International Journal of Research Studies in Science, Engineering and Technology
Volume 2, Issue 7, July 2015, PP 174-184
ISSN 2349-4751 (Print) & ISSN 2349-476X (Online)

The Application of Real-Time Voice Recognition to Control Critical Mobile Device Operations

Omyonga Kevin[1], Kasamani Bernard Shibwabo[2]

[1]Faculty of Information Technology, Strathmore University, Nairobi, Kenya
[1]komyonga@gmail.com, [2]bshibwabo@strathmore.edu

Abstract: *The consistent increase in the number and ownership population of mobile devices introduces a variety of limitations. A set of this limitations revolve around interactivity. The overly dependent haptic mechanism of interaction has caused device falls, slower time to interaction, health concerns, and limited support for the disabled among other problems. There is need to formulate innovative techniques that facilitate our interaction with these devices for users. In order to achieve this, a Real-time Voice Recognition Algorithm is formulated that lets users of mobile devices acquire freedom to move about and reduce the need for constantly glancing at their screen. This is achieved by allowing users to verbally command their devices to carry out ordinary tasks such as setting an alarm, making a call, or even starting any application. An added unique feature is that it also offers offline access as any commands given by a user are processed and executed locally on the device.*

Keywords: *Voice Recognition, Mobile Digital Assistant, Audio Pattern Matching, Audio Control, Mobility.*

1. Introduction

The smartphone market is one of the most competitive markets in the world today with various competitors such as Samsung, Apple, Huawei, Google, Sony, Microsoft, and Research in Motion among others being locked in a tight race to maintain or emerge in the top position. At first, it was all about the devices' hardware but now software has arisen as one of the key determinants of a devices failure or success [1].

One key software that is making waves is the Mobile Digital Assistant. An application that allows one to interact with their devices through speech. The most well-known of these applications are Apple's Siri and Google Now [2]. These applications allow one to accomplish tasks such as making phone calls, setting alarms and checking for data such as weather.

Though these applications do get the job done, there are a few drawbacks. For starters, most of them often require that a Wi-Fi/mobile data connection be present in order for them to work. This is because they first capture the user's speech as the user verbally issues a command and then proceed to upload this data to an online server which then interprets the data and deciphers the command which is then sent back to the device and finally executed [3].

Table 1 shows some of the most popular Mobile Digital Assistants and their respective firms.

Table1. *Mobile Digital Assistants (Adapted from Reddy, 2014)*

FIRM	APPLICATION
Apple	Siri
Google	Google Now
Samsung	S Voice
Microsoft	Cortana

Without internet, the functionality of such internet-dependent applications is greatly crippled. These applications also require a high bandwidth so as to be able to execute the user's commands in a timely fashion. If this is not available, the applications tend to operate at a sluggish pace and can often inconvenience the user. There are also additional modules which are not available on the closed-source solution sin the market that need to be developed. Additionally, these vendors often have a bias to mainly target the environment that supports their interests.

Additionally, internet connection is not only costly especially in developing countries but it is a major concern in terms of battery use especially when shared with other active applications. The limitation of battery capacity has been a major challenge in mobile devices and technology companies are working on technologies to double, triple or generally expand the current capacity.

2. Review of Related Literature

2.1. Mobile Digital Assistants

Mobile Digital Assistants are mobile applications that act as software agents/ middle men to allow easier interaction between users and their mobile devices. These applications can perform tasks, or services, for an individual based on user input, location awareness, and the ability to access information from a wide variety of online sources such as news, stock market, traffic updates, user schedules and retail prices among others. Examples of such applications are Google Now, Apple's Siri, Microsoft Cortana, Samsung's S Voice and HTC's Hidi [4].

Figure 1 shows two devices, a Google Nexus 4 and an Apple iPhone 4S with each displaying their respective company's Mobile Digital Assistant.

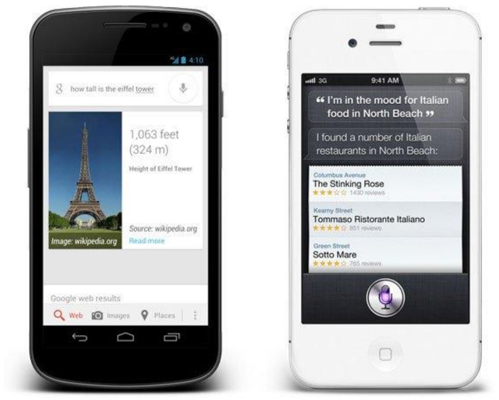


Fig1. *Google Now and Apple's Siri*

Mobile Digital Assistants are designed to use speech input. This is because speech is ideally suited to mobile computing since a single verbal command can accomplish a variety of tasks that would normally require a multitude of swipes and presses. This makes using a mobile device much easier especially in cases where a user may have their hands and eyes occupied with another more pressing task such as driving [5].

2.2. Mobile Digital Assistants General User Interface Design and Functionality

Up until recently, the idea of interacting with a mobile device through speech was deemed pure science fiction. Such applications were only seen in movies (Bosker, 2013). That has all changed. Various companies like Google, Apple and Microsoft are spearheading the development of this applications with each attempting to deliver the most efficient user experience. These applications are designed to use a voice-controlled interface. As a result, a growing number of people now talk to their mobile smart phones asking them to carry out various activities such as sending emails, making phone calls and sending text messages [5].

