SMART SYSTEM CONTROL WITH VOICE COMMAND USING FIREBASE

A PROJECT REPORT

Submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR, ANANTHAPURAMU

In partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING

By

Cheepati Nithin Kumar Reddy	**19BF1A0533**
Goduguchinta Sandeep	**19BF1A0553**
Erepalli Rekhasree	**19BF1A0548**
Boyapati Rahul	**19BF1A0523**

Under the Esteemed Guidance of

S. Mrudula, M.Tech

Assistant Professor

Department of Computer Science and Engineering

SRI VENKATESWARA COLLEGE OF ENGINEERING

Affiliated to JNTUA & Approved by AICTE Recognized under Sections 2(f) & 12(B) of UG act 1956.Accredited by NBA, New Delhi & NAAC Bangalore with ‘A’ Grade, Tirupati-517507, A.P.

2019-2023

SRI VENKATESWARA COLLEGE OF ENGINEERING

Affiliated to JNTUA & Approved by AICTE Recognized under Sections 2(f) & 12(B) of UGC act 1956. Accredited by NBA, New Delhi & NAAC Bangalore with 'A' Grade, Tirupati-517507, A.P.

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the project entitled **"SMART SYSTEM CONTROL WITH VOICE COMMAND USING FIREBASE"** is a bonafide work done by **"Goduguchinta Sandeep 19BF1A0553, Erepalli Rekhasree 19BF1A0548, Cheepati Nithin Kumar Reddy 19BF5A0533, Boyapati Rahul 19BF1A0523",** in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering, from Jawaharlal Nehru Technological University Anantapur, Ananthapuramu during the year 2022-2023.

Project Guide	**Head of the Department**
S. Mrudula, M. Tech	Dr. K. Santhi,M.Tech, Ph.D (CSE)
Assistant Professor	Professor and Head of the Department
Department of CSE	Department of CSE
SV College of Engineering	SV College of Engineering
Tirupati - 517 507	Tirupati – 517 507

Viva-Voce date: ________________

Internal Examiner **External Examiner**

DECLARATION

We hereby declare that the project report entitled **“SMART SYSTEM CONTROL WITH VOICE COMMAND USNG FIREBASE”** done by us under the esteemed guidance of **S Mrudula, Assistant Professor** and is submitted in partial fulfilment of the requirements for the award of /the Bachelor’ s of Technology in **Computer Science and Engineering.**

Date :	Goduguchinta Sandeep	(19BF1A0553)
Place :	Erepalli Rekhasree	(19BF1A0548)
	Cheepati Nithin Kumar Reddy	(19BF1A0533)
	Boyapati Rahul	(19BF1A0523)

ACKNOWLEDGEMENT

We are thankful to our guide **Miss. S Mrudula, M.Tech, Assistant Professor** for his valuable guidance and encouragement. Her helping attitude and suggestions have helped us in the successful completion of the project.

We are thankful to project coordinator **Mr. Reddicharala Venkata Ramana, M.Tech, Asst Professor** and **Mr. N Narendra**, **Asst Professor** for his guidance and regular schedules.

We would like to express our gratefulness and sincere thanks to **Dr. K. Santhi Ph.D**, **Head of Dept of CSE**, for her kind help and encouragement during the course of our study and in the successful completion of the project work.

We have great pleasure in expressing our hearty thanks to our beloved **Principal Dr. N. Sudhakar Reddy** for spending his valuable time with us to complete this project.

Successful completion of any project cannot be done without proper support and encouragement. We sincerely thank the **Management** for providing all the necessary facilities during the course of study.

We would like to thank our parents and friends, who have the greatest contributions in all our achievements, for the great care and blessings in making us successful in all our endeavours.

Goduguchinta Sandeep	(19BF1A0553)
Erepalli Rekhasree	(19BF1A0548)
Cheepati Nithin Kumar Reddy	(19BF1A0533)
Boyapati Rahul	(19BF1A0523).

TABLE OF CONTENTS

ABSTRACT

This project presents a system for controlling a computer system remotely using voice commands. The system uses a mobile interface built on Scratch, which can capture voice commands and transmitting them to a remote computer via Firebase. The computer system is programmed using Python to execute the commands received from the mobile interface. The proposed system is an improvement over traditional socket-based connections that are limited to a specific range. The system allows for remote control of a computer system from anywhere in the world. This project presents the implementation of the system, experimental results, evaluation metrics, and future work.

KEYWORDS : Voice commands, Mobile Interface, Scratch, Firebase, Python

LIST OF ABBREVIATIONS

1.	PC	Personal Computer
2.	IDE	Integrated Development Environment
3.	IOT	Internet of Things
4.	NLP	Natural Language Processing
5.	WI-FI	Wireless Fidelity
6.	iOS	iPhone Operating System
7.	CLI	Command Line Interface
8.	GUI	Graphical User Interface
9.	URL	Uniform Resource Locator
10.	SMS	Short Message Service
11.	UML	Unified Modeling Language
12.	API	Application Programming Interface
13.	OTP	One Time Password

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

Nowadays, PC's, Laptop's and all other electronic gadgets are inseparable part of our everyday life. Personal computers are not any longer meant for working purpose, but more and more used for entertainment in people's spare time. This is also applicable to the mobile phones, which have transformed into multifunctional devices with almost same features as computer have. Smartphones are common and commercially used device all over the world, user-friendly interface, and lots of features such as Wi-Fi, Internet access, Bluetooth, Camera, Video recording etc. add-on to the Android smartphone to be popular all over the world with cheap cost.

Knowingly, or unknowingly, personal assistants have become an integral part of our lives these days. It is because of all the features and ease of use they provide. Personal Assistants are also capable of automating some day-to-day tasks, so that a user can focus on what matters the most to them. Features like, making calls, writing messages, taking photographs, storing to-dos on the go, browsing internet etc., are offered by personal assistants. So, utilization of these features of a virtual assistant will save an individual a lot of time, and effort. It is important to focus more on what matters the most for an individual, whether it could be personal work, or professional work.

People often spend more time on doing routine tasks, and they can be automated with these types of personal assistants. When someone works in an environment with which he/she is not familiar with, they often find it difficult to locate applications that they need, like browser, any IDE or nay other software. Most of the time, they will end up wasting hours of time, searching for the application alone. This results in unnecessary time wastage. Therefore, a voice enabled personal assistant will help automating this process. User is expected just to give a voice command, and the assistant will take care of the rest. The paper indicates the usage of a Voice enabled personal assistant and it can enable an individual to get things done with voice commands and can save a lot of their time as well.

It is easy to use a computer while a person is in front of it but controlling a system from a distance is more difficult. For example, you can quickly adjust the sound or brightness when watching a movie on a computer from a close distance. But is not possible from distances. For example, you are away from computer while watching movie but if you want to increase sound it is not possible because you not connected with system.

1.2 LITERATURE SURVEY

There are many implementations to the solution applied to Android software stack. It has an open protocol, and it is widely deployed in the open-source community. This solution adapts very well to provide part of the functionality of the architecture, and it will be studied further.

They envisioned that someday computers will recognize natural language and count on what we need, whilst and where we need it, and proactively whole responsibilities on our behalf. However, speech recognition and machine getting to know have persevered to be refined, and based records served through packages and content providers have emerged. We agree with that as computer systems turn out to be smaller and greater ubiquitous [e.g., wearables and Internet of Things (IoT)].

The recognizer is designed to change a verbal articulation from a individual into an alternate method of data (e.g., text). A hand held individual colleague including a voice-recognizer and a characteristic dialect processor is disclosed. This snippet of data can be a plan for the day, data in the individual’s logbook or data from the individual’s address book, Such as a telephone number.

The Most well-known utilization of iPhone is “SIRI “which causes the end client to impart end client versatile with voice and it additionally reacts to the voice charges of the client. It is named as Personal Assistant with Voice Recognition Intelligence, which takes the client contribution to type of voice or content and process it and returns the yield in different structures like activity to be performed or the item is directed to the end client. Furthermore, this proposed framework can change the method for communications between end client and the cell phones.

As virtual assistants move toward becoming more intelligent and the IVA bio-logical community of administrations and gadgets extends, there’s a developing need to comprehend the security and protection dangers from this rising innovation. A few late occurrences feature noteworthy vulnerabilities in IVAs. Better demonstrative testing can uncover such vulnerabilities and prompt more reliable frameworks.

It enables the objective clients to connect with PCs and web based administrations with a wide cluster of usefulness in light of different web administrations and social media. There are four standard parts of the system; the voice recognition module, the natural language processing (NLP) module, conversational agent and the content extraction module. The current screen per client writing computer programs are not fitting for getting to Internet in perspective of the base help they give for web content and the nonattendance of voice affirmation. The Virtual Right hand programming open in the market are not especially given everything and unfit to utilize it similarly. Some may confront issue now too.

Lingyan Bi proposed a novel method to Design a Android based Remote Control System e with JNI Interface for providing convenience for the user.

Michael Spreitzen barth proposed analysis based Smartphone Mobile Malware for forensic Analyses.

Xinfang Lee, presented a novel Android based Forensic System.

Enck, W proposed a secure Android Remote controlling mechanism for performing secure transaction form the Remote location.

T. Richardson proposed a novel method of Internet based Android application to demonstrate working of Internet Computing.

CHAPTER 2

PROJECT DESCRIPTION

2.1 PROBLEM DEFINITION

Personal computers are not any longer meant for working purpose, but more and more used for entertainment in people's spare time. This is also applicable to the mobile phones, which have transformed into multifunctional devices with almost same features as computer have. Smartphones are common and commercially used device all over the world, user-friendly interface, and lots of features such as Wi-Fi, Internet access, Bluetooth, Camera, Video recording etc. add-on to the Android smartphone to be popular all over the world with cheap cost. It is easy to use a computer while a person is in front of it but controlling a system from a distance is more difficult. For example, you can quickly adjust the sound or brightness when watching a movie on a computer from a close distance. But is not possible from distances. For example, you are away from computer while watching movie but if you want to increase sound it is not possible because you not connected with system.

Despite the various benefits provided by speech recognition, the system is also plagued with limitations. By implication, the development of speech recognition applications also inherits these limitations. The existing Voice Assistants use pattern recognition techniques of python which lack in the context, Lack of accuracy, and misinterpretations, Time, costs and productivity, User accents. They operate only on online mode. They store the data in database servers which leads to increase in Time and Space Complexity. Some of them use cloud to store the data which leads to security issues. Background noise interference is also another daunting problem with speech recognition software.

2.2 PROJECT DETAILS

The implemented application consists of two parts, the first one is an application for Android smartphone and the second one is a server application that executes the command selected by user's application.

Application for android smartphone is created using the Scratch and firebase**.**

2.2.1 Scratch

Scratch is an event-driven visual programming language developed by MIT. In Scratch, we can create our own interactive stories, games, and animations using building blocks. In this platform, we do not need to write code to perform operations, things are done just by drag and drop, just like visual basic. It is the best platform to start basic programming by creating attractive animation effects. There are so many features available in Scratch, such as video games, animations, stories, sound, events, etc. It is a free platform created by the Lifelong Kindergarten group at MIT in the Media lab. It is developed in ActionScript and JavaScript and is compatible with any operating system. It has been translated into more than 70 languages and used in most parts of the world.

Uses of Scratch: Scratch is made to learn basics programming concepts with fun. It is a tool for creating interesting games, stories, and more block-based programming. It has its own paint editor and sound builder. Anyone can start learning programming from Scratch even they do not have previous knowledge about programming languages and programming concepts.

Elements of Scratch:

There are the following four main elements of Scratch, and they are:

1. Programming Palette
2. Stage Area
3. Sprites
4. Script

Application using Scratch:

Using Scratch create an application for user which can take voice commands from user and convert it into text store it on firebase for the further assistance when you click on the Jarvis button it calls the speech recognition and take the voice command from the user and store it on firebase.

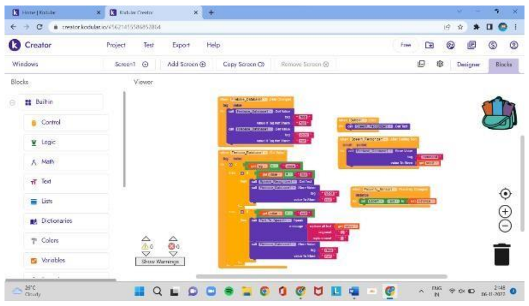


Fig of 2.2.1.1: Scratch interface

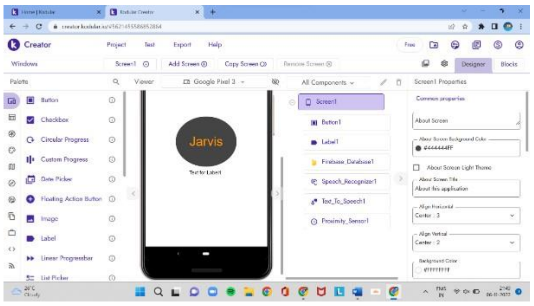


Fig of 2.2.1.2 : Application using scratch

2.2.2 Firebase

Firebase is a product of Google which helps developers to build, manage, and grow their apps easily. It helps developers to build their apps faster and in a more secure way. No programming is required on the firebase side which makes it easy to use its features more efficiently. It provides services to android, iOS, web, and unity. It provides cloud storage. It uses NoSQL for the database for the storage of data.

