

VISUALLY IMPAIRED ASSISTANCE APP (VIAA)



By

Khurram Azeem Malik

Waleed Muhammad

Muhammad Asad Ullah

Submitted to the Faculty of Computer Software Engineering Department,
Military College of Signals, National University of Sciences and Technology,
Islamabad in partial fulfilment for the requirement of a B.E. Degree in Computer
Software Engineering

June 2015

CERTIFICATE FOR CORRECTNESS AND APPROVAL

Certified that work mentioned in this thesis “Visually Impaired Assistance App” carried out by Khurram Azeem Malik, Waleed Muhammad, Muhammad Asad Ullah, under the supervision of Dr. Sarmad Sadik for partial fulfillment of Degree of Bachelor of Software Engineering is correct and approved.

Approved by

Dr. Sarmad Sadik

CSE Department

MCS

Dated: 12th June 2015

ABSTRACT

Visually Impaired Assistance Application (VIAA) is specially designed for specific domain i.e. visually impaired people that are eager to use android phone's functionalities but are unable because of the interface provided by android. VIAA will be embedded in the system and will guide visually impaired people to commence specific tasks. As visually impaired doesn't have to do anything with the interface but mostly with the functionalities that android platform provides. Most of the functionalities we will be focusing on are already implemented in different ways but to make the system more user friendly and efficient for visually impaired people, we will focus on basic functionalities like attend or reject call, send or receive messages, alarm settings, control calendar events, weather updates and current location of the visually impaired person.

All functions will be based on voice command that will facilitate the user in a better manner. No skills are required. VIP will only be needed to memories small commands that will be stimulus to the system and application will response accordingly. System will response in the form of speech as we are using Application Programming Interfaces (Text to Speech and Speech to text). Furthermore, it will have only one tap interface that will be opened by clicking VIAA and rest by clicking main screen, commands will be manipulated by the application. Incoming calls automatically pop up and will speak the incoming caller number where as making call to a specific number will receive number from VIPs. In message module, incoming message alerts the VIP that he has new message and by taping screen API translates message to speech and will narrate it for VIP. Sending message module accepts number along with number. In short, its one touch interface that facilitates VIPs and novice users to android plate form that facilitate them to use android phone and its functionalities efficiently and effectively.

DECLARATION

We solemnly declare that the work presented herewith is the result of sole effort of our group, comprising of Khurram Azeem Malik, Waleed Muhammad, Muhammad Asad Ullah, and is free of any kind of plagiarism in part or whole. We also declare that the dissertation has never been submitted previously in part or whole in support of another award or qualification either at this institution or elsewhere.

DEDICATION

To our respected teachers whose kind guidance and unfailing support made this mammoth task easy for us and to our very dear parents whose unceasing prayers gave us strength and courage to complete the work of this magnitude.

ACKNOWLEDGEMENT

We are, very humbly, grateful to Almighty Allah for bestowing us with the strength and resolve to undertake and complete the project.

We owe a special debt of gratitude to our supervisor, Dr. Sarmad Sadik for the continuous supervision, motivation and support provided to us and for their continuous and valuable suggestions, guidance, and instructions from time to time right through the project.

We would also like to acknowledge and thank the beginning faculty members of the department of Computer Software Engineering for their continuous support. It was their help and guidance which helped us complete the project in due time.

Table of Contents

Chapter 1	1
1. Introduction	1
1.1. Overview	1
1.2. Problem Statement	1
1.3. Project Scope	1
1.3.1. Project Vision	2
1.3.2. Project Objectives	2
1.4. Deliverables	2
Chapter 2	4
2. Literature Review	4
2.1. Introduction	4
2.2. Similar Applications	4
Chapter 3	7
3. Overall Description	7
3.1. Product Perspective	7
3.2. Project Modules and Functions	8
3.2.1. Project Modules	8
3.2.2. Project Functions	8
3.3. User Classes and Characteristics:	10
3.4. Operating Environment	10
3.5. Design and Implementation Constraints:	10
3.6. User Documentation:	11
3.7. Assumptions and Dependencies	12
Chapter 4	13
4. Software Requirement Specifications	13
4.1. System Features	13
4.1.1. Make Call:	14
4.1.1.1. Description and Priority	14
4.1.1.2. Stimulus and Response Sequence	14
4.1.1.3. Functional Requirements	14
4.1.2. Answer Call:	14

4.1.2.1.Description and Priority:	14
4.1.2.2.Stimulus and Response Sequence:	15
4.1.2.3.FunctionalRequirements(FR):	15
4.1.3.Send Messages:	15
4.1.3.1.Description and Priority:	15
4.1.3.2.Stimulus and Response Sequence:	15
4.1.3.3.FunctionalRequirements(FR):	16
4.1.4.Read Message:	16
4.1.4.1.Description and Priority:	16
4.1.4.2.Stimulus and Response Sequence:	16
4.1.4.3.FunctionalRequirements(FR):	16
4.1.5.Set Alarm:	17
4.1.5.1.Description and Priority:	17
4.1.5.2.Stimulus and Response Sequence:	17
4.1.5.3.FunctionalRequirements(FR):	17
4.1.6.Delete Alarm:	17
4.1.6.1.Description and Priority:	17
4.1.6.2.Stimulus and Response Sequence:	18
4.1.6.3.FunctionalRequirements(FR):	18
4.1.7.Set Calendar:	18
4.1.7.1.Description and Priority:	18
4.1.7.2.Stimulus and Response Sequence:	18
4.1.7.3.FunctionalRequirements(FR):	19
4.1.8.Delete Calendar Events:	19
4.1.8.1.Description and Priority:	19
4.1.8.2.Stimulus and Response Sequence:	19
4.1.8.3.FunctionalRequirements(FR):	19
4.1.9.Find Current Location:.....	20
4.1.9.1.Description and Priority:	20
4.1.9.2.Stimulus and Response Sequence:.....	20
4.1.9.3.Functional Requirements (FR):.....	20
4.1.10.Weather Updates:	20
4.1.10.1.Description and Priority:.....	20

4.1.10.2.Stimulus and Response Sequence:.....	20
4.1.10.3.Functional Requirements (FR):	20
4.2. Non Functional Requirements	21
4.2.1.Performance Requirements	21
4.2.2.Security Requirements	21
4.2.3.Software Quality Attributes	21
4.2.3.1.Reliability:.....	21
4.2.3.2.Usage Easiness:.....	21
4.2.3.3.Compatibility:	21
4.2.3.4.Overall Satisfaction:.....	22
4.2.3.5.Trust:	22
4.2.3.6.Understandability and Politeness:.....	22
4.2.3.7.Learning:.....	22
4.2.3.8.Maintenance:.....	22
4.2.3.9.Supportability:.....	22
4.2.3.10Adaptability:	22
Chapter 5	23
5. System Design Specifications.....	23
5.1. Overview of Modules/Components.....	23
5.2. Structure and Relationship	24
5.2.1.Basic Layout Diagram:	24
5.2.2.Architecture Diagram:	25
5.2.3.Class Diagram:.....	26
5.2.4.Use Case Realization:	27
5.2.5.Activity Diagram:	28
5.2.6.Sequence Diagrams:.....	28
5.2.7.Interface specification	29
5.2.7.1.Main Screen:	29
5.2.7.2.Call:.....	29
5.2.7.3.Messages:.....	30
5.2.7.4.Alarm Settings:	31
5.2.7.5.Calendar events:.....	31
5.2.7.6.Location:	32

5.2.7.7.Weather:	32
Chapter 6	34
6. System Implementation	34
6.1. Technology Used	34
6.1.1.Programming Language Used.....	34
6.1.2.Development Tools	34
6.1.3.Operating System	34
6.2. Complete System Implementation	34
6.2.1.Calls	34
6.2.2.Messages	34
6.2.3.Alarms.....	35
6.2.4.Events.....	35
6.2.5.Location	35
6.2.6.Weather	35
Chapter 7	36
7. System Testing.....	36
7.1. Overview	36
7.2. Approach	36
7.3. Test Cases	36
7.3.1.Accept Incoming Call Testing.....	36
7.3.2.Reject Incoming Call Testing.....	37
7.3.3.Make Call Testing	38
7.3.4.Read Message Testing.....	38
7.3.5.Send Message Testing	39
7.3.6.Set Alarm Testing.....	40
7.3.7.Delete Alarm Testing	40
7.3.8.Set Calendar Event Testing.....	41
7.3.9.Delete Calendar Event Testing.....	42
7.3.10.Location Finder.....	42
7.3.11.Weather Update Testing	43
Chapter 8	44
8. Conclusion and Future Work.....	44
8.1. Conclusion.....	44

8.2. Future Work.....	44
Appendices.....	46
Appendix A: Use Cases Description:.....	46
Appendix B : Sequence Diagrams	55
Appendix C : Detailed Description of Components	64
Appendix D : User Manual	67
Bibliography	72
Glossary	73

List of Figures

Fig 3.1 : Basic Layout Diagram.....	7
Fig 3.1 : Functions of VAAA.....	9
Fig 4.1 : Use Case Diagram.....	13
Fig 5.1 : Basic Layout Diagram.....	24
Fig 5.2 : Architecture Diagram.....	25
Fig 5.3 : Class Diagram.....	26
Fig 5.4 : Use Case Diagram.....	27
Fig 5.5 : Activity Diagram.....	28
Fig 5.6 : Main Screen.....	29
Fig 5.7 : Call.....	30
Fig 5.8 : Message.....	30
Fig 5.9 : Alarm.....	31
Fig 5.10 : Calendar Events.....	31
Fig 5.11 : Location.....	32
Fig 5.12 : Weather.....	33
Fig B.1 : Make Call Sequence Diagram.....	55
Fig B.2 : Answer Call Sequence Diagram.....	55
Fig B.3 : Send Message Sequence Diagram.....	56
Fig B.4 : Read Message Sequence Diagram.....	57
Fig B.5 : Set Alarm Sequence Diagram.....	58
Fig B.6 : Delete Alarm Sequence Diagram.....	59
Fig B.7 : Set Calendar Event Sequence Diagram.....	60
Fig B.8 : Delete Calendar Event Sequence Diagram.....	61
Fig B.9 : Find Location Sequence Diagram.....	62
Fig B.10 : Weather Update Sequence Diagram.....	63
Fig D.1 : VAAA.....	67
Fig D.2 : Main Screen.....	67
Fig D.3 : Call.....	68
Fig D.4 : Message.....	69

Fig D.5 : Alarm.....	69
Fig D.6 : Calendar Event.....	70
Fig D.7 : Location.....	70
Fig D.8 : Weather.....	71

List of Tables

Table 1.1: Project Vision.....	2
Table 1.2: Deliverables.....	3
Table 3.1: Operating Environment.....	10
Table 3.2: Schedule.....	11
Table 7.1: Accept Incoming Call Testing.....	36
Table 7.2: Reject Incoming Call Testing.....	37
Table 7.3: Make Call Testing.....	38
Table 7.4: Read Message Testing.....	38
Table 7.5: Send Message Testing.....	39
Table 7.6: Set Alarm Testing.....	40
Table 7.7: Delete Alarm Testing.....	40
Table 7.8: Set Calendar Event Testing.....	41
Table 7.9: Delete Calendar Event Testing.....	42
Table 7.10: Location Finder Testing.....	42
Table 7.11: Weather Updates Testing.....	43
Table A.1: Make Call Use Case.....	46
Table A.2: Answer Call Use Case.....	46
Table A.3: Send Message Use Case.....	47
Table A.4: Read Message Use Case.....	48
Table A.5: Set Alarm Use Case.....	49
Table A.6: Delete Alarm Use Case.....	49
Table A.7: Set Calendar Events Use Case.....	50
Table A.8: Delete Calendar Events Use Case.....	51
Table A.9: Location.....	53
Table A.10: Weather Updates.....	53
Table C.1: Component Template Description	64
Table C.2: Speech Recognition API.....	65
Table C.3: Text-to-Speech API.....	66

Chapter 1

1. Introduction

1.1. Overview

Android is rapidly growing technology that covers various aspects of human life. It makes human life far easier whether in the field of medical or science. Visually Impaired Assistance App (VIAA) will focus on disable people that would enable them to use android phone effectively. It will enable them to attend call using voice recognition APIs and will reject on command. Furthermore, sending receiving messages will also be manipulated using voice recognition APIs.

1.2. Problem Statement

Android technology is growing rapidly and it's been used by almost all age of people. Its interface is user friendly that makes it easier for educated as well as illiterate people. As it's the need of today's era, we had an interview with visually impaired people, they are keen to use android phone but the current interface provided by android phone doesn't fulfill the requirements. We are introducing an application that would enable visually impaired people to use android phone's functionalities effectively.

1.3. Project Scope

VIAA will focus on specific domain i.e. visually impaired people who're eager to use android phones but are unable because of the interface provided by android. VIAA will be embedded in the system and will guide visually impaired people to attend or reject call. It will also have the facility to send, receive messages, alarm settings through voice controller, calendar events and current location of the visually impaired person. Furthermore if there's any message and visually impaired person wants to read it, app will have the converter embedded in it that would read message. Similarly if visually impaired wants to send message, speech to text APIs will be embedded in

the system that would target the spoken voice to text and will send on the command of visually impaired person.

1.3.1. Project Vision

For	The app is intended for VIP (Visually Impaired People) who is interested in using android phone
What	To facilitate VIP in effective manner
Name	Visually Impaired Assistance App
Is	Android application
That	Facilitate the VIP to attend and reject calls. Send and receive messages, alarm setting, calendar event setting and current location of VIP. Controlling the system through voice that will enable the user to use in easy way
Unlike	Apps are available with less features that doesn't fulfill the requirements of VIP
Product	Easy interface manipulated by VIP makes it feasible and effective

Table 1.1: Project Vision

1.3.2. Project Objectives

The objectives of the project are:-

- To facilitate visually disable people to use android phone comfortably
- To develop android based application.
- To learn about visually impaired disability specific support features in mobile app development.

1.4. Deliverables

The deliverables of our project are:-

Deliverable Name	Deliverable Description
Software Requirement Specification (SRS)	Complete Description of WHAT system will do, who will use it. Detailed description of functional and non-functional requirements and system features
Software Design Document (SDD)	Complete description of How the system will do. Design models are included.
Code	Complete code with the APIs
Testing Document	Whole system is tested corresponding to the specifications. System is tested at all levels of Software Development Life Cycle (SDLC)
Android App	Complete working android app

Table 1.2: Deliverables

Chapter 2

2. Literature Review

2.1. Introduction

We have come across group of visually impaired people, by interviewing we came to know they are keen to use android phone as it is the need of today's era. We conducted a survey that gave us data on how they wish to use android, depiction of the interface and other necessary things that would facilitate visually impaired people to use android phone effectively. It has been observed that they wish the interface of android phone to be fully voice controlled that need commands and execute it according to the will of visually impaired. Furthermore they wish to use very few functionalities as we have described earlier (chapter 1) that would be manipulated and controlled through voice and the response of the system should be in the form of voice that would be familiar with it in a little spam of time.

2.2. Similar Applications

Android is a vast domain that has variety of different applications that facilitates end user. Applications for visually impaired people has also been developed but that doesn't control the entire system as a whole. By now, we have restricted our domain to some basic modules but we will modify and embed the maximum functionalities in our application that would benefit visually impaired people in better manner. Some of the existing products and their description is given below.

- **Just speak**

JustSpeak is an android accessibility service that enables voice control of Android device. Once it is enabled, one can activate on-screen controls, launch installed applications, and trigger other commonly used Android actions using spoken commands. Once Just Speak is installed and enabled on device, one can initiate voice commands by either performing an up-swipe from the Home button i.e. if the device

has soft keys or by performing multiple taps on the Home button. Just Speak can be configured to take multiple voice commands at once and perform them sequentially. This chaining works with both local and global commands, performing the preceding action after the previous action has been executed.