Figure 2 shows Apple's Siri user interface which comprises of help text to give suggested commands and one button which is used to activate the application.

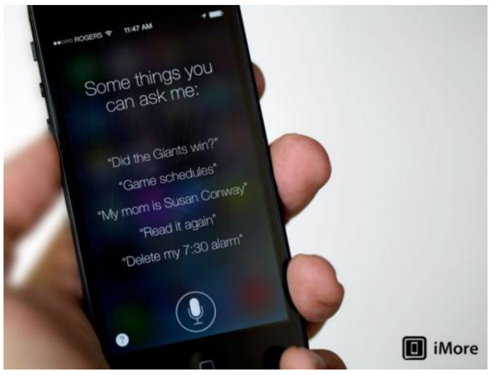


Fig2. *Siri showing user voice-controlled interface*

A Mobile Digital Assistant allows hands free utilization of a mobile device's resources. Most Mobile Digital Assistants operate by receiving user commands, processing them and performing/executing the action requested by delegating requests to a set web services. This allows the applications to have access to a vast database of knowledge which is their greatest strength. However, this strength does have its drawbacks in that these applications often require an internet connection for them to function. Without the connection, their functionality is limited and in some cases they will not function at all [2]. There are some companies however that are working on eliminating this extreme reliance on internet connection [3].

Figure 3 shows Apple's Siri help interface which gives a list of commands that a user can execute using the application.

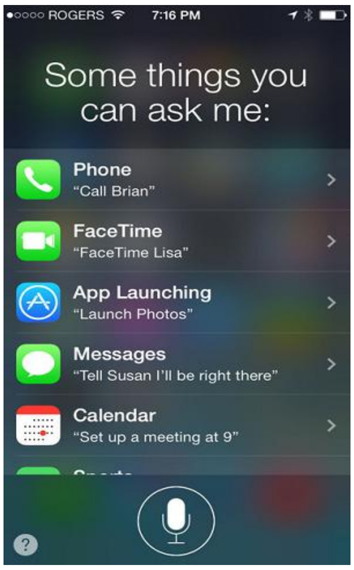


Fig3. *Screenshot of Apple's Siri detailing some functionalities on iOS 7*

Mobile Digital Assistants are quickly becoming a popular trend in the smartphone industry. These applications have allowed what was once perceived to be science fiction to become a reality. Users are now able to interact with their devices using only their voice.

Despite this promising developments in the quest for ergonomic simplicity, the applications that are currently available do have their short comings. Having experimented with Google Now and Apple's Siri, it was concluded that the existing applications are "data guzzlers" due to the fact that they require a lot of internet connection for them to properly function. Another drawback is that these applications are only available on high end devices which are not affordable by everyone on the market.

3. The Proposed Real-Time Voice Recognition Solution

3.1. The Way Forward

In order to develop the proposed solution, a high quality object oriented application object-oriented analysis and design methodology was applied. In an object oriented analysis and design complex software system can be broken down into various objects, combining the data and the functions that operate on the data into a single unit, the object. This form of object decomposition provides a natural way of breaking the problem down into isolated, manageable parts. In many cases, the development effort shifts from writing a new code, to assembling existing objects in new and innovative ways to solve a problem. Thus, object-oriented analysis and design methodology cuts down development time and costs, leading to faster time to market and significant competitive advantage, and enables producing more flexible, modifiable, easily maintainable object-oriented systems [6].

The life cycle of development of object oriented systems development involves three phases which are analysis phase, design phase and implementation phase. Analysis phase is when model of a real world application is developed showing its properties and functional behaviours. Design phase when an analysis model is refines and adopted to the system environment and implementation phase when the design model is implemented using programming language or database management system [6]. Table 2 presents the stages that coming up with the solution involved.

Table2. *Methodology Stages*

Stages	Activity	Outcome
System Analysis	Requirements gathering.	Functional and Non-functional requirements. Use Case Diagram
System Design	Database Schema Design. Class Diagram Design.	Database Schema. Class Diagram.
Testing	Test running the application with basic commands.	A basic voice recognition application.
Debugging	Eliminating bugs encountered during the testing of the application.	Error handling procedures to prevent the application failing when executing commands.
Maintenance and Expansion	Expanding the commands that the application is capable of executing.	An application capable of executing more complex commands and wake up when called.
Documentation	Documenting the application.	Project Documentation

3.2. Analysis and Design

3.2.1. Functional Requirements

Table 3 lists the functional requirements of the project as well as their descriptions. These are the functions that the complete application can perform.

Table3. *Functional Requirements*

Functionality	Description
Offline Voice Recognition	The application is able to recognize a user’s verbal commands without the need for an internet connection.
Making Calls	The application can receive a contact name or number verbally and make a phone call.
Search	The application is able to search the device for a user defined search term.
Social Media Interaction	The application is able to post to social media and read out notifications verbally.
Updates	The application is capable of giving updates on factors like time, weather and traffic.
Opening Applications	The application allows the user to access other applications using verbal commands.
Wake Up Trigger	The application allows a user to activate it without having to push a button.

