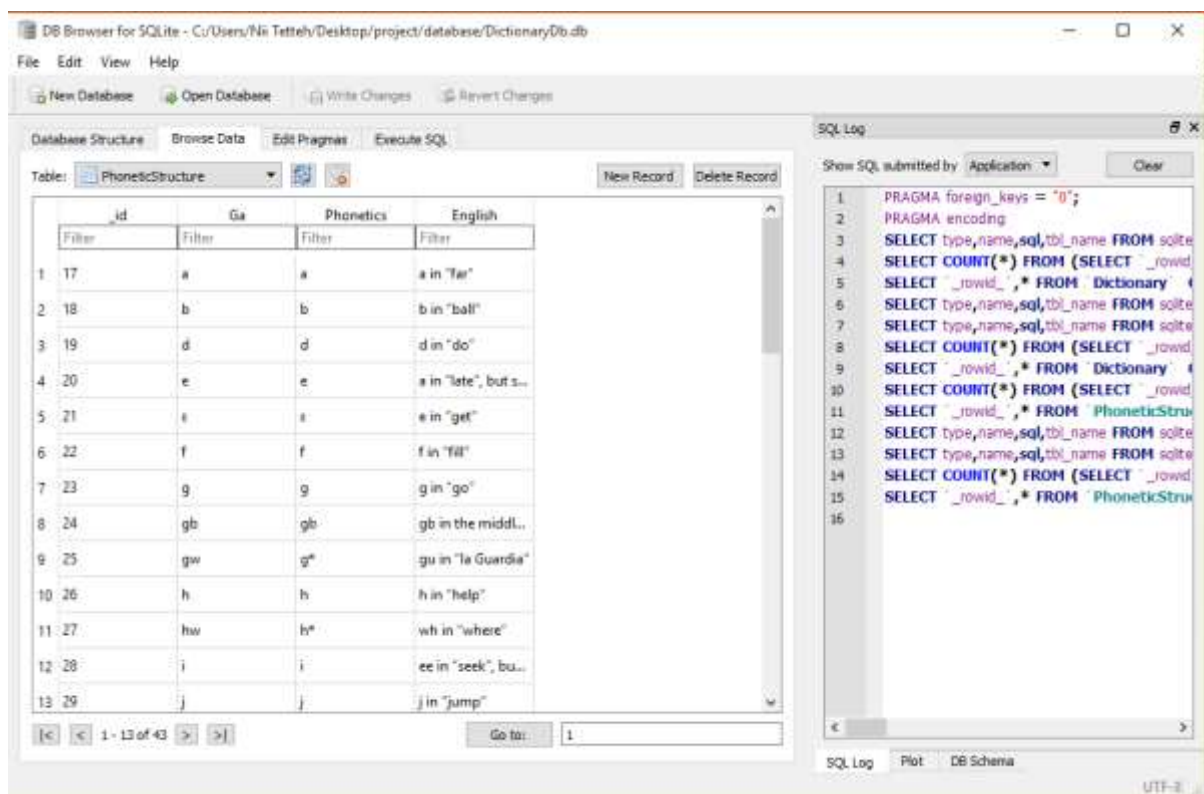


Figure 4.1 Phonetic Structure

Figure 4.1 shows the **PhoneticStructure** table designed to teach the user how to pronounce Ga alphabets.