- **Talkback**

Talkback is an Accessibility Service that helps blind and vision-impaired users interact with their devices. Talkback adds spoken, audible, and vibration feedback to android device. It is a system application that was pre-installed on most devices and is updated when the accessibility service is improved. Talkback was installed on android device when it is bought it as part of Google's Android application suite, and it is routinely updated with improvements and new features through Google Play.

- **Kickback**

Kickback is android apps for the Android Accessibility Service to help blind and vision-impaired users use their devices more easily. These apps add spoken, audible, and vibration functionality services to your device. They are system applications that were pre-installed on most device and are updated when the accessibility service is improved. Kickback is one of the official accessibility services provided by Google. It provides haptic feedback by vibrating the device briefly when you perform an action on the phone.

It can be especially useful when using the touch screen. The feedback will inform you when you have found a valid control, like the slide-to-unlock handle. Additionally, it can be useful when navigating your device using a directional controller, as you know when your movement has resulted in an action. For example, when navigating through a menu you will feel a bump every time you hit a new menu item. When you stop receiving feedback, you know you have reached the end of the list. Talkback cannot be uninstalled.

- **Eyes free**

Eyes-Free aims to enable fluent eyes-free use of mobile devices running Android. Target uses range from eyes-busy environments like driving, to use by people who are unwilling or unable to look at the visual display. You can get a high-level overview of more potential use cases for Eyes-Free from this recent New York Times article. As described in the article, we are releasing components from project Eyes-Free as they become ready for end-user deployment.

Though the underlying source code has been available for some time from our repository on Google Code, we've now posted the first public release of the eyes-free shell on the Android Marketplace. Users of the eyes-free shell can conveniently launch talking applications. Along with this release, we've also made available a collection of applications to turn mobile devices running Android into eyes-free communication devices. Each of these applications have been written to be useful both to end users and as a means of helping the developer community come up to speed quickly as they develop eyes-free applications for Android

Chapter 3

3. Overall Description

3.1. Product Perspective

VIAA will be a product that will have sufficient features required by VIPs. The conception of its idea was originated with the aim of providing combined features in a single product. Most of the products are used in the market that has very limited features like just calling and receiving feature or just text writer and reader. VIAA will have embedded five modules that are essential need of a person using mobile phone. Easy interface will make it a product useful for the VIPs as well as normal users that wishes to control his android phone through voice.

Following diagram gives an overview of the VIAA.

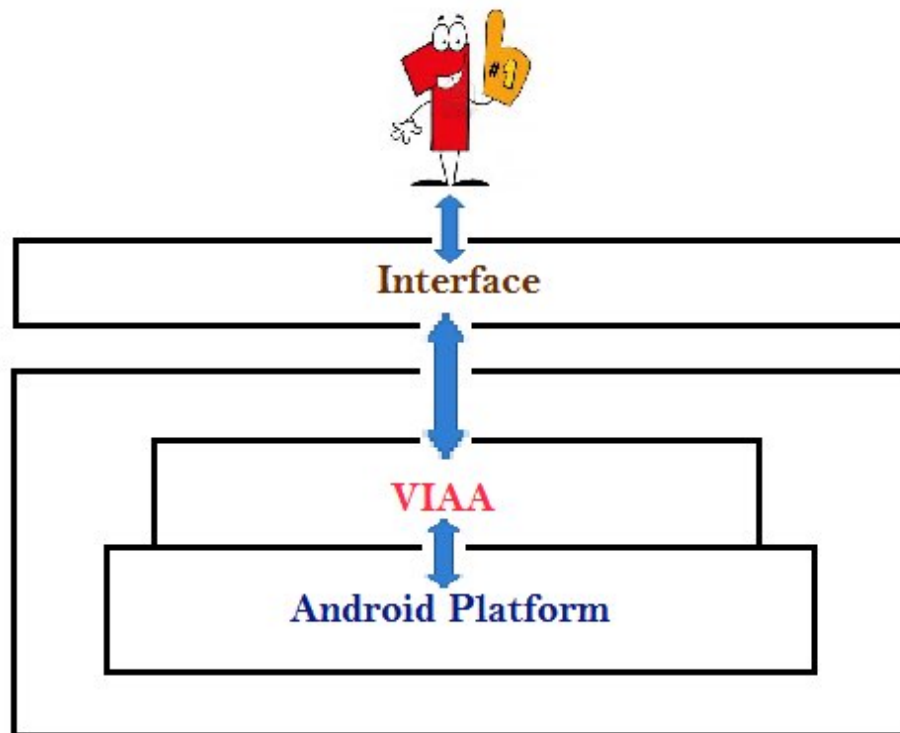


Fig 3.1 : Basic Layout

3.2. Project Modules and Functions

3.2.1. Project Modules

1. Response to the Calls:

The module enables VIPs to response to the call. Response will be based on voice recognition, APIs will be embedded to manipulate the response and act accordingly.

2. Response to the Messaging:

The module enables VIPs to response to the messages and deal with the message options whether it is to write message or read it from the system. Response will be based on voice recognition, APIs will be embedded to manipulate the response and act accordingly.

3. Alarm Settings:

The module enables VIPs to set up an alarm through voice command. APIs will be embedded to manipulate the response and act accordingly.

4. Calendar Events:

The module enables VIPs to set up calendar events through voice command. APIs will be embedded to manipulate the response and act accordingly.

5. Current Location:

This module will tell the VIPs their current location using GPS.

6. Weather Update:

This module will tell VIPs about the weather information.

3.2.2. Project Functions

- **Accepting Calls:** It helps VIPs to attend the call through voice command.
- **Rejecting Calls:** It helps VIPs to reject the call through voice command.
- **Read Messages:** It helps VIPs to extract message from the system using text to

speech APIs.

- **Write Messages:** It helps VIPs to enter message to the system using speech to text APIs.
- **Create Alarm:** It helps VIPs to setup new alarm through voice command.
- **Delete Alarm:** It helps VIPs to delete its previous alarms through voice command.
- **Location Viewer:** It extracts VIPs current location.
- **New Calendar Event:** Setting up new calendar events through voice command.
- **Delete Calendar Events:** It allows VIPs to delete its calendar events.
- **Wether Update:** It allows VIPs to get weather updates.

Following diagram demonstrates the product functions of VIAA.

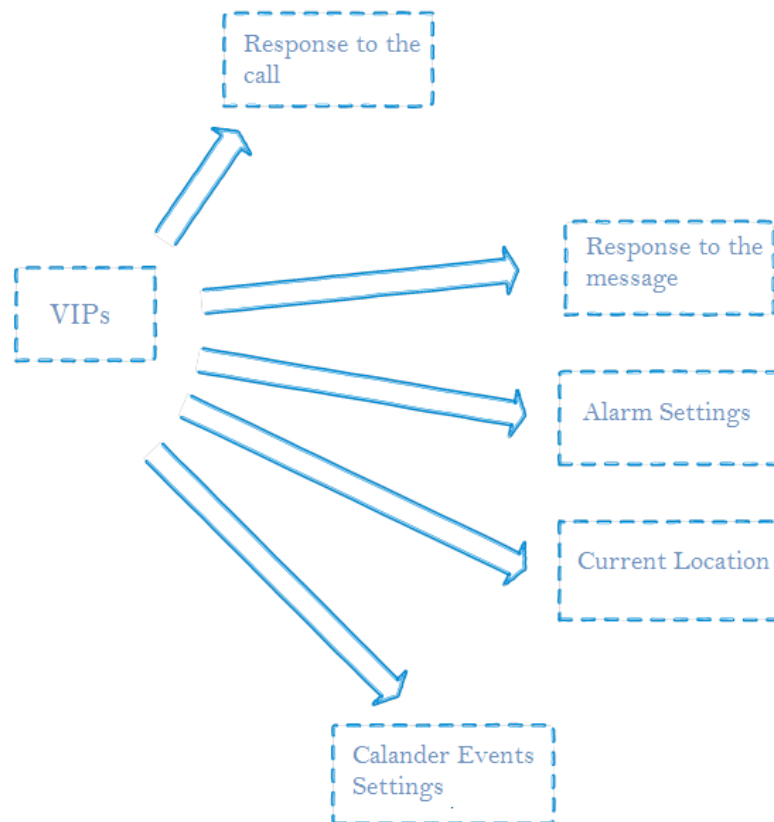


Fig 3.2 : Functions of VIAA

3.3. User Classes and Characteristics:

The target end-user of the system is:

- Anyone who wishes to control his android device using voice controller especially VIPs.
- VIPs are the main of focus of this product that wishes to use android devices but are unable to use it because of their visual disability.

- **End-users**

Most end users will be VIPs but the domain is not restricted to VIPs only, normally we don't have much people who doesn't have technical background of android platform. In order to use this system they only need to be familiarized with few very basic functions and they will be able to capture maximum functionalities of android. This will enable VIPs as well as less educated people to use android device efficiently and effectively.

3.4. Operating Environment

Operating System	Android based mobile operating system
Client	Android devices
Internet connectivity	GPS
Hardware Platform	Android enabled device (Mobile and tablets)

Table 3.1: Operating Environment

3.5. Design and Implementation Constraints:

The application will not be usable on Apple and other mobile phone sets which do not work on Android OS. The application is being developed only for Android based mobile phone/tablets and APIs have GPS activated for his current location.

Schedule:

Time is limited i.e. 11 months.

	2014						2015				
Activity	July	August	September	October	November	December	January	February	March	April	May
Requirement Elicitation	█										
Analysis & Design of the Project			█								
Implementation					█						
Evaluation							█				
Documentation	█										

Table 3.2: Schedule

Hardware:

Comprehensive testing on all types of Android Phone sets may not be feasible due to a vast variety of models available in the market. Testing will have to be done on more popular mobile phone sets / other phone sets with such specifications.

Cost:

Android mobile phones are generally expensive. For the moment, project may be started with only one phone.

3.6. User Documentation:

The documentation provided to the final user will be complete and exhaustive in each functionality of the software that will guide the user to use get familiar with the interface as well as functionalities. Functionalities will have all detailed description of the modules, its working, its constraints and limitation as well as its usage methodologies. Furthermore it will provide sufficient information about the interface to make it user friendly and effective.

3.7. Assumptions and Dependencies .

Assumptions

AS-1: Assume that the project started on September 21, 2013.

AS-2: Assume that the faculty members of the Department of CSE are the client party for this project.

AS-3: Assume that the individuals in MCS form the user population, which for the moment, does not exceed 5 users.

Dependencies

D-1: There will be a permanent dependency on Google Maps and reliability of system will depend on continuous / uninterrupted data from Google-Maps to get the current location of the APIs.

D-2: Application will be running all the time on the background and will provide an interface when there's any incoming calls or messages .

D-3: System is dependent on APIs already developed that will be embedded in the app to provide its services.

Chapter 4

4. Software Requirement Specifications

4.1. System Features

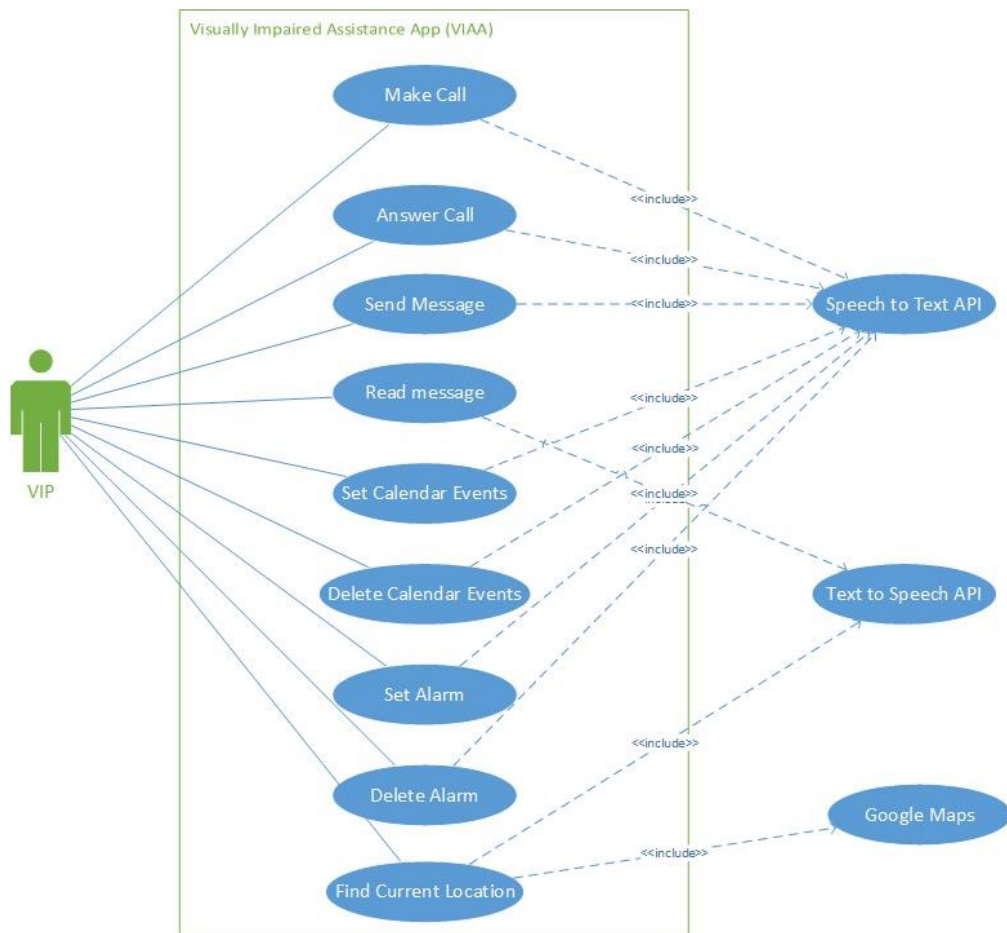


Fig 4.1 : Use Case Diagram

The system features are enlisted below:

- Make Call
- Answer Call
- Send Message
- Read Message
- Set Alarm
- Delete Alarm
- Set Calendar Events

- Delete Calendar Events
- Find Current Location
- Weather Update

4.1.1. Make Call:

4.1.1.1. Description and Priority

This use case describes that when VIP will have a call and he wants to attend that call, how he will be able to respond the call through voice commands.

Priority = High

4.1.1.2. Stimulus and Response Sequence

Stimulus: The VIP gets a call on cell phone.

Response: The system will generate an interface that will have options to receive and reject call.

Stimulus: VIP speaks a keyword to accept the call.

Response: The system will verify the keyword from its data set and then after verification it will accept call.

4.1.1.3. Functional Requirements

1. System shall get an interface for making a call
2. System shall have two options
 - a. Make call using voice command
 - b. Make call using screen flip

4.1.2. Answer Call:

4.1.2.1. Description and Priority:

This use case describes that when VIP will have a call and he wants to reject that call, how he will be able to respond the call through voice commands.

Priority = High

4.1.2.2.Stimulus and Response Sequence:

Stimulus: The VIP gets a call on cell phone.

Response: The system will generate an interface that will have options to receive and reject call.

Stimulus: VIP speaks a keyword to reject the call.

Response: The system will verify the keyword from its data set and then after verification it will reject call.

4.1.2.3.FunctionalRequirements(FR):

1. The system shall end call that will have two options
 - a. Disconnect using voice command (Keyword)
 - b. Disconnect using screen flip
2. System will accept / reject call with the following options
 - a. Voice command (Keyword)
 - b. Manual rejection method

4.1.3.Send Messages:

4.1.3.1.Description and Priority:

This use case describes that when VIP wants to send message to anyone then how he will be able to send messages through voice commands.

Priority = Medium

4.1.3.2.Stimulus and Response Sequence:

Stimulus: VIP speaks a keyword to open write message option.

Response: The system will verify the keyword from its data set and then after verification it will open a screen to write a message.

Stimulus: VIP speaks his message.

Response: The system will convert the spoken message into text through Speech Recognition API.

Stimulus: VIP speaks a keyword to send the message.

Response: The system will verify the keyword from its data set and then after verification it will send the message.