3.2.2. Non-functional Requirements

Table 4 lists the non-functional requirements of the project as well as their descriptions. These are the features and behaviours that the application will be able to provide to the users.

Table4. *Non-Functional Requirements*

Non-Functionality	Description
Availability	The application is installed on the user’s smartphone and can be accessed any time the user wishes to interact with it.
Reliability	The application is able to function even in the absence of an internet connection.
Efficiency	Voice recognition is quite fast as the processing of commands is done on the device itself without any need for cloud processing.

Additionally, in order to explain the functional requirements further, a Use case diagrams is formulated. Use cases are useful for presentations to management and/or project stakeholders, but for actual development you will find that use cases provide significantly more value because they describe "the meat" of the actual requirements [7]. Figure 4 summarizes the operations on a Use case diagram.

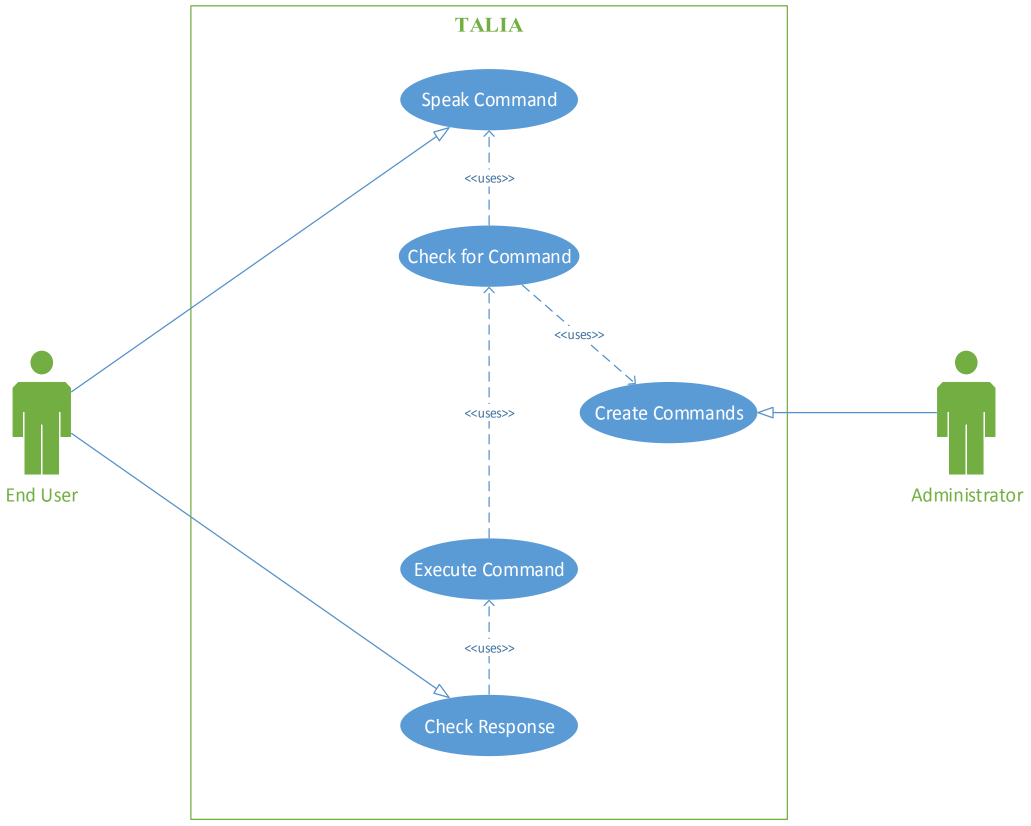


Fig4. *The Use Case Diagram for the Application*

Class diagrams show the classes of the system, their interrelationships (including inheritance, aggregation, and association), and the operations and attributes of the classes [7]. Figure 5 presents the class diagram for the Afya App application.