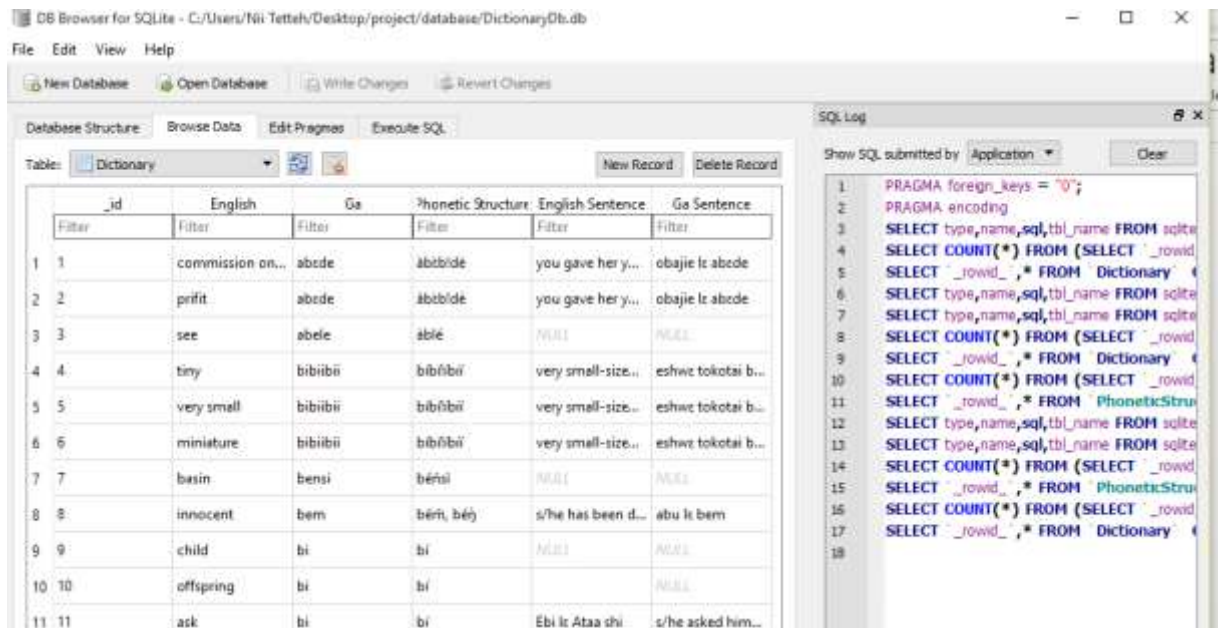


Figure 4.2 Dictionary content

Figure 4.2 above shows the words within the dictionary.

4.3.2 Importing SQLite Database into Android Studios

This involves the use of the `android.database.sqlite` API but more specifically, extending `SQLiteAssetHelper` class. This allows one to use an external SQLite database stored within an assets resource file within the android project. A quintessential aspect of this process is to include the compile **'com.readystatesoftware.sqliteasset:sqliteassethelper:2.0.1'** dependency into the `build.gradle` block of code. Upon completion, a class is created that extends the `SQLiteAssetHelper`, and within this class a connection is made to the database via the `getWritableDatabase` method.

Below are snapshots for importing database into android studios

```

package com.example.niitetteh.gadictionaryapp.Database;

/**
 * Created by Nii Tetteh on 2/17/2016.
 */

import android.content.Context;

import com.readystatesoftware.sqliteasset.SQLiteAssetHelper;

public class DatabaseOpenHelper extends SQLiteAssetHelper {
    private static final String DATABASE_NAME = "DictionaryDb.db";
    private static final int DATABASE_VERSION = 1 ;

    public DatabaseOpenHelper(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }
}

```

Figure 4.3 Database connection

Figure 4.3 above indicates the SQLiteAssetHelper extension which facilitates the connection to the DictionaryDb.db.zip database stored within the databases file, within assets.

```

    * @return the instance of DatabaseAccess
    */
    public static DatabaseAccess getInstance(Context context) {
        if (instance == null) {
            instance = new DatabaseAccess(context);
        }
        return instance;
    }

    /**
     * Open the database connection.
     */
    public void open() {
        this.database = openHelper.getWritableDatabase();
    }

    /**
     * Close the database connection.
     */
    public void close() {
        if (database != null) {
            this.database.close();
        }
    }

    /**
     * Read all quotes from the database(testing the databases success)
     *
     * @return a list of quotes
     */

    public List getEnglishWords() {
        List<String> list = new ArrayList<>();
    }
}

```

Figure 4.4 Database connection

Figure 4.4 above shows the connection to the database through the instance of the DatabaseAccess class created previously which extends the SQLiteAssetHelper class.

```

dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    testCompile 'junit:junit:4.12'

    compile 'com.android.support:appcompat-v7:23.1.1'
    compile 'com.android.support:design:23.1.1'
    compile 'com.readystatesoftware.sqliteasset:sqliteassethelper:2.0.1'
    compile 'com.android.support:recyclerview-v7:23.1.1'
    compile 'com.android.support:support-v4:23.1.1'
}

```

Figure 4.5 Graddle App

Figure 4.5 above indicated the addition of the **com.readystatesoftware.sqliteasset:sqliteassethelper:2.0.1** dependency into the gradle file. It also shows additional dependencies such as '**com.android.support:recyclerview-v7:23.1.1**' for the RecyclerView and the '**com.android.support:design:23.1.1**' for the implementation of material design.

