A PROJECT REPORT ON

”VOICE CONTROLLED PERSONAL ASSISTANT DEVICE AND CONNECTING IOT DEVICES”

SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE

OF

BACHELOR OF ENGINEERING (COMPUTER ENGINEERING)

BY

KAPISH KAITH	ROLL NO:13CO028
ROHAN KILLEDAR	ROLL NO:13CO032
CHAITANYA KULKARNI	ROLL NO:13CO033
ABHAY DEKATE	ROLL NO:13CO401

DEPARTMENT OF COMPUTER ENGINEERING
AISSMS COLLEGE OF ENGINEERING

सत्याला मरण नाही

Kennedy Road, Near R.T.O Office, Pune, Maharashtra 411001

2016-2017

CERTIFICATE

सत्याला मरण नाही

This is to certify that the project report entitles

”VOICE CONTROLLED PERSONAL ASSISTANT DEVICE AND CONNECTING IOT DEVICES”

Submitted by

KAPISH KAITH	**ROLL NO:13CO028**
ROHAN KILLEDAR	**ROLL NO:13CO032**
CHAITANYA KULKARNI	**ROLL NO:13CO033**
ABHAY DEKATE	**ROLL NO:13CO401**

is a bonafide work carried out by them under the supervision of **Prof N. R. Talhar** and it is approved for the partial fulfillment of the requirement of Savitribai Phule Pune University Pune for the award of the degree of Bachelor of Engineering (Computer Engineering)

This project work has not been earlier submitted to any other Institute or University for the award of any degree or diploma.

Prof. N.R. Talhar
Guide
Dept. of Computer Engineering

Prof. D.P. Gaikwad
Head,
Dept. of Computer Engineering

Dr. S.P. Danao
Principal
AISSMS College of Engineering

Place : Pune
Date :

ACKNOWLEDGEMENT

We would like to extend my sincere gratitude and thanks to my guide **Prof. N.R. Talhar**, for his invaluable guidance and for giving us useful inputs and encouragement time and again, which inspired us to work harder. Due to his forethought, appreciation of the work involved and continuous imparting of useful tips, this report has been successfully completed.

We are extremely grateful to **Prof. D.P. Gaikwad**, Head of the Department of Computer Engineering, for his encouragement during the course of the project work.

We also extend our heartfelt gratitude to the staff of Computer Engineering Department for their cooperation and support.

We also take this opportunity to thank all our classmates,friends and all those who have directly or indirectly provided their overwhelming support during our project work and the development of this report.

Kapish Kaith
Rohan Killedar
Chaitanya Kulkarni
Abhay Dekate

ABSTRACT

In the Modern Era of fast moving technology we can do things which we never thought we could do before. But to achieve and accomplish these thoughts there is a need for a platform which can automate all our tasks with ease and comfort.

Thus we need to develop a Personal AI having brilliant powers of deduction and the ability to interact with our surroundings just by one of the materialistic form of human interaction, our VOICE. The Hardware device captures the personals audio request through microphone and processes the request so that the device can respond to the individual using in-built speaker module. For Example, if you ask the device 'what's the weather?' or 'how's traffic?' using its built-in skills, it looks up the weather and traffic status respectively and then returns the response to the customer through connected speaker.

The platform is open to all and it can connect all IOT devices in the vicinity. This feature makes it distinctive from already existing personal AIs and separates it from the flock. It uses open source software to process natural language, to determine the intent of the query and to perform the action. The platform features include : Turn on the lights, do the mathematics, play your favorite song, check the weather, connect IoT devices, basically answer to all your logical questions. Just speak naturally and the platform is there to do your bidding.

The Platform also extends an Android Application in which you can add a To-Do-list or set a reminder or alarm through the app, push notifications and personalize according to your needs. The device has a lot of native skills and abilities baked in it and more can be added to extend its capabilities.

There is still a lot of ground to be covered up in the world of automation but the skills of our device can help to build a new generation of voice controlled devices and bring a new sustaining change in the field of automation.

LIST OF FIGURES

LIST OF TABLES

CONTENTS

Chapter 1

INTRODUCTION

1.1 BACKGROUND

In 21st century, everything is leaning towards automation; may it be your home or car. There is an unbelievable change in; rather advancement in technology over the last few years. Believe it or not, in today's world you can interact with your machine. What is interacting with a machine? Obviously giving it some input, but what if the input is not in the conventional way of typing, rather it is your own Voice. What if you are talking to the machine, giving it commands and wanting the machine to interact with you like your assistant. What if the machine is not giving you answers just by showing you the best results but also by advising you with a better alternative. An easy access to machine with voice commands is the revolutionary way of human-system interaction.

To achieve this, we need to use speech to text API for understanding the input. Many companies like Google, Amazon and Apple are trying to achieve this in generalized form. Isnt it amazing that you can set reminders by just saying remind me to .... or set alarm with wake me up at .

Understanding the importance of this we have decided to make a system that can be placed anywhere in vicinity and you can ask it to help you do anything for you just by speaking with it. Not only this you can also connect two such devices through WiFi and make them communicate with each other in future. This device can be very handy for day to day use and it can help you function better by constantly giving you reminders and updates. Why would we need it? Because your own voice is turning into a best input device than a conventional Enter Key.

1.2 PROBLEM STATEMENT

To develop a hardware system which takes the input through voice commands and perform the various actions and keeps learning the context of these commands to further improvise the responses in future and help humans with day-to-day workforce.

1.3 PURPOSE

The purpose of this project to ease the human effort by helping them out with whatever they need. May it be there reminders, setting their alarms, helping them out with their navigation, weather updates, etc. It represents a detailed description of the Voice Based Personal Assistant System. It will explain the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli? This project can act as a prototype for many advanced applications.

1.4 SCOPE

This system will be a Voice Based Personal Assistant System for a home and personal use. It will be designed to minimize the human efforts to interact with many other subsystem, which would otherwise have to be performed manually. By achieving this, the system will make human life comfortable.

More specifically, this system is designed to interact with other subsystems intelligently and control those other devices or subsystems, this includes IoT devices or getting news from Internet, providing other information, getting personalized data saved previously on the system, etc. The software will facilitate ease of access to various other devices and platforms. The system will also contain an android application to add more details to system from the smart-phone over the Internet.

It always keeps listing for its name and wakes up to response upon calling with the assigned functionality. It can be a very crucial device in future. It will be useful for accessing any kind of information. For children, they can just ask for What is the highest peak in the world and itll simply reply as Mount Everest. You can just click the picture of anything from the app and itll response you with its contents. It keeps learning the sequence of questions asked to it related to its context which it remembers for the future. So when the same context is mentioned, it starts a conversation with you asking relevant questions.

Chapter 2

LITERATURE SURVEY

2.1 PRESENT WORK

1. Interacting With Computers by Voice: Automatic Speech Recognition and Synthesis
Authors : DOUGLAS OSHAUGHNESS

This paper has examined the basic aspects of humancomputer interactions via voice. The basic approaches and methods of ASR and synthesis have been discussed. Microprocessors can easily handle the computation speeds needed for many synthesizers, and many function entirely in software. However, the memory requirements of modern waveform concatenation systems strain the capacities of some practical systems.

2. Analysis of Machine Translation and Speech Synthesis Speech-To-Speech Trnaslation System
Authors : KEI HASHIMOTO, JUNICHI YAMAGISHI, WILLIAM BYRNE, SIMON KING, KEIICHI TOKUDA

This paper has provided an analysis of the impacts of machine translation and speech synthesis on speech-to-speech translation. We have shown that the naturalness and intelligibility of the synthesized speech are strongly affected by the fluency of the translated sentences. The intelligibility of synthesized speech is improved as the translated sentence become more uent.

3. On the track of Artificial Intelligence: Learning with Intelligent Personal Assistants
Authors : NIL GOKSEL-CANBEK, MEHMET EMIN MUTLU

The paths of this study regarding IPAs is intended to reveal an overview on how and to what extent these devices might be used in human-computer interaction and learning. In this connection, the working systems of the IPAs namely Apples Siri, Google Now and Microsoft Cortana are revised within the context of AI. Although there have been several works related to IPAs in education. The potential use of IPAs for second language learning within Natural Language Processing (NLP) should be focused particularly.

4. Chatbot-Assiting: SIRI
Authors : HARSHITA PHATNANI, Mr. JYOTIPRAKASH PATRA, ANKIT SHARMA

Siri's first Apple iteration opens minds and speaks loudly to Siri's potential. What struck us is that even with this initial release one can readily imagine a sea change in the way humans interact with mobile. My voice application business allowed us to see the potential of truly great natural language voice technologies. The few great applications we found left us believing that someday, voice will handle large numbers of everyday tasks and, where appropriate, even more complex things.

5. An Intelligent Voice Assistant Using Android Platform
Authors : SUTAR SHEKHAR, POPHALI SAMEER, KAMAD NEHA, Prof. DEVKATE LAXMAN

We have developed application in which user can easily send a message with their voice commands and also tried to use the most of the inbuilt application with voice command. But all these applications have adaptions for the English language. We are also surveying to use the mailing and calendar where user will be able to mail and also create their event using voice command.

6. Home Automation Using IOT
Authors : VINAY SAGAR, KUSUMA SM

Home automation using IOT has been experimentally proven to work satisfactorily by connecting simple appliances to it and the application appliances were successfully controlled remotely through internet. The designed system not only monitors the sensor data like temperature, gas, light, motion sensors but also actuates a process according to the requirement. For instance, switching on the lights when it gets dark. It also stores the sensor parameters in the cloud in a timely manner. This will help user to analyze the condition of various parameters in the home anytime anywhere.

7. RASPBERRY PI BASED ROBOT WITH CLOUD TECHNOLOGY
Authors : Prof. GOKILAVANI R, NAVANEETHAN S

The data monitored has been updated to the firebase cloud server with the help of wifi and the continuous access from anywhere is feasible. The proposed method contains the raspberry pi and the sensor devices. The output sensor data is updated to the cloud server; the updated sensor values can be viewed on its own smart phone. The commands can be sent to server by using WiFi using the IP address of raspberry pi toolkit.

8. Exploring IOT Application Using Raspberry Pi
Authors : CHEAH WAI ZHAO, JAYANAND JEGATHEESAN, SON CHEE LOON

Raspberry PI is useful for small application development because it can be used to integrate with many components such as speakers, LED lights, sensors, cameras and wireless communication units to develop smart applications. It is very important to protect the file in order to maintain data consistency and accuracy. File sharing is widely used ranging from small-medium sized company to big company. One of the reason is that, admin can easily manage the file and client can access the file efficiently.

9. Natural Language Processing
Authors : ABHIMANYU CHOPRA, ABHINAV PRASHAR, CHANDRESH SAIN

The strength of the capabilities to use natural language for query specification and retrieval bags over the keyword, key-phrase approaches. We believe that the restricted use of natural language in captions for multimedia data abstraction is less cumbersome task than the full natural language fact abstraction and feel that can be judged and build upon not only for abstracting images but also the form so multimedia (audio, video, text, data etc) data as input sources.

10. A study of techniques for facial detection and expression classification
Authors : G. HEMALATHA, C.P. SUMATHI

Although human recognize facial expressions virtually without effort or delay, reliable expression recognition by machine is still a challenge, in achieving optimal pre-processing, feature extraction or selection, and classification, particularly under conditions of input data variability.To attain successful recognition performance, most current expression recognition approaches require some control over the imaging conditions because many real-world applications requires operational flexibility.

2.2 PROPOSED WORK

Our proposed system will provide following features:

- Performing Arithmetic calculations based on voice commands and giving back the computed solution through a voice.
- Searching Internet based on user's voice input and giving back the reply through a voice with further interactive questions by the machine.
- Checking the weather conditions of the users location.
- Playing music as per voice command of the user.
- Notify using push notification.
- Wakes you up by setting your alarm.
- It will connect itself with smart-phone through WiFi.
- It will keep itself updated by the data on its cloud server.
- It sets your reminders.
- It spells your word correctly for you.
- It shows you the distance between two locations.
- It switch ONs and switch OFFs your light.