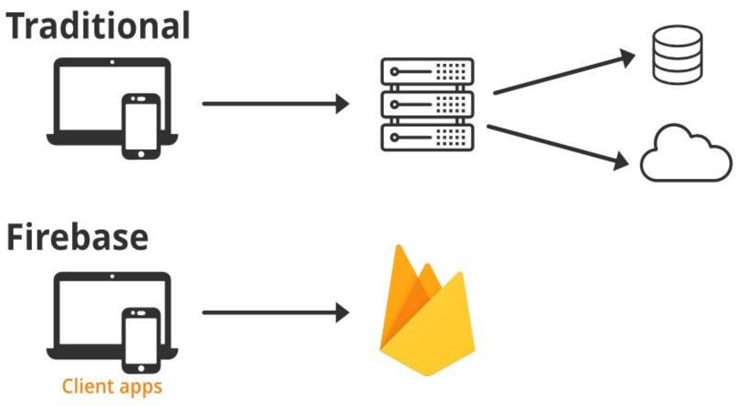


Fig 2.2.2.1: Data handling: Traditional vs Firebase system

2.2.3 Python:

Python is a high-level, general-purpose and a very popular programming language. Python programming language (latest Python 3) is being used in web development, Machine Learning applications, along with all cutting-edge technology in Software Industry. Python Programming Language is very well suited for Beginners, also for experienced programmers with other programming languages like C++ and Java.

This specially designed Python tutorial will help you learn Python Programming Language in most efficient way, with the topics from basics to advanced (like Web-scraping, Django, Deep Learning, etc.) with examples.

Below are some facts about Python Programming Language:

1. Python is currently the most widely used multi-purpose, high-level programming language.
2. Python allows programming in Object-Oriented and Procedural paradigms.
3. Python programs generally are smaller than other programming languages like Java. Programmers must type relatively less and indentation requirement of the language, makes them readable all the time.
4. Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc.
5. The biggest strength of Python is huge collection of standard libraries which can be used for the following:
    - Machine Learning
    - GUI Applications (like Kivy, Tkinter, PyQt etc.)
    - Web frameworks like Django (used by YouTube, Instagram, Dropbox)
    - Image processing (like OpenCV, Pillow)
    - Web scraping (like Scrapy, BeautifulSoup, Selenium)
    - Test frameworks
    - Multimedia
    - Scientific computing
    - Text processing and many more.

List of Modules used in Project:

1.Webbrowser

2.Pyautogui

3.Pywhatkit

4.Firebaseadmin

5.Keyboard

6.Time

7.os

Webbrowser*:*

In Python, *webbrowser module* is a convenient web browser controller. It provides a highlevel interface that allows displaying Web-based documents to users. webbrowser can also be used as a CLI tool. It accepts a URL as the argument with the following optional parameters: -n opens the URL in a new browser window, if possible, and -t opens the URL in a new browser tab.

Pywhatkit:

Python offers numerous inbuilt libraries to ease our work. Among them *pywhatkit* is a Python library for sending WhatsApp messages at a certain time, it has several other features too.

Following are some features of pywhatkit module:

1. Send WhatsApp messages.
2. Play a YouTube video.
3. Perform a Google Search.
4. Get information on a particular topic.

The pywhatkit module can also be used for converting text into handwritten text images.

In Python3 pywhatkit module will not come pre-installed, so you can install it by using the command:pip install pywhatkit

Send Whatsapp Messages:

Syntax: pywhatkit.sendmsg(“receivermobilenumber”,”message”,hours,minutes)

Play a YouTube video:

Syntax: pywhatkit.playonyt(“url/topic name”)

Perform Google Search:

Syntax: pywhatkit.search(“Keyword”)

Keyword: It open your browser and performs search operation.

Get information on particular topic:

Syntax: pywhatkit.info (“topic”, lines=x)

pyautogui:

PyAutoGUI is a Python module which can automate your GUI and programmatically control your keyboard and mouse. This article illustrates the GUI functions to create display boxes.If you want to know more about PyAutoGUI and its ability to automate your keyboard and mouse, follow this article : Mouse and Keyboard Automation using PyAutoGUI PyAutoGUI does not come with python, so go to command prompt and type the following: pip3 install PyAutoGUI

Keyboard Module:

Python provides a library named keyboard which is used to get full control of the keyboard. It’s a small Python library which can hook global events, register hotkeys, simulate key presses and much more.

- It helps to enter keys, record the keyboard activities and block the keys until a specified key is entered and simulate the keys.
- It captures all keys, even onscreen keyboard events are also captured.
- Keyboard module supports complex hotkeys.
- Using this module, we can listen and send keyboard events.
- It works on both windows and linux operating system.

Install using this command:

```
pip install keyboard
```

Example:

```
# Using Keyboard module in Python import
keyboard

# It writes the content to output
keyboard.write("GEEKS FOR GEEKS\n")


# It writes the keys r, k and endofline
keyboard.press_and_release('shift + r, shift + k, \n')
keyboard.press_and_release('R, K')

# it blocks until ctrl is pressed keyboard.wait('Ctrl')
```

CHAPTER 3

COMPUTATIONAL ENVIRONMENT

3.1 SOFTWARE REQUIREMENTS

Operating System	:	Windows 10, Android
Server-side Script	:	Python 3.6
IDE	:	PyCharm
Libraries Used	:	PyAutoGUI, PyWhatKit, Psutil

3.2 HARDWARE REQUIREMENTS

Hard Disk	:	Minimum 2gb of space
Processor	:	Intel i5 or above
RAM	:	Minimum 255mb or more
Input Device	:	Microphone
Output Device	:	Monitor Screen

3.3 SOFTWARE FEATURES

3.3.1 INTRODUCTION TO PYTHON

Python is a high-level, interpreted, and object-oriented programming language that is easy to learn and use. It was first released in 1991 by Guido van Rossum and has since become one of the most popular programming languages in the world. Python is used for a variety of applications, including web development, data analysis, machine learning, and scientific computing.

Python's syntax is simple and easy to read, making it an ideal language for beginners. It has a large and active community of developers who contribute to the development of libraries and modules, making it easier to accomplish complex tasks.

Python is an interpreted language, which means that code written in Python is not compiled but is instead interpreted by the Python interpreter. This makes it easier to write and debug code because developers can test their code as they write it. Python is also an object-oriented language, which means that it uses objects to represent data and perform actions. This makes it easier to organise code and reuse existing code.

Overall, Python is a powerful and versatile language that is widely used in the technology industry and beyond. Whether you are a beginner or an experienced developer, Python is an excellent language to learn and use.

3.3.2. SYNTAX AND SEMANTICS

Python has a simple and easy-to-understand syntax that emphasises readability and a clean, uncluttered code style. Here are some basic syntax rules for Python:

- Statements are typically written one per line.
- Statements are terminated with a newline character, but can also be terminated with a semicolon (;) to put multiple statements on the same line.
- Indentation is used to define blocks of code instead of curly braces ({ }).
- Comments start with a hash (#) and extend to the end of the line.

Python's semantics are also straightforward and intuitive. Here are some key aspects of Python's semantics:

- Python is dynamically typed, which means that variable types are determined at runtime.
- Python is strongly typed, which means that once a variable is assigned a type, it cannot be changed.
- Variables are created when they are first assigned a value.
- Functions are objects, which means they can be passed as arguments to other functions and returned as values from functions.
- Python uses garbage collection to automatically manage memory.

Overall, Python's syntax and semantics are designed to be easy to learn and use, making it a popular choice for beginners and experienced programmers alike.

3.3.3. INDENTATION

In Python, indentation is used to define blocks of code. Indentation is used to group together statements that belong to the same block, such as a function, a loop, or a conditional statement. The amount of indentation is not specified by the language specification but is typically four spaces or one tab.

Here's an example of how indentation is used in Python:

```
def my_function():
print("This is the first line of my function.") print("This is the second line of my function.")
```

In this example, the two print statements are indented by four spaces, indicating that they are part of the my_function function.

Indentation errors are a common issue in Python programming. If the indentation is incorrect, Python will raise an error. For example, consider the following code:

```
def my_function():
print("This is the first line of my function.")
print("This is the second line of my function.")
```

In this example, the second print statement is indented with two spaces instead of four spaces. This will result in an Indentation Error when the code is executed.

It's important to be consistent with indentation in Python to avoid errors and make your code easy to read and understand. Many text editors and IDEs have features that can automatically adjust indentation to help make your code more consistent.

3.3.4. STATEMENTS AND CONTROL FLOW

In Python, statements are instructions that are executed by the interpreter. Here are some of the basic types of statements in Python:

1. Assignment statements: These are used to assign values to variables. The syntax for an assignment statement is variable = value.
2. Expression statements: These are used to evaluate an expression and do not produce any output. The syntax for an expression statement is expression.
3. Print statements: These are used to display output to the console. The syntax for a print statement is print(expression).
4. Conditional statements: These are used to execute different code depending on the value of a condition. The syntax for a conditional statement is:

5. Loop statements: These are used to execute a block of code repeatedly. There are two main types of loop statements in Python:

- for loops: These are used to iterate over a sequence of values. The syntax for a for loop is:

```
for variable in sequence:
# code to execute for each value in the sequence
```

- while loops: These are used to execute a block of code repeatedly as long as a condition is true. The syntax for a while loop is:

```
while condition:
# code to execute while condition is true
```

6. Exception handling statements: These are used to handle errors and exceptions that occur during the execution of a program. The syntax for an exception handling statement is:

```
try:
# code that may raise an exception except exception_type:
# code to handle the exception
```

These are some of the basic types of statements in Python. Control flow statements are used to control the order in which statements are executed. Some examples of control flow statements include if statements, for loops, while loops, and break and continue statements. Control flow statements allow you to create more complex programs by branching and looping through code.

3.3.5. EXPRESSIONS

In Python, an expression is a combination of values, variables, operators, and function calls that produces a result. Expressions can be as simple as a single value or variable, or as complex as a mathematical equation or a function call.

Expressions are used extensively in Python programming to perform calculations, evaluate conditions, and manipulate data. They are a fundamental building block of any Python program.

Python has a large variety of built-in operators and functions that can be used in expressions, making it a powerful language for performing complex calculations and data manipulations.

Here are some examples of expressions in Python:

- 5 is a numeric expression that evaluates to the value 5.
- 3 + 4 is a mathematical expression that evaluates to the value 7.
- x = 2 is an assignment expression that assigns the value 2 to the variable x.
- len("hello") is a function call expression that returns the length of the string "hello".
- x + y * z is a complex expression that combines variables and operators to produce a result.

Python has a wide variety of built-in operators that can be used in expressions, including arithmetic operators (+, -, *, /, //, %), comparison operators (==, !=, <, >, <=, >=), logical operators (and, or, not), and many others.

In addition to built-in operators, Python also has a large standard library of built-in functions that can be used in expressions. These functions include mathematical functions like sqrt and pow, string functions like split and join, and many others.

Overall, expressions are a fundamental part of Python programming and are used extensively in creating complex programs.

3.3.6. PYTHON HAS VARIOUS KINDS OF STRING LITERALS

In Python, a string literal is a sequence of characters enclosed within quotation marks. String literals can be written using either single quotes ('...') or double quotes ("..."). In addition to single-quoted and double-quoted string literals, Python also supports triple-quoted string literals. Triple-quoted string literals are enclosed within triple quotes ('''...''' or """...""") and can span multiple lines.

String literals are often used to represent text data in Python programs. They can be assigned to variables, passed as arguments to functions, and manipulated using various string methods. Python provides a rich set of operations and functions for working with string literals, making it easy to perform a wide range of text processing tasks.

3.3.7. METHODS

In Python, a method is a function that is defined within a class and is used to perform operations on objects of that class. Methods are associated with objects and can access and modify the object's data.

Here are some of the most commonly used methods in Python:

- str(): This method converts the given object into a string.
- int(): This method converts the given object into an integer.
- float(): This method converts the given object into a float.
- len(): This method returns the length of the given object.
- type(): This method returns the type of the given object.
- append(): This method adds an item to the end of a list.
- pop(): This method removes and returns the last item from a list.
- sort(): This method sorts the items in a list.
- replace(): This method replaces a substring with another substring in a string.
- strip(): This method removes leading and trailing whitespace from a string.
- upper(): This method returns a string in all uppercase letters.
- lower(): This method returns a string in all lowercase letters.
- join(): This method concatenates a list of strings into a single string.
- split(): This method splits a string into a list of substrings.
- range(): This method generates a sequence of numbers.

These are just a few examples of the many methods available in Python. By using methods, you can perform a wide variety of operations on objects, making Python a powerful and versatile programming language.