4.3.3 Creating the User Interface

The user interface was created in full accordance with material design principles. It involves a primary colour; #F44336, secondary colour; #E64A19 and a colour accent #03A9F4. The home screen (activity_main.xml) implements a Search View for word searching, a Toolbar to house other activity icons or menu options, an Image View to display the background image and a Floating Action Button. The menu icons are flat in nature. The application makes room for affordance with 'back-arrow' icons, which helps the user return to the previous activity.

The diagrams below illustrate implementation of the user interface;

```

<Button
    android:layout_width="300dp"
    android:layout_height="85dp"
    android:text="Dictionary"
    android:id="@+id/button"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true"
    android:onClick="dictionary"
    android:textColor="#fff"
    android:background="#E64A19"
    style="?metaButtonBarButtonStyle"
    android:textSize="50sp"/>/>

<Button
    android:layout_width="300dp"
    android:layout_height="85dp"
    android:text="Proverbs"
    android:id="@+id/button2"
    android:layout_marginTop="96dp"
    android:textSize="50sp"
    android:onClick="proverbs"
    android:textColor="#fff"
    android:background="#E64A19"
    style="?metaButtonBarButtonStyle"
    android:layout_below="@+id/button"
    android:layout_centerHorizontal="true" />/>

<Button
    android:layout_width="300dp"
    android:layout_height="85dp"
    android:text="Games"
    android:id="@+id/button3"
    android:layout_marginTop="104dp"
    android:onClick="games"
    android:textColor="#fff"

```

Figure 4.7 Main Activity interface

Figure 4.7 above illustrates the design for the home screen.

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.niitetteh.gadictionaryapp">

    <uses-permission android:name="android.permission.INTERNET" />
    "
    <application
        android:allowBackup="true"
        android:icon="@drawable/app2"
        android:label="@string/app_name"
        android:largeHeap="true"
        android:supportRtl="true"
        android:theme="@style/AppTheme">
        <activity
            android:name=".MainActivity"
            android:label="@string/app_name"
            android:theme="@style/AppTheme.NoActionBar">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name=".Subfunctions.AboutUs"
            android:label="@string/title_activity_about_us"
            android:parentActivityName=".MainActivity"
            android:theme="@style/AppTheme.NoActionBar">
            <meta-data
                android:name="android.support.PARENT_ACTIVITY"
                android:value="com.example.niitetteh.gadictionaryapp.MainActivity" />
        </activity>
        <activity android:name=".Dictionary.DictionaryList">
            <meta-data
                android:name="android.support.PARENT_ACTIVITY"
                android:value="com.example.niitetteh.gadictionaryapp.MainActivity" />

```

Figure 4.8 Android Manifest screenshot

Figure 4.8 above indicates the flow of activities back to the home page. The content circled with the red marker indicate an activity or page. The parts highlighted in green however represent the parent activity and as such the destination of the ‘back-arrow’ button on the activity.

4.3.4 Creating the Proverbs Page

The proverbs activity implements a sliding tab and as such takes full advantage of fragments. Each fragment represents an image which contains a proverb. ProverbTab.java, which is the name of the java class extends the `FragmentManager`. This is because the `FragmentManager` is used for instances with a considerably larger amount of fragments as

compared to extending the Fragment class. The getItem() function based on a switch state returns an instance of each fragment class and in effect, as is the case with this application, displays from the corresponding fragment's XML file an image.

The code snippets below represent the creation of the proverbs activity.