4.1.3.3.FunctionalRequirements(FR):

1. The system shall provide message interface that makes VIP enable to interact with interface.
2. The system shall type voice message for VIPs through APIs.
3. The system should fetch the contact requested by VIP.
4. The system should send message on VIP's command.

4.1.4.Read Message:

4.1.4.1.Description and Priority:

This use case describes that when VIP wants to read a message then how he will be able to read it through voice commands.

Priority = Medium

4.1.4.2.Stimulus and Response Sequence:

Stimulus: VIP speaks a keyword to read the message.

Response: The system will verify the keyword from its data set and then after verification it will open that message and read it with the help of Text to Speech API.

4.1.4.3.FunctionalRequirements(FR):

1. The system shall provide message interface that makes VIP enable to interact with interface.
2. The system shall read message for VIPs through API (Text to Voice).
3. The system should narrate the sender name for VIP.
4. The system should read message on VIP's command.

4.1.5.Set Alarm:

4.1.5.1.Description and Priority:

This use case describes that when VIP wants to set an alarm then how he will be able to do it.

Priority = Low

4.1.5.2.Stimulus and Response Sequence:

Stimulus: VIP speaks a keyword to open alarm option.

Response: The system will verify the keyword from its data set and then after verification it will open alarm option.

Stimulus: VIP speaks to set the alarm time.

Response: The system will set the alarm time.

Stimulus: VIP speaks a keyword to activate alarm.

Response: The system will verify the keyword from its data set and then after verification it will activate the alarm.

4.1.5.3.FunctionalRequirements(FR):

1. The system shall come up with an interface on VIP's alarm command.
2. The system should add an alarm on VIP's voice command.
3. The system shall alter an alarm on VIP's voice command.

4.1.6.Delete Alarm:

4.1.6.1.Description and Priority:

This use case describes that when VIP wants to delete/deactivate an alarm then how he will be able to do it.

Priority = Low

4.1.6.2.Stimulus and Response Sequence:

Stimulus: VIP speaks a keyword to open alarm option.

Response: The system will verify the keyword from its data set and then after verification it will open alarm option.

Stimulus: VIP speaks a keyword to delete/deactivate alarm.

Response: The system will verify the keyword from its data set and then after verification it will deactivate the specific alarm.

4.1.6.3.FunctionalRequirements(FR):

1. The system shall come up with an interface on VIP's alarm command
2. The system shall delete alarm on VIP's voice command

4.1.7.Set Calendar:

4.1.7.1.Description and Priority:

This use case describes that when VIP wants to set a calendar event then how he will be able to do it.

Priority = Low

4.1.7.2.Stimulus and Response Sequence:

Stimulus: VIP speaks a keyword to open calendar events option.

Response: The system will verify the keyword from its data set and then after verification it will open calendar events option.

Stimulus: VIP speaks to set the event date and time.

Response: The system will set the event date and time.

Stimulus: VIP speaks a keyword to activate event.

Response: The system will verify the keyword from its data set and then after verification it will activate the event.

4.1.7.3.FunctionalRequirements(FR):

1. The system shall come up with an interface on VIP's calendar event command.
2. The system should add calendar event on VIP's voice command.
3. The system shall alter calendar event on VIP's voice command.

4.1.8.Delete Calendar Events:

4.1.8.1.Description and Priority:

This use case describes that when VIP wants to delete a calendar event then how he will be able to do it.

Priority = Low

4.1.8.2.Stimulus and Response Sequence:

Stimulus: VIP speaks a keyword to open calendar events option.

Response: The system will verify the keyword from its data set and then after verification it will open calendar events option.

Stimulus: VIP speaks a keyword to delete/deactivate event.

Response: The system will verify the keyword from its data set and then after verification it will deactivate the specific event.

4.1.8.3.FunctionalRequirements(FR):

1. The system shall come up with an interface on VIP's calendar event command.
2. The system shall delete calendar event on VIP's voice command

4.1.9.Find Current Location:

4.1.9.1.Description and Priority:

This use case describes when VIP wants to know where he is at that time then how he will be able to know about his location.

Priority = Low

4.1.9.2.Stimulus and Response Sequence:

Stimulus: VIP speaks a keyword to know about his location.

Response: The system will verify the keyword from its data set and then after verification it will find the current location through GPS and speak it.

4.1.9.3.Functional Requirements (FR):

1. The system shall always be connected to the internet using GPS
2. The system shall fetch the location whenever VIP changes its coordinates.
3. The system shall update the location of VIP as soon as he changes its location.

4.1.10.Weather Updates:

4.1.10.1.Description and Priority:

This use case describes that when VIP wants to get weather updates, how he will be able to get weather updates through voice commands.

Priority = High

4.1.10.2.Stimulus and Response Sequence:

Stimulus: VIP speaks a keyword to get weather updates.

Response: The system will verify the keyword from its data set and then after verification it will fetch the weather information and speak it.

4.1.10.3.Functional Requirements (FR):

1. System shall get an interface for getting weather updates.

2. System shall have two options
 - a. Get weather updates using voice command
 - b. Get weather updates using screen flip

4.2. Non Functional Requirements

4.2.1. Performance Requirements

Since the system is a mobile application, there shouldn't be much delay in processing and responding to the user's requests. The requests should be responded at most 2 seconds.

4.2.2. Security Requirements

1. The system should not be accessed by any unauthorized user.
2. The data set should only be allowed to be manipulated by valid users.

4.2.3. Software Quality Attributes

4.2.3.1. Reliability:

This system should be reliable and respond accurately according to the commands without errors. Since it is an android based application, the system is highly portable and can be used on large number of android devices. The system is easily extendible since we can add functionalities.

4.2.3.2. Usage Easiness:

- 90% of a test panel of non-experienced users should be able to use the system within 5 minutes.
- After having used the application once, 95% of users are able to locate the experienced functionality within 1 minute.

4.2.3.3. Compatibility:

Android applications must be able to communicate with the rest of the system, and to handle all of the functionalities.

4.2.3.4.Overall Satisfaction:

After conducting a survey, 80% of the users should keep using the application after a two week exploitation period.

4.2.3.5.Trust:

After having used all the features three times, 90% of the users should feel confident about the reliability, robustness and be convinced that the product does what it is expected to do.

4.2.3.6.Understandability and Politeness:

Our system uses only those symbols and words that are immediately understandable by users. All the technical details should be hidden from the user.

4.2.3.7.Learning:

A VIP should be able to attend and reject calls, send and receive messages, set and delete alarms, set and delete calendar events and find his current location within the first 15 minutes of usage without referring to the user manual.

4.2.3.8.Maintenance:

The first version of the system is open source and its maintenance is not provided. Future commercial versions of the software may offer maintenance.

4.2.3.9.Supportability:

The required support level needed by the system should be low, even up to the point it could be handled by the service provider's help desk.

4.2.3.10Adaptability:

The application should be portable on Android version higher than 2.1.

Chapter 5

5. System Design Specifications

5.1. Overview of Modules/Components

Our system have 6 main modules, which are as below

- **Calls**

The VIPs with android phones having VIAA installed will be able to make and respond calls using voice commands. Their voice commands will be converted to the system commands and in response, the system will act according to the given voice command.

- **Messages**

This module will help VIPs to send and read messages using voice commands. Their voice commands will be converted to the system commands and in response, the system will act according to the given voice command whereas for reading the messages the system will use the text to speech API.

- **Alarms**

VIPs will be able to set and delete alarms using the voice commands. The voice commands will be converted to the system commands.

- **Calendar Events**

This module will help VIPs to set and delete calendar events using the voice commands. The VIAA will recognize the voice commands and convert them to the system commands.

- **Finding Current Location**

The app will inform VIP about his current location, by taking the current location from GPS and then speaking the location so that VIP can hear that where he is at that

moment.

- **Weather Info**

The app will inform VIP about his weather of his/her location, by taking the current location from GPS and extracting the weather from the internet and then speaking the weather info so that VIP can hear that what is the situation of weather at that moment.

5.2. Structure and Relationship

5.2.1. Basic Layout Diagram:

VIAA will be installed on Android platform and it will provide a user friendly interface to VIPs. The interface will help the user to use his/her android phone using the voice commands and taking the text output in speech form.

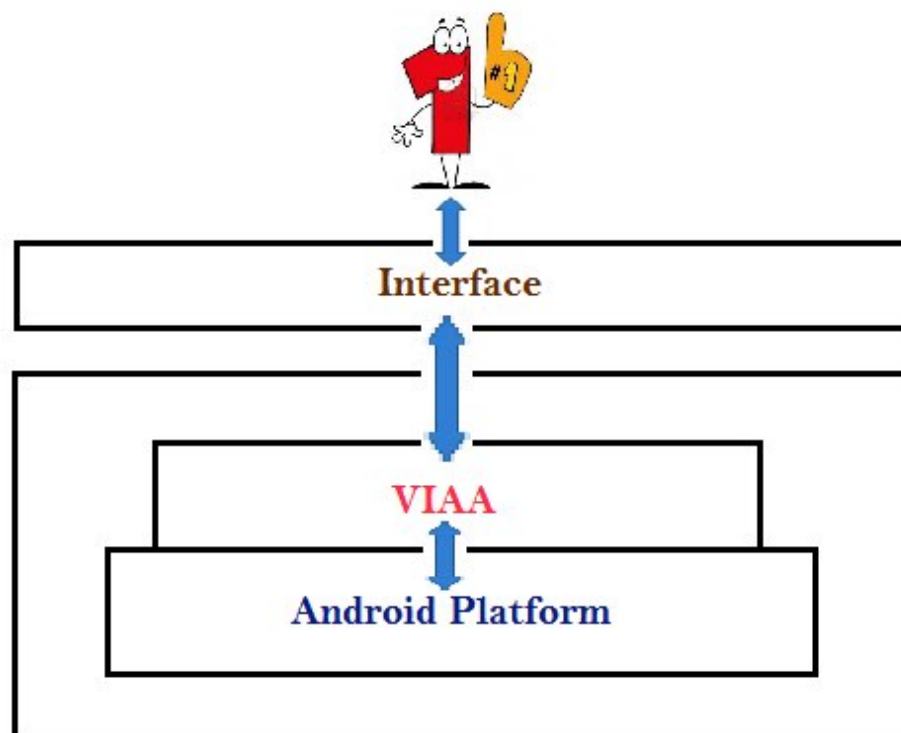


Fig 5.1 : Basic Layout Diagram

5.2.2. Architecture Diagram:

The architecture of VIAA is LAYERED. It shows that it will help the user to communicate with the system through voice commands. In the data store, the keywords are stored on which the app take some actions or give the system some commands corresponding to those keywords. GPS is used by VIAA to know about the current location of the user.

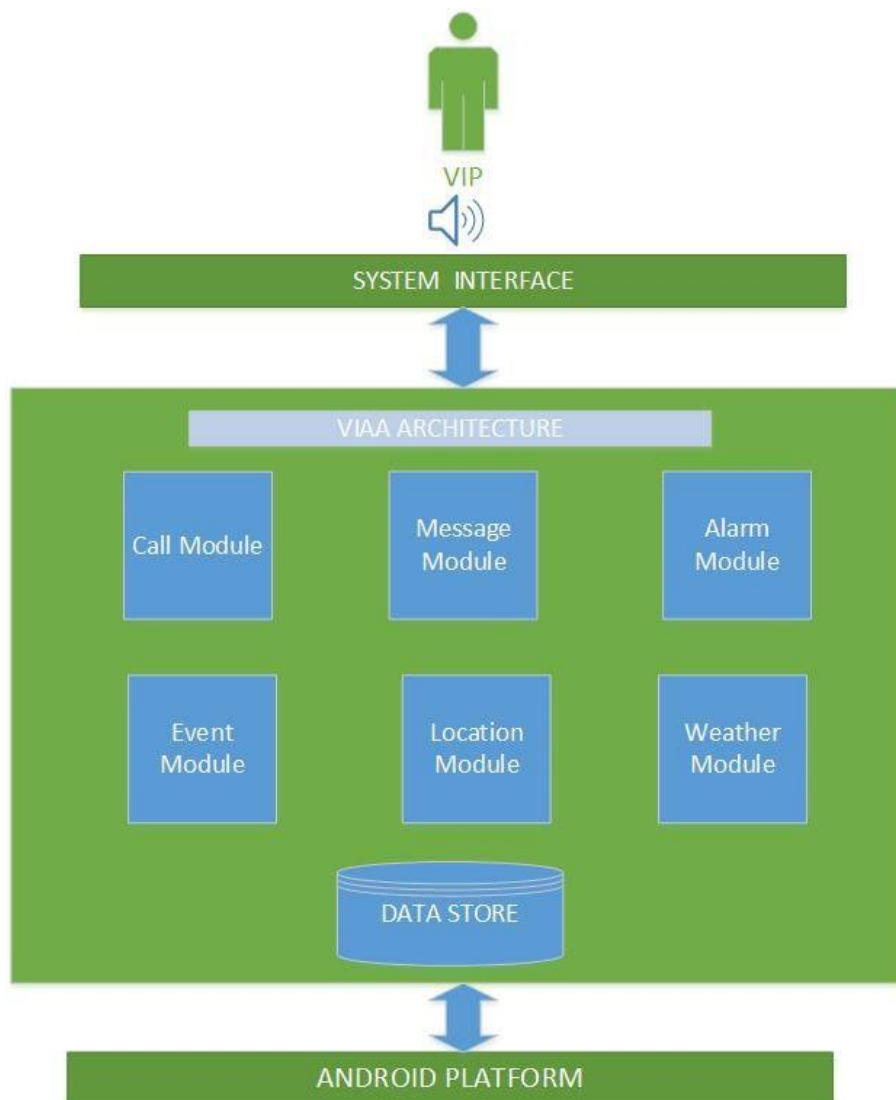


Fig 5.2 : Architecture Diagram

5.2.3. Class Diagram:

Class diagram below shows the overall functionalities of VIAA. All the classes use speech-to-text and text-to-speech API to perform their functions. It shows how classes are interrelated in between the system, how these modules interact with each other and how the system interacts with external modules like APIs and GPS. It shows the functions in each class as well as the attributes related to those classes. Diagram shows the overall structure of VIAA

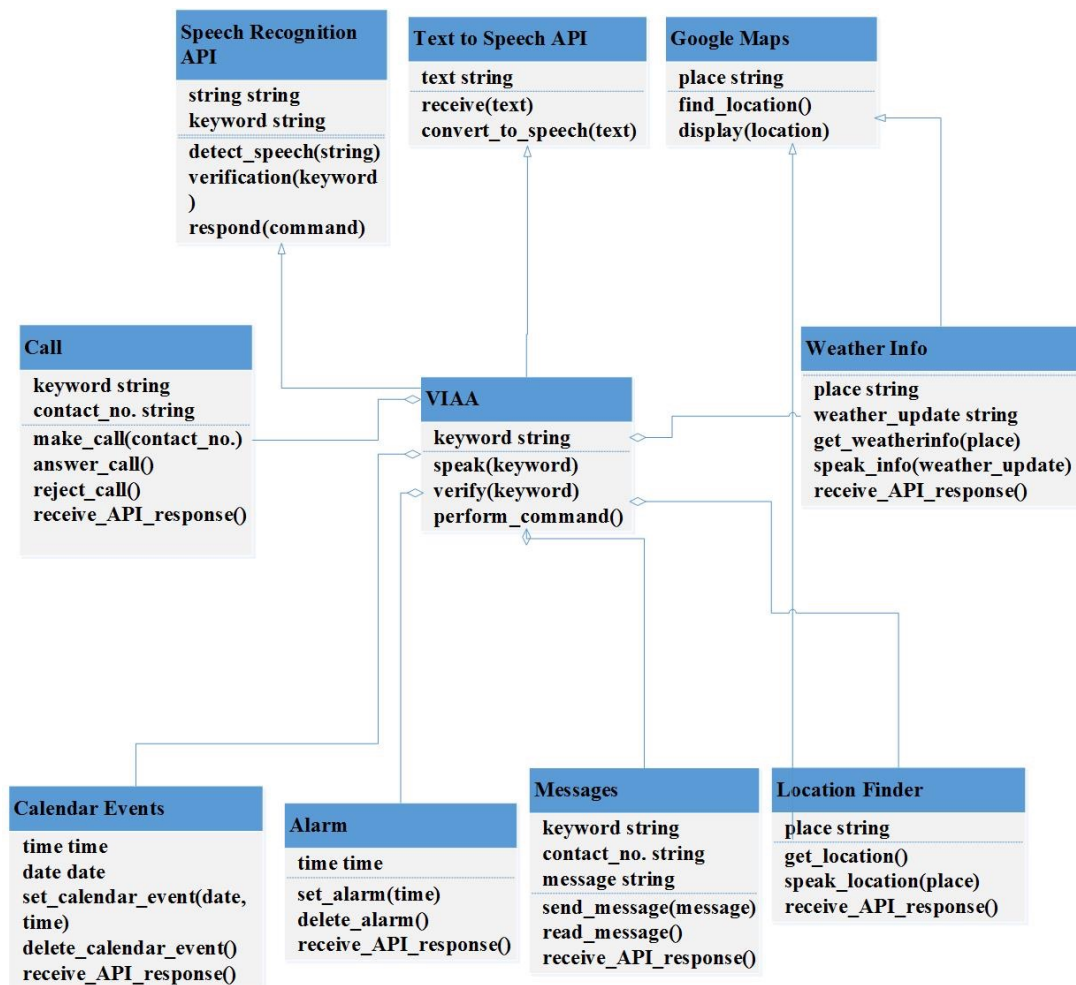


Fig 5.3 : Class Diagram

5.2.4. Use Case Realization:

Use case diagram below shows the overall system, its basic modules and external entities involve that needs to interact with the system to perform its functionalities. All modules use speech-to-text and text-to-speech API to perform their functions. It shows how classes are interrelated in between the system, how these modules interact with each other and how the system interacts with external modules like APIs and GPS. More detail is in Appendix A.