3.3.8. APPLICATIONS OF PYTHON

Python is a versatile and widely used programming language with a broad range of applications. Here are some of the most common areas where Python is used:

Web Development: Python is used extensively in web development to create web applications, dynamic websites, and server-side scripting. Popular web frameworks such as Django, Flask, and Pyramid are built using Python.

Data Science and Analytics: Python is a popular choice for data analysis, machine learning, and artificial intelligence. Libraries such as NumPy, Pandas, Matplotlib, and Scikit-learn provide powerful tools for data manipulation, visualisation, and modelling. **Scientific Computing:** Python is widely used in scientific computing and research, with libraries such as SciPy and Biopython providing a range of functions for scientific computations, simulations, and data analysis.

Desktop Application Development: Python can also be used to create desktop applications with graphical user interfaces (GUIs) using libraries such as PyQt and Tkinter.

Game Development: Python can be used to create games with the help of libraries such as Pygame and PyOpenGL.

System Administration: Python is used extensively in system administration and automation tasks, such as network programming, system monitoring, and scripting.

Education: Python is a popular choice for teaching programming to beginners due to its simple syntax and easy-to-learn structure.

These are just a few examples of the many areas where Python is used. Its versatility, simplicity, and wide range of libraries and tools make it a powerful language for a variety of applications.

3.3.9. INTRODUCTION TO PYCHARM

PyCharm was first released in 2010 by JetBrains, a Czech software development company. JetBrains had previously developed popular IDEs for other programming languages, such as IntelliJ IDEA for Java and ReSharper for C#. PyCharm was developed as a dedicated IDE for Python programming, with a focus on productivity and ease of use. The first version of PyCharm included features such as code completion, syntax highlighting, and code analysis, as well as support for various web frameworks such as Django and Flask.

Over the years, PyCharm has evolved to become one of the most popular and widely used IDEs for Python development. It has continued to add new features and tools, such as integrated version control, remote development support, and integration with popular data analysis and machine learning libraries. PyCharm has also introduced various editions, such as the Community Edition (which is free and open-source) and the Professional Edition (which includes additional features and tools for advanced Python development).

Today, PyCharm is used by millions of developers worldwide and is considered one of the most powerful and versatile IDEs for Python programming.

3.3.10. WHAT IS PYCHARM

PyCharm is an Integrated Development Environment (IDE) for the Python programming language. It is developed by JetBrains and provides a range of features and tools that make Python development more efficient and productive.

PyCharm provides a powerful and user-friendly interface for developing, debugging, and deploying Python applications.

Here are some features of PyCharm:

Code completion and suggestion: PyCharm provides intelligent code completion and suggestion based on the context of the code.

Debugging: PyCharm has a powerful debugger that allows you to easily debug your code and find and fix errors.

Code analysis: PyCharm has built-in code analysis tools that can detect errors and suggest improvements in your code.

Refactoring: PyCharm makes it easy to refactor your code, allowing you to easily rename variables, extract methods, and more.

Version control: PyCharm has built-in support for popular version control systems such as Git, SVN, and Mercurial.

Deployment: PyCharm provides tools for deploying your applications to servers and cloud services such as AWS, Google Cloud, and Microsoft Azure.

PyCharm is used by Python developers for a variety of tasks such as web development, data science, scientific computing, game development, and more. It is especially useful for complex projects where managing code and dependencies can become difficult. PyCharm is typically used when developing Python applications that require a powerful and comprehensive IDE. It is especially useful for larger projects and for teams working on collaborative projects. However, for smaller projects or quick scripting tasks, a simple text editor or lightweight IDE may suffice.

CHAPTER 4

FEASIBILITY STUDY

The feasibility of the project is analysed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ♣ TECHNICAL FEASIBILITY
- ♣ OPERATIONAL FEASIBILITY
- ♣ ECONOMICAL FEASIBILITY
- ♣ SOCIAL FEASIBILITY

9.1 TECHNICAL FEASIBILITY:

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system. So, we are using python technology which is an open source and feasible to implement in the System.

9.2 OPERATIONAL FEASIBILITY:

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

Be given for quick that specific and money associated assets are each settled on a choice about adequate. In case clients are in each manner that actually subjects with the cutting-edge system, see no issues with it, and usually aren't associated with referencing any other structure, safety from completing the brand new shape will be strong.

9.3 ECONOMIC FEASIBILITY:

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

9.2 SOCIAL FEASIBILITY:

The social feasibility of "Smart System Control with Voice Command Using Firebase" would depend on various factors such as cultural, societal, and economic factors.

From a cultural perspective, the adoption of voice-activated smart systems is becoming increasingly common and widely accepted in many societies. People are becoming more comfortable with the idea of interacting with technology through their voice. However, some cultures or individuals may prefer to use physical controls or may feel uncomfortable with the idea of constantly being monitored by a smart system.

From a societal standpoint, the use of smart systems with voice command could lead to greater accessibility and convenience for people with disabilities or mobility issues. This technology could provide an easier and more intuitive way for them to interact with their environment. However, concerns around privacy and security could arise, particularly in situations where the system is constantly listening to and recording people's voices.

From an economic perspective, the feasibility of implementing a smart system with voice command using Firebase would depend on factors such as the cost of the hardware and software required to run the system, as well as the cost of implementing and maintaining the system. The availability of reliable and affordable internet connectivity could also impact the feasibility of this technology in certain regions.

Overall, the social feasibility of "Smart System Control with Voice Command Using Firebase" would depend on various factors such as cultural acceptance, societal benefits, and economic feasibility. With proper consideration and implementation of these factors, this technology could have significant potential to improve the lives of many people.

CHAPTER 5

SYSTEM ANALYSIS

5.1 EXISTING SYSTEM:

Access Control is an Android Application that works on the concepts of wireless socket programming. This application is to provide the user with a remote for his/her PC in the form of an android device. Using this application, the user may perform various actions on his PC such as controlling the mouse movement and operations, sliding through various PPT slides, managing media, entering text on any application on their PC, from a reasonable distance, all just with the help of their android device. For this application to work, the PC and the given android device need only be connected to a common network.

Client-side Application: Client/server architecture is a computing model which consist of multiple client computer or process and single computer or process. Generally, architecture of this type has one or more client computers connected to a central server via network or Internet connection. This system generally shares computing resources among multiple client computer, so in a client server application, client depends upon the server for resource such as a file, device and also for processing ability. Similarly, this application is fragmented in two parts, client-side application and server side application. Client-side application involves the smart phone with android application install on it. Client side is visible to the user and user interact with client side by using some specific action and receiving the response as a outcome of that action. Server-side application responsible for processing the request from client side and returns desired result to the client side. Like traditional client-server application, the android client application is installed on smart phone which is fully depends on the server application for service such as a making calls/SMS, retrieving call logs/SMS and providing operator information. Smartphone which having a sturdy android operating system install on it (KitKat 4.4.2) is located at client side.

Server-side Application: In a client-server application, server is the powerful computers or the process which are responsible for managing resource. Server is also responsible for processing a request which are coming from the client and also for providing a outcome to the client which is the result of that request. Server-side application involves a “java executable” file, which is situated on computing device like desktop computer or laptops. Server side is responsible for receiving the request from client side and for providing the service to the user by processing the request from client side, similar like traditional client-server architecture. Server-side application is not visible to the user but, it provides the requested service to the user through the android application.

DISADVANTAGES OF EXISTING SYSTEM:

- **Security risks**: Since the application involves wireless socket programming, there may be potential security risks associated with its use. For example, if the common network is not secure, it could allow unauthorized access to the PC by malicious actors.
- **Compatibility issues**: The application may not be compatible with all versions of Android devices or PCs, which could limit its usefulness.
- **Performance issues**: The application may experience performance issues such as lag or connectivity problems, especially if the network connection is weak or unstable.
- **Limited functionality**: The application may not offer the full range of capabilities that a user needs to control their PC remotely, which could limit its usefulness for certain tasks.
- **Learning curve**: The user may need to spend some time learning how to use the application effectively, which could be a drawback for those who prefer a more intuitive user experience.

5.2 PROPOSED SYSTEM

The proposed system is a voice-controlled system that allows users to control their system remotely from anywhere in the world using voice commands. The system utilizes Firebase, Scratch, and Python to establish a connection between the user's mobile device and their system.

The system consists of two components: the mobile interface and the system interface. The mobile interface is built using Scratch and allows the user to input voice commands that are sent to Firebase. Firebase then sends the commands to the system interface, which is built using Python. The system interface receives the voice commands from Firebase and processes them using a speech recognition library. Once the command is recognized, the system executes the appropriate action. For example, the user could say "turn on the lights," and the system would execute the command by sending a signal to the connected lights to turn on.

The system also has security features to ensure that only authorized users can control the system. Users must log in to the mobile interface using a secure username and password, and the system interface only accepts commands from authorized users. The proposed system provides several benefits over traditional socket connections. Users can control their system from anywhere in the world as long as they have an internet connection. The system is also more secure since it uses secure login credentials and encrypted communication channels.

In summary, the proposed system is a voice-controlled system that allows users to control their system remotely from anywhere in the world using voice commands. The system utilizes Firebase, Scratch, and Python to establish a secure and reliable connection between the user's mobile device and their system.

ADVANTAGES OF PROPOSED SYSTEM

There are several advantages of the proposed system for controlling a computer system remotely using voice commands:

- **Ease of use**: The system is designed to be easy to use, with a mobile interface built on Scratch that can capture voice commands. This makes it accessible to a wide range of users, including those who may not be familiar with more complex remote-control systems.
- **Flexibility**: The system is not limited by range, unlike traditional socket-based connections. This means that it can be used to control a computer system from anywhere in the world, as long as there is an internet connection.
- **Convenience**: The ability to control a computer system remotely using voice commands can be more convenient than using a traditional keyboard and mouse, especially for users who may have limited mobility or prefer a more natural interface.
- **Efficiency**: The use of Python to execute commands received from the mobile interface can make the system more efficient and responsive than other remote-control systems.
- **Potential for future development**: The paper presents the implementation of the system and outlines future work, suggesting that the system has potential for further development and refinement. This could lead to even greater usability, flexibility, and efficiency in the future.

CHAPTER 6

SYSTEM DESIGN

System design is a transition from a user-oriented document to programmers or database personnel. The design is a solution, how to approach the creation of a new system. This is composed of several steps. It provides the understanding and procedural details necessary for implementing the system recommended in the feasibility study. Designing goes through logical and physical stages of development, logical design reviews the present physical system, prepare input and output specification, details of the implementation plan and prepare a logical design walkthrough. During Detailed Design, the internal logic of each of the modules specified in system design is decided. During this phase, the details of the data of a module are usually specified in a high level design description language, which is independent of the target language in which the software will eventually be implemented. The unnecessary fields should be avoided because it affects the storage areas of the system. Then in the input and output screen design, the design should be made user friendly. The menu should be precise and compact.

Elements of a System

● Architecture - This is the conceptual model that defines the structure, behaviour and more views of a system. We can use flowcharts to represent and illustrate the architecture.

● Modules - These are components that handle one specific task in a system. A combination of the modules makes up the system.

6.1. SYSTEM ARCHITECTURE

A system architecture or systems architecture is the computational design that defines the structure and/or behaviour of a system.

An architecture description is a formal description of a system, organised in a way that supports reasoning about the structural properties of the system. It defines the system components or building blocks and provides a plan from which products can be procured, and systems developed, that will work together to implement the overall system.