Chapter 3

SOFTWARE REQUIREMENTS SPECIFICATION

3.1 INTRODUCTION

Nowadays, everything is leaning towards automation; may it be your home or car. To achieve this, one of the way is to use voice commands to control and interact with the objects around us, get news or any information over the internet. An easy access to machine with voice commands is the revolutionary way of human-system interaction. Many companies like Google, Amazon and Apple are trying to achieve this in generalized form. Isnt it amazing that you can set reminders by just saying remind me to .... or set alarm with wake me up at . Understanding the importance of this we have decided to make a system that can be placed anywhere in vicinity and you can ask it to help you do anything for you just by speaking with it. Why would we need it? Because your own voice is turning into a best input device than a conventional Enter Key.

3.1.1 PROJECT SCOPE

- It always keeps listing for its name and wakes up to response upon calling with the assigned functionality.
- It can be a very crucial device in future. It will be useful for accessing any kind of information. For children, they can just ask for What is the highest peak in the world and itll simply reply as Mount Everest.
- You can just click the picture of anything from the app and itll response you with its contents.
- It keeps learning the sequence of questions asked to it related to its context which it remembers for the future. So when the same context is mentioned, it starts a conversation with you asking relevant questions.

3.1.2 USER CLASSES AND CHARACTERISTICS

This project is meant to provide a faster, easier, convenient and sophisticated way to interface with our environment and providing quick results for voice commands. Consequently, the application will be able to predict more accurately the user related data as the usage of system increases i.e; it gathers or analyses more amount of data related to similar type of usage. The user interface will be as intuitive as possible. The concept is to let the user feel as having a personal assistant for their jobs that are performed daily. Any user can use this system as long as the system is able to understand the user's voice command.

3.1.3 OPERATING ENVIRONMENT

The main component of our project is the hardware-software system that we are designing to ease the process of interfacing with various devices and extracting information over Internet. The system will require to store and acquire significant amount of data per user over the Internet so it requires a decent amount of storage space and good network connectivity.

3.1.4 DESIGN AND IMPLEMENTATION CONSTRAINTS

- Primary constraint is that it is network dependent since it uses API's from different providers to fetch information.
- It works only with English language.
- One of the constraints is background noise in fetching commands through voice.
- Data flow is unidirectional as Firebase cloud server only sends the data and does not receive any type of data.
- The device is not designed for a particular voice and hence it can be less secure since anyone can use it.

3.1.5 ASSUMPTIONS AND DEPENDENCIES

This project will work on the minimum system specifications as follow:

- Raspberry pi controller.
- Arduino Uno microprocessor.
- 16 GB MicroSD flash memory.
- Minimum 512GB LPDDR3 RAM.
- Internet connection and smooth access to a network of computers of same physical system configuration.
- Data stored and accessed from Firebase cloud server database.
- Natural language toolkit.

3.2 EXTERNAL INTERFACE REQUIREMENTS

3.2.1 USER INTERFACES

- User interface includes voice commands.
- Another interface includes updated data through android application.
- Device will have a screen which will show speech to text data.

3.2.2 HARDWARE INTERFACES

- Raspberry pi Model B
- Arduino Uno
- Neo-pixel ring
- nRF24L01 Transcievers
- Servo motor

3.2.3 SOFTWARE INTERFACES

- NLTK
- C++
- Python
- Java
- XML
- Firebase API
- Google API

3.3 OTHER REQUIREMENTS

3.3.1 DATABASE REQUIREMENT

- Firebase Cloud database
- Ease of Development and Integration of Various Modules of applications
- Data Availability for Platform
- Real-time Performance of the Firebase Cloud Services

3.4 NON-FUNCTIONAL REQUIREMENTS

These requirements don't affect the system features but play an important role in deciding other factors that are important for a software application to be reliable.

3.4.1 PERFORMANCE REQUIREMENTS

The system needs to be alert always and it should respond as soon as it gets initialization command through voice. The system should perform its tasks efficiently with minimum amount of time. System should be reliable and robust.

3.4.2 SOFTWARE QUALITY ATTRIBUTES

The application software gives justice to important quality attributes such as:

- **Flexibility:**
  commands related to various domains should be performed by the system.

- **Reliability:**
  System should perform tasks according to its specifications.
- **Usability:**
  Provides simple user interface easily accessible by the users.
- **Scalability:**
  System can be used for variable data as well as is scalable on multiple systems over same network.
- **Security:**
  Secure as the system asks for user's credentials to provide access to system.
- **Dependability:**
  Application software is dependable on the online data collection by Firebase cloud server.

3.5 ANALYSIS MODEL

3.5.1 DATA FLOW DIAGRAMS

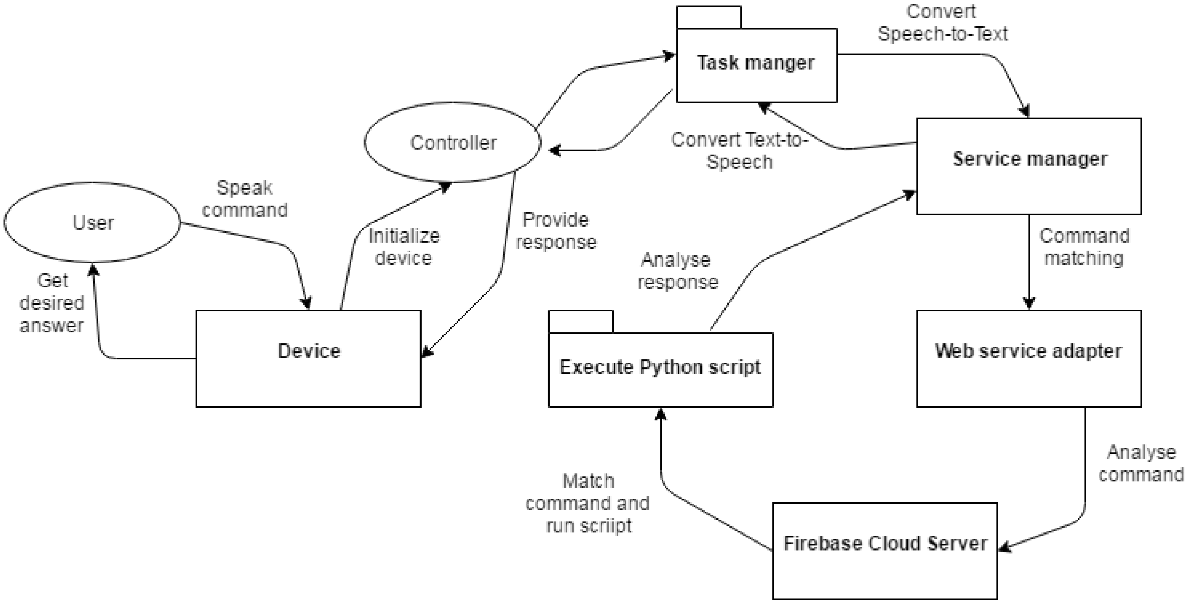


Figure 3.1: Data-flow Diagram

3.5.2 MATHEMATICAL MODEL

Mathematical Formulation:
Let the M i the universal states which contains,
M = {Q, S, F, Q1 , Qf }
where,

Q = No. of states {Q1,Q2,Q3,Q4,Q5,Q6,Q7,Q8}
X = No. of states {X1, X2, X3, X4}
Q1 = Initial State.
S = Success state.
F = Failure state.

where,

Q1= Start.

Q2 = Initialize assistant by calling action word.

Q3 = Once initialized, give input in the speech format.

Q4 = Processing of speech into text.

Q5 = Text is compared with commands.

Q6 = Perform action according to command.

Q7 = Convert the action into speech.

Q8 = Give output in the speech format.

X1 = Connect Personal AI Device to mobile application.

X2 = Update To-Do list, Calender, Alarm, reminder, etc.

X3 = Updated data is sent to cloud server.

X4 = According to the data sent, AI Device will take action.

Failure Conditions: F = {F1,F2,F3}

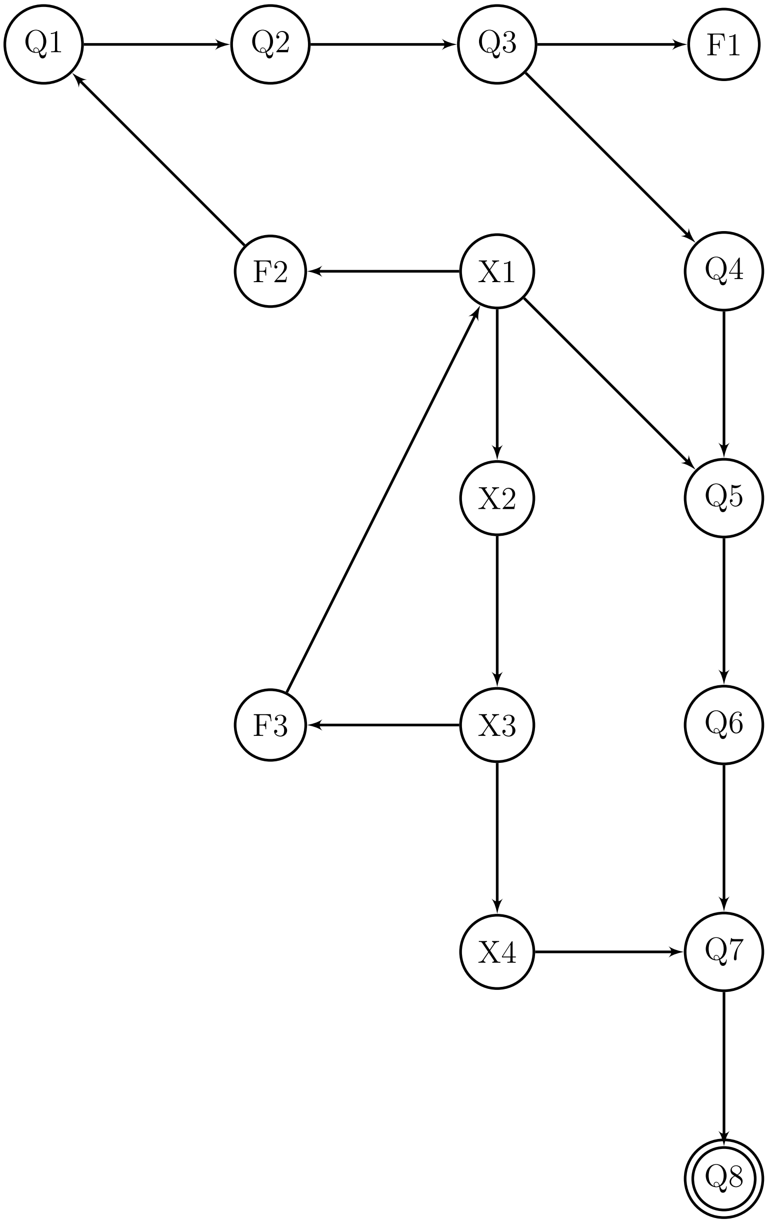
- F1 = Failure if the device is not initialised due to noisy background.
- F2 = Connection with android app failed.

- F3 = Server down failure or cannot retrieve data from server.

Success Conditions: S = {S1}

- S1 = Success after the data is assimilated and output is produced accordingly.

3.5.3 ACITIVITY DIAGRAM



3.6 SYSTEM IMPLEMENTATION PLAN

Table 3.1: System Implementation Plan Phase-I

SR. NO.	TASK NAME	DURATION	COMPLETION
1.	Project Topic Selection	12 days	✓
2.	Literature Survey	15 days	✓
3.	Study Of Existing System	10 days	✓
4.	Synopsis & Abstract Submission	15 days	✓
5.	SRS	15 days	✓
6.	Design Of System Architecture	8 days	✓
7.	Design Of UML Diagrams	3 days	✓
8.	Planning Of System Modules & Interface	10 days	✓

Table 3.2: System Implementation Plan Phase-II

SR. NO.	TASK NAME	DURATION	COMPLETION
9.	Implementation Of Neo-pixels ring & Servo motor	10 Days	
10.	Implementation Of nRF24L01 Transciever	5 Days	
11.	Implementation Of IoT box	7 Days	
12.	Implementation Of Jasper voice assistant	12 Days	
13.	Get desired output	10 Days	
14.	Testing Of Above Modules After Completion of Each Module	3 Days Per Module	
15.	Project Review	5 Days	

Chapter 4

SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE

The overall system design consists of following modules:

(a) Data collection in the form of speech.