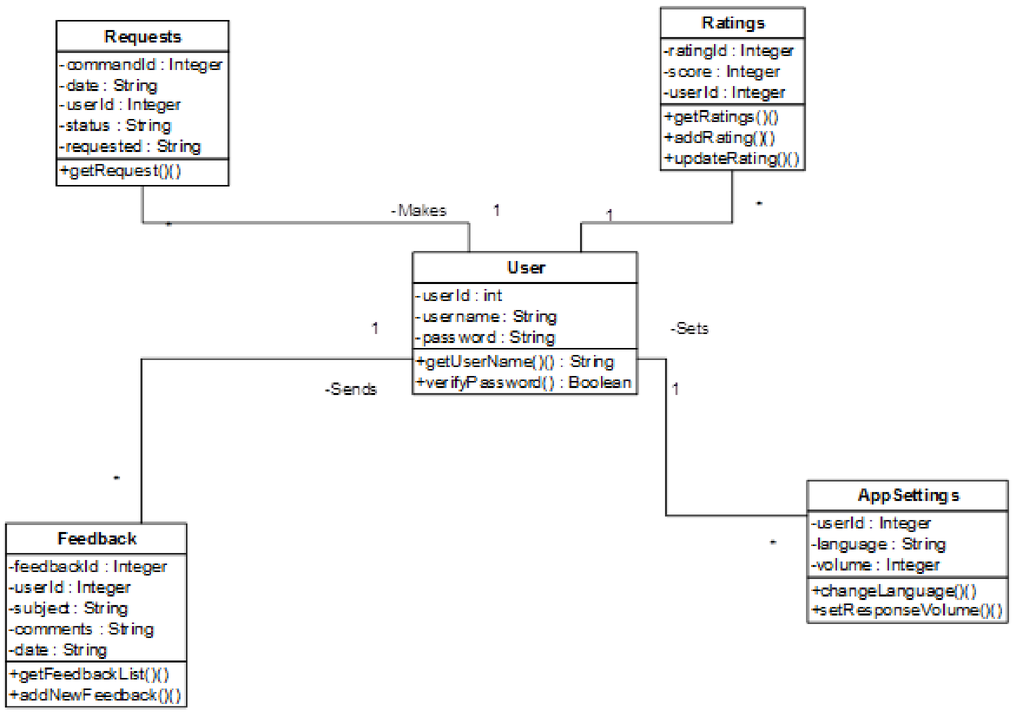


Fig5. *The Class Diagram for the Application*

A Schema diagram represents the logical view of entire database. It tells about how the data is organized and how relation among them is associated. It formulates all database constraints that would be put on data in relations, which resides in database [8]. Figure 6 presents the schema diagram for the proposed solution.

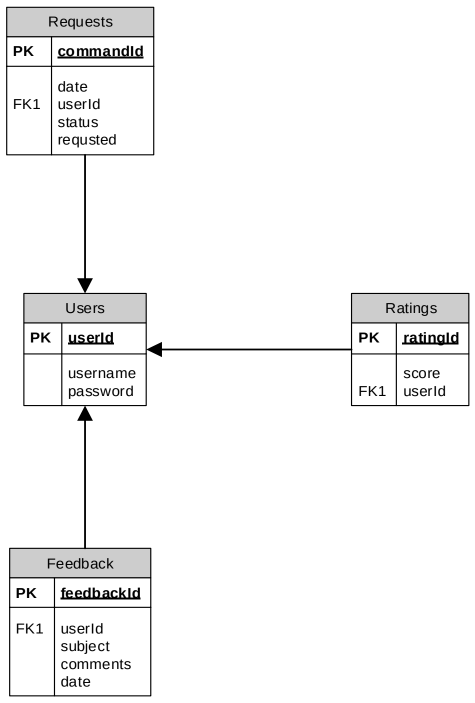


Fig6. *The Database Schema for the Application*

3.3. Implementation and Testing

For the development of this application Eclipse was initially the Integrated Development Environment (IDE) of choice due to its ease of compatibility with the Android Software Development Kit (SDK). However, during the course of this project, Google perfected and released an official IDE for the Android platform named Android Studio that was aimed at replacing Eclipse. This events necessitated a shift that halted progress temporarily, but helped to greatly improve the application thereafter due to smart code completion, optimization tips and better module simulation.

The programming language used is Java. This is because it is the recommended language for the development of native android applications. It is used to implement the functionality of the application. XML (Extensible Mark-up Language) was used to design the graphical layout of the application due to it being much easier to use as it allowed the definition of custom tags which in turn aided in the design of a beautiful user interface that corresponds to the functionality defined in the application’s Java classes.

In order to store data, there is often need to use a standard way of storing data, for this purpose an SQLite database is built into an android platform to store and retrieve data that is relevant to the application such as the song names on the user’s device. This is because it is lightweight and less resource intensive compared to MySQL, Oracle, PostgreSQL among other alternatives.

The solution has been implemented and tested on the android platform. The specific version of android that was targeted for testing is Android version 4.2.2 (Jelly Bean).

3.3.1. System Modules

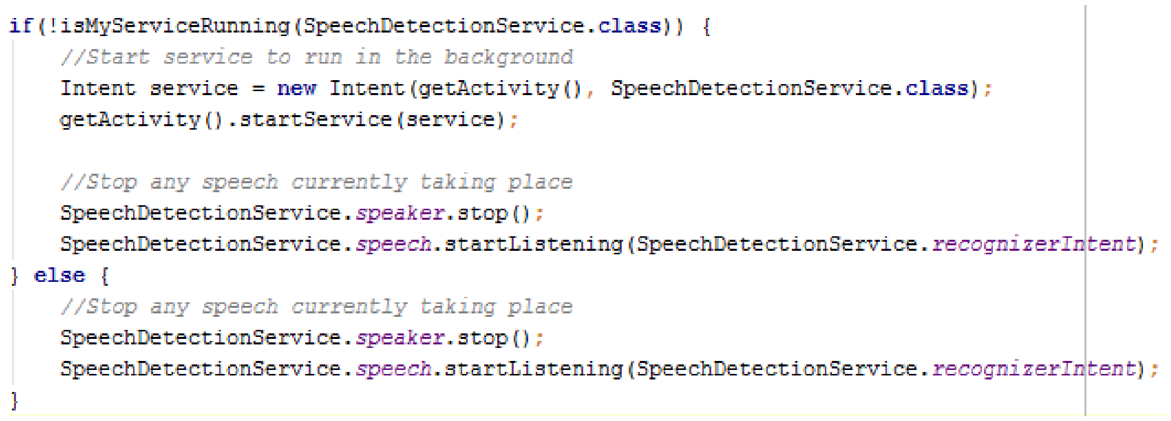
The application consists of the following modules:

- Chat Room Module: This module was designed to mimic the interface of most messaging and chat applications like ‘WhatsApp’. This was intended to give the user a more personal and unique user experience of actually chatting with the device. A user’s command is to be received and be displayed on the right side of the screen in a purple chat bubble. The application’s response is then displayed to the left of the screen in a light blue chat bubble.
- Commands Module: This module was designed to act as a reference point for users who wish to know the type of commands the application can accept, process, and execute. It encompasses all the commands that are coded into the application’s knowledge base and comes in handy when a user is accessing the app for the first time.