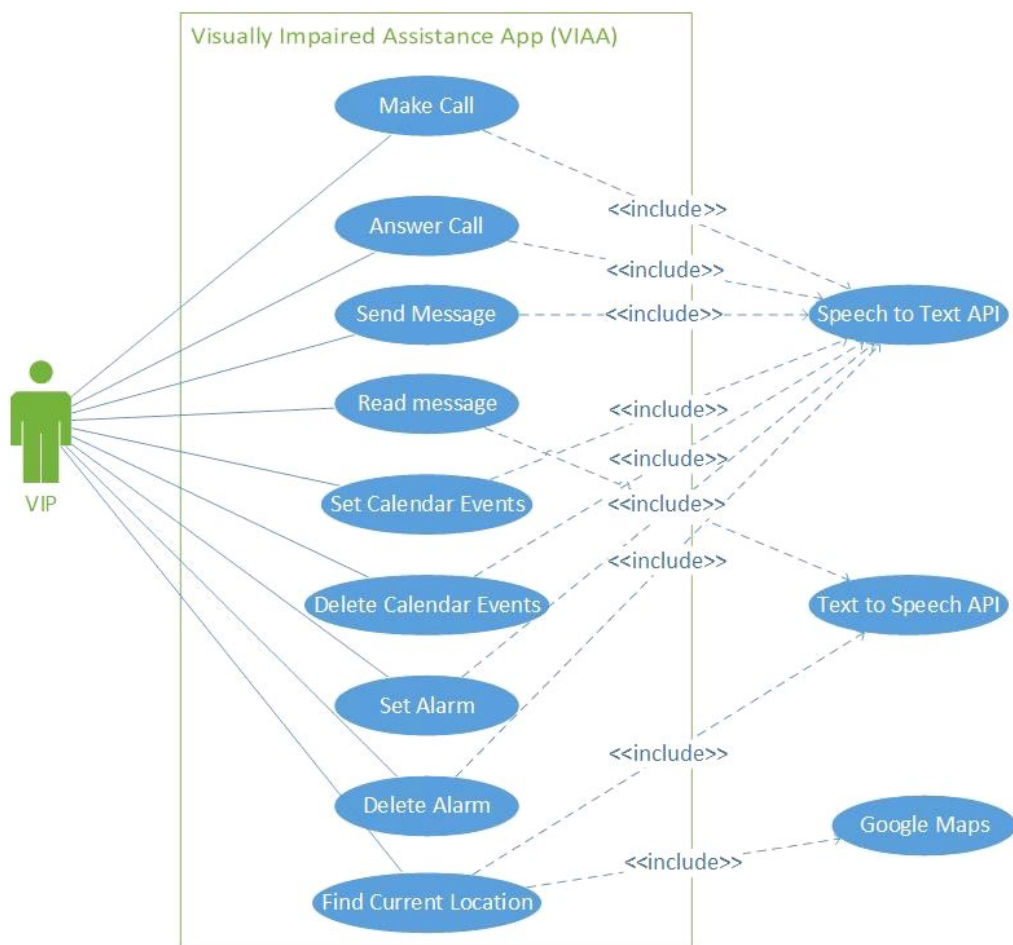


Fig 5.4 : Use Case Daigram

5.2.5. Activity Diagram:

Activity diagram below shows the overall function, how the system will react to the visually impaired voice command and how the flow of commands will be handled in by the system. All stimulus and response are description is shown in the diagram. Initially when the VIP will command to the system, it will be manipulated by API and will be converted to text. Further it will be checked across all possible modules and will look for the matching modules. As soon as voice and text matches, an interface will be displayed based on VIPs command.

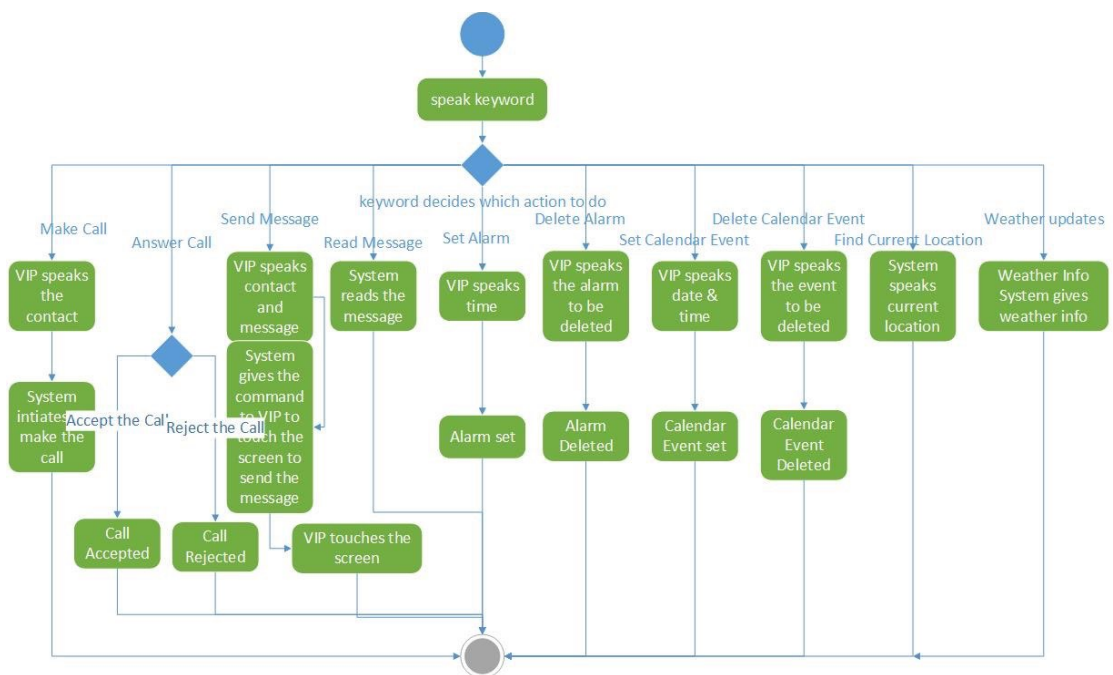


Fig 5.5 : Activity Diagram

5.2.6. Sequence Diagrams:

Sequence diagram below shows the flow of commands and response to the commands between VIP, system and android system. Sequence diagrams for each modules is shown below that shows the action of VIP to the system. Further how these commands are handled by the system. In case command doesn't match, system will ask again the VIP to command again that is shown in alt scenario. When the command is recognized by the system, it will interact with the device and will respond accordingly. More detail is in Appendix B.

5.2.7.Interface specification

Please note that the interfaces provided are just for demonstration purposes. Actual interfaces may be different.

5.2.7.1.Main Screen:

Main screen is very simple. VIP have to just touch the screen and speak the command.

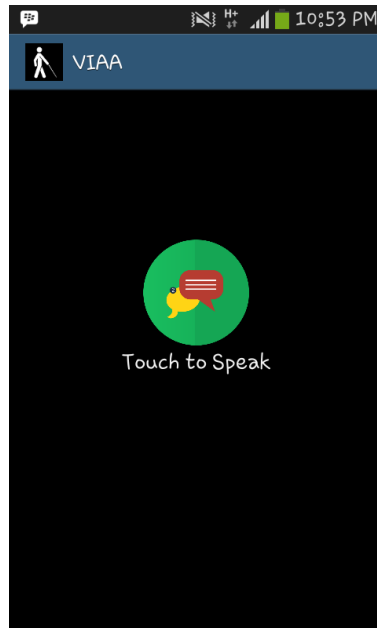


Fig 5.6 : Main Screen

5.2.7.2.Call:

This screen will appear to the VIP for incoming calls. The screen is same but now the VIP will not have to touch the screen, he just needs to speak a command and his call will be accepted or rejected depending on what he speak. Receiver will have both the options whether attend/reject manually or through voice command. Likewise, if VIP wants to call a number, he will narrate the name and system will search for contact in the list and will call that specific number on command.

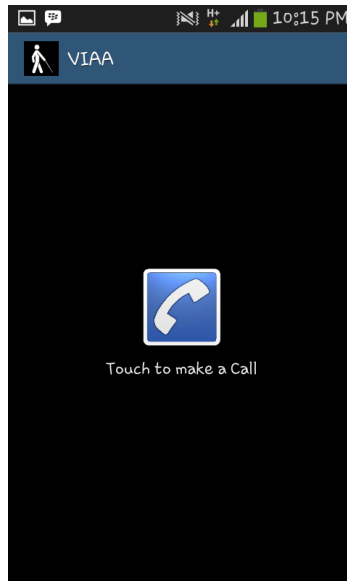


Fig 5.7 : Call

5.2.7.3.Messages:

Such screen as shown in figure will appear to the VIP. It is similar as the traditional screen but an embedded feature is, VIP will narrate message and speech to text APIs will convert it to text. Likewise, if VIP want to read message, text to speech API will provide its services and will narrate it to the VIP.



Fig 5.8 : Message

5.2.7.4. Alarm Settings:

This module will be designed to facilitate VIPs to set up an alarm if there's any important task and want a reminder. Alarm will have default ringtone that will ring on desired set time. Figure shows description of the setting.

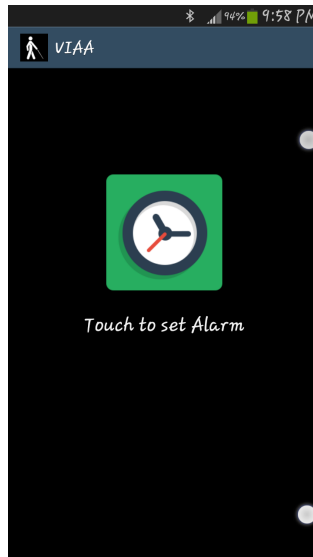


Fig 5.9 : Alarm

5.2.7.5. Calendar events:

This module will be designed to facilitate VIPs to set up an event if there's any important task and want a reminder on specific date. Reminder will setup and default ringtone that will ring on desired set time. Figure shows description of the setting.

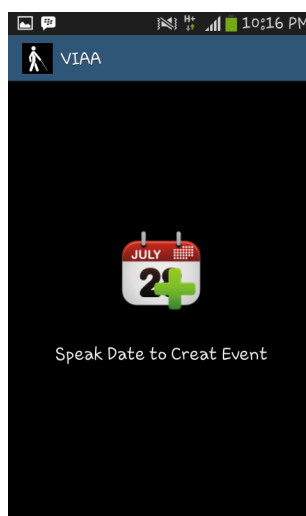


Fig 5.10 : Calendar Events

5.2.7.6.Location:

This module will be designed to facilitate VIPs to view its current location. It will narrate the current location of the VIPs. It will give information about the current city VIP is in and text to speech API will provide its services to narrate it for VIPs.

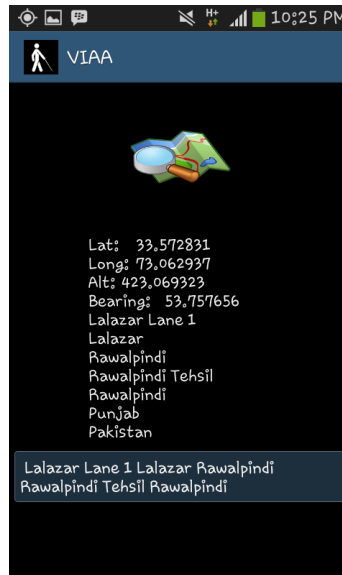


Fig 5.11 : Location

5.2.7.7.Weather:

Weather module will focus on the weather current state and will narrate it to the VIPs. VIP will ask for weather information and system will tell him the current status of weather fetched through GPS whether the day is sunny, cloudy or rainy. Furthermore, system will tell the temperature of the day, humidity, precipitation and wind.

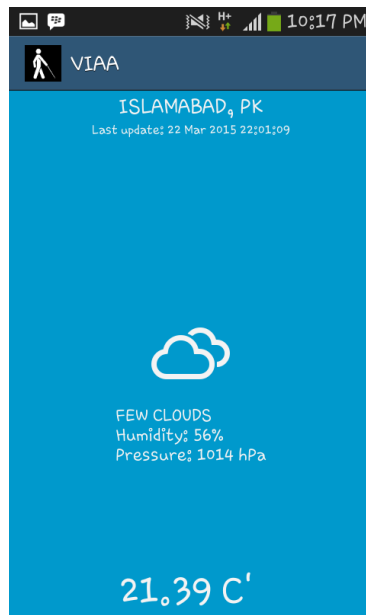


Fig 2.12 : Weather

Chapter 6

6. System Implementation

6.1. Technology Used

6.1.1. Programming Language Used

Java is used as the programming language for the development of the application. SQLite is the database which is used to manage the alarm and calendar events.

6.1.2. Development Tools

Eclipse is the tool which is used to develop the application. Android Studio is the tool which is used for Android application development.

6.1.3. Operating System

This application is designed for the android platform. So, the application will run on all android phones and tablets.

6.2. Complete System Implementation

VIAA consists of six main modules. All the modules involve either speech recognition or text-to-speech or both APIs of android to perform their functionalities. For the speech recognition, google dictionary is used to recognize the words spoken by the user.

6.2.1. Calls

Call module uses the speech recognition API. User will speak the command and after verifying the command VIAA will accept, reject the incoming call or make a call.

6.2.2. Messages

Message module is implemented by using speech recognition and text-to-speech API.

In case of Read Message, whenever a message is received VIAA will be activated and text-to-speech API will convert the message from text to speech and when a user will touch screen, system will speak that message. While the Send Message module needs speech recognition API to function. User will speak the number and message, which the speech recognition API will convert to text and on the touch by the user system will send the message.

6.2.3.Alarms

Alarm module is implemented using speech recognition API and SQLite Database. User will speak the command which the speech recognition API will convert to the system command. SQLITE is the database used to store the alarms.

6.2.4.Events

Event module is also implemented using speech recognition API and SQLite Database. User will speak the command which the speech recognition API will convert to the system command. SQLITE is the database used to store the events.