```

* A {@link FragmentPagerAdapter} that returns a fragment corresponding to
* one of the sections/tabs/pages.
*/
public class SectionsPagerAdapter extends FragmentStatePagerAdapter {

    public SectionsPagerAdapter(FragmentManager fm) {
        super(fm);
    }

    @Override
    public Fragment getItem(int position) {
        // getItem is called to instantiate the fragment for the given page.
        // Return a PlaceholderFragment (defined as a static inner class below).
        switch (position) {
            case 0:
                return Proverb10.newInstance();
            case 1:
                return Proverb1.newInstance();
            case 2:
                return Proverb2.newInstance();
            case 3:
                return Proverb3.newInstance();
            case 4:
                return Proverb4.newInstance();
            case 5:
                return Proverb5.newInstance();
            case 6:
                return Proverb6.newInstance();
            case 7:
                return Proverb7.newInstance();
            case 8:
                return Proverb8.newInstance();
            case 9:
                return Proverb9.newInstance();
        }
    }
}

```

Figure 4.9: Proverbs activity code snippets

Figure 4.9 above illustrates the use of the FragmentStateAdopter as well as illustrates the switch statement which instantiates the necessary fragments.



Figure 4.10 Proverbs UI

4.3.5 Creating the Feedback Activity.

The Feedback activity utilize an email intent. It requires the selection of an email client. In the absence of an email client the function notifies the user.

The image below represents the creation of the Feedback and Word Suggestion functions.


```

import android.widget.Toast;

import com.example.niitetteh.gadictionaryapp.MainActivity;
import com.example.niitetteh.gadictionaryapp.R;

public class AddWord extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_feedback);

        try {
            Intent emailIntent = new Intent(Intent.ACTION_SEND);
            emailIntent.putExtra(Intent.EXTRA_EMAIL, new String[]{"teaty5.0@gmail.com"});
            emailIntent.putExtra(Intent.EXTRA_SUBJECT, "Feedback");
            emailIntent.putExtra(Intent.EXTRA_TEXT, "");
            emailIntent.setType("message/rfc822");
            startActivityForResult(Intent.createChooser(emailIntent, "Choose an Email client"), 0);
        }
        catch (ActivityNotFoundException anfe){
            Toast toast = Toast.makeText(AddWord.this, "Sorry your mobile device does not have an email client installed or logged into ", Toast.LENGTH_LONG);
            toast.show();
        }

    }

    @Override public void onActivityResult(int requestCode, int resultCode, Intent data) {
        // TODO Auto-generated method stub if(requestCode == 0) {
        // You will get callback here when email activity is exited
        // perform you task here
        startActivity(new Intent(this, MainActivity.class));
        }
    }
}

```

Figure 4.11 Feedback activity code snippet

4.3.6 Creating the Guessing game activity

The guessing game is based on a simple logic. It displays 4 pictures in a two column grid by support of the GridView and a word beneath. Upon clicking a picture SetOnItemClickListener method is evoked and as such obtains the name of the image clicked. This name is then compared with a predefined or static name which corresponds to the word on the bottom of the activity. Upon successful selection, a success page is displayed and the user is navigated to another game. However, the user is asked to retry upon failure.

The snippets of code illustrates part of the Guessing Game activity

```

setContentView(R.layout.activity_game2);

final MediaPlayer playsound = MediaPlayer.create(this, R.raw.book);
Button playsound_button = (Button) this.findViewById(R.id.playsound);
playsound_button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        playsound.start();
    }
});

CustomGrid adapter = new CustomGrid(GameActivity2.this, test, imgId);
grid=(GridView)findViewById(R.id.grid);
grid.setAdapter(adapter);
grid.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view,
        int position, long id) {
        Toast.makeText(GameActivity2.this, "You Clicked at " + test[position], Toast.LENGTH_SHORT).show();