- Feedback Module: This module is designed to give users a means to send feedback on the app to the developer as well as rate their experience with the app. This helps the developer know the users' opinions about the app and make improvements where necessary.

3.3.2. Sample Code for the Algorithm

The code in Figure 7 is used to activate the Speech Detection Service of the application. When it is being executed, any speech that the device was making is stopped so as to avoid clashing with the user's own voice when issuing the command. Figure 7 shows Speech Detection Service Activation.

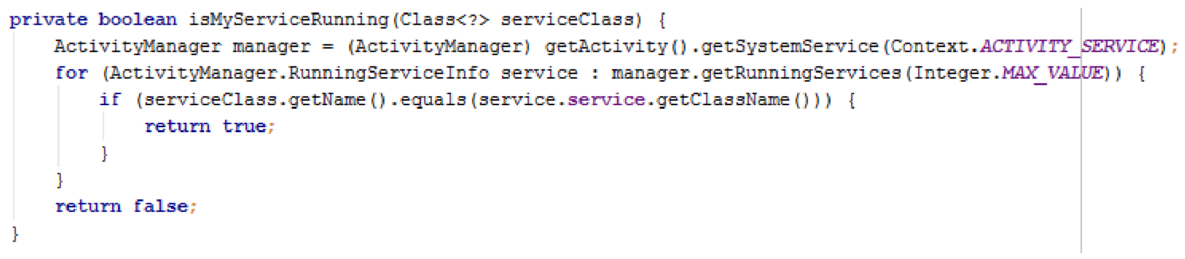


```
if(!isMyServiceRunning(SpeechDetectionService.class)) {
    //Start service to run in the background
    Intent service = new Intent(getActivity(), SpeechDetectionService.class);
    getActivity().startService(service);

    //Stop any speech currently taking place
    SpeechDetectionService.speaker.stop();
    SpeechDetectionService.speech.startListening(SpeechDetectionService.recognizerIntent);
} else {
    //Stop any speech currently taking place
    SpeechDetectionService.speaker.stop();
    SpeechDetectionService.speech.startListening(SpeechDetectionService.recognizerIntent);
}
```

Fig7. *Speech Detection Service Activation*

Figure 8 is the Speech Detection Service State Checker. It is is a function that is called in the if-else statement shown in Figure 7. It is used to check if the Speech Detection Service is already running or not. Its return value is Boolean (true or false). If the service is running, it returns true. If not, the value is false.



```
private boolean isMyServiceRunning(Class<?> serviceClass) {
    ActivityManager manager = (ActivityManager) getActivity().getSystemService(Context.ACTIVITY_SERVICE);
    for (ActivityManager.RunningServiceInfo service : manager.getRunningServices(Integer.MAX_VALUE)) {
        if (serviceClass.getName().equals(service.service.getClassName())) {
            return true;
        }
    }
    return false;
}
```

Fig8. *Speech Detection Service State Checker*

3.3.3. How to Interact with the System

Once the device is set up and ready, the application can then be installed into the device for the user to operate.

- Open the device app drawer and locate the 'TALIA' app launcher icon.

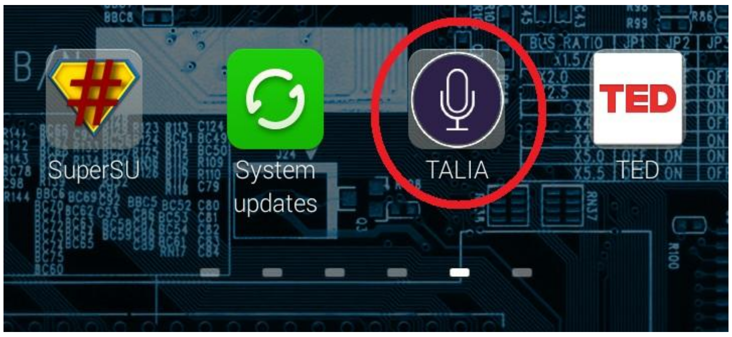
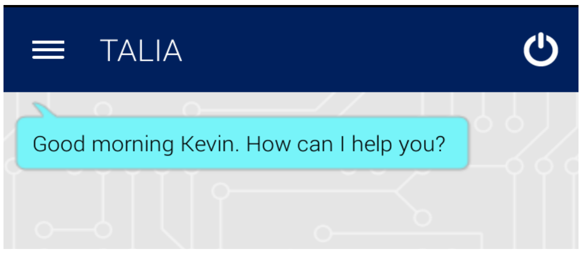


Fig14. *TALIA App Launcher Icon*

- The app will start up and prepare the voice recognition service as well as the text to speech service. Once completed, a greeting will be issued to the user.