6.2.5.Location

Location module uses speech recognition and text-to-speech APIs and GPS to find the location. Speech recognition API will identify the command and after verifying it will extract the location from GPS. The location will be displayed on the screen and when the user will touch the screen, it will initiate text-to-speech API which will speak the extracted data.

6.2.6.Weather

Weather module is incorporated to help the user to find out the weather conditions of any city. This module will fetch the data from the internet and show it on the screen and when the user will touch the screen, it will initiate the text-to-speech API which will speak the extracted information.

Chapter 7

7. System Testing

7.1. Overview

Testing of the software projects involve different levels of testing to make sure that the software which is being developed is error and fault free. Visually Impaired Assistance App has different modules which were developed separately depending up on the functionalities. Therefore testing of all the modules has to be done and testing while integrating all the modules. The different levels at which testing was done are discussed here.

7.2. Approach

No tool is used to test the application. Code is tested on Eclipse and the features are tested on the running application.

7.3. Test Cases

7.3.1. Accept Incoming Call Testing

Test Case Name	Accept incoming call
Test Case ID	1
Description	This feature will give first priority to incoming call if there is any other application running, this feature will resume all activities and will pop up the incoming call.
Testing Technique used	Black Box Testing
Preconditions	System is running and google dictionary has already been installed on the phone.

Input Values	System is running and google dictionary has already been installed on the phone.
Valid Inputs	Speak keyword “Accept”
Expected output	Call should be accepted when VIP speaks valid command
Actual output	App accepts call when VIP speaks valid command
Status	PASS

Table 7.1: Accept Incoming Call Testing

7.3.2.Reject Incoming Call Testing

Test Case Name	Reject incoming call
Test Case ID	2
Description	This feature gives first priority to incoming call if there is any other application running, this feature will resume all activities and will pop up the incoming call.
Testing Technique used	Black Box Testing
Preconditions	System is running and google dictionary has already been installed on the phone.
Input Values	Keywords
Valid Inputs	Speak keyword “Reject”
Expected output	Call should be rejected when VIP speaks valid command
Actual output	App rejects call when VIP speaks valid command
Status	PASS

Table 7.2: Reject Incoming Call Testing

7.3.3. Make Call Testing

Test Case Name	Make call
Test Case ID	3
Description	This feature enables the user to make call on specific spoken number
Testing Technique used	Black Box Testing
Preconditions	The System is running and google dictionary has already been installed on the phone.
Input Values	<ul style="list-style-type: none">• Keyword• Phone Number
Valid Inputs	Speak keyword “Call” and then phone number
Expected output	App should make a call to that specific number
Actual output	App makes a call to that specific number
Status	PASS

Table 7.3: Make Call Testing

7.3.4. Read Message Testing

Test Case Name	Read message
Test Case ID	4
Description	This feature pops up incoming message and speaks it for VIP when touches the screen
Testing Technique used	Black Box Testing
Preconditions	System is running and google dictionary has already been installed on the phone.
Input Values	Touch

Valid Inputs	Touch the screen
Expected output	App should read message for VIP
Actual output	App reads message for VIP
Status	PASS

Table 7.4: Read Message Testing

7.3.5. Send Message Testing

Test Case Name	Send message
Test Case ID	5
Description	This feature sends message to any valid number that is provided by VIP
Testing Technique used	Black Box Testing
Preconditions	System is running and google dictionary has already been installed on the phone.
Input Values	<ul style="list-style-type: none"> • Keyword • Phone Number • Message
Valid Inputs	Speak keyword “Message” and then phone number and message
Expected output	Message should be sent when VIP speaks valid phone number and message
Actual output	App sends message to that specific valid phone number
Status	PASS

Table 7.5: Send Message Testing

7.3.6.Set Alarm Testing

Test Case Name	Set Alarm
Test Case ID	6
Description	This feature creates an alarm on a specific time provided by VIP
Testing Technique used	Black Box Testing
Preconditions	System is running and google dictionary has already been installed on the phone and SQLite database is connected.
Input Values	<ul style="list-style-type: none">• Keyword• Time
Valid Inputs	Speak keyword “Set Alarm” and then time
Expected output	Alarm should be set for the desired time and stored in the database
Actual output	Alarm is set for the desired time and stored in the database.
Status	PASS

Table 7.6: Set Alarm Testing

7.3.7.Delete Alarm Testing

Test Case Name	Delete Alarm
Test Case ID	7
Description	This feature creates an alarm on a specific time provided by VIP
Testing Technique used	Black Box Testing

Preconditions	System is running and google dictionary has already been installed on the phone and SQLite database is connected.
Input Values	<ul style="list-style-type: none"> • Keyword
Valid Inputs	Speak keyword “Delete Alarm”
Expected output	Alarm should be deleted from the database
Actual output	Alarm is deleted from the database
Status	PASS

Table 7.7: Delete Alarm Testing

7.3.8.Set Calendar Event Testing

Test Case Name	Set Calendar Event
Test Case ID	8
Description	VIP will be able to set calendar events.
Testing Technique Used	Black Box Testing
Preconditions	Application is running and SQLite database is connected
Input Values	<ul style="list-style-type: none"> • Keyword • Date • Event Title
Valid Inputs	Speak keyword “Events” and event title and date.
Expected Output	Calendar Event should be set for the desired date.
Actual Output	Calendar Event is set and stored in the database.
Ststus	PASS

Table 7.8: Set Calendar Event Testing

7.3.9.Delete Calendar Event Testing

Test Case Name	Delete Calendar Event
Test Case ID	9
Description	VIP will be able to delete calendar events.
Testing Technique Used	Black Box Testing
Preconditions	Application is running and SQLite database is connected
Input Values	<ul style="list-style-type: none">• Keyword• Event Title
Valid Inputs	Speak keyword / “Delete Event” and event title and date.
Expected Output	Calendar Event should be deleted.
Actual Output	Calendar Event is deleted from the database.
Ststus	PASS

Table 7.9: Delete Calendar Event Testing

7.3.10.Location Finder

Test Case Name	Location Finder
Test Case ID	10
Description	VIP will be able to find his / her current location
Testing Technique Used	Black Box Testing
Preconditions	Application is running and GPS is turned on.
Input Values	Keyword

Valid Inputs	Speak keyword “Location”
Expected Output	Location should be extracted and on touch system should speak the extracted data.
Actual Output	Location is extracted and displayed on the screen and on touch system speaks the extracted data.
Status	PASS

Table 7.10: Location Finder Testing

7.3.11. Weather Update Testing

Test Case Name	Weather Updates
Test Case ID	11
Description	VIP will be able to know the weather updates of different city
Testing Technique Used	Black Box Testing
Preconditions	Application is running and internet connection is required
Input Values	Keyword
Valid Inputs	Speak keyword “Weather:
Expected Output	Weather information should be extracted and on touch system should speak the extracted data.
Actual Output	Location is extracted and displayed on the screen and on touch system speaks the extracted data.
Status	PASS

Table 7.11: Weather Updates Testing

Chapter 8

8. Conclusion and Future Work

8.1. Conclusion

VIAA is an android application designed for visually impaired people. It has ten main modules that consists of incoming call module, outgoing call module, send message, receive message, create an alarm, delete an alarm, create an event, delete an event, find location of VIPs and weather of the current city as well as the city spoken by VIPs.

Application is implemented in Java using Eclipse IDE with the inclusion of Application Programing Interfaces (Text to Speech & Speech to Text) are included for speech to text and vice versa conversion. SQLite database has been integrated for the complete performance of application that has all information of alarms and events modules.

8.2. Future Work

As mentioned in the document, we have developed ten main modules of VIAA that are somehow necessary functions for visually impaired person. Moreover it covers maximum aspects of application that includes all functionalities a visually impaired person needed to have. Furthermore it can be extended to control entire android mobile through voice commands. Like fetching a media player from menu and play song for visually impaired, turning on camera to capture an object, switching its controls to front and back camera. It can also have whatsapp, viber, radio control and other such functionalities that can be usefull for visually impaired people to facilitate them with better visual interface and multiple functionalities to make them use application in better way.

It can be extended for blinds as well to provide them with maximum functionalities and full control over android device that would require image processing along with some other techniques to kill the background disturbance and other interruption that disturbs input to the application.

Appendices

Appendix A: Use Cases Description:

USE CASE NAME	Make call
ACTOR	VIP
NORMAL COURSE	<ol style="list-style-type: none"> 1. The process is initiated when a VIP wants to make a call. 2. VIP speaks a keyword and contact. 3. Speech recognition API verifies that keyword from its data set. 4. System fetches the contact after verification of the making call command from VIP. 5. System will make call.
ALTERNATE COURSE	<ol style="list-style-type: none"> 3. a) Verification Denied <ol style="list-style-type: none"> 1. Speech Recognition API doesn't verify the keyword. 2. VIP needs to speak again the keyword.
PRE CONDITION	<ul style="list-style-type: none"> • Keywords for taking some particular action are already defined in the data set of API.
POST CONDITION	<ul style="list-style-type: none"> • VIP will be able to make call through voice commands and without touching the screen.

Table A.1: Make Call Use Case

USE CASE NAME	Answer the call
ACTOR	VIP

NORMAL COURSE	<ol style="list-style-type: none"> 1. The process is initiated when a VIP have any call to answer. 2. VIP speaks a keyword. 3. Speech recognition API verifies that keyword from its data set. 4. System accepts or rejects the call after verification of the rejecting command from VIP.
ALTERNATE COURSE	<ol style="list-style-type: none"> 3. a) Verification Denied <ol style="list-style-type: none"> 1. Speech Recognition API doesn't verify the keyword. 2. VIP needs to speak again the keyword.
PRE CONDITION	<ul style="list-style-type: none"> • Keywords for taking some particular action are already defined in the data set of API.
POST CONDITION	<ul style="list-style-type: none"> • VIP will be able to attend or reject the call through voice commands and without touching the screen.

Table A.2: Answer Call Use Case

USE CASE NAME	Send Message
ACTOR	VIP
NORMAL COURSE	<ol style="list-style-type: none"> 1. The process is initiated when VIP wants to send a message. 2. VIP speaks his message. 3. Speech Recognition API converts his spoken words into text. 4. VIP speaks a keyword. 5. Speech recognition API verifies the keyword from it data set. 6. System sends the message after verification of the sending voice command.

ALTERNATE COURSE	<p>3) a) Verification Denied</p> <ol style="list-style-type: none"> 1. Speech Recognition API doesn't verify the keyword. 2. VIP needs to speak again the keyword. <p>5) a) Message Discarding</p> <ol style="list-style-type: none"> 1. System discards the message after verification of the discarding voice command.
PRE CONDITION	<ul style="list-style-type: none"> • Keywords for taking some particular action are already defined in the data set of API.
POST CONDITION	<ul style="list-style-type: none"> • VIP will be able to send a message to anyone through voice commands.

Table A.3: Send Message Use Case

USE CASE NAME	Read Message
ACTOR	VIP
NORMAL COURSE	<ol style="list-style-type: none"> 1. The process is initiated when VIP wants to read a message. 2. VIP speaks keyword. 3. Speech Recognition API verifies the keyword from its data set. 4. Text to Speech Conversion API converts the message into speech after verification of the reading message voice command.
ALTERNATE COURSE	<p>3) a) Verification Denied</p> <ol style="list-style-type: none"> 1. Speech Recognition API doesn't verify the keyword. 2. VIP needs to speak again the keyword.
PRE CONDITION	<ul style="list-style-type: none"> • Keywords for taking some particular action are already defined in the data set of API.
POST CONDITION	<ul style="list-style-type: none"> • VIP will be able to know what is in the message.

Table A.4: Read Message Use Case

USE CASE NAME	Set Alarm
ACTOR	VIP
NORMAL COURSE	<ol style="list-style-type: none"> 1. The process is initiated when VIP wants to set an alarm. 2. VIP speaks keyword to open alarm options to set alarm. 3. Speech Recognition API verifies the keyword from its data set. 4. VIP speaks the time for setting the alarm. 5. VIP speaks keyword to set alarm. 6. Speech Recognition API verifies the keyword from its data set. 7. After verification of the voice command, alarm is set.
ALTERNATE COURSE	<p>3) a) Verification Denied</p> <ol style="list-style-type: none"> 1. Speech Recognition API doesn't verify the keyword. 2. VIP needs to speak again the keyword.
PRE CONDITION	<ul style="list-style-type: none"> • Keywords for taking some particular action are already defined in the data set of API.
POST CONDITION	<ul style="list-style-type: none"> • VIP will be able to set alarm.

Table A.5: Set Alarm Use Case

USE CASE NAME	Delete Alarm
ACTOR	VIP

<p>NORMAL COURSE</p>	<ol style="list-style-type: none"> 1. The process is initiated when VIP wants to delete an alarm. 2. VIP speaks keyword to open alarm options to delete alarm. 3. Speech Recognition API verifies the keyword from its data. 4. VIP speaks keyword to delete alarm. 5. Speech Recognition API verifies the keyword from its data. 6. After verification of the voice command, alarm will be deleted.
<p>ALTERNATE COURSE</p>	<p>3) a) Verification Denied</p> <ol style="list-style-type: none"> 1. Speech Recognition API doesn't verify the keyword. 2. VIP needs to speak again the keyword. <p>5) a) Verification Denied</p> <ol style="list-style-type: none"> 1. Speech Recognition API doesn't verify the keyword. 2. VIP needs to speak again the keyword.
<p>PRE CONDITION</p>	<ul style="list-style-type: none"> • Keywords for taking some particular action are already defined in the data set of API.
<p>POST CONDITION</p>	<ul style="list-style-type: none"> • VIP will be able to delete alarm

Table A.6: Delete Alarm Use Case

<p>USE CASE NAME</p>	<p>Set Calendar Event</p>
<p>ACTOR</p>	<p>VIP</p>

<p>NORMAL COURSE</p>	<ol style="list-style-type: none"> 1. The process is initiated when VIP wants to set a calendar event. 2. VIP speaks keyword to open calendar to set event. 3. Speech Recognition API verifies the keyword from its data. 4. VIP speaks the date and time for setting the event. 5. VIP speaks keyword to set event. 6. Speech Recognition API verifies the keyword from its data. 7. After verification of the voice command, event is set.
<p>ALTERNATE COURSE</p>	<p>3) a) Verification Denied</p> <ol style="list-style-type: none"> 1. Speech Recognition API doesn't verify the keyword. 2. VIP needs to speak again the keyword. <p>6) a) Verification Denied</p> <ol style="list-style-type: none"> 1. Speech Recognition API doesn't verify the keyword. 2. VIP needs to speak again the keyword.
<p>PRE CONDITION</p>	<ul style="list-style-type: none"> • Keywords for taking some particular action are already defined in the data set of API.
<p>POST CONDITION</p>	<ul style="list-style-type: none"> • VIP will be able to set calendar events.

Table A.7: Set Calendar Event Use Case

<p>USE CASE NAME</p>	<p>Delete Calendar Events</p>
<p>ACTOR</p>	<p>VIP</p>

NORMAL COURSE	<ol style="list-style-type: none"> 1. The process is initiated when VIP wants to delete a calendar event. 2. VIP speaks keyword to open events to delete event. 3. Speech Recognition API verifies the keyword from its data set. 4. VIP speaks keyword to delete event. 5. Speech Recognition API verifies the keyword from its data set. 6. After verification of the voice command, event will be deleted.
ALTERNATE COURSE	<ol style="list-style-type: none"> 3) a) Verification Denied <ol style="list-style-type: none"> 1. Speech Recognition API doesn't verify the keyword. 2. VIP needs to speak again the keyword. 5) a) Verification Denied <ol style="list-style-type: none"> 1. Speech Recognition API doesn't verify the keyword. 2. VIP needs to speak again the keyword.
PRE CONDITION	<ul style="list-style-type: none"> • Keywords for taking some particular action are already defined in the data set of API.
POST CONDITION	<ul style="list-style-type: none"> • VIP will be able to delete calendar event.