        if (test[position].equals(ANSWER)) {
            //testing onclick listener
            //Toast.makeText(GameActivity2.this, "You Clicked at " + test[position], Toast.LENGTH_SHORT).show();
            correctResults();
        } else {
            wrongResults();
        }
    }
});
}

```

Figure 4.12: Guessing game code snippet

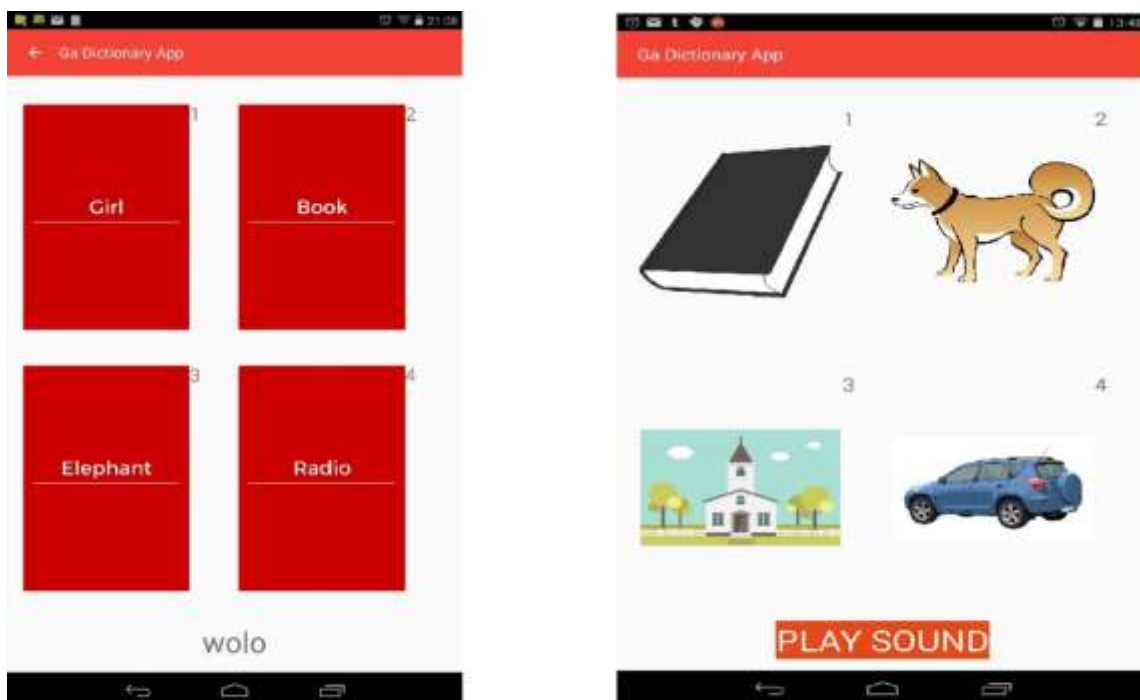


Figure 4.13 Guessing game UI

4.3.7 Creating the main search functionality

The main dictionary activity starts off as a list of all the available English words with Ga translations in a ListView. It is however paired with a search view which filters though the list upon user input. An onItemClickListener is set on the list adopter and the id of the word clicked is sent within an intent to the display activity.

The display activity deploys a tabbed view with three sections. One which contains the word, the other the translation and the last, its sound. This employs dynamic fragments.

```

this.listView = (ListView) findViewById(R.id.listView);
this.searchView= (SearchView) findViewById(R.id.searchView2);
DatabaseAccess databaseAccess = DatabaseAccess.getInstance(this);
databaseAccess.open();
final List<String> quotes = databaseAccess.getEnglishWords();
databaseAccess.close();

final ArrayAdapter<String> adapter = new ArrayAdapter<String>(this, android.R.layout.simple_list_item_1, quotes);
this.listView.setAdapter(adapter);

listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    public void onItemClick(AdapterView<?> parent, View view,
        int position, long id) {

        // testing setOnItemClickListener
        String val =(String) parent.getItemAtPosition(position);
        System.out.println("Value is "+val);

        // sending the item clicked to a new activity
        Intent dictionaryListIntent = new Intent(DictionaryList.this,DictionaryContent.class);

        dictionaryListIntent.putExtra(INTENT_STRING, val);
        startActivity(dictionaryListIntent);