Good morning Kevin. How can I help you?

Fig15. *TALIA App Greeting*

- To issue a command to the app, tap on the microphone button at the bottom of the 'Chat Room' or say 'Device On' within earshot of the device. Once this is done, you should hear a short beep to indicate that the app is listening and awaiting your command.

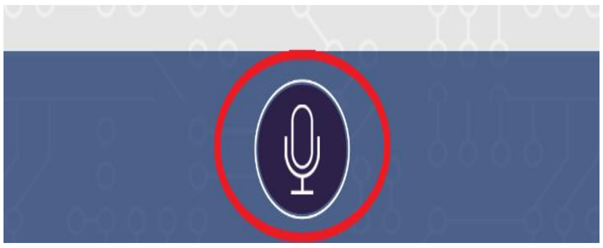
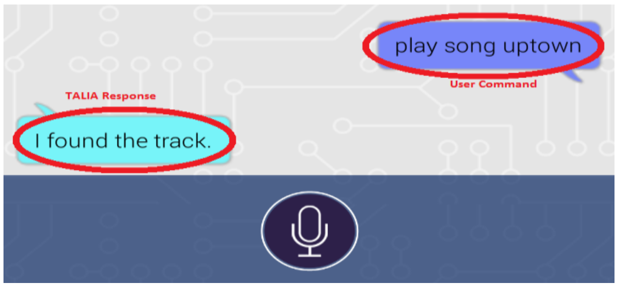


Fig16. *TALIA App Activation Button*

- Once the command is received, the app will process it and issue a response.



play song uptown

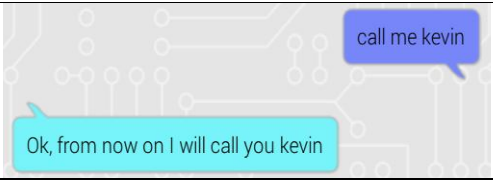
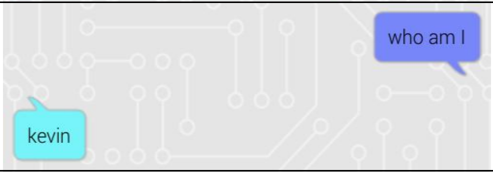
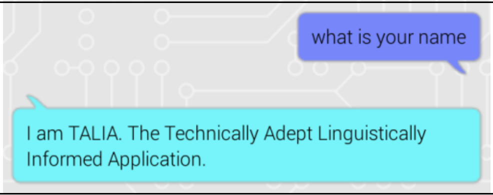
I found the track.

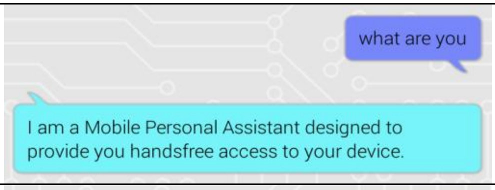
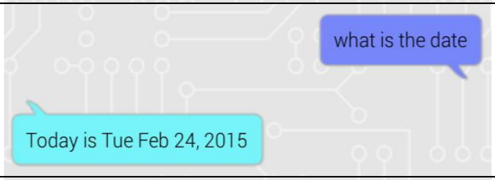
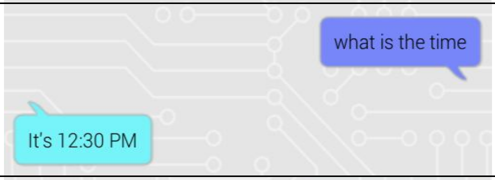
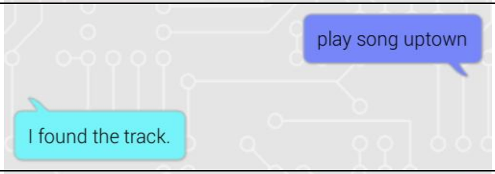
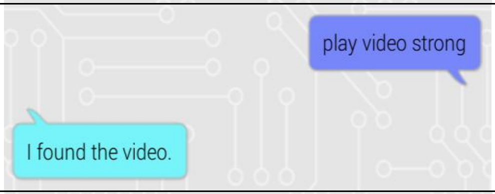
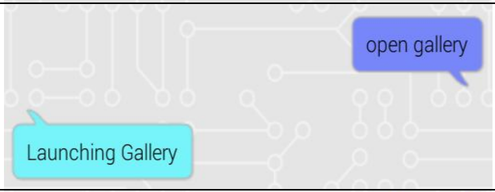
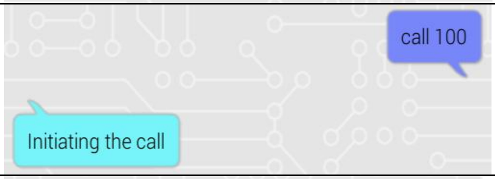
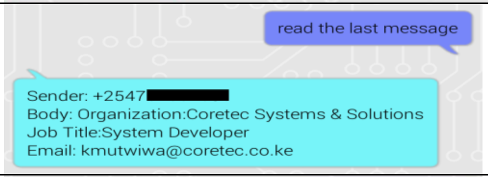
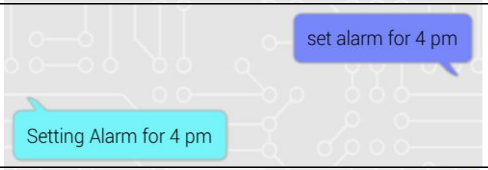
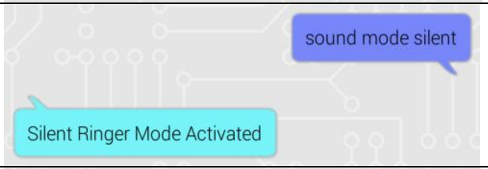
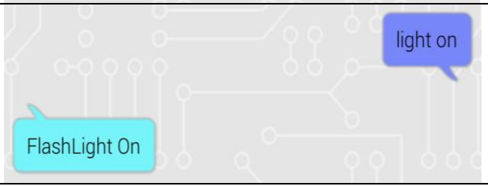
Fig17. *TALIA Command Execution*