(b) Voice analysis and conversion to text

(c) Data storage and processing

(d) Generating speech from the processed text output

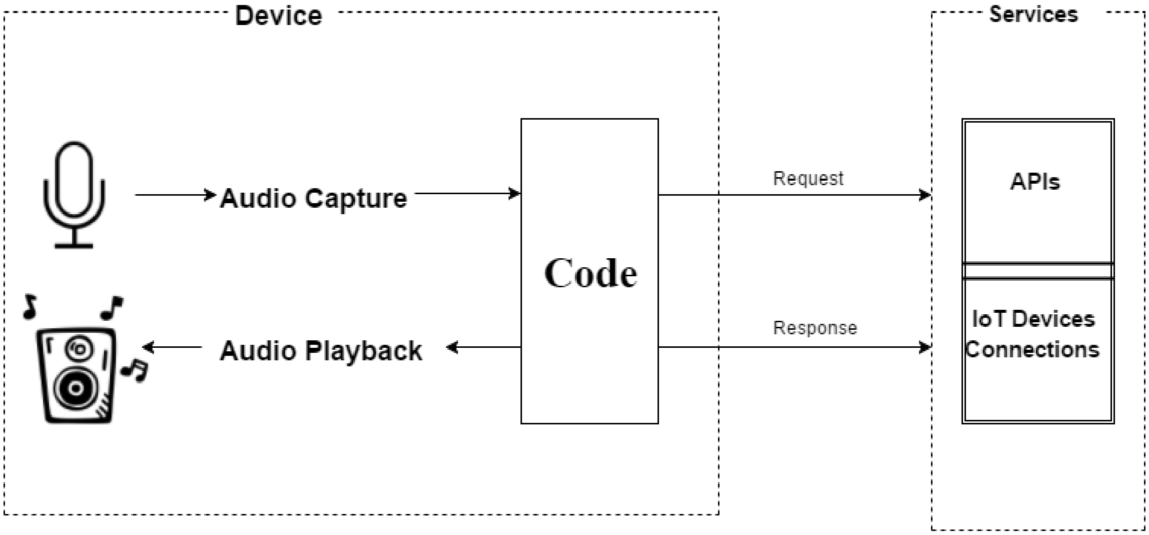


Figure 4.1: System Architecture

System architecture consists of 4 phases, namely; Data collection in the form of voice, voice analysis and conversion to text, data storage and processing, and generating

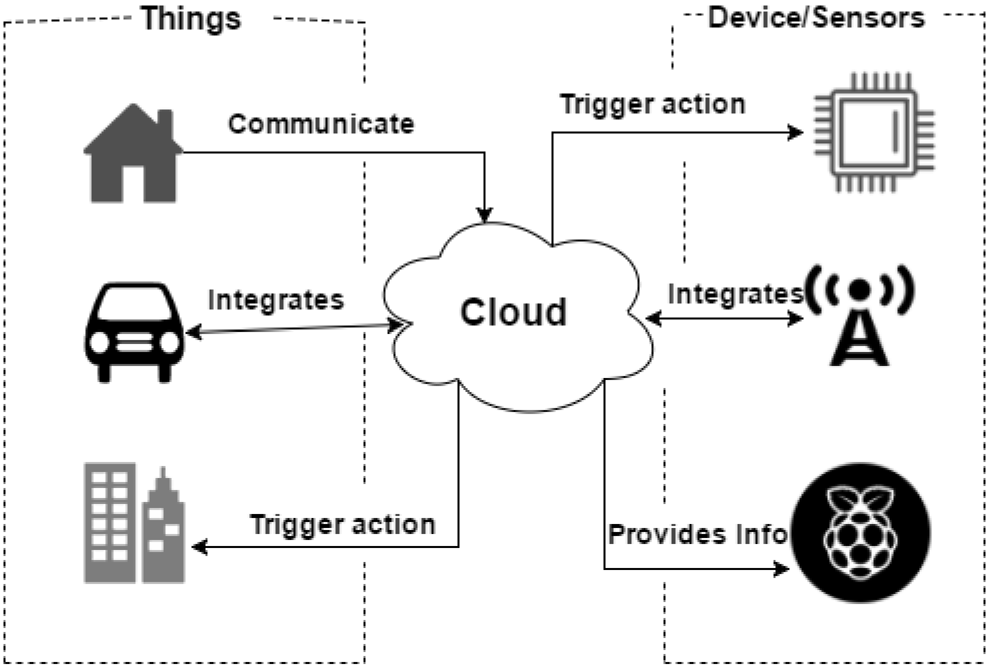


Figure 4.2: IoT architrcture

speech from the processed text output. Now, in Data collection in the form of voice, the data is stored as input for next step.

In IoT box, tasks are divided in three modules and data transmission takes place through nRF24L01 transceiver. Task modules used in IoT box are

- Servo Motor for mechanical movement for object, in our case: Curtains.
- Switching on and off Neo-pixel ring.
- LED display : Displays the data.

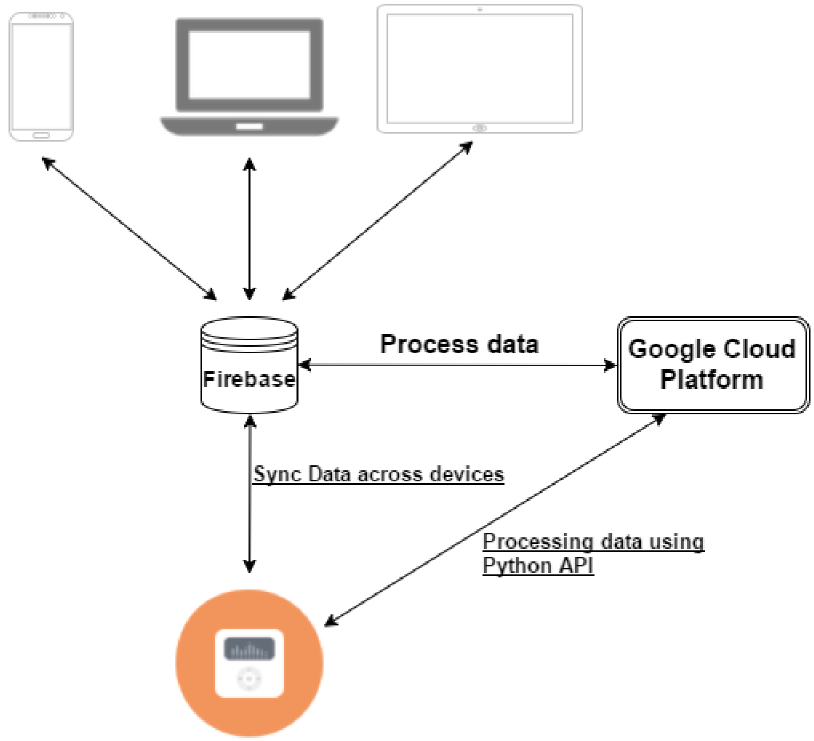


Figure 4.3: Firebase cloud server architecture

Android application helps the user to share his personalized data through android application with ease from anywhere. The data transfer and processing is done through network adapter inbuilt APIs. This data generated is stored in firebase cloud storage and is available for the main system to access. All these tasks are being done in parallel to each other.

All the data stored in Firebase cloud DB is accessible to main system and can be retrieved and process as per required. The need of given data is also processed in parallel to continuous fetching of data from the server.

The input voice is continuously processed and converted to text using STT along with the identifying and processing the commands using Python Script in background. The output then generated is converted from simple text to speech using TTS.

4.2 UML DIAGRAMS

4.3 DEFINITION

The Unified Modelling Language (UML) is a general purpose,developmental, modelling language in the field of software engineering, that is intended to provide a standard way to visualize the design of a system. UML was originally motivated by the desire to standardize the disparate notational systems and approaches to software design.

4.4 DESIGN

The Unified Modelling Language (UML) offers a way to visualize a system's architectural blueprints in a diagram including elements such as:

- Any activities.
- Individual components of the system: And how they can interact with other software components.
- How the system will run.
- How entities interact with others (components and interfaces)
- External user interface

Although originally intended solely for object oriented design documentation, the Unified Modelling Language (UML) has been extended to cover a larger set of design documentation.

4.5 UML SYSTEM MODEL

- **Static (or structural) view:**
  Emphasizes the static structure of the system using objects, attributes, operations and relationships. The structural view includes class diagrams and composite structure diagrams.
- **Dynamic (or behavioural) view:**
  Emphasizes the dynamic behaviour of the system by showing collaborations among objects and changes to the internal states of objects. This view includes sequence diagrams, activity diagrams and state machine diagrams.

4.6 UML MODELS

(a) **Use case diagram:**
To model a system, the most important aspect is to capture the dynamic behaviour of the system. Hence, use case diagrams are used for designing of the Dynamic nature of the system i.e. nature of the system when it is operating. Use case diagrams consists of actors, use cases and their relationships.

The purposes of use case diagrams can be as follows:

- Used to gather requirements of a system.
- Used to get an outside view of a system.
- Identify external and internal factors influencing the system.
- Show the interacting among the requirements and actors.

Places where use case diagrams are used:

- Requirement analysis and high level design.
- Model the context of a system.
- Reverse engineering.
- Forward engineering.

(b) **Deployment Diagram:**
Deployment diagrams are used to picture the topology of the physical components i.e. Hardware components of a system where the software components are deployed. Hence, deployment diagrams are used to sketch the static deployment view of a system. It consists of nodes and their relationships.

The purpose of deployment diagrams can be described as:

- Visualize hardware topology of a system.
- Describe the hardware components used to deploy software components.
- Describe run-time processing nodes.

Usage of deployment diagrams can be described as follows:

- To model the hardware topology of a system.
- To model embedded system.
- To model hardware details for a client/server system.
- To model hardware details of a distributed application.
- Forward and reverse engineering.

(c) **Activity Diagram:**
Activity diagram is a flow chart to represent the flow of one activity to another activity i.e. an operation of the system. So the control flow is drawn from one operation to another. This flow can be sequential, branched or concurrent. Activity diagrams deals with all type of flow control by using different elements like fork, join etc.

So the purposes can be described as:

- Draw the activity flow of a system.
- Describe the sequence from one activity to another.
- Describe the parallel, branched and concurrent flow of the system.

Following are the main usages of activity diagram:

- Modelling work flow by using activities.
- Modelling business requirements.
- High level understanding of the system's functionality.
- Investigate business requirements at a later stage.

(d) **Sequence Diagram:**
The sequence diagram is used to describe some type of interactions among the different elements in the model. So this interaction is a part of dynamic behaviour of the system. Sequence diagram emphasizes on time sequence of data flow across the system.

Purposes of sequence diagram can be described as:

- To capture dynamic behaviour of a system.
- To describe the message flow in the system.
- To describe structural organization of the objects.
- To describe interaction among objects.

Following are the usages of sequence diagrams:

- To model flow of control by time sequence.
- To model flow of control by structural organizations.
- For forward engineering.
- For reverse engineering.

(e) **State Chart Diagram:**
The name itself clarifies the purpose of the diagram and other details. It describes different states of a component in a system. The states are specific to a component/object of a system. A State-chart diagram describes a state machine. state machine can be defined as a machine which defines different states of an object and these states are controlled by external or internal events.

Following are the main purposes of using State-chart diagrams:

- To model dynamic aspect of a system.
- To model life time of a reactive system.
- To describe different states of an object during its life time.
- Define a state machine to model states of an object.

The main usages of state-chart diagrams are as follows:

- To model object states of a system.
- To model reactive system. Reactive system consists of reactive objects.
- To identify events responsible for state changes.
- Forward and reverse engineering.

(f) **Class Diagram:**
The class diagram is a static diagram. It represents the static view of an application. Class diagram is used for visualizing, describing, documenting different aspects of a system and for constructing executable code of the software application. The class diagram shows a collection of classes, interfaces, associations, collaborations and constraints. It is also known as a structural diagram.

Purpose of the class diagram can be summarized as:

- Analysis and design of the static view of an application.
- Describe responsibilities of a system.
- Base for component and deployment diagrams.
- Forward and reverse engineering.

Class diagrams are used for:

- Describing the static view of the system.
- Showing the collaboration among the elements of the static view.
- Describing the functionality performed by the system.
- Construction of software applications using object oriented languages.