    }
});

```

4.3.8 Creating the Suggest a Word Activity

This activity involves a connection to a PHP server and as such utilizes an adapter class which extends `AsyncTask`. When a user suggests a word, the word is stored on the server and awaits verification from an administrator—in this case the administrator could be a lecturer or individuals who has in-depth knowledge in the Ga language. This however is paired with another functional requirement which allows the user to see newly added words. The user can only view words that have been certified by the administrators. It is important to note that the newly added words exist as a separate database on a database server from the primary database of words stored on the device. After some time the newly added words are added to the primary word database together with sound, pronunciation, proper use in a sentence and phonetic structure and the user gets direct access to via an update.

Below a diagram indicating code snippets.

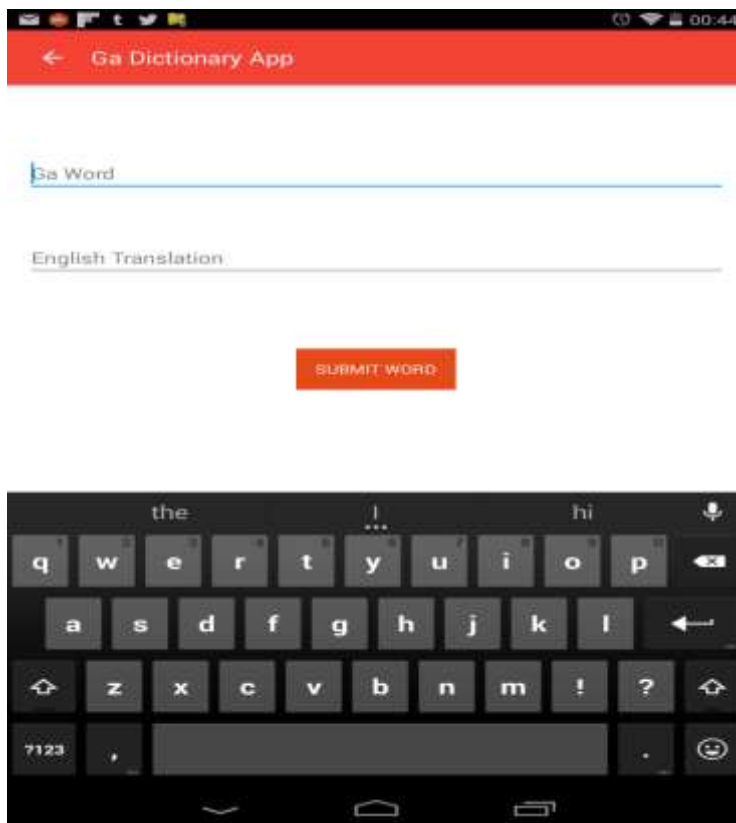
```
String method = params[0];
if (method.equals("suggestWord")) {
    String ga = params[1];
    String english = params[2];

    try {
        URL url = new URL(suggestword_url);
        try {
            HttpURLConnection httpURLConnection = (HttpURLConnection)url.openConnection();
            httpURLConnection.setRequestMethod("POST");
            httpURLConnection.setDoOutput(true);

            OutputStream OS = httpURLConnection.getOutputStream();
            BufferedWriter bufferedWriter = new BufferedWriter(new OutputStreamWriter(OS, "UTF-8"));
            String data= URLEncoder.encode("ga", "UTF-8") + "=" + URLEncoder.encode(ga, "UTF-8")+ "&" +
                URLEncoder.encode("english", "UTF-8") + "=" + URLEncoder.encode(english, "UTF-8");