Table A.8: Delete Calendar Event Use Case

USE CASE NAME	Find Current Location
ACTOR	VIP

NORMAL COURSE	<ol style="list-style-type: none"> 1. The process is initiated when VIP wants to know about his current location. 2. VIP speaks keyword to find his location. 3. Speech Recognition API verifies the keyword from its data set. 4. After verification of the voice command, system tells VIP about his location.
ALTERNATE COURSE	<p>3) a) Verification Denied</p> <ol style="list-style-type: none"> 1. Speech Recognition API doesn't verify the keyword. 2. VIP needs to speak again the keyword.
PRE CONDITION	<ul style="list-style-type: none"> • Keywords for taking some particular action are already defined in the data set of API.
POST CONDITION	<ul style="list-style-type: none"> • VIP will be able to know about his location.

Table A.9: Location Use Case

USE CASE NAME	Weather Updates
ACTOR	VIP
NORMAL COURSE	<ol style="list-style-type: none"> 1. The process is initiated when a VIP wants to get weather updates. 2. VIP speaks a keyword. 3. Speech recognition API verifies that keyword from its data set. 4. System fetches the weather info after verification of the weather info command from VIP. 5. System will speak the wether information.

ALTERNATE COURSE	<p>3) a) Verification Denied</p> <ol style="list-style-type: none"> 1. Speech Recognition API doesn't verify the keyword. 2. VIP needs to speak again the keyword.
PRE CONDITION	<ul style="list-style-type: none"> • Keywords for taking some particular action are already defined in the data set of API.
POST CONDITION	<ul style="list-style-type: none"> • VIP will be able to get the weather info through voice commands and without touching the screen.