4.6.1 USE CASE DIAGRAM

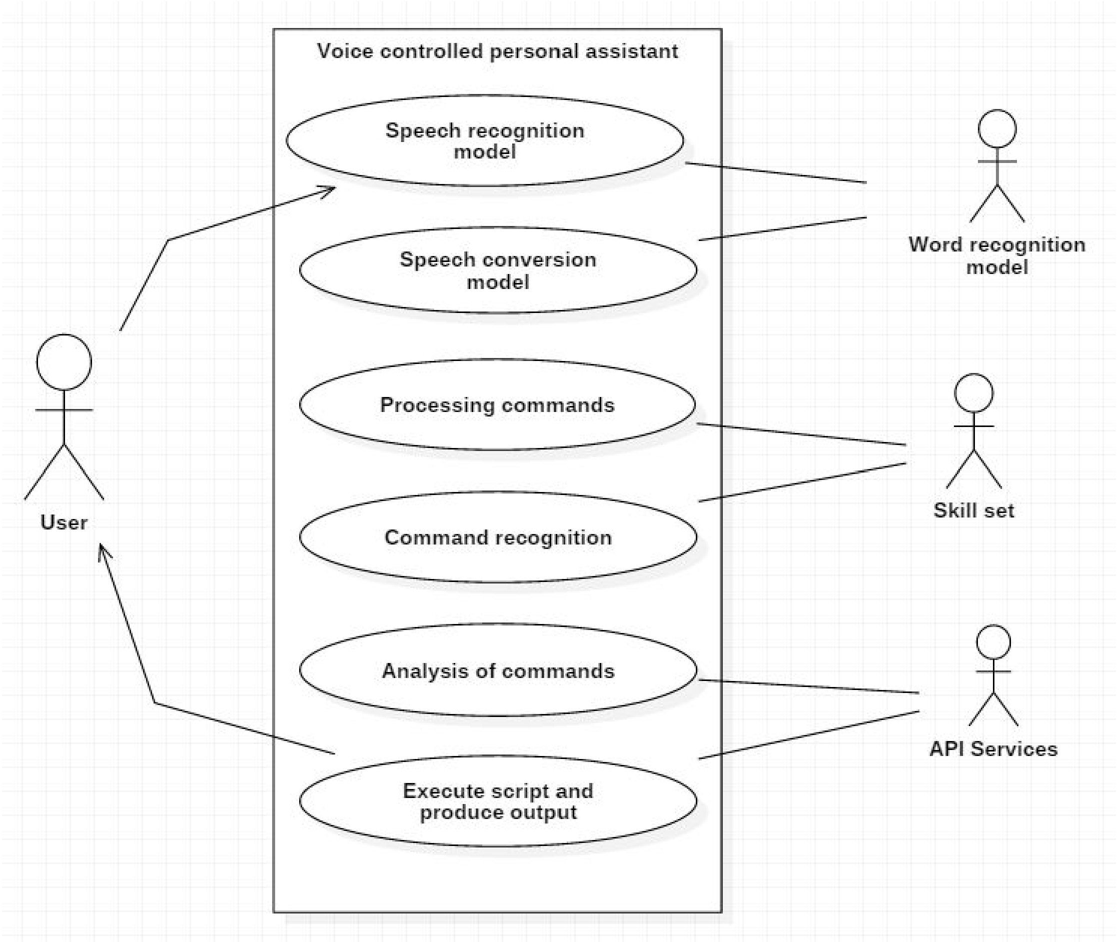


Figure 4.4: Use Case Diagram

4.6.2 CLASS DIAGRAM

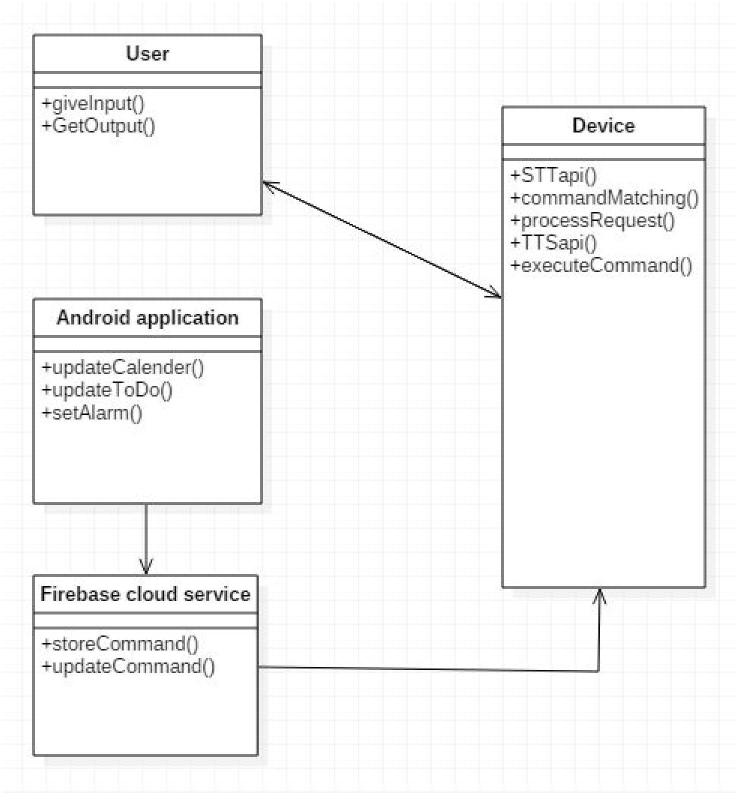


Figure 4.5: Class Diagram

4.6.3 ACTIVITY DIAGRAM

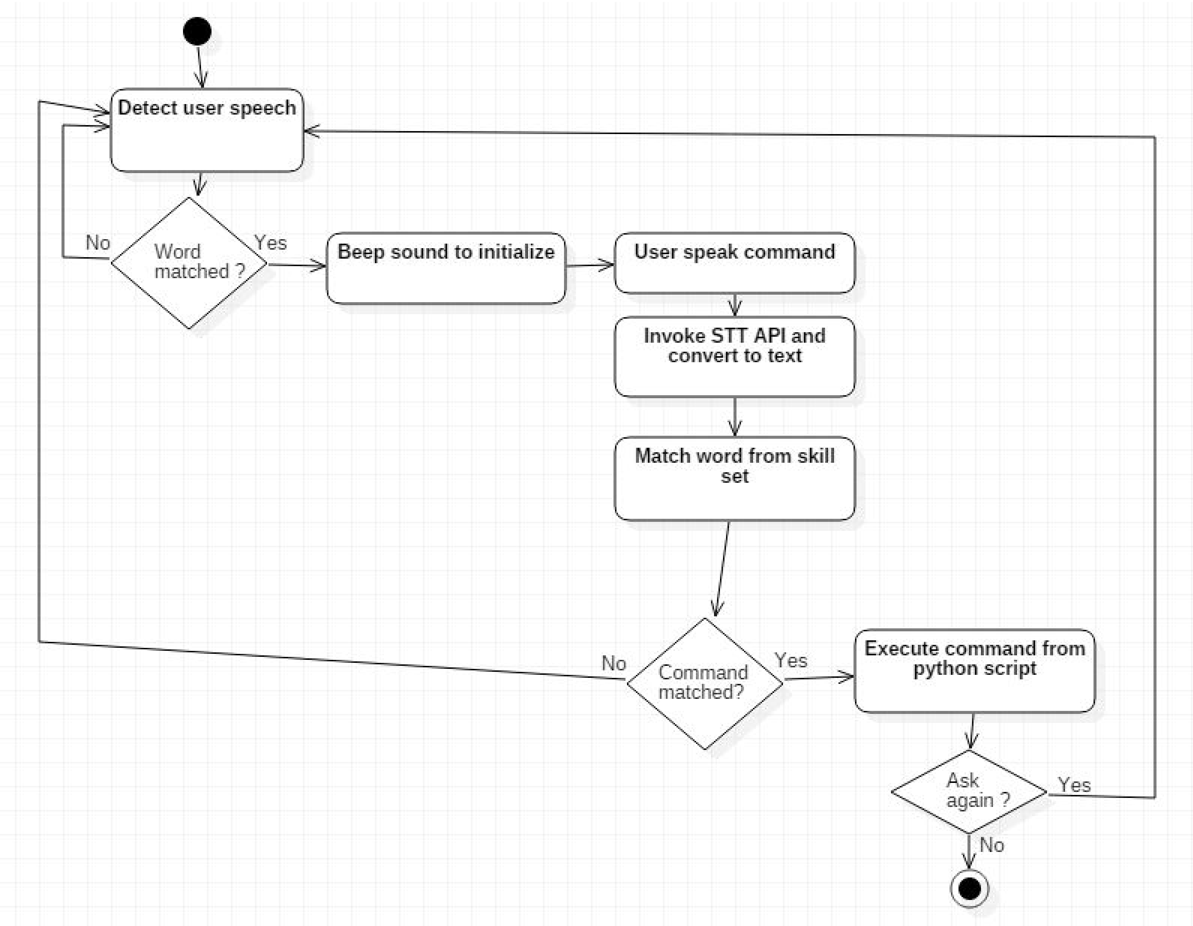


Figure 4.6: Activity Diagram

4.6.4 DEPLOYMENT DIAGRAM

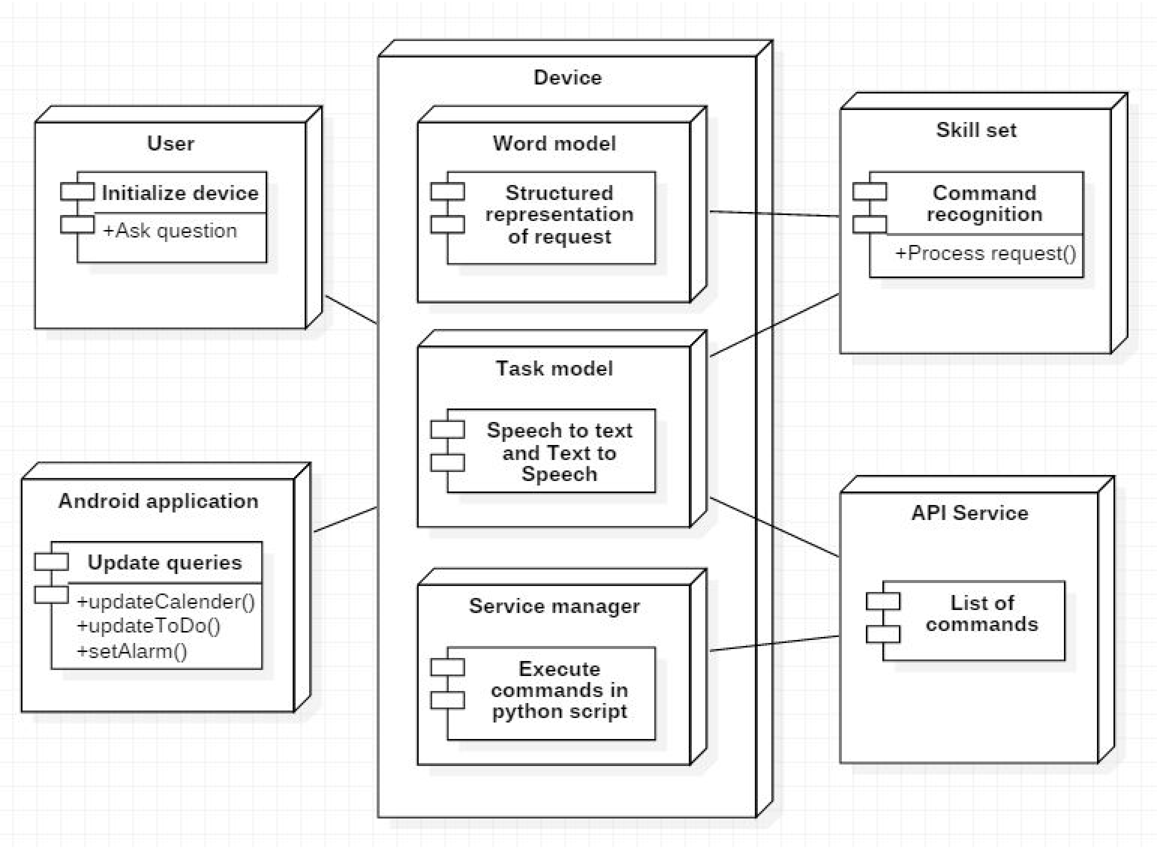


Figure 4.7: Deployment Diagram

4.6.5 SEQUENCE DIAGRAM

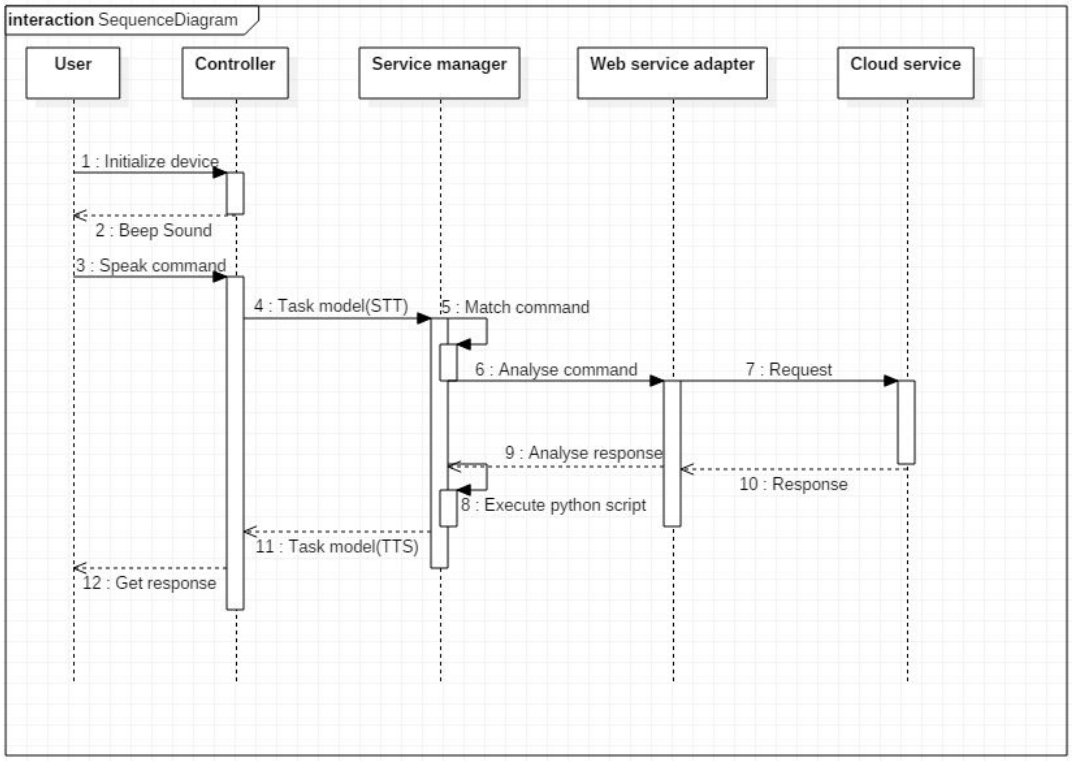


Figure 4.8: Sequence Diagram

4.6.6 STATE CHART DIAGRAM

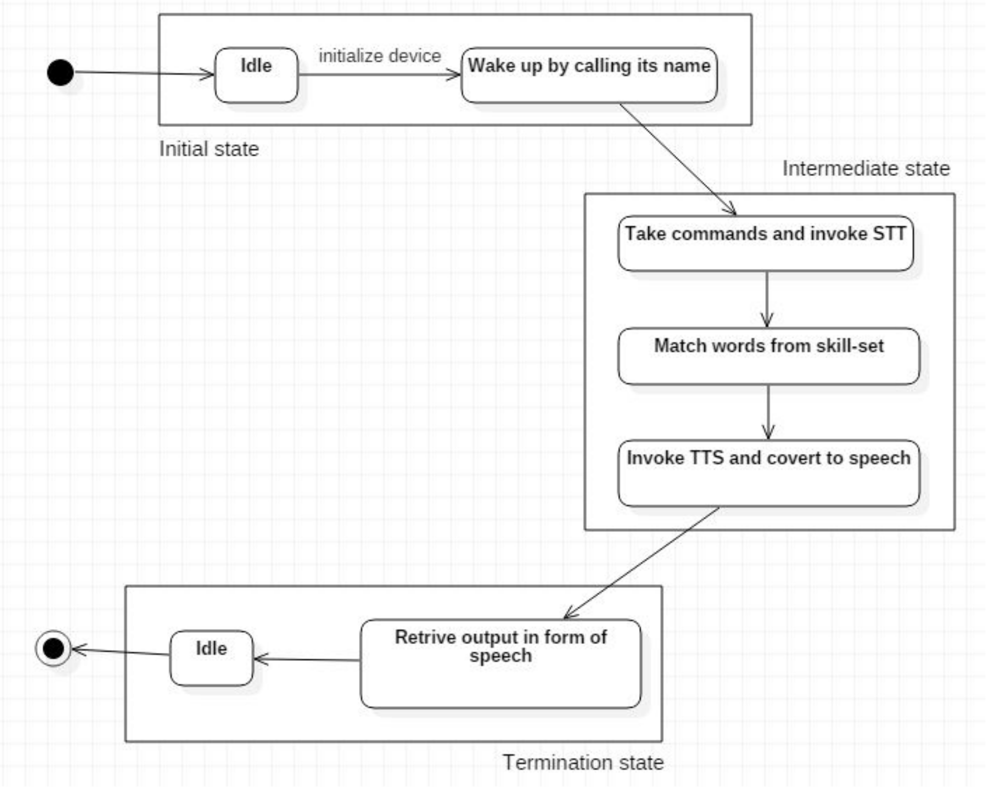

Idle
initialize device
Wake up by calling its name
Initial state
Intermediate state
Take commands and invoke STT
Match words from skill-set
Invoke TTS and covert to speech
Idle
Retrive output in form of speech
Termination state

Figure 4.9: State Chart Diagram

Chapter 5

TECHNICAL SPECIFICATION

5.1 ADVANTAGES

- User Friendly interface.
- Hands-free experience to interact between humans and computers.
- It is portable and easy to use.
- Connectivity with multiple devices.
- Good interface and accessibility.

5.2 DISADVANTAGES

- Primary disadvantage is that it is network dependent since it uses API's from different providers to fetch information.
- It works only with English language.
- One of the constraints is background noise in fetching commands through voice.
- Data flow is unidirectional as Firebase cloud server only sends the data and does not receive any type of data.
- The device is not designed for a particular voice and hence it can be less secure since anyone can use it.

5.3 APPLICATIONS

- Source of entertainment and information for blind/visually impaired.
- Voice based calculator can be used to teach visually impaired students or it can be a game for visually sound students.
- Voice based spell corrector.

Chapter 6
RESULTS

6.1 EXPECTED RESULTS

The project system is supposed to have voice commands as inputs, voice analyser that converts the speech to text, a processor that analyses this text based commands and generate output as the required action performed. There should be minimum noise in the input so as to avoid any error and make the system more efficient hence must have good noise cancellation. The speech to text engine is expected to work in English robustly. The system should respond to users request every time and in case of failure, should report the same. The android application should let the user add data such as calendar entries, set alarm, or even reminders. This data should be sent over to system cloud server and help the system keep records by using both input medium. The system should be able to implement home automation using IoT devices.

Chapter 7

CONCLUSION

The developed system will have the following phases:

- Data collection in the form of voice
- Voice analysis and conversion to text
- Data storage and processing
- Generating speech from the processed text output.

The data generated at every phase can further be used to find patterns and suggest user later. This can be a major base for artificial intelligence machines that learns and understand users.

Thus, on the basis of literature survey and by analysing the existing system, we have come to a conclusion that the proposed system will not only ease to interact with the other systems and modules but also keeps us organised.

REFERENCES

[1] Douglas OShaughnessy, Senior Member, IEEE
Interacting With Computers by Voice: Automatic Speech Recognition and Synthesis

[2] Kei Hashimoto, Junichi Yamagishi, William Byrne, Simon King, Keiichi Tokuda
Analysis Of Machine Translation Aand Speech Synthesis In Speech-To-Speech Translation System
Nagoya Institute of Technology, Department of Computer Science and Engineering, Japan
University of Edinburgh, Centre for Speech Technology Research, United Kingdom
Cambridge University, Engineering Department, United Kingdom, ISSN 5108-5111

[3] Nil Goksel-Canbek ,Mehmet Emin Mutlu
On the track of Artificial Intelligence: Learning with Intelligent Personal Assistants
ISSN:1303-5134

[4] Harshita Phatnani, Mr. Jyotiprakash Patra and Ankit Sharma
Chatbot Assisting: Siri
E-ISSN22498974

[5] Harshita Phatnani, Mr. Jyotiprakash Patra, Ankit Sharma
An Intelligent Voice Assistant Using Android Platform

[6] Vinay Sagar, Kusuma SM
Home Automation Using IOT
ISSN 2395-0056.

[7] Prof Gokilavani R, Navaneethan S
Raspberry Pi Based Robot With Cloud Technology
ISSN 2321 3361

[8] Cheah Wai Zhao, Jayanand Jegatheesan, Son Chee Loon
Exploring IOT Application Using Raspberry Pi

[9] Abhimanyu Chopra, Abhinav Prashar, Chandresh Sain
Natural Language Processing
ISSN 2347-4289

[10] G. Hemalatha, C.P. Sumathi