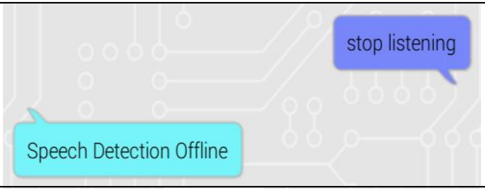
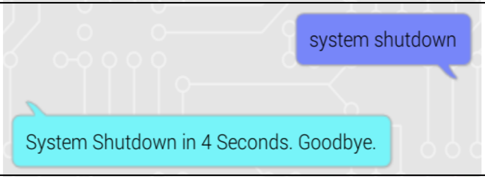
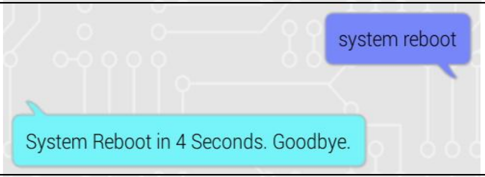
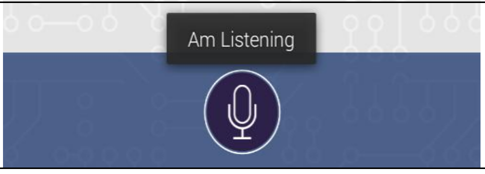
3.3.4. System Sample Test Results

All the commands in the system's knowledge base as listed in Table 5 were tested for documentation purposes so as to give a glimpse of how the app would respond when the commands were received and executed.

Table5. *Test Results for the Commands*

Command	**Function**	**Demo**
"Call me" + Preferred Name	This command allows the user to give the app a name by which to refer to them so as to give the app a more personalized feel.	 call me kevin Ok, from now on I will call you kevin
"Who am I?"	Used to enquire the name that the app will use to identify the user.	 who am I kevin
"What is your name?"	Gives the app an opportunity to introduce itself to the user.	 what is your name I am TALIA. The Technically Adept Linguistically Informed Application.

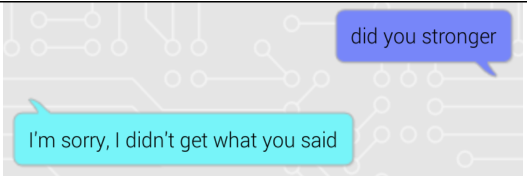
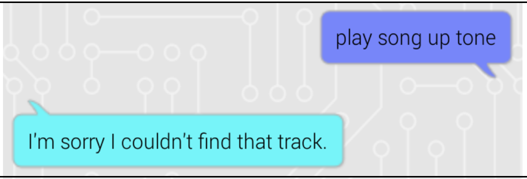
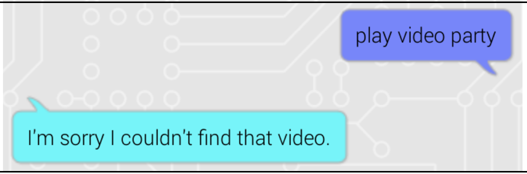
“What are you?”	This lets the app give a brief description about itself and what it can do.	 what are you I am a Mobile Personal Assistant designed to provide you handsfree access to your device.
“What’s the date?” “What is the date?”	This query yields the current date of the user’s device.	 what is the date Today is Tue Feb 24, 2015
“What’s the time?” “What is the time?”	This query yields the current time of the user’s device.	 what is the time It's 12:30 PM
“Play song” + song name	Starts up a music player on the user’s device and plays the requested song if it is located on the device.	 play song uptown I found the track.
“Play video” + video name	Starts up a video player on the user’s device and plays the requested video if it is located on the device.	 play video strong I found the video.
“Open” + app name	Launches the requested app if it is located among the installed apps on the user’s device.	 open gallery Launching Gallery
“Call” + phone number	Initiates a call to the number spoken by the user.	 call 100 Initiating the call
“Read last message” “Read the last message”	Reads the latest message received on the device out loud.	 read the last message Sender: +2547[redacted] Body: Organization:Coretec Systems & Solutions Job Title:System Developer Email: kmutwiwa@coretec.co.ke
“Set alarm for” + time (am/pm)	Sets an alarm based on the time specified by the user.	 set alarm for 4 pm Setting Alarm for 4 pm
“Sound mode” + preferred state e.g. Normal, Silent, Vibrate	Changes the sound mode of the user’s device to either normal, silent or vibrate based on the preferred state.	 sound mode silent Silent Ringer Mode Activated
“Light on/off”	Turns the device’s flashlight on or off. This command only works on devices that have a flashlight.	 light on FlashLight On

"Stop Listening"	Stops the app's background voice recognition service.	 stop listening Speech Detection Offline
"System Shutdown"	Powerful command that makes the device shutdown completely until the power button is pressed.	 system shutdown System Shutdown in 4 Seconds. Goodbye.
"System Reboot"	Powerful command that makes the device reboot/restart.	 system reboot System Reboot in 4 Seconds. Goodbye.
"Device On"	Activates the app and makes it start listening for a command without the user having to press a button.	