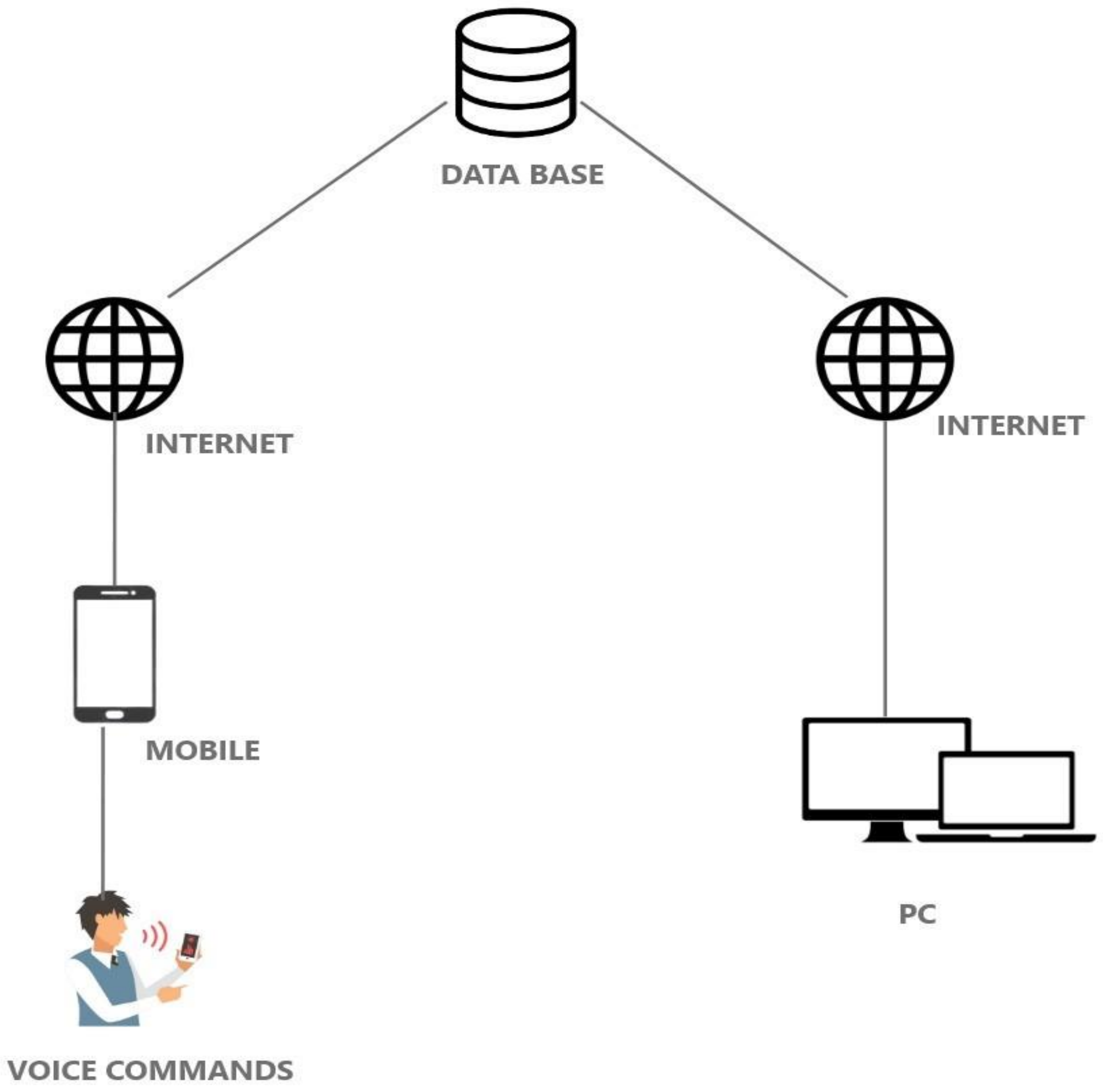


Fig 6.1.1: System Architecture

6.2. UML DIAGRAMS

What is UML?

The Unified Modelling Language (UML) is a standard language for specifying, Visualizing, constructing and documenting the artifacts of software systems, as well as for business modelling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modelling of large and complex systems. The UML is a very important part of developing objects-oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects. Using the UML helps project teams communicate, explore potential designs, and validate the architectural design of the software.

Goals:

The primary goals in the design of the UML are as follows:

1. Be independent of particular programming languages and development processes.
2. Provide a formal basis for understanding the modelling language.
3. Encourage the growth of the OO tools market.
4. Support higher-level development concepts such as collaborations, frameworks, patterns and components.
5. Integrate best practices.

In this project six basic UML diagrams have been explained

1. Use Case Diagram
2. Class Diagram
3. Sequence Diagram
4. Collaboration Diagram
5. Activity Diagram
6. Deployment Diagram

6.2.1 USE CASE DIAGRAM:

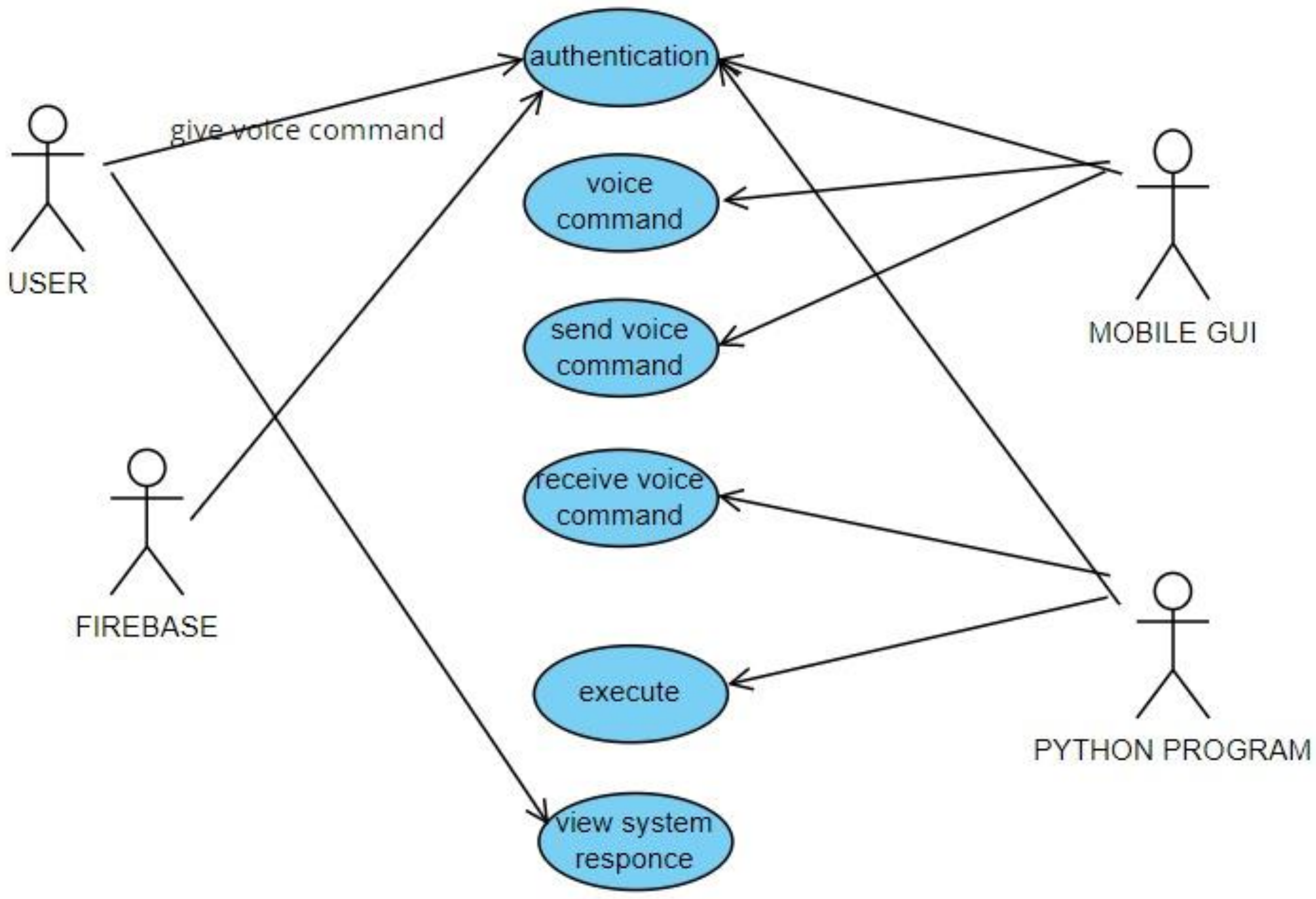


Fig 6.2.1.1 : Use Case Diagram

In this diagram, there are two primary actors: User and MobileDeviceGUI. The User is responsible for giving the voice command and viewing the system response. The Mobile Device GUI is responsible for recording the voice command, sending the voice command to the Python Program, receiving the response from the PythonProgram, and displaying the response to the User.

The Python Program is responsible for executing the command received from the MobileDeviceGUI and sending the response back to the MobileDeviceGUI.

The arrows in the diagram represent the interactions between the actors and the system components. The User initiates the voice command, which is recorded by the MobileDeviceGUI and sent to the Python Program. The Python Program executes the command and sends the response back to the MobileDeviceGUI, which displays the response to the User.

6.2.2 CLASS DIAGRAM:

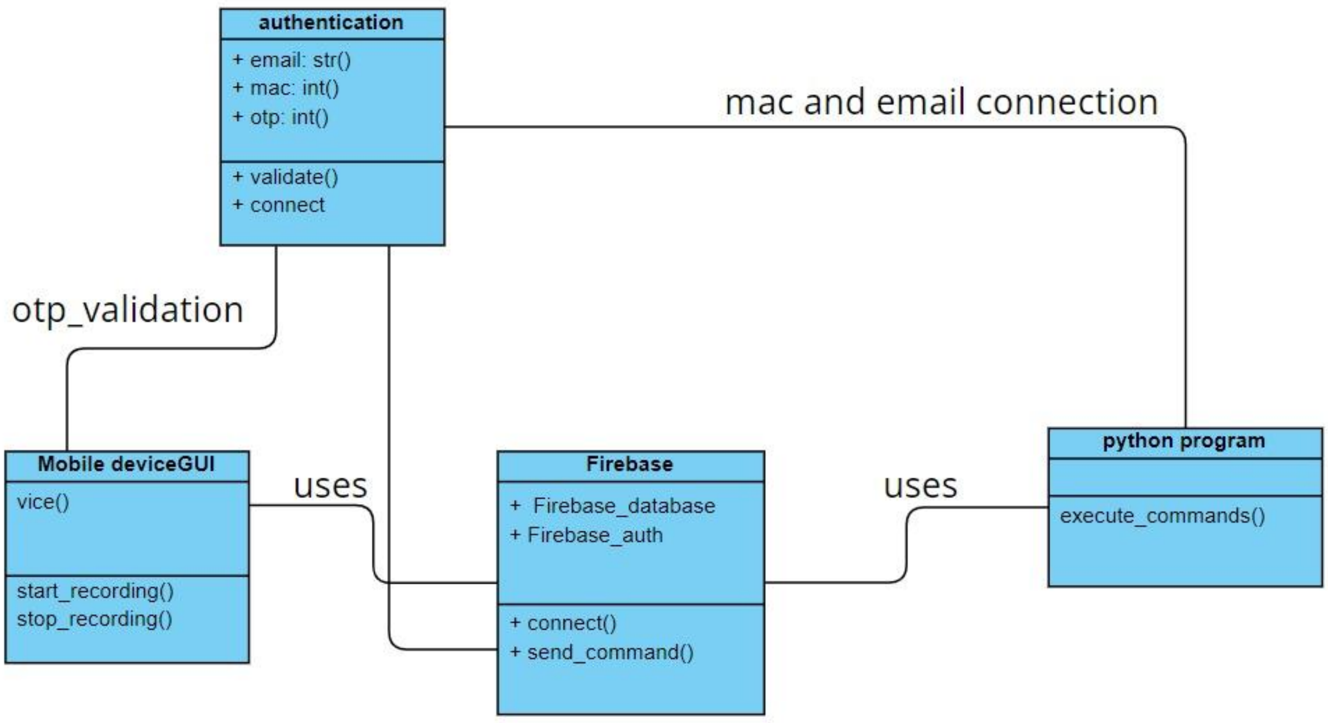


Fig 6.2.2.1 : Class Diagram

In this diagram, there are three classes: MobileDeviceGUI, FirebaseConnection, and PythonProgram.

MobileDeviceGUI represents the graphical user interface of the mobile device, which allows the user to give voice commands. It has two methods: start_recording() and stop_recording(), which begin and end the recording of the user's voice, respectively.

FirebaseConnection represents the Firebase connection between the mobile device and the laptop running PythonProgram. It has two attributes: firebase_database and firebase_auth, which are used to establish the connection to the Firebase server. It also has two methods: connect(), which establishes the connection to the Firebase server, and send_command(), which sends the user's voice command to the laptop running PythonProgram.

PythonProgram represents the program running on the laptop that executes the user's voice command. It has one method: execute_command(), which executes the user's voice command. This method is called when FirebaseConnection sends the user's voice command to the laptop running PythonProgram.

The arrows in the diagram represent the relationships between the classes. The MobileDeviceGUI class uses the FirebaseConnection class to send the user's voice command to the laptop running PythonProgram. The FirebaseConnection class uses the PythonProgram class to execute the user's voice command.