A Study Of Techniques For Facial Detection and Expression Classification

Appendix A

GLOSSARY

STT	Speech-to-Text
TTS	Text-to-Speech
NLTK	Natural language toolkit
API	Application Programming Interface
JDK	Java Development Kit
GUI	Graphical User Interface
UML	Unified Modelling Language
DB	DataBase
IOT	Internet Of Things
LED	Light Emitting Diode
XML	Extensible Markup Language
IPA	Intelligent Personal Assistant
ASR	Automatic Speech Recognition

Appendix B
ASSIGNMENTS

LAB ASSIGNMENT 01

Aim

Refer Chapter 7 of first reference to develop the problem under consideration and justify feasibility using concepts of knowledge canvas and IDEA Matrix.

Project

Voice Controlled Personal Assistant Device and Connecting IOT Devices

Table B.1: Project Canvas

Purpose	**Goals**	**Users**
To reduce the human efforts	To produce precise desired output with given minimal input and greater time-efficiency	All Age-Group Humans
To Connect IOT Devices in the vicinity	To interface the device with the IOT devices for home automation	Industries and Enterprises

Actions	**Deliverables**	**Risks**
To define functionality of device along with respective vocal commands	SRS & Project Design	Features should work in harmony and both time and space efficiency
Work on different Modules of Project to build an integrated system	Project Report & Project	Device may not be able to perform the service according to command

Milestones	**Constraints**	**Scope**
Synopsis, Abstract, Report	All Functionality well defined with all possible voice commands which	Commands limited to only English
Coding and Final Software	Device should work in any environment	Limited to quality of the vocal commands

LAB ASSIGNMENT 02

Aim

Project problem statement feasibility assessment using NP-Hard, NP-Complete or satisfy ability issues using modern algebra and/or relevant mathematical models.

Feasibility Theory

The feasibility of the project can be defined as the measure of our project whether it is viable or not.It includes various different types of feasibility as follows:

- **Performance:**
  In this we check whether the proposed system is capable of performing all the functional requirements as mentioned in system features in SRS.If our system is performing the functional requirements appropriately then it's performance is feasible.Here we also check the accuracy and efficiency of the system based on various STT-TTS Engines.
- **Technical:**
  In this we check whether the technical specification provided that is hardware and software requirements are minimum requirements for our platform to run successfully without any error regarding the system configuration. Also the voice commands are suffice to call a predefined functionality module and features are performed effectively.
- **Economical:**
  In this we check the cost per line of code, also the cost for storage of hardware required and cost related to the run time of the system.Apart from this since no large database transactions are needed apart from minimum system data which will be required for profilication.

Feasibility on basis of Class of Problem

Complexity classes are one way to talk about how difficult or easy a problem is. Complexity theory gets very technical but the basics are actually extraordinarily intuitive, and it's possible to understand the P versus NP issue with very little math background.

If there is a fast solution to the search version of a problem then the problem is said to be Polynomial time, or P for short. If there is a fast solution to the verification version of a problem then the problem is said to be Non deterministic Polynomial time, or NP for short. The question of ”P=NP” is then the question of whether these sets are identical.

Some problems can be translated into one another in such a way that a fast solution to one problem would automatically give us a fast solution to the other. There are some problems that every single problem in NP can be translated into, and a fast solution to such a problem would automatically give us a fast solution to every problem in NP. This group of problems are known as NP Hard.Some problems in NP Hard are actually not themselves in NP the group of problems that are in both NP and NP Hard is called NP Complete

Classes of problems

- **NP**
  A lot of programs that don't (necessarily) run in polynomial time on a regular computer, but do run in polynomial time on a non deterministic Turing machine. These programs solve problems in NP, which stands for non deterministic polynomial time.An equivalent way to define NP is by pointing to the problems that can be verified in polynomial time.