            bufferedWriter.write(data);
            bufferedWriter.flush();
            bufferedWriter.close();
            OS.close();
            InputStream IS = httpURLConnection.getInputStream();
            IS.close();
            return "Word Successfully Added";
        }
    }
}
```

Figure 4.12 Word suggestion activity code snippets.



Figure

4.3.8 Creating and Implementing Sounds (word pronunciations)

Each Ga word available in the database was pronounced by a sound model by name Bridget Asantewa Acheampong. These wave files were later converted to mp3 format and stored with the raw folder within the resources directory.

Chapter 5: Tests

This chapter examines the tests put in place to ensure that the application meets the functional and non-functional requirements.

5.1 Feature Tests

These tests involve a simulation of possible user inputs and interactions to fully meet the user's need without errors.

The tables below give an overview of the tests carried out.

Table 5.1 : Search Performance Test

Search Tests

Test	Results : Display in List View		Description
	Yes	No	
Full word (word exists)	✓		The word is displayed in the list
Full word (word does not exist)		✓	The word is not displayed in the list
Partial word (beginning)	✓		Displays the a list of words—desired word inclusive, that start with the typed in letters
Partial word (middle or end of desired word)		✓	Does not display desired word. It however displays other words that have the

			inserted words as the beginning
Word does not exist		✓	Displays an empty list.
White space		✓	Displays an empty list if the white space is before any word.

The table above indicates the test results on the primary word search function. Upon search via text or speech of a desired word (if exists within the database). The desired word is shown within the list can be clicked to display the next screen which contains its translation and sound among other discussed outputs.

Table 5.2 Game Tests

Gaming Tests

Test	Results: Displays necessary results		Description
	Yes	No	
Click on correct answer	✓		Displays success and links via button to the next page
Click on wrong answer	✓		Displays failure and links the same activity upon hitting retry

The game activity tests here indicate the possible eventualities tied to the guessing game.

5.2 Hardware tests

Beyond the tests run within the Android Studio emulator, the following hardware tests were carried out.

5.2.1 Screen size and form factor test

The application was run on the Samsung Galaxy S6 edge smartphone, which had a 5.1 inch screen as well as the Dell Venue 8 Android table with an 8 inch screen. This was to test for distortions in the user interface and user experience. The conclusions however were favorable in terms of a uniform graphical user interface though the Samsung Galaxy S6 edge run relatively faster and smoother.

The image below illustrates screen size success



Figure 5.1 Screen size test

5.2.3 Speed Tests

Two mobile devices were used for this comparison. One, a high end Samsung Galaxy S6 edge with 3GB RAM and a quad-core CPU, and the other, a Dell Venue 8 with a RAM size 2GB and a duo-core processor.

Very little differences could be observed in general. However the Samsung Galaxy S6 edge proved smoother of the two when it came to sliding through high resolution proverb tabs.

5.2.4 Time Tests

The application adheres to less than a second of activity transitioning. It however contains a number of images thus causing a little lag on when using devices such as the Samsung Galaxy Pocket which has 289MB of RAM.

5.3 User Tests

A sample of 9 students of Ashesi University College were selected for the user tests. These individuals were broken into 3 equal categories to obtain unbiased insights. The first category of students were students who had studied a local language during their primary education, the second international students and the third computer science students.

The following were the insights obtained from each group.

Table 5.3 Insights

Group	Insights (Emphasis of interest)
Students who studied a local language in basic school	<ul style="list-style-type: none"> • Sound pronunciation • Word translation • A list of all the words to provide a nostalgic feel • Proverbs
International students	<ul style="list-style-type: none"> • Sound pronunciation • Word translation • Phonetic structure • Guessing game • Preferred step-by-step lessons
Computer science students	<ul style="list-style-type: none"> • Speed • Possible errors

The information obtained was relevant to the confirmation and updating of user requirements.

Chapter 6: Conclusion

The current state as of the submission of this documentation accounts for 80% of functional and non-functional requirement completion. It however concludes on a successful note with the absence of a tourist mode.

6.1 Limitations

The limitations of the project at its current state can be found within its scope, that is, the provision of single word translations. There however exists much more sophisticated tools and technologies such as Rule-based Machine Translation or Statistical Machine Translation employed within other language translators such as the Word Lens applications, which translates in real time as the user hovers his or her camera's lens over an image containing content that needs to be translated. This however is an indicator that there is more room for improvement.