6.2.3 SEQUENCE DIAGRAM:

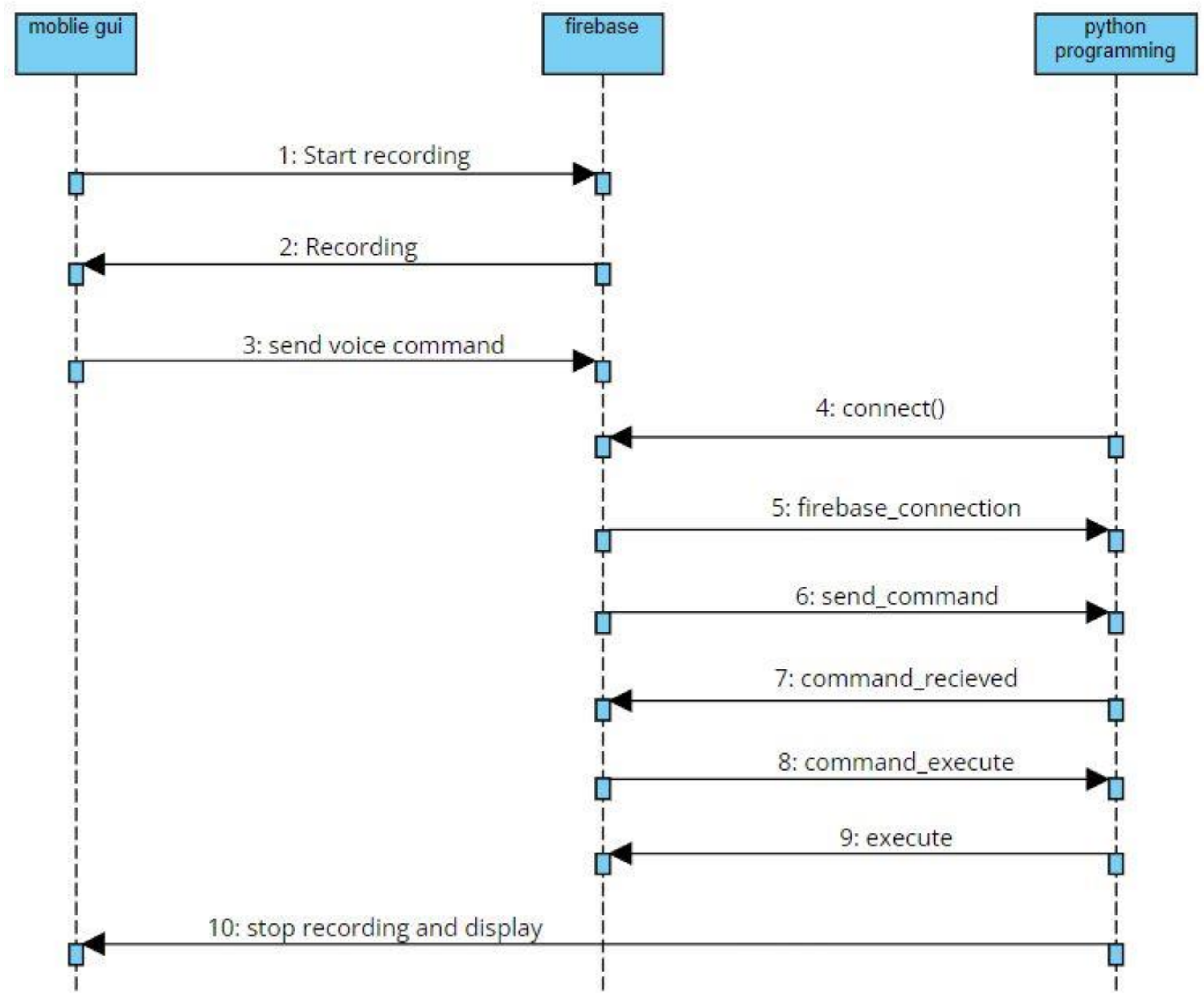


Fig 6.2.3.1 : Sequence Diagram

The user initiates the voice command by starting the recording using the start_recording() method of the MobileDeviceGUI class.

The mobile device begins recording the user's voice.

The user speaks the voice command.

The mobile device sends the voice command to the FirebaseConnection class using the send_command() method.

The FirebaseConnection class establishes a connection to the Firebase server using the connect() method.

The FirebaseConnection class sends the voice command to the Firebase server.

The PythonProgram class receives the voice command from the Firebase server using the send_command() method of the FirebaseConnection class.

The PythonProgram class executes the command using the execute_command() method.

The PythonProgram class sends the response to the Firebase server.

The Firebase server sends the response to the MobileDeviceGUI class.

The MobileDeviceGUI class stops recording and displays the response to the user.

6.2.4 COLLABORATION DIAGRAM:

A Collaboration diagram is an interaction diagram that emphasizes the structural organization of the objects that send and receive messages. Interaction diagrams address the dynamic view of a system.

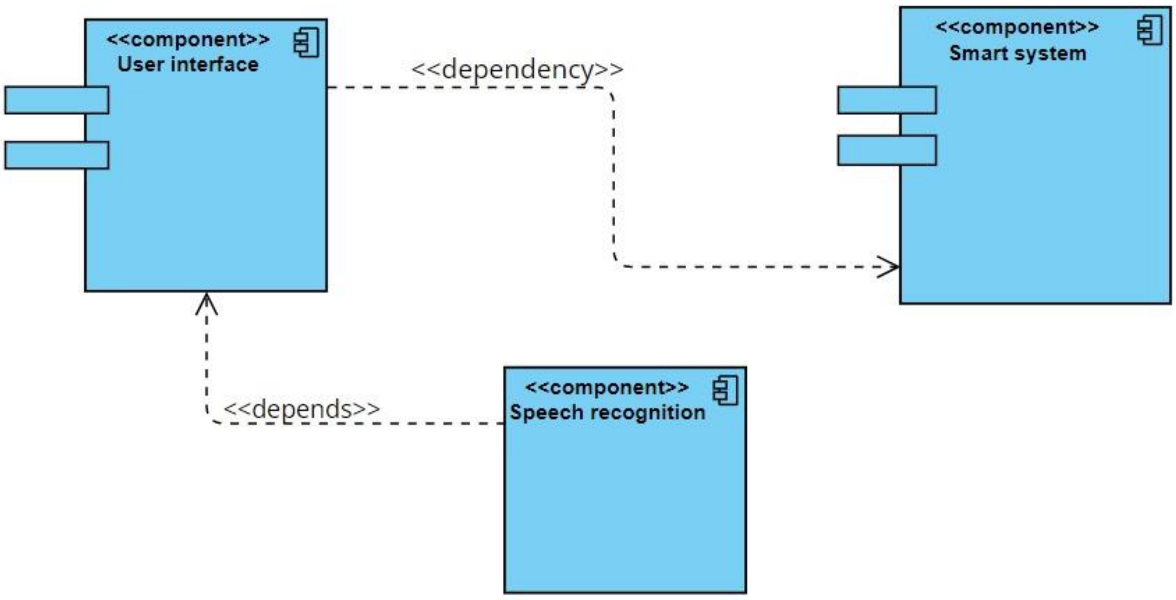


Fig 6.2.4.1 : Collaboration Diagram

6.2.5 ACTIVITY DIAGRAM:

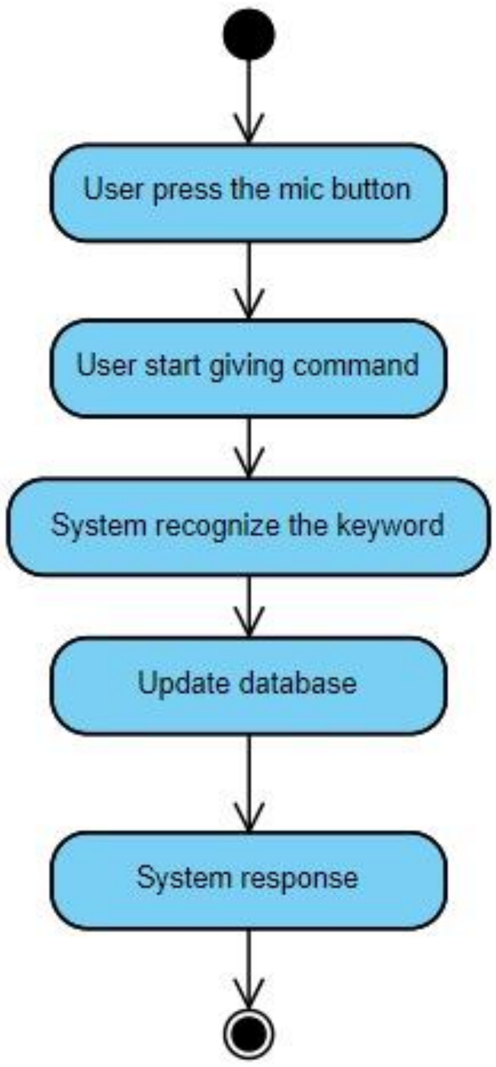

User press the mic button
User start giving command
System recognize the keyword
Update database
System response

Fig 6.2.5.1 : Activity Diagram

The activity diagram is another important diagram in UML to describe the dynamic aspects of the system. The activity diagram is a flowchart to represent the flow from one activity to another activity.

6.2.6 DEPLOYMENT DIAGRAM :

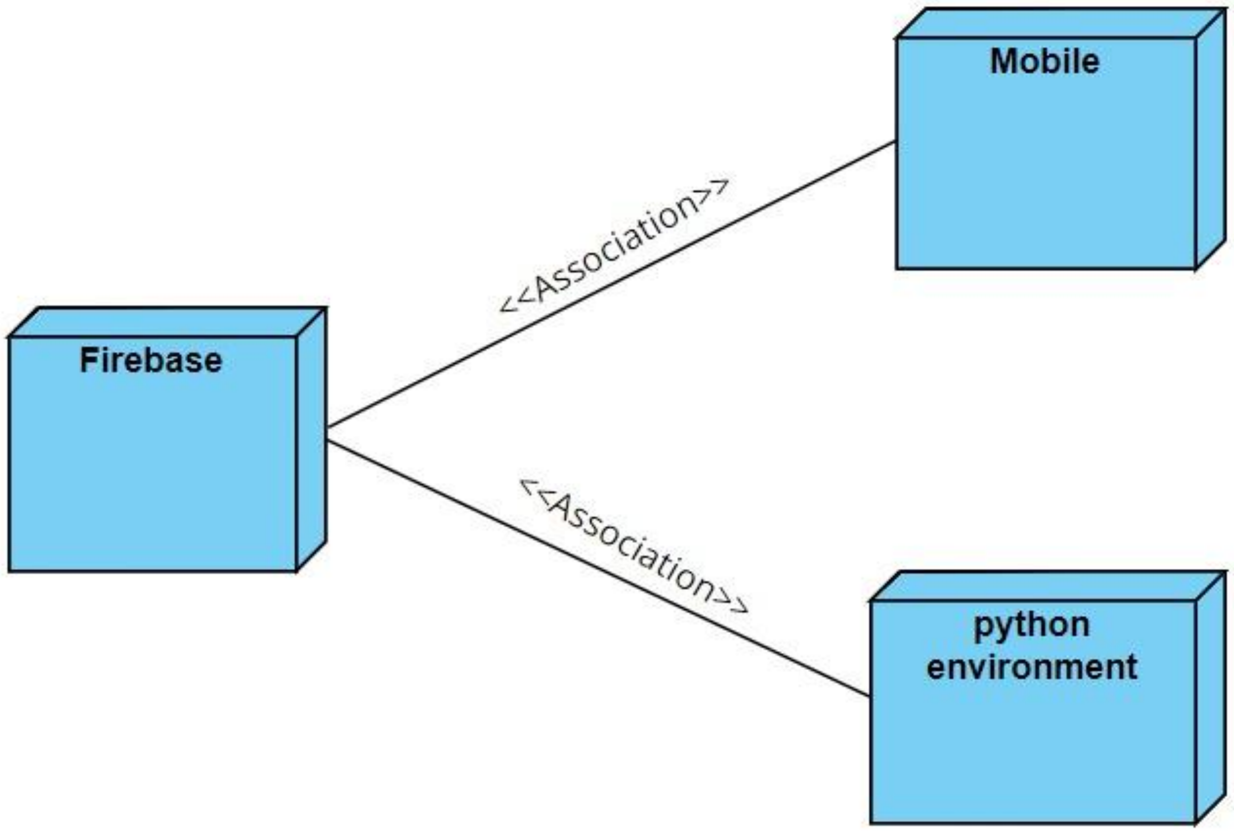


Fig 6.2.6.1 : Deployment Diagram

In this diagram, there are three main components: the Mobile Device, Firebase Cloud, and Python Environment. The Mobile Device contains the microphone and mobile app used to record and send the voice command to the Firebase Cloud. The Firebase Cloud is used to receive the voice command, store it in the database, and forward it to the Python Environment. The Python Environment receives the voice command, interprets it using the interpreter and libraries, executes the command, and sends the response back to the Firebase Cloud.

The arrows in the diagram represent the communication between the components. The Mobile Device sends the voice command over the internet to the Firebase Cloud, which stores it in the database and forwards it to the Python Environment. The Python Environment executes the command and sends the response back to the Firebase Cloud, which then sends the response back to the Mobile Device.

CHAPTER 7

SYSTEM IMPLEMENTATION

The proposed system consists of two parts: a mobile interface built on Scratch and a remote computer system programmed using Python. The mobile interface captures the voice commands and transmits them to the remote computer system using Firebase. The remote computer system executes the commands received from the mobile interface. The system can be used to perform various tasks, such as opening applications, controlling media players, and adjusting system settings.

The proposed system for controlling a computer system with voice commands using Firebase and Scratch can be implemented through the following steps:

Mobile Interface Development:

First, develop a mobile interface using Scratch. The mobile interface should have a user-friendly design and be capable of capturing voice commands from the user. The voice commands should be transmitted to the remote computer system using Firebase.