- **NP Hard**
  If a problem is NP hard, this means I can reduce any problem in NP to that problem. This means if I can solve that problem, I can easily solve any problem in NP. If we could solve an NP hard problem in polynomial time, this would prove P = NP.

- **NP Complete**
  A problem is NP complete if the problem is both NP hard, and in NP.

Our system satisfies not only problems of bringing different services under one platform as well as we are adding many functionality which are not provided by other personal assistant devices and connecting IOT devices as well as extending usability by providing and Android application as well.

Since we have a fast solution to convert the speech into a recognizable functionality it is said to be P type problem capable to solve a part in polynomial time also the servicing the functionality doesn't has a fast solution so it is not a NP complete problem.

Relation Between Classes of Problems

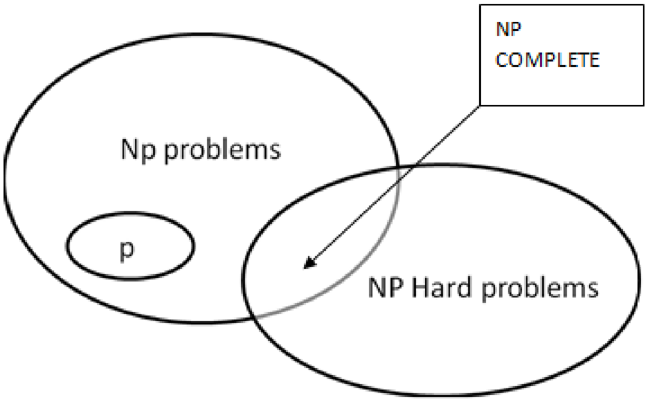


Figure B.1: Relation between Classes of Problems

Mathematical Model

Mathematical Formulation:
Let the M i the universal states which contains,
M = {Q, S, F, Q1 , Qf }
where,

Q = No. of states {Q1,Q2,Q3,Q4,Q5,Q6,Q7,Q8}
X = No. of states {X1, X2, X3, X4}
Q1 = Initial State.
S = Success state.
F = Failure state.

where,

Q1= Start.

Q2 = Initialize assistant by calling action word.

Q3 = Once initialized, give input in the speech format.

Q4 = Processing of speech into text.

Q5 = Text is compared with commands.

Q6 = Perform action according to command.

Q7 = Convert the action into speech.

Q8 = Give output in the speech format.

X1 = Connect Personal AI Device to mobile application.

X2 = Update To-Do list, Calender, Alarm, reminder, etc.

X3 = Updated data is sent to cloud server.

X4 = According to the data sent, AI Device will take action.

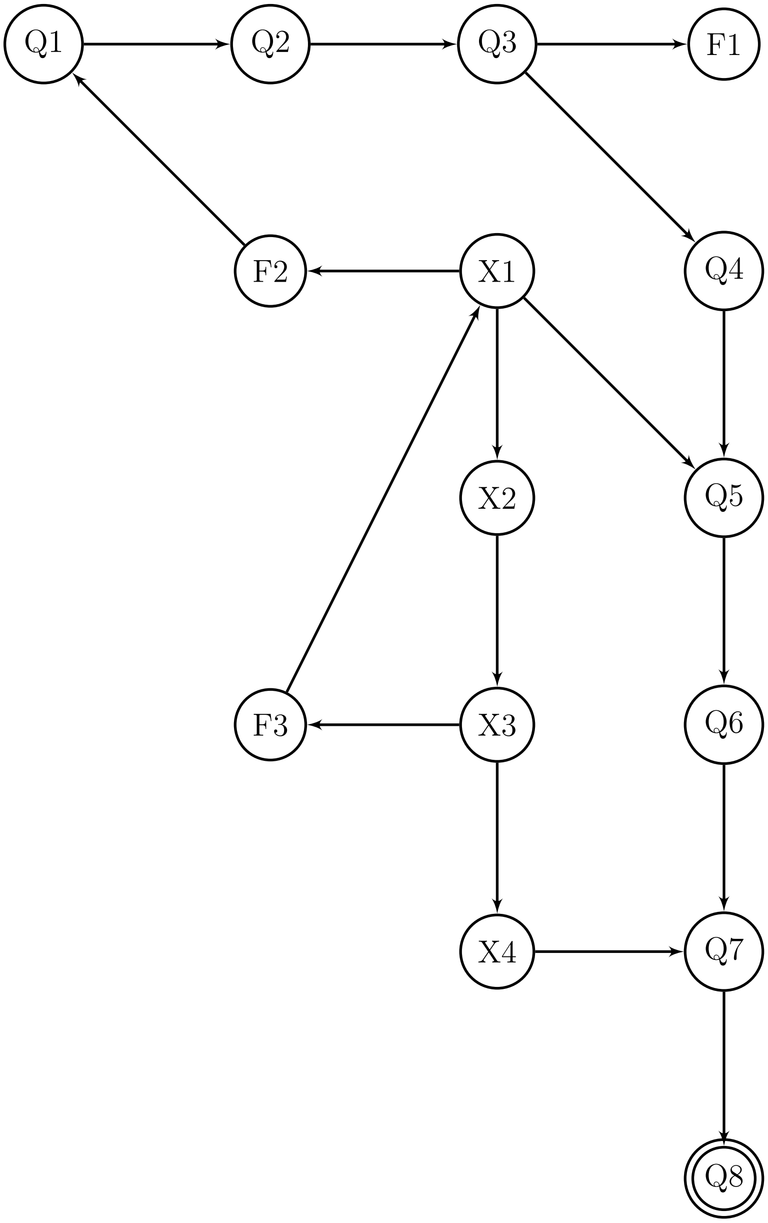
Failure Conditions: F = {F1,F2,F3}

- F1 = Failure if the device is not initialised due to noisy background.
- F2 = Connection with android app failed.
- F3 = Server down failure or cannot retrieve data from server.

Success Conditions: S = {S1}

- S1 = Success after the data is assimilated and output is produced accordingly.

Activity Diagram



Lab Assignment 03

Aim

Use of divide and conquer strategies to exploit distributed/parallel/concurrent processing of the above to identify objects, morphisms, overloading in functions (if any), and functional relations and any other dependencies (as per requirements).

Concept

A divide and conquer algorithm works by recursively breaking down a problem into two or more sub-problems of the same (or related) type (divide), until these become simple enough to be solved directly (conquer). So have divided our problem based on modules used and those are:

- IOT box
- Android Smart-phone application
- Voice input and processing.
- Get data from Firebase cloud DB.
- Generating speech using TTS task model.

Divide and conquer (D&C) is an algorithm design paradigm based on multi-branched recursion. So we have to recursively divide our problem into sub-problems of the same (or related) type (divide), until these sub-problem become simple enough to be solved directly (conquer). The solutions to the sub-problems are then combined to give a solution to the original problem.

In our project we have divided our project into 4 phases which are : Data collection in the form of voice, voice analysis and conversion to text, data storage and processing, and generating speech from the processed text output. Now, in Data collection in the form of voice, the data is stored as input for next step.

In IoT box, tasks are divided in three modules and data transmission takes place through nRF24L01 transceiver. Task modules used in IoT box are

- Servo Motor for mechanical movement for object, in our case: Curtains.
- Switching on and off Neo-pixel ring.
- LED display : Displays the data.

Android application helps the user to share his personalized data through android application with ease from anywhere. The data transfer and processing is done through network adapter inbuilt APIs. This data generated is stored in firebase cloud storage and is available for the main system to access. All these tasks are being done in parallel to each other.

All the data stored in Firebase cloud server is accessible to main system and can be retrieved and process as per required. The need of given data is also processed in parallel to continuous fetching of data from the server.

The input voice is continuously processed and converted to text using STT along with the identifying and processing the commands using Python Script in background. The output then generated is converted from simple text to speech using TTS.

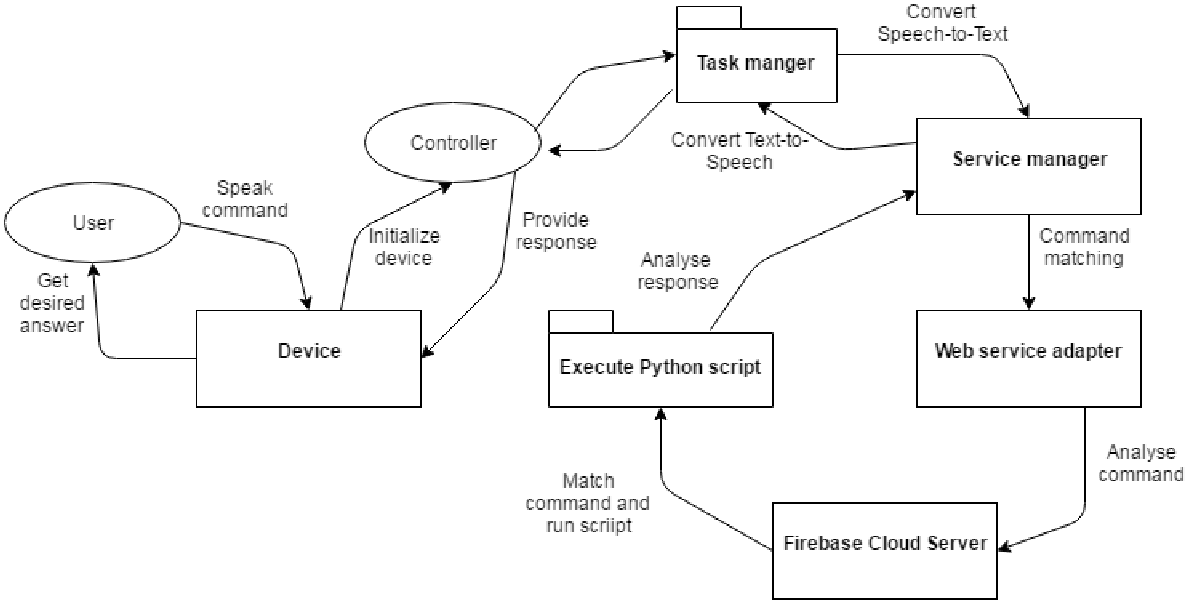


Figure B.2: Data-flow Diagram

Lab Assignment 04

Aim

Use of above to draw functional dependency graphs and relevant Software modelling methods, techniques including UML diagrams or other necessities using appropriate tools.