3.3.5. Error Handling

Measures were taken to ensure that the application could offer user friendly responses as shown in Table 6 whenever an error was encountered when processing or executing commands.

Table6. Error Handling Test Results

Error	Description	Demo
Unknown Command	A user may speak a command that for one reason or another may not be recognized by the application. When this happens, the application issues a polite response alerting the user to its inability to understand/ execute the command.	 did you stronger I'm sorry, I didn't get what you said
Unknown Song	A user may ask the application to play a song that is not available in the device database. When this occurs, the application will inform the user that the song they requested could not be found.	 play song up tone I'm sorry I couldn't find that track.
Unknown Video	A user may ask the application to play a video that is not available in the device database. When this occurs, the application will inform the user that the song they requested could not be found.	 play video party I'm sorry I couldn't find that video.

4. Conclusion

The formulation of algorithms that facilitate interactivity in multimedia devices is critical to the usability success of these devices. This challenge has been addressed to a large extent in this paper. The formulated solution is able to process voice commands offline allowing users to cut down on the cost of data bundles. This also helps to make it faster in comparison to alternative applications like Apple's Siri. It can also operate in the background thanks to the inclusion of a wake up service that allows the user to activate it without having to push any buttons thus establishing a fully hands free interface between the user and the device. Moreover, the solution is capable of carrying out a variety of tasks with ease such as setting alarms, telling the date and time, playing music/videos, making phone calls and even shutting down or rebooting a user's device.

Future work needs to be done in addressing accuracy especially if the application is used by speaking commands in noisy environments. This challenge was found to however be partially countered by using a headset / earphones with a microphone. The application also greatly relies on the sensitivity of a device's microphone. Additionally, new language packs for local languages could be developed thus allowing users such as those living in rural areas to have access to voice operated software without necessarily having to learn a new language and/or accent.

REFERENCES

[1] F. Fenzi (2013). Need a Personal Assistant? Try These 4 Apps. [Online]. Available: Inc.: http://www.inc.com/francesca-fenzi/meet-your-new-personal-assistant.html

[2] A. Jesdanun (2013). Strengths and weaknesses of Apple's Siri and Google Now. [Online]. Available: San Jose Mercury News: http://www.mercurynews.com/ci_22740979/strengths-and-weaknesses-apples-siri-and-google-now

[3] C. Mims (2014). Intel's voice recognition will blow Siri out of the water—because it doesn't use the cloud. [Online]. Available: Quartz: http://qz.com/170668/intels-voice-recognition-will-blow-siri-out-of-the-water-because-it-doesnt-use-the-cloud/

[4] M. Elgan (2013). Why are virtual assistant apps so shy? [Online]. Available: Computerworld: http://www.computerworld.com/s/article/9242026/Why_are_virtual_assistant_apps_so_shy_

[5] W. Knight (2012). Where Speech Recognition Is Going. [Online]. Available: MIT Technology Review: http://www.technologyreview.com/news/427793/where-speech-recognition-is-going/

[6] El-Ghareeb, H. (2014, December 5). Object Oriented Analysis and Design. Retrieved from Slideshare: http://www.slideshare.net/helghareeb/object-oriented-analysis-and-design-12164752

[7] S. W. Ambler (2014). UML 2 Class Diagrams: An Agile Introduction. [Online]. Available: Agile Modeling: http://www.agilemodeling.com/artifacts/classDiagram.htm

[8] Tutorialspoint. (2014). DBMS Data Schemas. [Online]. Available: Tutorialspoint: http://www.tutorialspoint.com/dbms/dbms_data_schemas.htm

AUTHORS' BIOGRAPHY

Kevin Omyonga holds a Bachelors of Business and Information Technology from Strathmore University. He has created innovative solutions to day to day problems. At youthful age, he has already worked on: Pregnancy Tracker Android App, Whistle Flare Android App,V.K.O 2014 Campaign App, Strathmore University Graduation App, Kuna Happen App among other solutions.

Dr. Bernard Shibwabo Kasamani earned a Bachelor's Degree in Business and Information Technology, a Master of Science in Information Technology and a doctoral degree in IT from Strathmore University. He is currently serving as the Academic Director in the Faculty of Information Technology at Strathmore University. He is also a member of the steering committee for the Annual Strathmore ICT conference. He has developed major solutions that solve a wide range of societal challenges e.g. Banking, Legal, Academic, Trade, Health. His research interests include Systems Integration, Dataspaces, Data mining, Business Intelligence and Applications.