Table A.10: Weather Use Case

Appendix B : Sequence Diagrams

Make Call

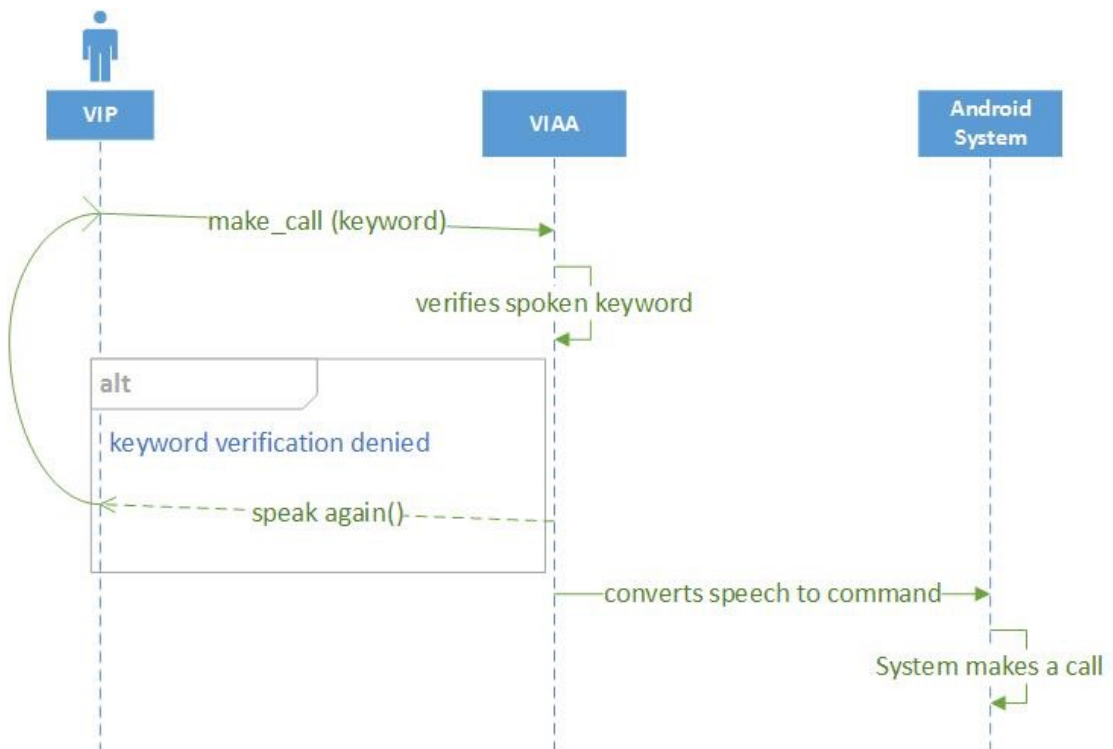


Fig B.1 : Make Call Sequence Diagram

Answer Call

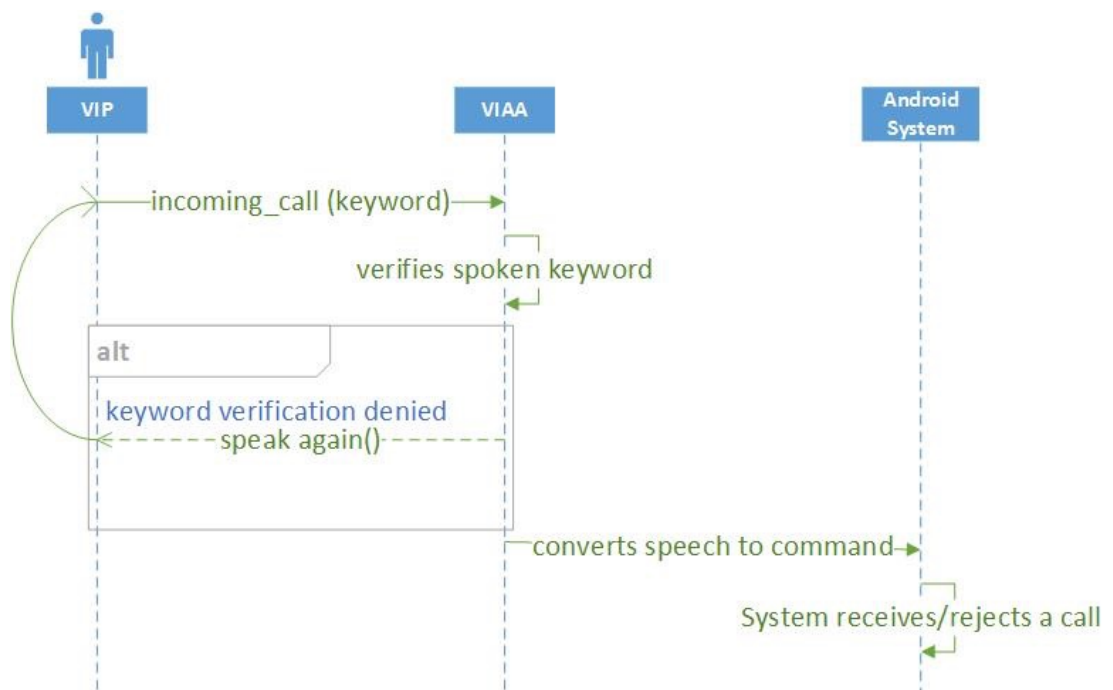


Fig B.2 : Answer Call Sequence Diagram

Send Message

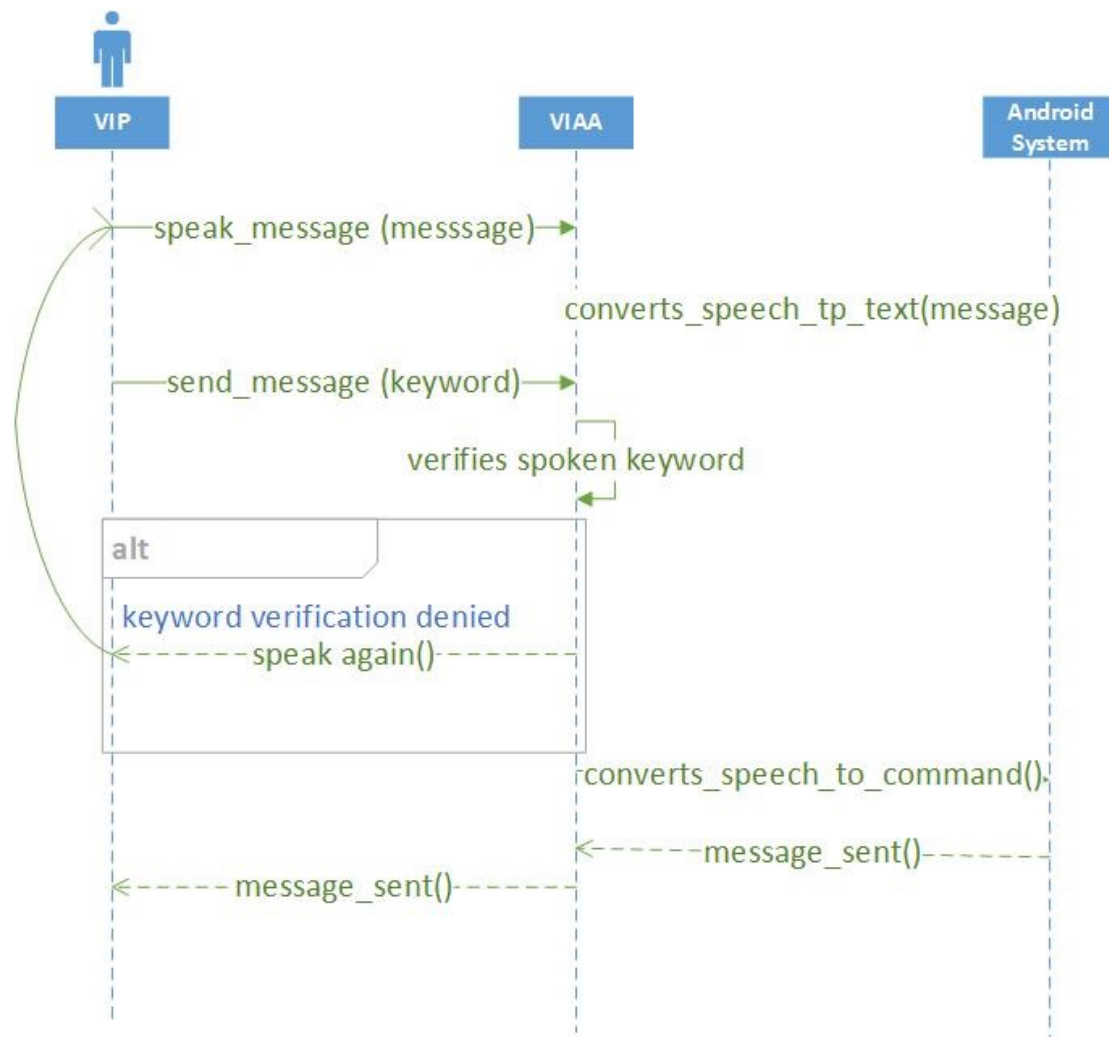


Fig B.3 : Send Message Sequence Diagram

Read Message

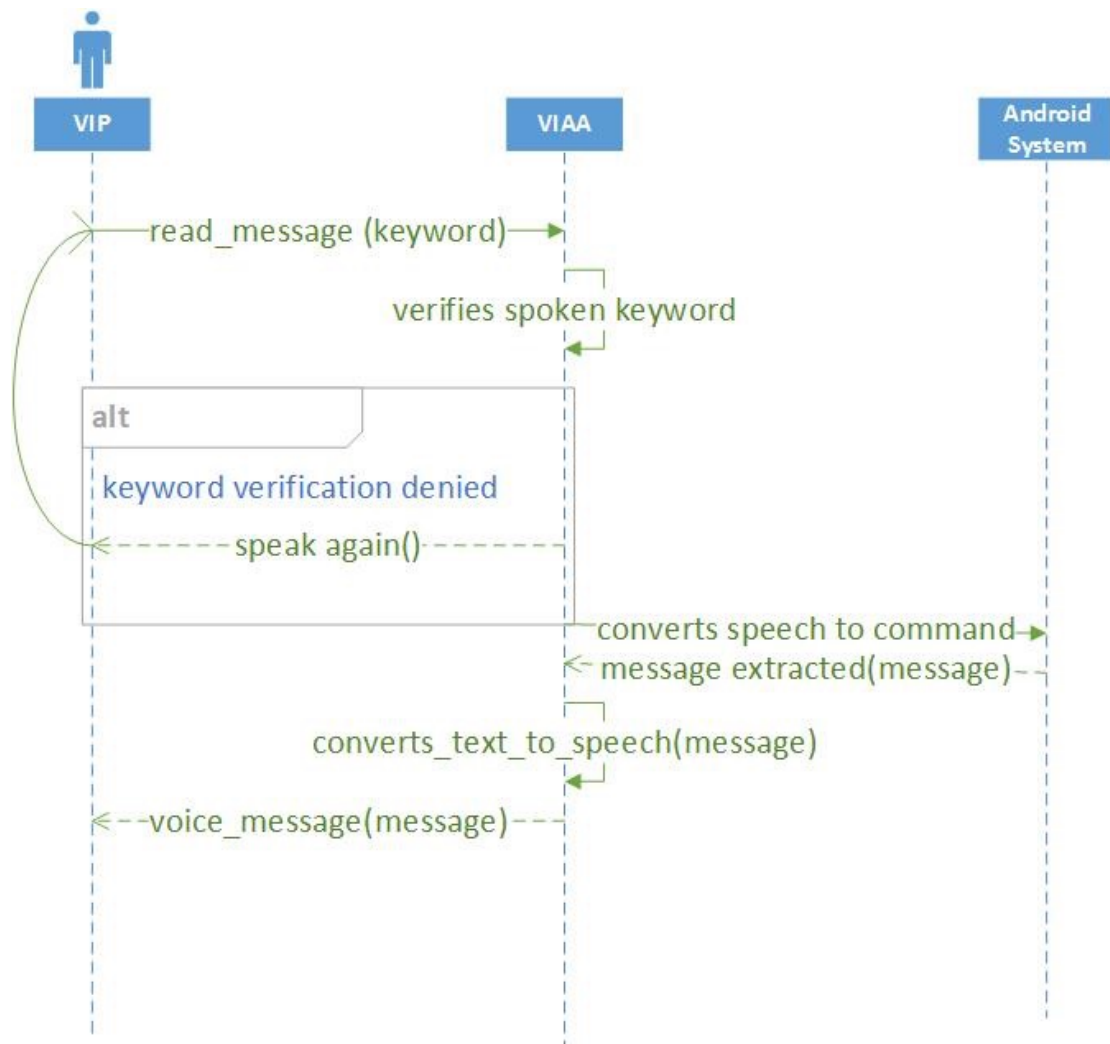


Fig B.4 : Read Message Sequence Diagram

Set Alarm

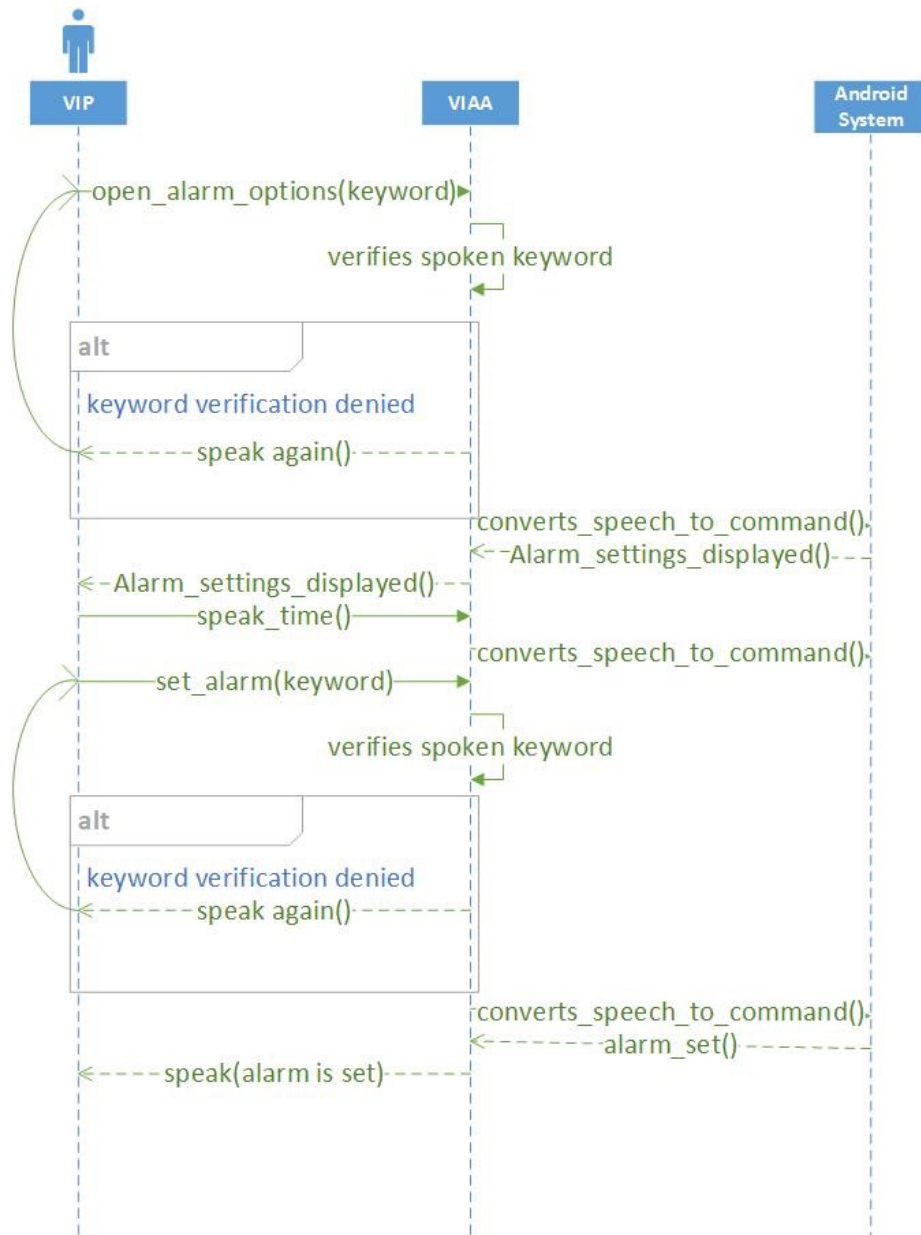


Fig B.5 : Set Alarm Sequence Diagram

Delete Alarm

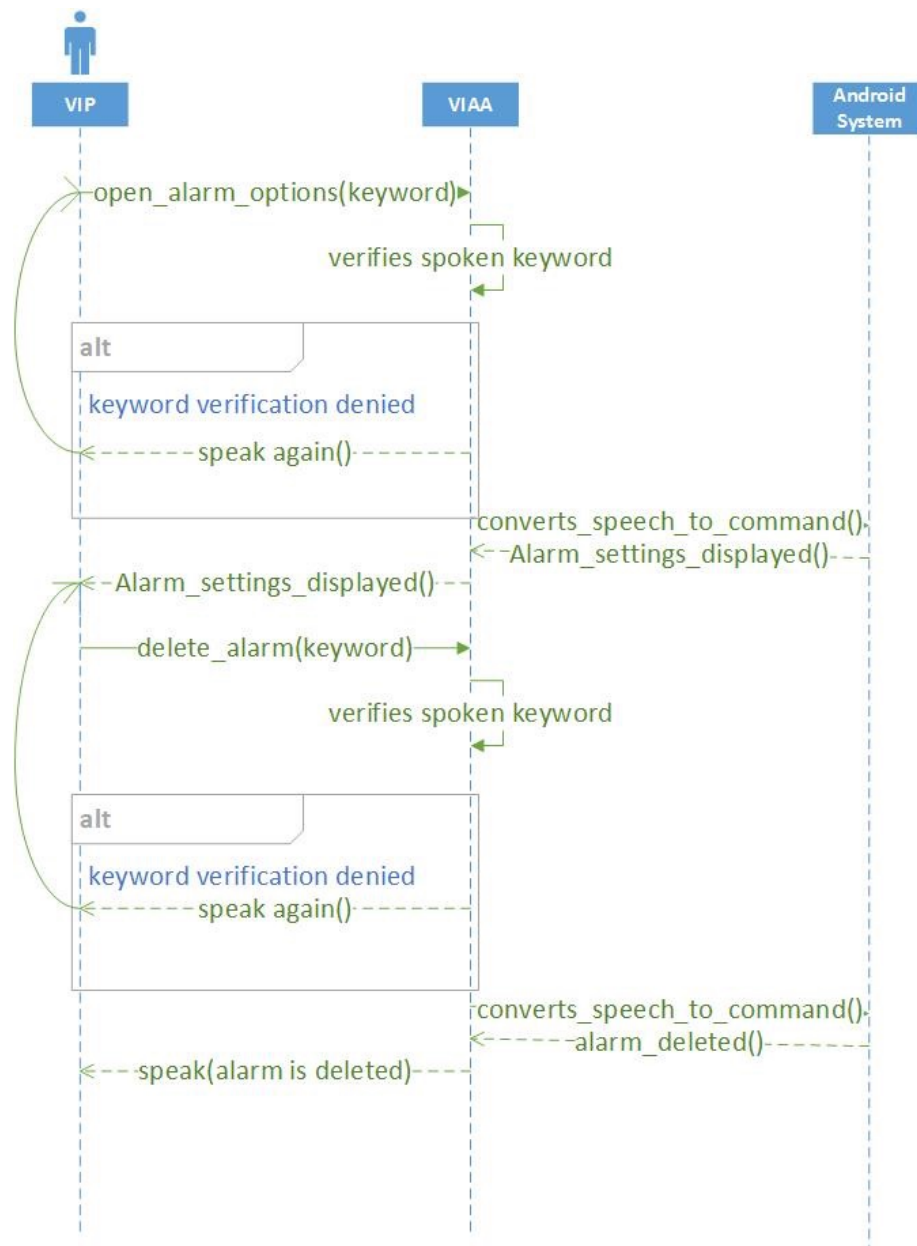


Fig B.6 : Delete Alarm Sequence Diagram

Set Calendar Event

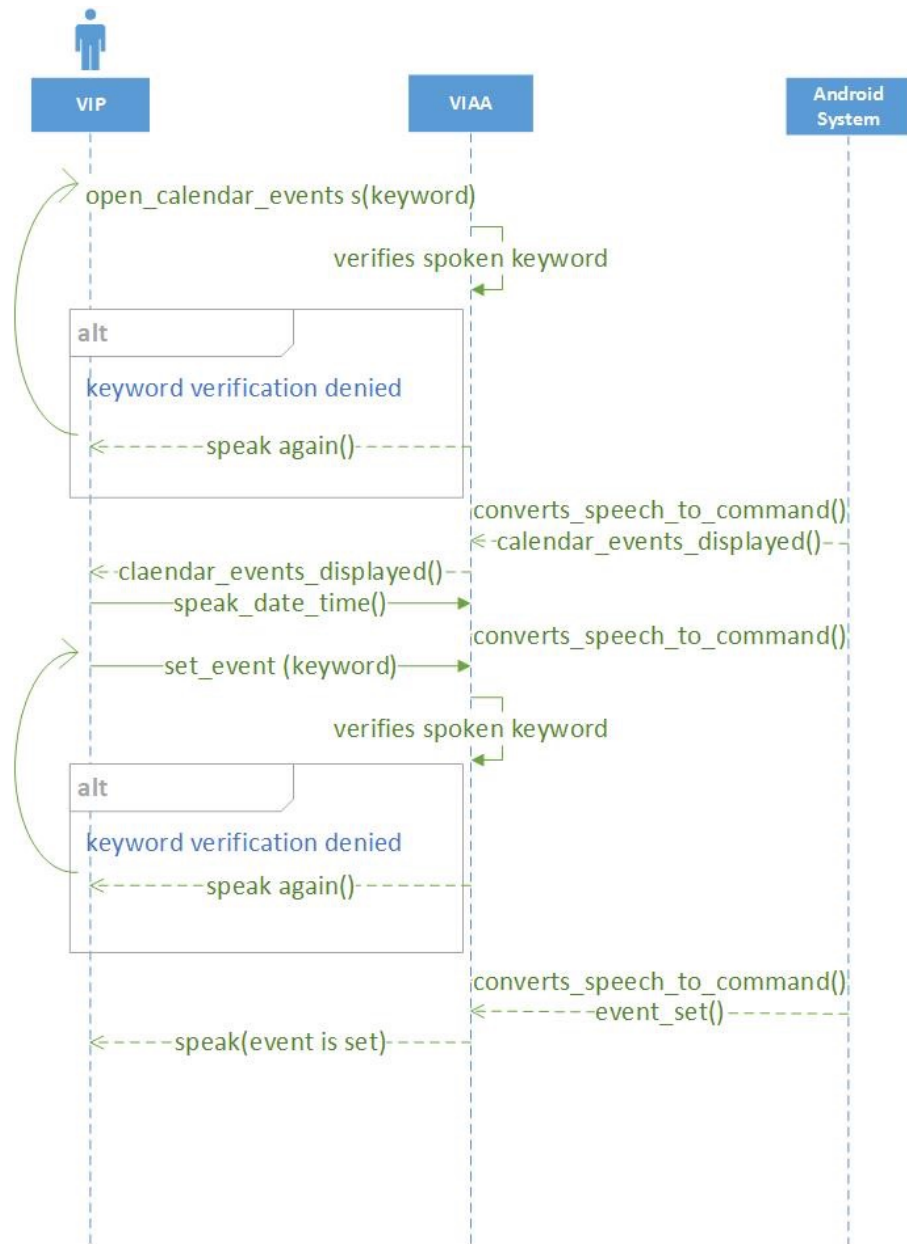


Fig B.7 : Set Calendar Event Sequence Diagram

Delete Calendar Event

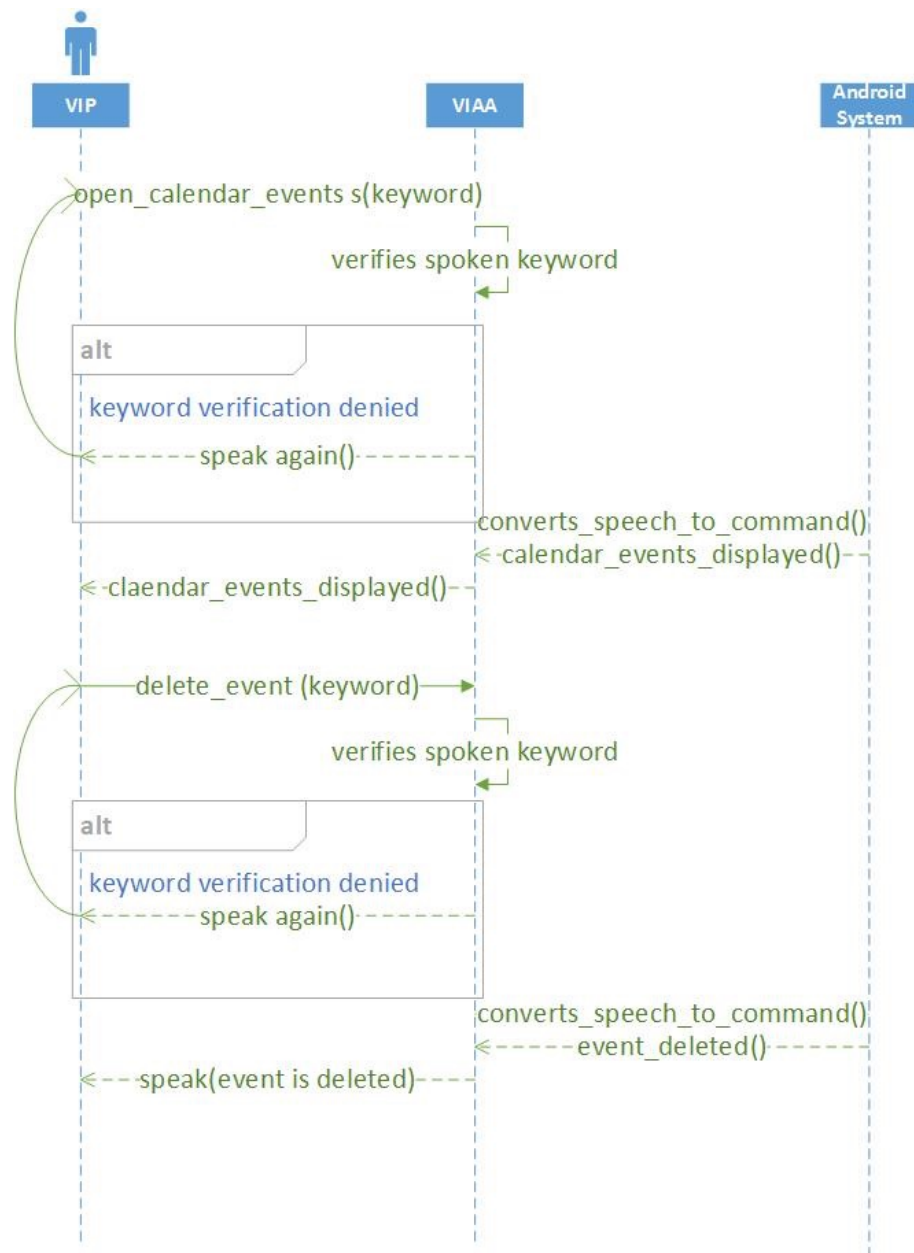


Fig B.8 : Delete Calendar Event Sequence Diagram

Find Location

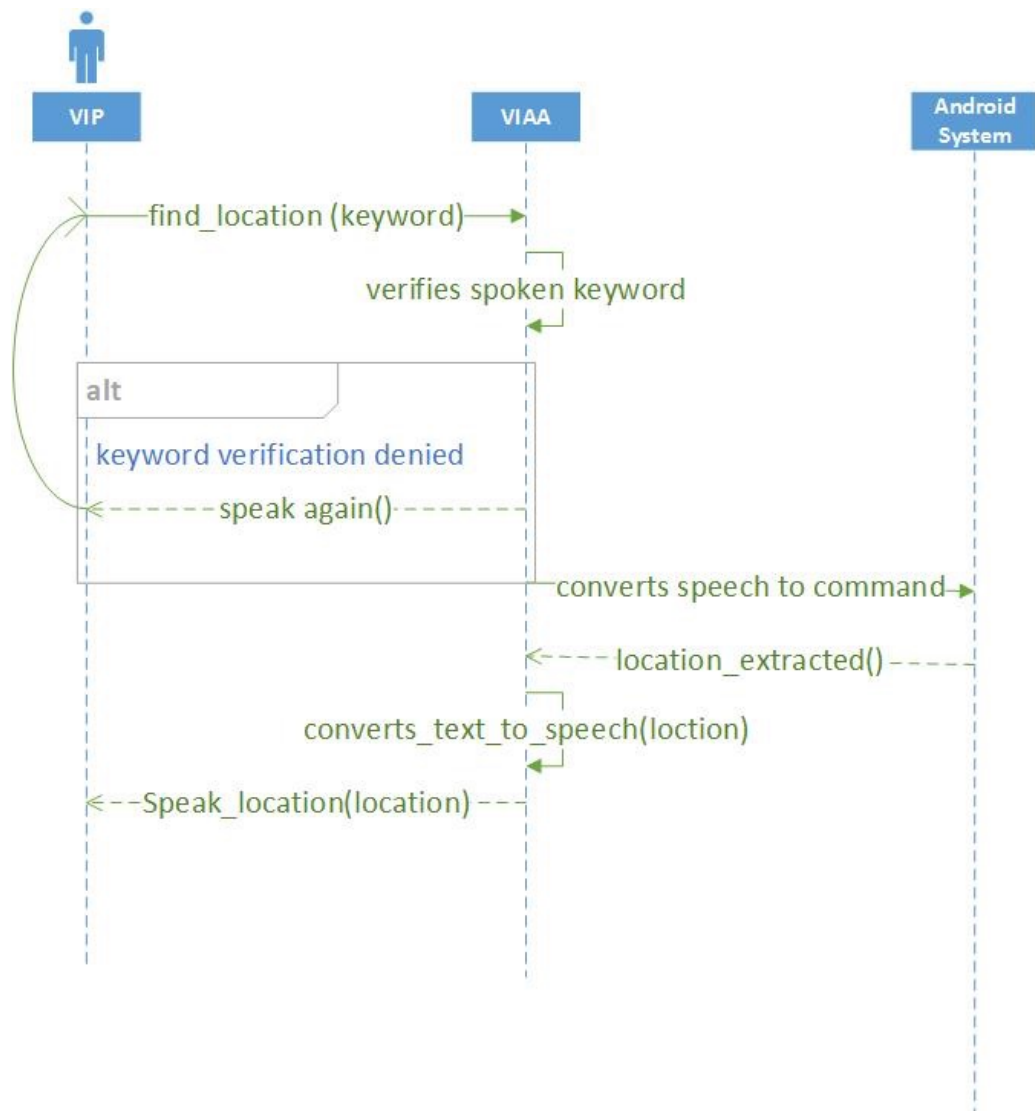


Fig B.9 : Find Location Sequence Diagram

Weather Update

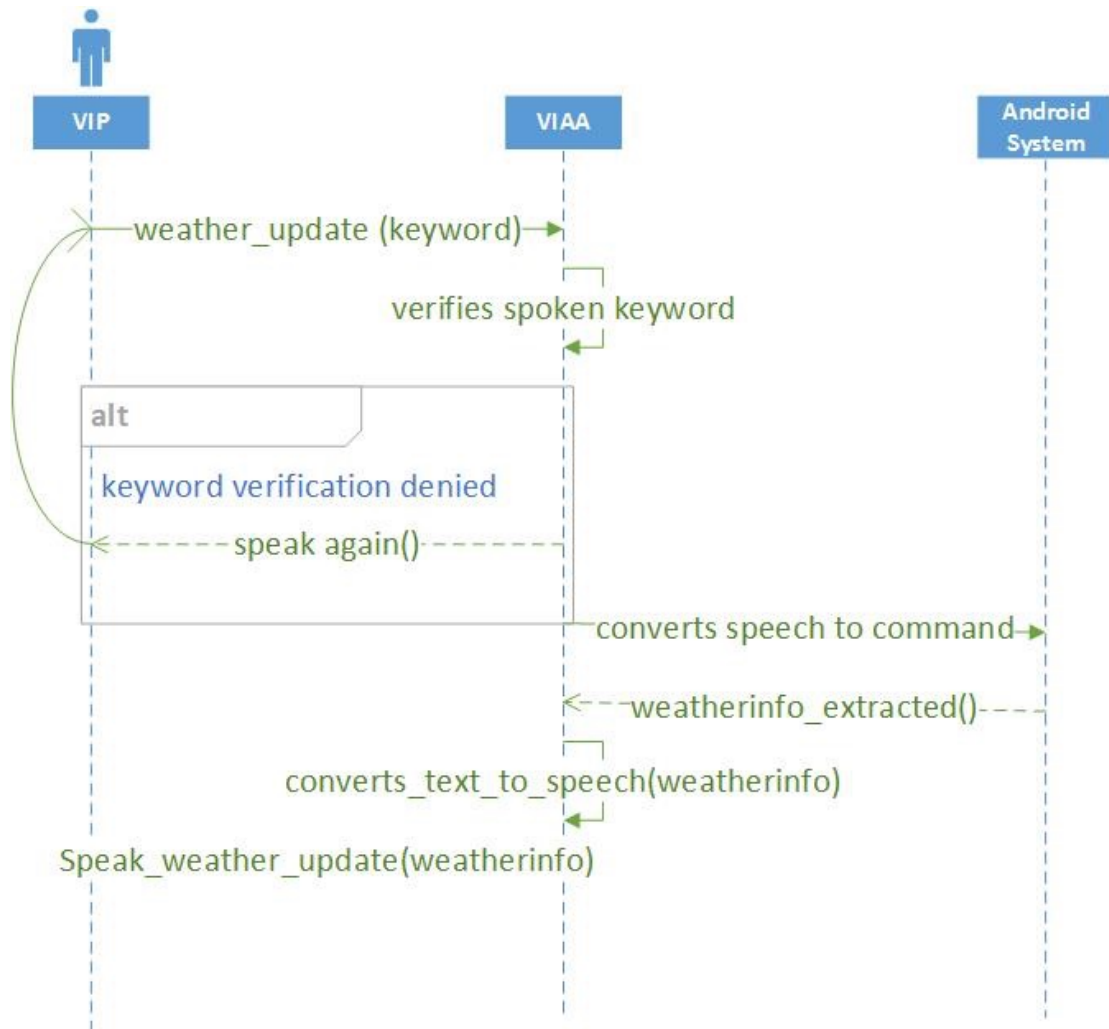


Fig B.10 : Weather Update Sequence Diagram

Appendix C : Detailed Description of Components

Component Template Description

Identification	The unique name for the component and the location of the component in the system.
Type	A module, a subprogram, a form, a data file, a control procedure, a class, etc
Purpose	Function and performance requirements implemented by the design component, including derived requirements. Derived requirements are not explicitly stated in the SRS - but are implied or adjunct to formally stated SDS requirements.
Subordinates	The internal structure of the component, the constituents of the component, and the functional requirements satisfied by each part.
Dependencies	How the component's function and performance relate to other components. How this component is used by other components. The other components that use this component. Interaction details such as timing, interaction conditions (such as order of execution and data sharing), and responsibility for creation, duplication, use, storage, and elimination of components.
Interfaces	Detailed description of all external or internal interfaces as well as of any mechanism for communicating through messages, parameters, or common data areas. All error messages and error codes should be identified. All screen formats, interactive messages, and other user interface components (originally defined in the SRS) should be given here.
Resources	A complete description of all resources (hardware or software) external to the component but required to carry out its functions.
Processing	A full description of the functions presented in the Function subsection. Pseudocode can be used to document algorithms, equations, and logic.

Data	For the data internal to the component, describes the representation method, initial values, use, semantics, and format.
------	--

Table C.1: Component Template Description

Speech Recognition API

Identification	Speech recognition API
Type	Package
Purpose	The purpose of this API is to enable VIPs to convey its commands to the system.
Function	It will convert user command understandable to the system and will act according to the command provided by the VIPs
Subordinates	Speech to text API is built in feature incorporated in the system that will facilitate the user to verify its commands and act accordingly
Dependencies	It will be dependent on Google dictionary to verify and understand the commands given by VIPs
Interfaces	There's no specific interface, its built in package that runs on the background of the system
Resources	It will be checked in the initial commands given by the users that whether Google dictionary is installed or not. If dictionary is not installed, application will automatically download it from Google play store