USE CASE DIAGRAM

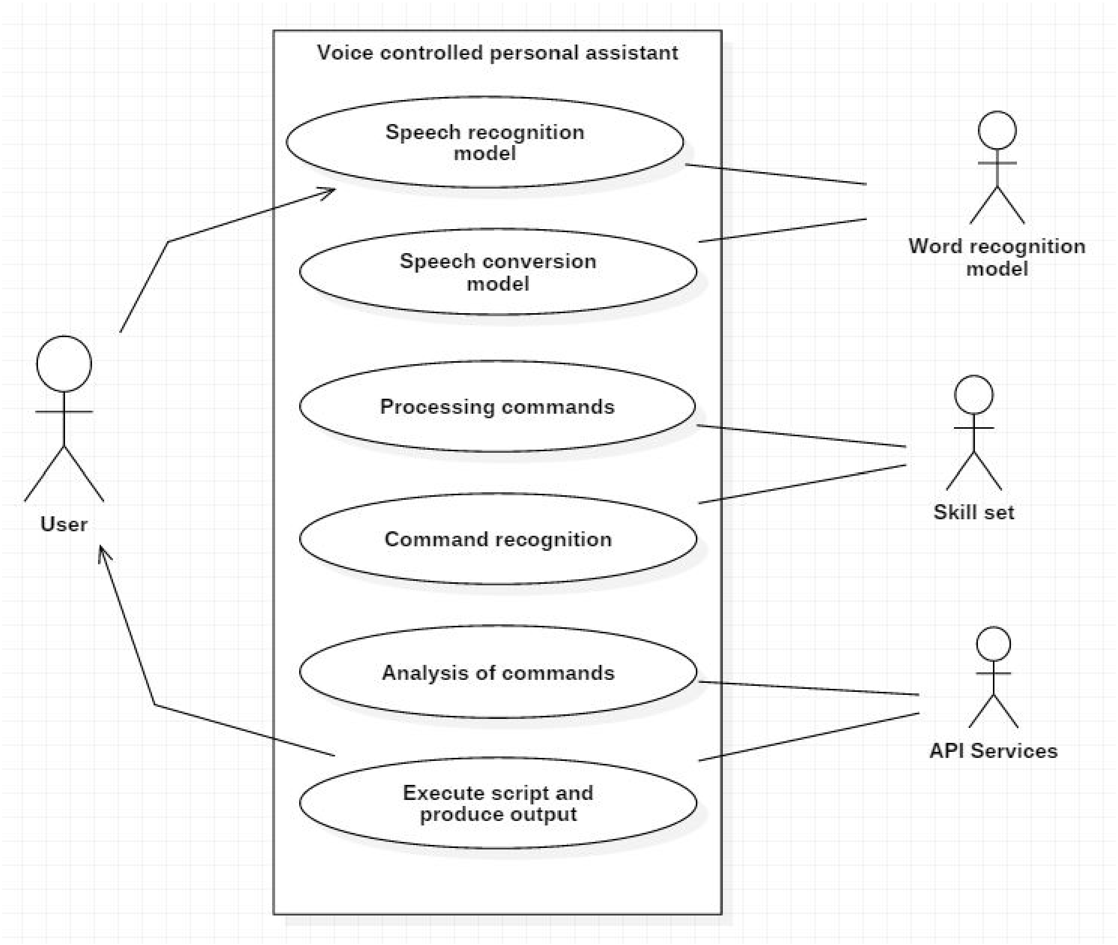


Figure B.3: Use Case Diagram

CLASS DIAGRAM

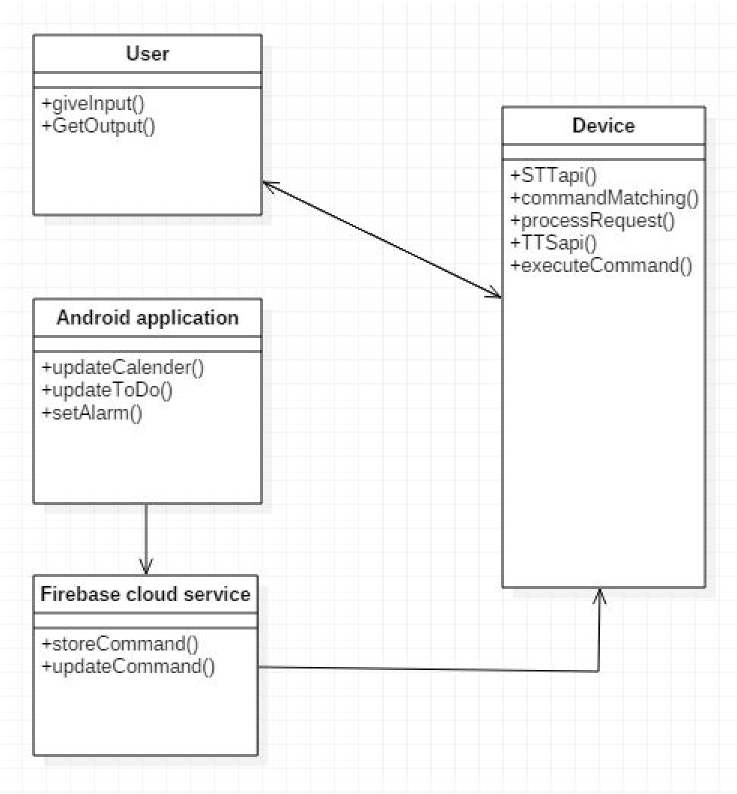


Figure B.4: Class Diagram

ACTIVITY DIAGRAM

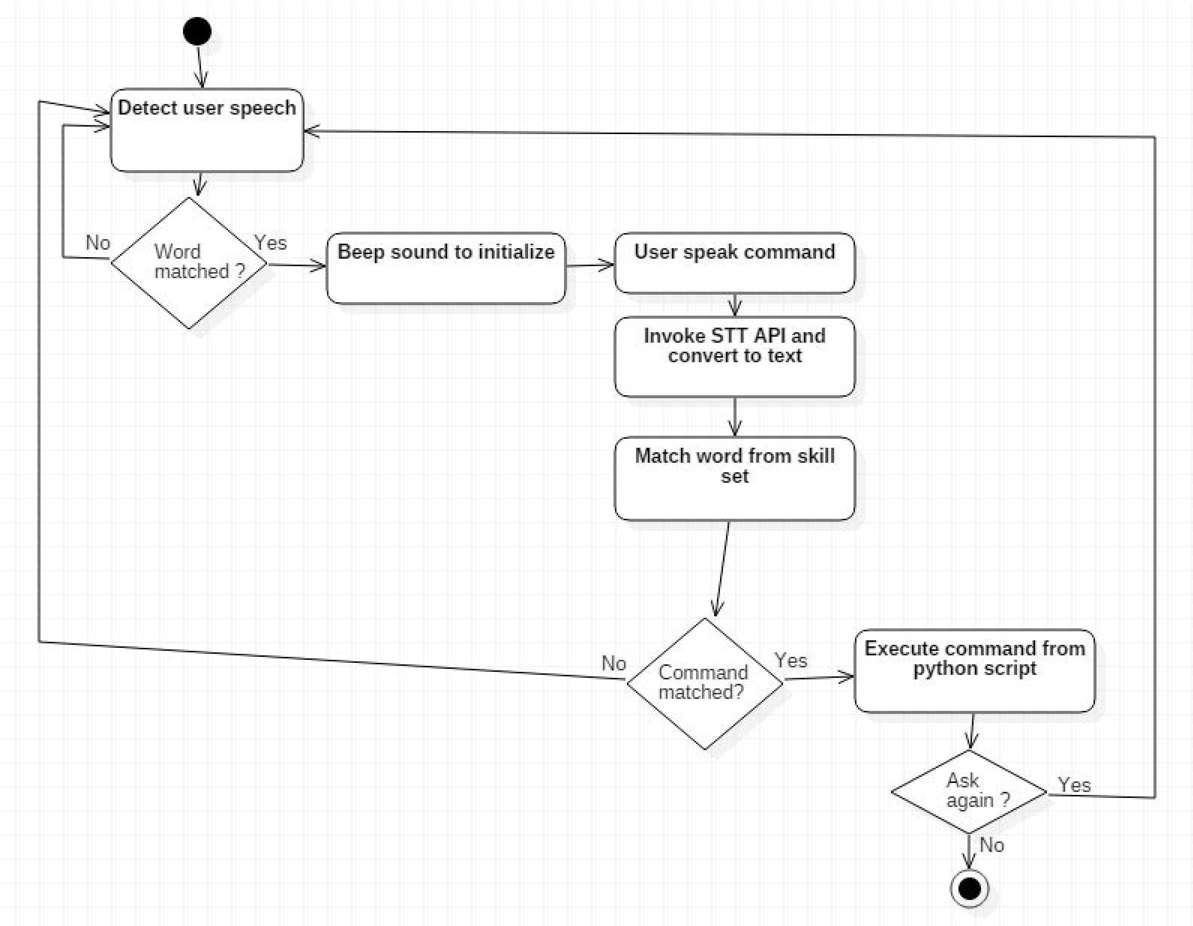


Figure B.5: Activity Diagram

DEPLOYMENT DIAGRAM

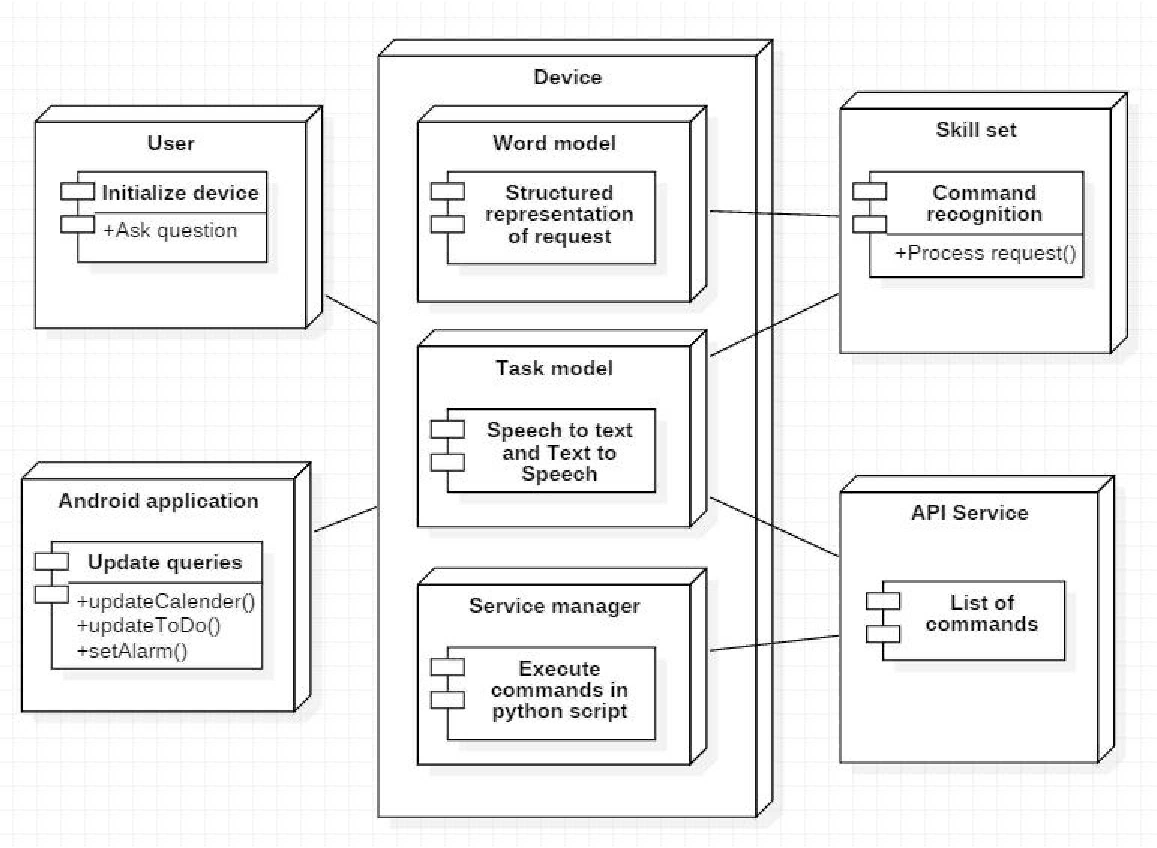


Figure B.6: Deployment Diagram

SEQUENCE DIAGRAM

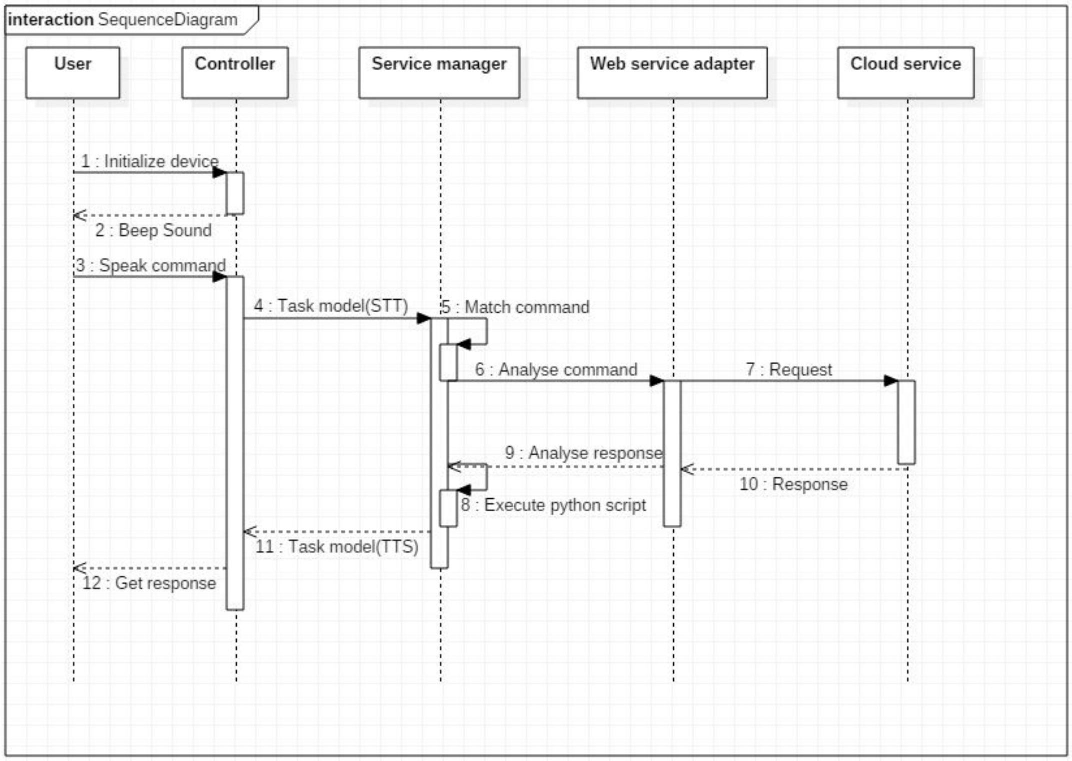


Figure B.7: Sequence Diagram

STATE CHART DIAGRAM

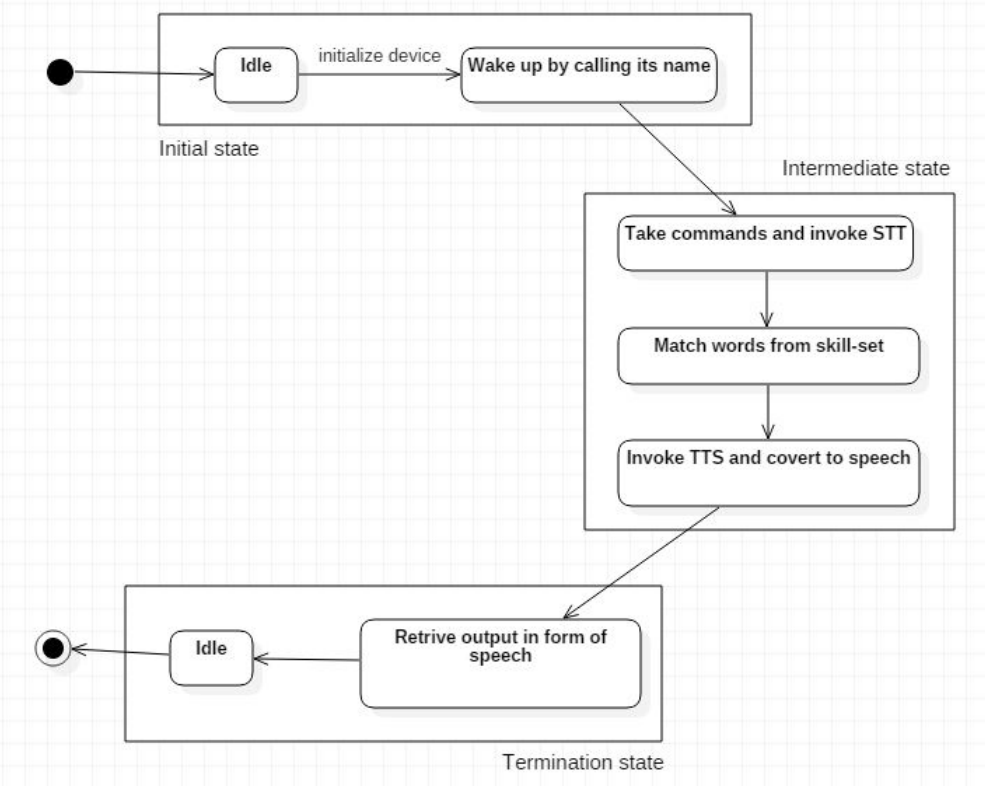

Idle
initialize device
Wake up by calling its name
Initial state
Intermediate state
Take commands and invoke STT
Match words from skill-set
Invoke TTS and covert to speech
Idle
Retrive output in form of speech
Termination state

Figure B.8: State Chart Diagram

Lab Assignment 05

Aim

Testing of project problem statement using generated test data (using mathematical models, GUI, Function testing principles, if any) selection and appropriate use of testing tools, testing of UML diagram's reliability.

Testing

WHAT IS SOFTWARE TESTING?
Software testing is a process of verifying and validating that a software application or program
1. Meets the business and technical requirements that guided its design and development, and
2. Works as expected.

Software testing also identifies important defects, flaws, or errors in the application code that must be fixed.

Software testing has three main purposes: verification, validation, and defect finding.

- The verification process confirms that the software meets its technical specifications. A specification is a description of a function in terms of a measurable output value given a specific input value under specific preconditions. A simple specification may be along the line of a SQL query retrieving data for a single account against the multi-month account-summary table must return these eight fields <list> ordered by month within 3 seconds of submission.

- The validation process confirms that the software meets the business requirements. A simple example of a business requirement is After choosing a branch office name, information about the branchs customer account managers will appear in a new window. The window will present manager identification and summary information about each managers customer base: <list of data elements>. Other requirements provide details on how the data will be summarized, formatted and displayed.

- A defect is a variance between the expected and actual result. The defects ultimate source may be traced to a fault introduced in the specification, design, or development (coding) phases.

Software testing answers questions that development testing and code reviews cant

(a) Does it really work as expected?

(b) Does it meet the users requirements?

(c) Is it what the users expect?

(d) Do the users like it?

(e) Is it compatible with our other systems?

(f) How does it perform?

(g) How does it scale when more users are added?

(h) Which areas need more work?

(i) Is it ready for release?

What can we do with the answers to these questions?

(a) Save time and money by identifying defects early

(b) Avoid or reduce development downtime

(c) Provide better customer service by building a better application

(d) Know that weve satisfied our users requirements

(e) Build a list of desired modifications and enhancements for later versions

(f) Identify and catalog reusable modules and components

(g) Identify areas where programmers and developers need training

The Test Plan

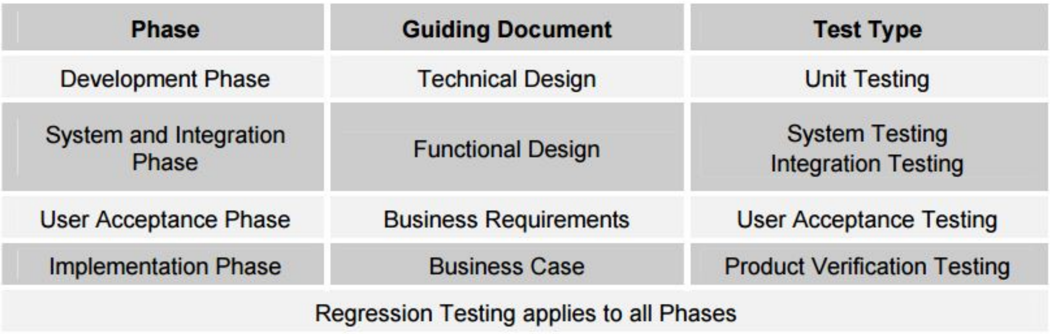
The test plan is a mandatory document. You cant test without one. For simple, straight-forward projects the plan doesnt have to be elaborate but it must address certain items. As identified by the American National Standards Institute and Institute for Electrical and Electronic Engineers Standard 829/1983 for Software Test Documentation, the following components should be covered in a software test plan.

Table B.2: Test Plan

Component	Description	Purpose
Responsibilities	Specific people who are and their assignments	Assigns responsibilities and keeps everyone on track and focused
Assumptions	Code and systems status and availability	Avoids misunderstandings about schedules
Test	Testing scope,schedule,duration, and prioritization	Outlines the entire process and maps specific tests
Communication	Communications plan–who, what, when, how	Everyone knows what they need to know it
Risk Analysis	Critical items that will be tested	Provides focus by identifying areas that are critical for success
Defect Reporting	How defect will be logged and documented	Tells how to document a defect so that it can be reproduced, fixed, and retested
Environment	The technical environment, data, work area, and interfaces used in testing	Reduces or eliminates misunderstandings and sources of potential delay

Types of Software Tests

It describes which testing types we might follow in our testing life cycle.



Phase	Guiding Document	Test Type
Development Phase	Technical Design	Unit Testing
System and Integration Phase	Functional Design	System Testing Integration Testing
User Acceptance Phase	Business Requirements	User Acceptance Testing
Implementation Phase	Business Case	Product Verification Testing
Regression Testing applies to all Phases		

Figure B.9: Types of software testing

- **Black Box Testing**
  Black box testing methods focus on the functional requirements in the software. That is, black box testing enables us to derive sets of input conditions that will