Other limitations include using static fragments for the proverbs activity, using a lot of high resolution pictures and using on-board sound rather than obtaining sound from a server, which may be sufficient for now but may cause storage problems as the database expands.

Future Development

Potential developments for the application lies within the areas of improvement and testing.

Preliminarily, alpha testing using the Google Play Developer Console to eliminate bugs would be a first step towards improvement. However, the next stage is to concentrate on ways to employ Rule-based machine translation technologies to ensure the strict adherence to rules and word combination, thus expanding to word combinations, phrases and sentences.

References

- AT&T launches an SMS-based language translation service called Muuzii. (2014, June 25). Retrieved from gigaom.com: <https://gigaom.com/2014/06/25/att-launches-an-sms-based-language-translation-service-called-muuzii/>
- Dakubu, M. K. (2006). *Ga- English Dictionary*. Accra: Black Mask Limited.
- Developer.Apple. (2015, April 28). *Swift*. Retrieved from developer.apple.com: <https://developer.apple.com/swift/>
- Duolingo. (2016, April 15). *duolingo*. Retrieved from duolingo.com: <https://www.duolingo.com/>
- Ghana Embassy. (2016, April 15). *Language and Religion*. Retrieved from ghanaembassy.org: <http://www.ghanaembassy.org/index.php?page=language-and-religion>
- Ghana National Communications Authority. (2016, April 20). *Mobile Data Figures For The Month Of December 2015*. Retrieved from nca.org.gh: <http://www.nca.org.gh/73/34/News.html?item=583>
- Gigaom. (2014, June 25). *AT&T launches an SMS-based language translation service called Muuzii*. Retrieved from gigaom.com: <https://gigaom.com/2014/06/25/att-launches-an-sms-based-language-translation-service-called-muuzii/>
- Google. (2016, April 15). *Translate*. Retrieved from translate.google.com.gh: http://translate.google.com.gh/about/intl/en_ALL/languages/
- Index Mundi. (2015, June 30). *Ghana Demographics Profile 2014*. Retrieved from <http://www.indexmundi.com/>: http://www.indexmundi.com/ghana/demographics_profile.html
- InfoWorld. (2015, February 14). *How to choose a database for your mobile apps*. Retrieved from infoworld.com: <http://www.infoworld.com/article/2887754/mobile-technology/how-to-choose-a-database-for-your-mobile-apps.html>
- Kramsch, C. (2003). *Language and Culture*. New York: Oxford University Press.
- Mahapatra, L. (2013, November 11). *Android Vs. iOS: What's The Most Popular Mobile Operating System In Your Country?* Retrieved from ibtimes.com: <http://www.ibtimes.com/android-vs-ios-whats-most-popular-mobile-operating-system-your-country-1464892>
- Merritt, A. (2013, August 5). *Learn a language for free: the five best websites*. Retrieved from telegraph.co.uk: <http://www.telegraph.co.uk/education/educationadvice/10218763/Learn-a-language-for-free-the-five-best-websites.html>
- Mushtaq, H. (2015, May 24). *Tutorial for Creating Android Swiping Application using Tabbed Activity on Android Studio*. Retrieved from <http://haidermushtaq.com/>: <http://haidermushtaq.com/tutorial-for-creating-android-swiping-application-using-tabbed-activity-on-android-studio/>
- Owocki, G. (1999). *Literacy Through Play*. March: Heinemann Educational Books.
- Pew Research Center: Global Attitudes & Trends. (2015, April 14). *Few Own Smartphones in Africa, But Cell Phones Common*. Retrieved from pewglobal.org:

<http://www.pewglobal.org/2015/04/15/cell-phones-in-africa-communication-lifeline/africa-phones-5/>

Safaba Translation Innovation. (2016, April 15). *Hybrid MT*. Retrieved from safaba.com:
<http://www.safaba.com/machine-translation/machine-translation-technologies/hybrid-machine-translation>

Safaba Translation Innovation. (2016, April 15). *Rule-Based MT*. Retrieved from safaba.com:
<http://www.safaba.com/machine-translation/machine-translation-technologies/rule-based-machine-translation>

Safaba Translation Innovation. (2016, April 15). *Statistical MT*. Retrieved from safaba.com:
<http://www.safaba.com/machine-translation/machine-translation-technologies/statistical-machine-translation>

Windows Dev Center. (2016, April 28). *Getting Started* . Retrieved from msdn.microsoft.com:
[https://msdn.microsoft.com/en-us/library/windows/apps/ff402529\(v=vs.105\).aspx](https://msdn.microsoft.com/en-us/library/windows/apps/ff402529(v=vs.105).aspx)

World Guides. (2016, April 15). *Kotoka International Airport (ACC) Information*. Retrieved from world-guides.com: http://www.world-guides.com/africa/western-africa/ghana/ghana_airport.html

Wulomei.bb. (2016, April 15). *The Ga Language*. Retrieved from wulomei.bb:
<http://www.wulomei.bb/language.php>

Ziflaj, A. (2014, August 15). *Native vs Hybrid App Development*. Retrieved from sitepoint.com:
<http://www.sitepoint.com/native-vs-hybrid-app-development/>