Processing	It will varies system to system. If system is fast enough, it will response back to the VIPs in 1 second
Data	The data that is used in it is related to VIPs that shall be verified by the Google dictionary. By using this API; we shall recognize the word

Table C.2: Speech Recognition API

Text to Speech API

Identification	Text to speech API
Type	Package
Purpose	The purpose of this API is to enable VIPs receive outputs from the system
Function	It will convert user command understandable to the VIP and will act according to the command provided by the VIPs
Subordinates	Text to speech API is built in feature incorporated in the system that will facilitate the user to get response to its command
Dependencies	It will be dependent on Google dictionary to verify and understand the commands given by VIPs and its response to the VIPs
Interfaces	There's no specific interface, its built in package that runs on the background of the system
Resources	It will be checked in the initial commands given by the users that whether Google dictionary is installed or not. If dictionary is not installed, application will automatically download it from Google play store
Processing	It will varies system to system. If system is fast enough, it will response back to the VIPs in 1 second
Data	The data that is used in it is related to VIPs that shall be verified by the Google dictionary. By using this API; we shall get response to the system

Table C.3: Text-to-Speech API

Appendix D : User Manual

1. Open VIAA

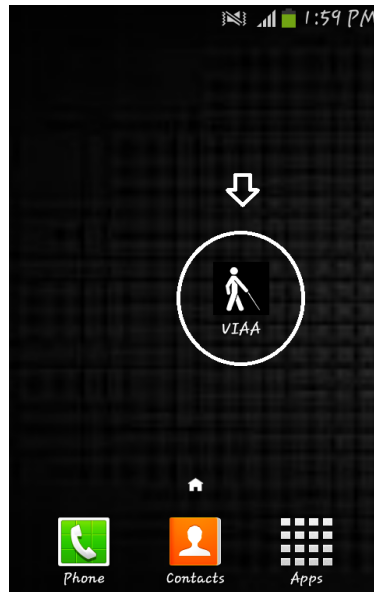


Fig D.1 :VIAA

2. Touch the screen to speak keyword (Call, Message, Alarm, Calendar, Weather,Location)

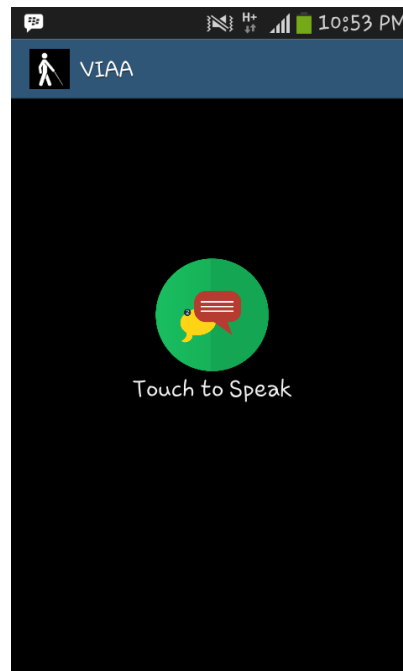


Fig D.2: Main Screen

2.1.Call

Touch the screen and speak phone number.

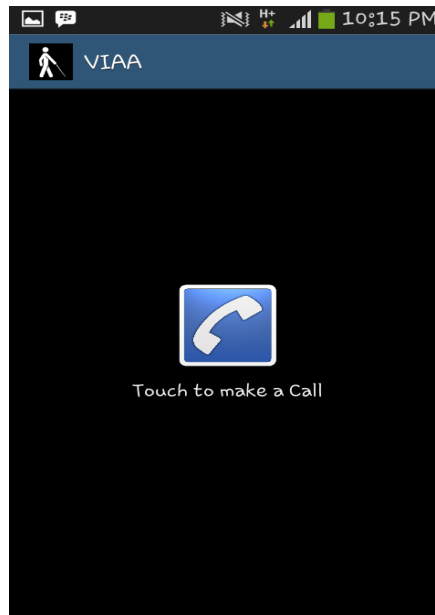


Fig D.3 : Call Screen

2.2.Message

- Touch screen and then speak phone number and message content.
- After speaking the message, touch again to send it.

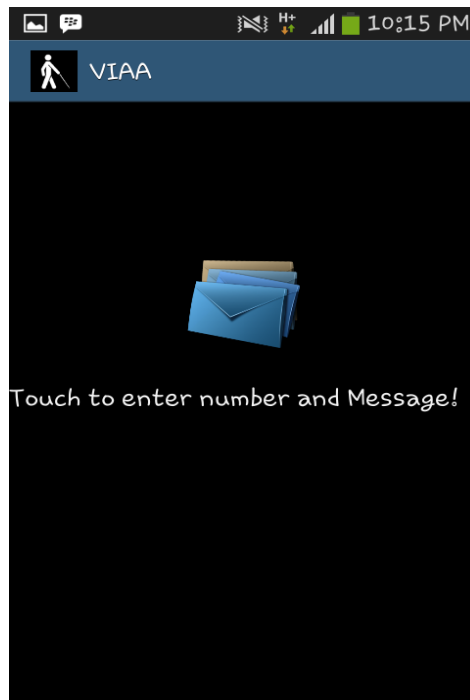


Fig D.4 : Message

2.3. Alarm

Touch the screen and then speak time to set alarm for desired time.

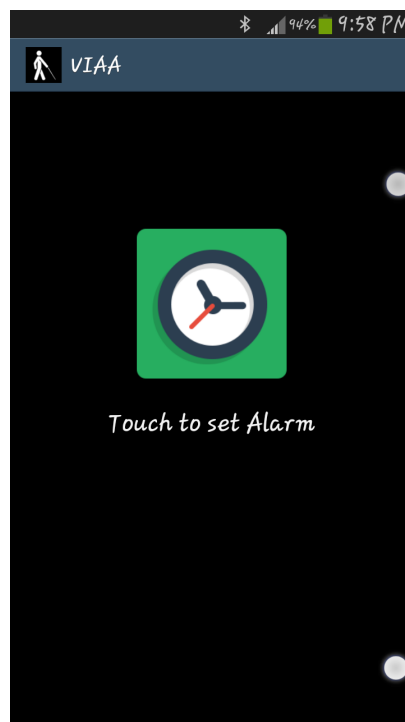


Fig D.5 : Alarm

2.4. Calendar Event

Touch the screen and speak first date and then time to set calendar event.



Fig D.6 : Calendar Event

2.5. Location

GPS should be turned on. When data is extracted just touch the screen it will speak the location.



Fig D.7 : Location

2.6. Weather Updates

Touch the screen and speak the city whose weather conditions user wants to know.

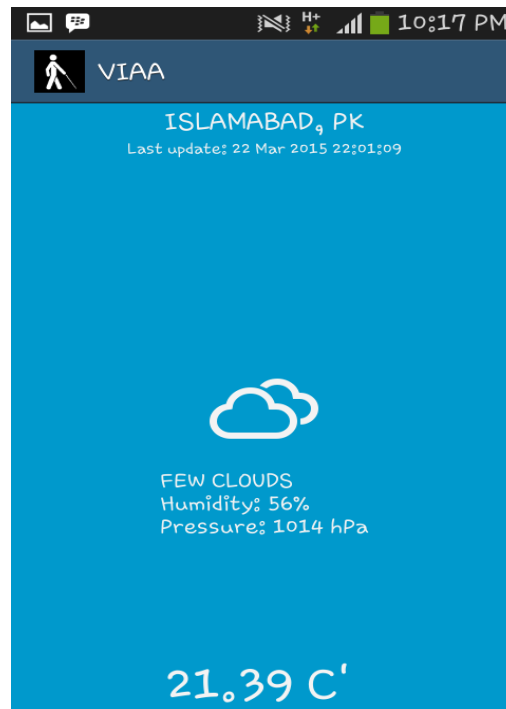


Fig D.8 : Weather

Bibliography

1. **Eyes-free.googlecode**, online: http://eyes-free.googlecode.com/svn/trunk/documentation/android_access/services.html (Accessed 2015)
2. **Androidcentral**, online: <http://www.androidcentral.com/what-google-talk-back> (Accessed 2015)
3. **Support.google**, online: <https://support.google.com/accessibility/android/answer/6007100?hl=en> (Accessed 2015)
4. **Accessibleandroid.blogspot**, online: <http://accessibleandroid.blogspot.com/2010/09/getting-to-know-kickback.html> (Accessed 2015)
1. **Androidforums**, online: <http://androidforums.com/threads/what-are-kickback-soundback-and-talkback.286001/> (Accessed 2015)
5. **Groups.google**, online : <https://groups.google.com/forum/#!forum/eyes-free> (Accessed 2015)
6. **Google-opensource.blogspot**, online:<http://google-opensource.blogspot.com/2009/04/announcing-eyes-free-shell-for-android.html> (Accessed 2015)
7. **Androidcentral**, online: <http://www.androidcentral.com/what-google-talk-back> (Accessed 2015)

Glossary

API: Application Programming Interface

VIP: Visually Impaired People

VIAA: Visually Impaired Assistance App

OS: Operating System

GPS: Global Positioning System

App: Application

SDLC: Software Development Life Cycle