Firebase Integration:

Integrate Firebase with the mobile interface to establish a connection between the mobile interface and the remote computer system. Firebase is a cloud-based service that allows for real-time data transmission and synchronization. Set up a Firebase project and configure it to enable real-time data transmission between the mobile interface and the remote computer system.

Computer System Programming:

Program the remote computer system using Python to receive the voice commands transmitted through Firebase and execute the corresponding actions. Use the Firebase Python library to establish a connection between the remote computer system and Firebase. The Python code should be designed to handle the incoming commands, process them, and execute the desired actions.

Voice Command Processing:

Design a voice command processing algorithm that can understand and interpret the voice commands captured by the mobile interface. Use a speech recognition library, such as Google Cloud Speech API or Python Speech Recognition, to convert the voice commands into text.

The algorithm should be designed to identify the specific actions required by the user and transmit them to the remote computer system.

Testing and Validation:

Test the system under various network conditions to ensure its reliability and accuracy. Conduct experiments to evaluate the system's response time, accuracy, and reliability. Validate the system against various use cases and user scenarios.

Deployment:

Deploy the system to the target environment and ensure that it is running smoothly. Monitor the system for any issues and resolve them as they arise.

7.1 Modules Explanation:

Implementation is the stage where the theoretical design is converted into programmatically manner. In this stage we will divide the application into a few modules and then coded for deployment. The implemented application consists of two parts, the first one is an application for Android smartphone and the second one is a server application that executes the command selected by user’s application.

The Android application is mainly divided into 3 modules:

1.making user interface using scratch

2.taking voice commands from user

3.convert voice commands into text and store it in database(firebase)

Server application is mainly divided into 4 modules:

1.import Necessary Libraries.

2.connection establish with user application and database.

3.retrive user commands from database.

4.execute these commands.

Authentication:

Authentication is the process of verifying the identity of a user or device. In the context of your project, authentication is used to ensure that only authorized users can access and control the system through voice commands.

Here's a detailed explanation of how authentication works in your project:

- When the system is run for the first time on a new device, it prompts the user to enter their email address.
- Once the email address is entered, the system sends an OTP (One-Time Password) to the email address provided by the user.
- The user then enters the OTP into the system to validate their email address.
- Once the email address is validated, the system establishes a connection between the user's device and the laptop using Firebase.
- When the user wants to control the system through voice commands, they must first authenticate themselves by entering the same email address and OTP.
- If the email address and OTP match the records in the system, the user is authenticated and can issue voice commands to control the system.

By using authentication, you can ensure that only authorized users can control the system and prevent unauthorized access. Additionally, using email address and OTP for authentication provides an extra layer of security to your project.

Making user interface using scratch:

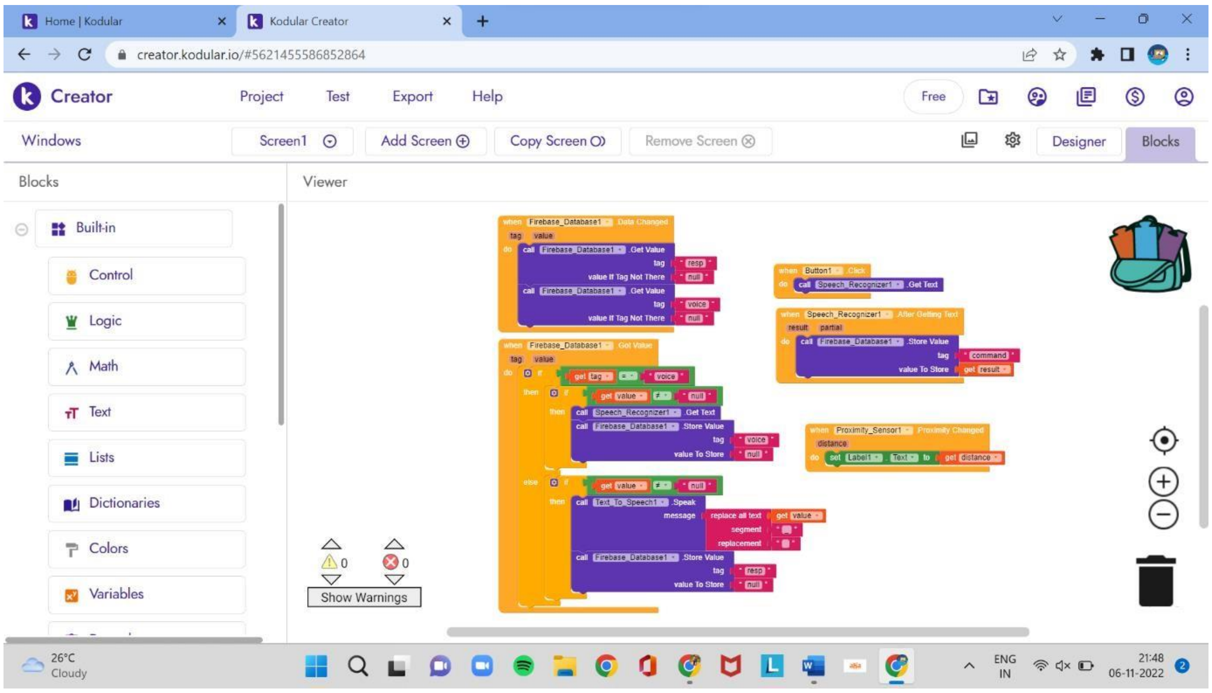
In this module create a user interface using scratch, which is taking voice commands from user.

Scratch:

Scratch is an event-driven visual programming language developed by MIT. In Scratch, we can create our own interactive stories, games, and animations using building blocks. In this platform, we do not need to write code to perform operations, things are done just by drag and drop, just like visual basic. It is the best platform to start basic programming by creating attractive animation effects. There are so many features available in Scratch, such as video games, animations, stories, sound, events, etc. It is a free platform created by the Lifelong Kindergarten group at MIT in the Media lab. It is developed in ActionScript and JavaScript and is compatible with any operating system. It has been translated into more than 70 languages and used in most parts of the world.

Application using Scratch:

Using Scratch create an application for user which can take voice commands from user and convert it into text store it on firebase for the further assistance when you click on the Jarvis button it calls the speech recognition and take the voice command from the user and store it on firebase.



```
when Firebase_Database1 .Data Changed
  tag  value
do  call Firebase_Database1 .Get Value
        tag "resp"
        value If Tag Not There "null"
    call Firebase_Database1 .Get Value
        tag "voice"
        value If Tag Not There "null"

when Firebase_Database1 .Got Value
  tag  value
do  if  get tag = "voice"
    then  if  get value ≠ "null"
          then  call Speech_Recognizer1 .Get Text
                call Firebase_Database1 .Store Value
                    tag "voice"
                    value To Store "null"
    else  if  get value ≠ "null"
          then  call Text_To_Speech1 .Speak
                    message  replace all text  get value
                             segment " "
                             replacement " "
                call Firebase_Database1 .Store Value
                    tag "resp"
                    value To Store "null"

when Button1 .Click
do  call Speech_Recognizer1 .Get Text

when Speech_Recognizer1 .After Getting Text
  result  partial
do  call Firebase_Database1 .Store Value
        tag "command"
        value To Store  get result

when Proximity_Sensor1 .Proximity Changed
  distance
do  set Label1 . Text to  get distance
```

Fig 7.1.1 Scratch Application

Taking voice commands from user:

For the further assistance when you click on the Jarvis button it calls the speech recognition and take the voice command from the user

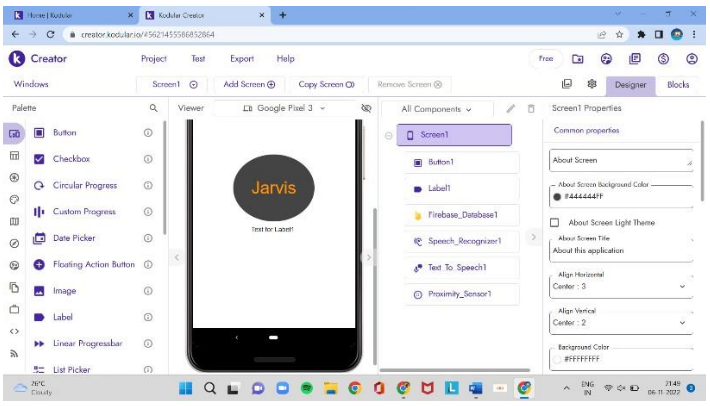


Fig 7.1.2: User interface

Convert voice commands into text and store it database(firebase):

A fter receiving voice commands from user convert these commands into the text then it will stored into database for the further use.

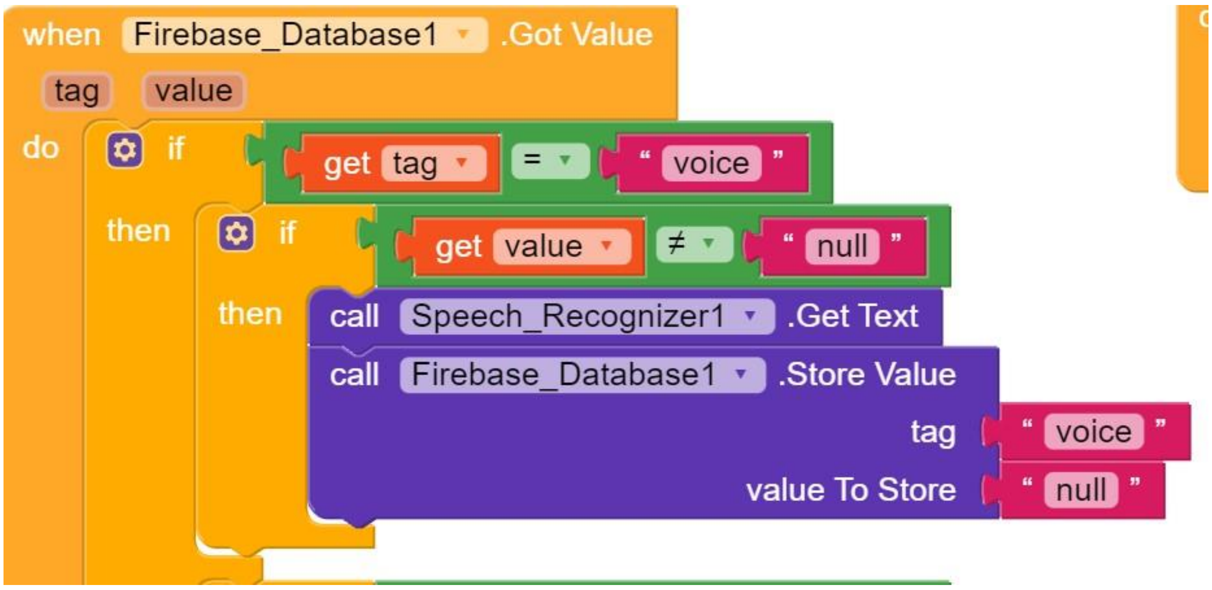
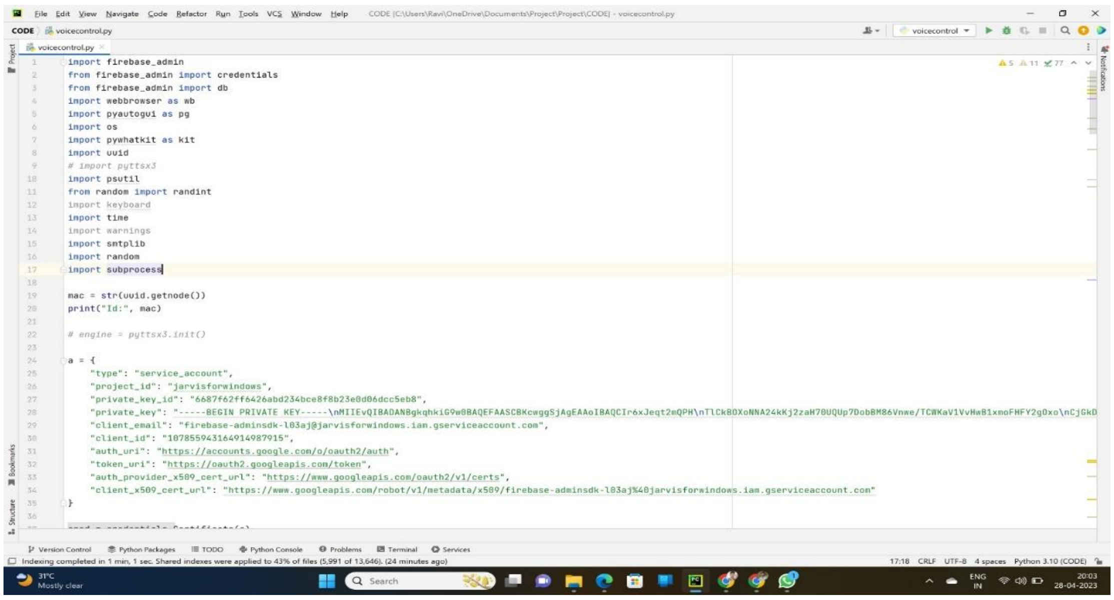

when Firebase_Database1 .Got Value
tag value
do if get tag = "voice"
then if get value ≠ "null"
then call Speech_Recognizer1 .Get Text
call Firebase_Database1 .Store Value
tag "voice"
value To Store "null"