fully exercise.
All functional requirements of the program Black box testing attempts to find errors in the following categories:

- Incorrect or missing function
- Interface errors
- Errors in data structure or external job access
- Performance errors
- Initialization and termination errors.

In the proposed application with the help of this technique, we do not use the code to determine a test suite; rather, knowing the problem that we are trying to solve, we come up with four types of test data:

(a) Easy-to-compute data,

(b) Typical data,

(c) Boundary / extreme data,

(d) Bogus data.

- **Unit Testing**
  Unit testing enables a programmer to detect error in coding. A unit test focuses verification of the smallest unit of software design. This testing was carried out during the coding itself. In this testing step, each module going to be work satisfactorily as the expected output from the module.

  The front end design consists of various forms. They were tested for data acceptance. Similarly, the back-end also tested for successful acceptance and retrieval of data. The unit testing is done on the developed code. Mainly the unit testing is done on modules.

- **System Testing**
  After performing the integration testing, the next step is output testing of the proposed system. No system could be useful if it doesn't produce the required output in a specified format. The outputs generated are displayed by the user. Here the output format is considered in to two ways. One in on screen and other in printed format.

- **Integration Testing**
  Through each program work individually, they should work after linking together. This is referred to as interfacing. Data may be lost across the interface; one module can have adverse effect on the other subroutines after linking may not do the desired function expected by the main routine. Integration testing is the systematic technique for constructing the program structure while at the

same time conducting test to uncover errors associated with the interface. Using integrated test plan prepared in the design phase of the system development as a guide, the integration test was carried out. All the errors found in the system were corrected for the next testing step.

- **User Acceptance Testing**
  User Acceptance Testing is also called Beta testing, application testing, and end-user testing. Whatever you choose to call it, its where testing moves from the hands of the IT department into those of the business users. Software vendors often make extensive use of Beta testing, some more formally than others, because they can get users to do it for free.

  By the time UAT is ready to start, the IT staff has resolved in one way or another all the defects they identified. Regardless of their best efforts, though, they probably dont find all the flaws in the application. A general rule of thumb is that no matter how bulletproof an application seems when it goes into UAT, a user somewhere can still find a sequence of commands that will produce an error.

- **Product verification Testing**
  Production verification testing is a final opportunity to determine if the software is ready for release. Its purpose is to simulate the production cut-over as closely as possible and for a period of time simulate real business activity. As a sort of full dress rehearsal, it should identify anomalies or unexpected changes to existing processes introduced by the new application. For mission critical applications the importance of this testing cannot be overstated.

  The application should be completely removed from the test environment and then completely reinstalled exactly as it will be in the production implementation. Then mock production runs will verify that the existing business process flows, interfaces, and batch processes continue to run correctly. Unlike parallel testing in which the old and new systems are run side-by-side, mock processing may not provide accurate data handling results due to limitations of the testing database or the source data.

Table B.3: Black-Box Test Plan of the Platform

Component	**Description**	**Input**	**Expected Output**
Proper Recognition of Command	To check command in English is recognised by Device as correctly	Vocal Command	Command is recognised by the device
Android Connectivity to Cloud DB	To check that data is transfered properly in both the direction to the Firebase Cloud DB with real-time results	Android App and Firebase Cloud DB	Android Application Updates are recognised by the platform
Performance of IOT Box	To check whether IOT Box performs correctly on the input of concerned commands	IOT Box Related Vocal Commands	IOT Box Works Seamlessly
Performance of Firebase Cloud	To check whether Firebase work with no downtime, real-time services and Provide API form Device and Android application	Firebase Cloud DB	Firebase Cloud Db works as expected with Python and Android API
Response Of AI Device	To check whether the AI Device Vocal response to commands are correct	Vocal Commands along with required parameters	Output vocal command is the correct response to the input command
Integration test of the Platform	To check whether all the modules of the platform is working properly when interfaced together as one	All the modules if the platform	Data flows throughout the platform from one end to another without any errors