Fig 7.1.3: Speech to text and store it in Database

Import necessary libraries:

In this module initially we need to import all the necessary libraries which are required for building the model. Here we try to use all the libraries which are used to convert the data into meaningful manner. Here the data is divided into numerical values which are easily identified by the system; hence we try to import NumPy module and for plotting the data in graphs and charts we used marplot library.



```
import firebase_admin
from firebase_admin import credentials
from firebase_admin import db
import webbrowser as wb
import pyautogui as pg
import os
import pywhatkit as kit
import uuid
# import pyttsx3
import psutil
from random import randint
import keyboard
import time
import warnings
import smtplib
import random
import subprocess

mac = str(uuid.getnode())
print("Id:", mac)

# engine = pyttsx3.init()

a = {
    "type": "service_account",
    "project_id": "jarvisforwindows",
    "private_key_id": "6687f62ff6426abd234bce8f8b23e0d06dcc5eb8",
    "private_key": "-----BEGIN PRIVATE KEY-----\nMIIEvQIBADANBgkqhkiG9w0BAQEFAASCBKcwggSjAgEAAoIBAQCIr6xJeqt2mQPH\nTlCkBOXoNNA24kKj2zaH70UQUp7DobBM86Vnwe/TCWKaV1VvHwB1xmoFHFY2gOxo\nCjGkD
    "client_email": "firebase-adminsdk-l03aj@jarvisforwindows.iam.gserviceaccount.com",
    "client_id": "107855943164914987915",
    "auth_uri": "https://accounts.google.com/o/oauth2/auth",
    "token_uri": "https://oauth2.googleapis.com/token",
    "auth_provider_x509_cert_url": "https://www.googleapis.com/oauth2/v1/certs",
    "client_x509_cert_url": "https://www.googleapis.com/robot/v1/metadata/x509/firebase-adminsdk-l03aj%40jarvisforwindows.iam.gserviceaccount.com"
}
```

Indexing completed in 1 min, 1 sec. Shared indexes were applied to 43% of files (5,991 of 13,646). (24 minutes ago)

File Edit View Navigate Code Refactor Run Tools VCS Window Help
CODE [C:\Users\Ravi\OneDrive\Documents\Project\Project\CODE] - voicecontrol.py
CODE voicecontrol.py
voicecontrol.py
Version Control Python Packages TODO Python Console Problems Terminal Services
17:18 CRLF UTF-8 4 spaces Python 3.10 (CODE)
31°C Mostly clear
Search
ENG IN 20:03 28-04-2023

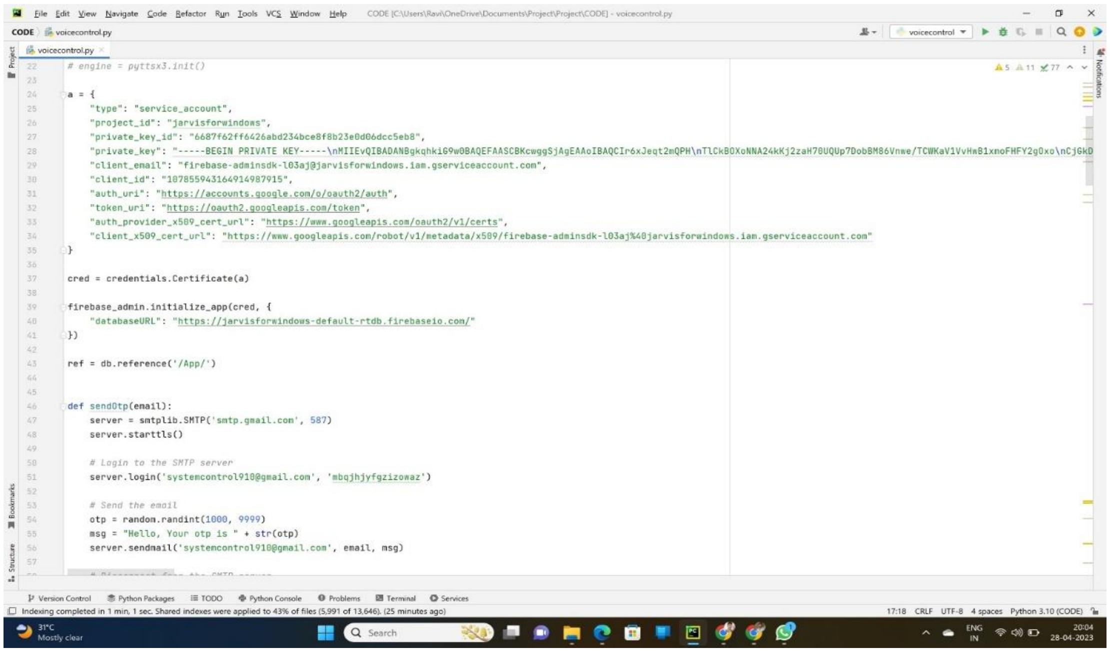
Fig 7.1.4 Modules import

Connection establishes with user and database:

In this module establish connection between the server and user using firebase

Firebase:

Firebase is a product of Google which helps developers to build, manage, and grow their apps easily. It helps developers to build their apps faster and in a more secure way. No programming is required on the firebase side which makes it easy to use its features more efficiently. It provides services to android, iOS, web, and unity. It provides cloud storage. It uses NoSQL for the database for the storage of data.



```
# engine = pyttsx3.init()

a = {
    "type": "service_account",
    "project_id": "jarvisforwindows",
    "private_key_id": "6687f62ff6426abd234bce8f8b23e0d06dcc5eb8",
    "private_key": "-----BEGIN PRIVATE KEY-----\nMIIEvQIBADANBgkqhkiG9w0BAQEFAASCBKcwggSjAgEAAoIBAQCIr6xJeqt2mQPH\nTlCkBOXoNNA24kKj2zaH70UQUp7DobBM86Vnwe/TCWKaV1VvHwB1xmoFHFY2g0xo\nCjGkD
    "client_email": "firebase-adminsdk-l03aj@jarvisforwindows.iam.gserviceaccount.com",
    "client_id": "107855943164914987915",
    "auth_uri": "https://accounts.google.com/o/oauth2/auth",
    "token_uri": "https://oauth2.googleapis.com/token",
    "auth_provider_x509_cert_url": "https://www.googleapis.com/oauth2/v1/certs",
    "client_x509_cert_url": "https://www.googleapis.com/robot/v1/metadata/x509/firebase-adminsdk-l03aj%40jarvisforwindows.iam.gserviceaccount.com"
}

cred = credentials.Certificate(a)

firebase_admin.initialize_app(cred, {
    "databaseURL": "https://jarvisforwindows-default-rtdb.firebaseio.com/"
})

ref = db.reference('/App/')


def sendOtp(email):
    server = smtplib.SMTP('smtp.gmail.com', 587)
    server.starttls()

    # Login to the SMTP server
    server.login('systemcontrol910@gmail.com', 'mbqjhjyfgzizowaz')

    # Send the email
    otp = random.randint(1000, 9999)
    msg = "Hello, Your otp is " + str(otp)
    server.sendmail('systemcontrol910@gmail.com', email, msg)
```

```
Indexing completed in 1 min, 1 sec. Shared indexes were applied to 43% of files (5,991 of 13,646). (25 minutes ago)
17:18  CRLF  UTF-8  4 spaces  Python 3.10 (CODE)
```

Fig 7.1.5 Firebase connection with server application

Overview of the system:

The overall system design consists of following phases:

1. Data collection in the form of speech.
2. Voice analysis and conversion to text.
3. Data storage and processing.
4. Execute the user request and Generating speech from the processed text output**.**

CHAPTER 8

TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the software system meets its requirements and user expectations and does not fail in an unacceptable manner.

- Unit Testing
- Integration Testing
- System Testing

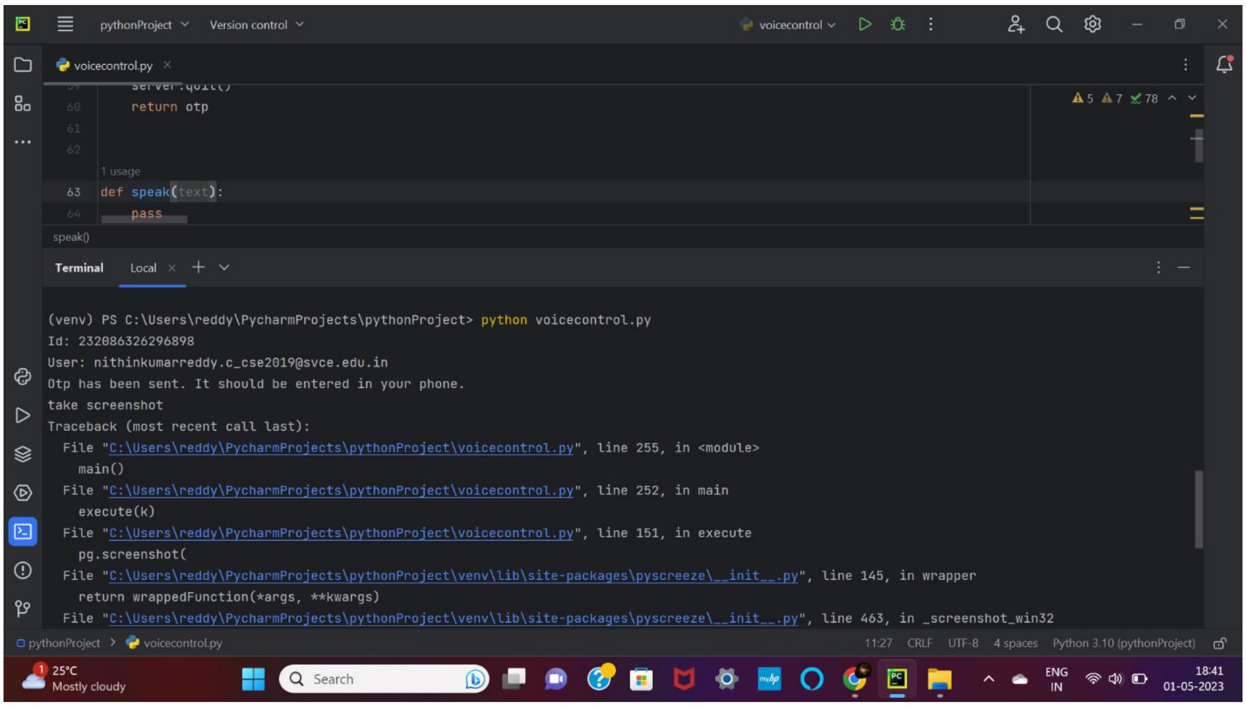
8.1 UNIT TESTING:

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Voice Recognition: Test the system's ability to accurately recognize voice commands. This can be done by providing the system with a variety of voice commands and ensuring that it responds appropriately.

- **Connectivity:** Test the connectivity of the system to Firebase. This can be done by ensuring that the system is able to access and retrieve data from Firebase.
- **Data Storage:** Test the system's ability to store data in Firebase. This can be done by adding data to the system and ensuring that it is stored correctly in Firebase.
- **Data Retrieval:** Test the system's ability to retrieve data from Firebase. This can be done by retrieving data that has been previously added to the system and ensuring that it is retrieved correctly.

Case 1:



```
    server.quit()
    return otp


def speak(text):
    pass
```

```
(venv) PS C:\Users\reddy\PycharmProjects\pythonProject> python voicecontrol.py
Id: 232086326296898
User: nithinkumarreddy.c_cse2019@svce.edu.in
Otp has been sent. It should be entered in your phone.
take screenshot
Traceback (most recent call last):
  File "C:\Users\reddy\PycharmProjects\pythonProject\voicecontrol.py", line 255, in <module>
    main()
  File "C:\Users\reddy\PycharmProjects\pythonProject\voicecontrol.py", line 252, in main
    execute(k)
  File "C:\Users\reddy\PycharmProjects\pythonProject\voicecontrol.py", line 151, in execute
    pg.screenshot(
  File "C:\Users\reddy\PycharmProjects\pythonProject\venv\lib\site-packages\pyscreeze\__init__.py", line 145, in wrapper
    return wrappedFunction(*args, **kwargs)
  File "C:\Users\reddy\PycharmProjects\pythonProject\venv\lib\site-packages\pyscreeze\__init__.py", line 463, in _screenshot_win32
```

Fig 8.1.1: Error for "take screenshot" command

8.2 INTEGRATION TESTING:

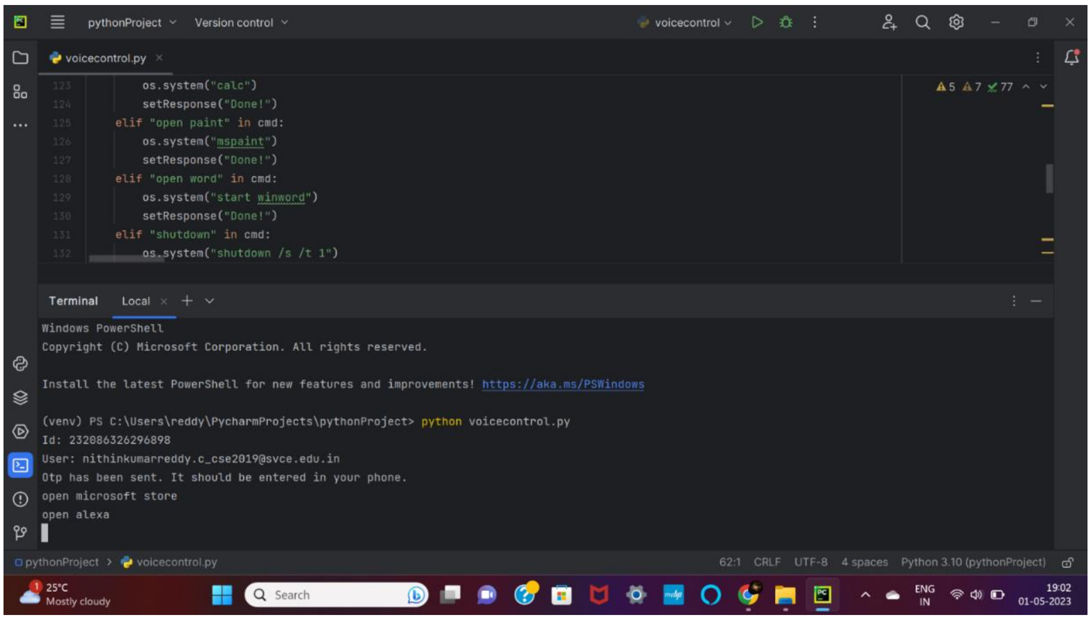
Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

- **Usability:** Test the system's usability by ensuring that it is easy to use and understand for users with varying levels of technical knowledge.
- **Integration:** Test the system's ability to integrate with other devices and services. This can be done by integrating the system with other smart devices and ensuring that it is able to control them appropriately.

8.3 SYSTEM TESTING :

After performing the validation testing, the next step is output testing of the proposed system. Since the system cannot be useful if it does not produce the required output. Asking, the user about the format in which the system is required, test output displayed or generated by the system under consideration. The output format is considered in two ways. One is on screen and another one is printed format. The output format on the screen is found to be correct as the format was designed in the system phase according to the user’s needs. For the hardcopy, the output comes according to the specifications requested by the user. Here the output testing does not give any result.

TEST CASES:



```
        os.system("calc")
        setResponse("Done!")
    elif "open paint" in cmd:
        os.system("mspaint")
        setResponse("Done!")
    elif "open word" in cmd:
        os.system("start winword")
        setResponse("Done!")
    elif "shutdown" in cmd:
        os.system("shutdown /s /t 1")
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

(venv) PS C:\Users\reddy\PycharmProjects\pythonProject> python voicecontrol.py
Id: 232086326296898
User: nithinkumarreddy.c_cse2019@svce.edu.in
Otp has been sent. It should be entered in your phone.
open microsoft store
open alexa
```

Fig 8.4.1: Error for unknown commands

Test cases Model building:

- **Command Execution:** Test the system's ability to execute commands based on voice commands received. This can be done by providing the system with voice commands and ensuring that it executes the appropriate command.
- **Error Handling:** Test the system's ability to handle errors. This can be done by intentionally providing the system with incorrect or invalid voice commands and ensuring that it responds appropriately.
- **Security:** Test the security of the system by ensuring that only authorized users are able to access and control the system.
- **Scalability:** Test the system's ability to handle many users and commands. This can be done by simulating many users and voice commands and ensuring that the system is still able to respond appropriately.

CHAPTER 9

SOURCE CODE

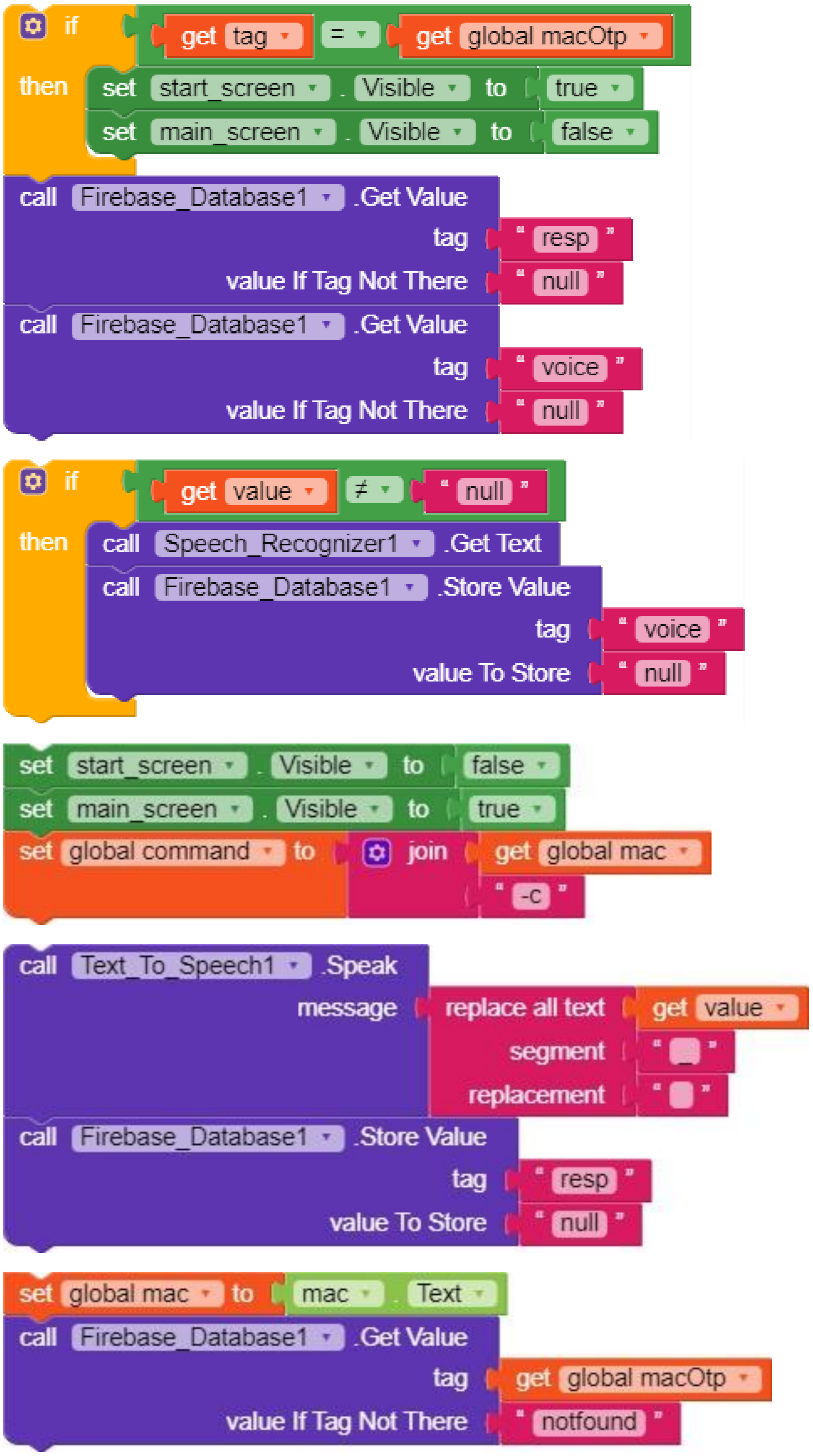
9.1 Python Code:

```
import firebase_admin
from firebase_admin import credentials
from firebase_admin import db
import webbrowser as wb
import pyautogui as pg
import os
import pywhatkit as kit
import uuid
import psutil
from random import randint
import keyboard
import time
import warnings
import smtplib
import random
import subprocess

mac = str(uuid.getnode())
print("Id:",mac)

#engine = pyttsx3.init()

a = {
 "type": "service_account",
 "project_id": "jarvisforwindows",
 "private_key_id": "6687f62ff6426abd234bce8f8b23e0d06dcc5eb8",
 "private_key": "-----BEGIN PRIVATE KEY-----
\nMIIEvQIBADANBgkqhkiG9w0BAQEFAASCBKcwggSjAgEAAoIBAQCIr6xJeqt2mQPH\nTlCkB
OXoNNA24kKj2zaH70UQUp7DobBM86Vnwe/TCWKaV1VvHwB1xmoFHFY2gOxo\nCjGkDBcWI
G2+K/EhWseqY/VOGxY7ICpirrLGnncwdEQ29mQReop/OgosFj0vuxRl\nJr9Ky29mJd5WE1PJZv6h
qJJyQUh4V2py9An3juAuc9Kf8r+wzNbX0eP8Sge1yXhv\naQQTXdbWTUtOq8O0uHnrsVUdId94nE
b8ATaqMMZCK3UJwWDTi/ilt4ORAhe2+Fbz\nfQeT5pe+rW62qnOtbad6AjIPzrA4Fnkw1pc+VvGP
MdXBi+WzqQfI9LR+pD8ic0FW\noGzaCuoFAgMBAAECggEAPjyC/YgR53PfoRHbLPuf9V2Kytq2
DibDyxxavYZpDNmj\nzT4JO9e1y8kTsQP//hNHIdlAr+gJp1KHkg3GMZRhtKz6WyNl7VKI5GAUM
0apFi3c\nrsct1rsTuSfPYZlJ0h2ST0DNnepYXNHZhP8iDbvYktG+TUIKngM8AL6hQ83O4h/9\nBxw
FAZ7HQxp2Sc3yTMFwLOJq22LwTPkt2T3bbYJaAQkDlUVRb7/FsvD9YRSdB25f\n2GIuuSAvJCo1
RAJqynhHsR661ybz9yPpo6ODVCfg7yU9Av2JYhx8RFarkzfWgy0W\nWCXOLptdTVLACxYspiEZ9
tOFSdN69b+LnKGPuqtpVwKBgQDAtxZzVPjQzAXAP6bJ\na+m8WWRzBVAeAoOE32gLbSqM8
W74qy6SMoO/sNWHN6VYlOsDBkfZGZdOVcOZMTHF\nHTN5B6ZJffNL9vJ0RADSZX8IbDxe5x
o33Wxr/IrRbJLdDliLCriekfBizMgnaZVt\n7i/aIGCptpjOE0VoffF+NwgjdwKBgQC1kmuYPhGKyhH9
mH24DqCmzdRCAnPK17Ma\nVNmnDCVVusnftm+9ZXTFDOh9YwAw+r/Koluio83jRwJco8gfruP
WGkc5GyUtZ0Zc\nyO3U9D7/ccDZULx1jUU0VASHEWalCcntgOKkD2Ag5m9zlnredXAv00e1wK
W3+Iw7\n2sDvsKMlYwKBgGzxE0fvaRjfvQCI+wgycNeA0UAUaM4OLbsXcAHFnKBAe7MnUhRj\
nagcbOBpQYrBIvvHeww7/YIFwCjq3jKMZdtecc2xoPvlaiIUhTDWkGsPwK9CaZD/g\ndEI3aWIqNn
uweG1hiixZ48J2cU+WaFrUo0hztTE7f/Y+/qWrTLFE+tzRAoGAP4Rf\nGzzrSg/yRzJnGFIVpQRv8j+
FXjoir11rXmKDVQAoype5cxngxWYElohhcsDlAu/U\n+oou5gjbLKkmwt6dWTKMI8/5K27rUF4Bx
PNEbnvOqLbzlnO699lEVDOkIqvP9cOW\nhSnTyO6Tom3LwbJ3cmOIvG4OCtNpyy51O+Qgzl0CgY
EAlELfD3Yhi04A+rWsQ/xC\nbVtvSIRY6Mk7InFHKDKBzk2d8qswCHKfJszy1k9JK4jrohhyXxq3z
AiOLbHadwQ2\nPAtYL4Pn4ARhmBSm4UOTtyx0q4wjJ/JASMopUVuOQnLbWveJ9bXCJb7dKrke
O24S\nDv8MuX8hNhRAyXT/+2ZrT7Y=\n-----END PRIVATE KEY-----\n",
 "client_email": "firebase-adminsdk-l03aj@jarvisforwindows.iam.gserviceaccount.com",
 "client_id": "107855943164914987915",
 "auth_uri": "https://accounts.google.com/o/oauth2/auth",
 "token_uri": "https://oauth2.googleapis.com/token",
 "auth_provider_x509_cert_url": "https://www.googleapis.com/oauth2/v1/certs",
 "client_x509_cert_url":         "https://www.googleapis.com/robot/v1/metadata/x509/firebase-adminsdk-
l03aj%40jarvisforwindows.iam.gserviceaccount.com"
}

cred = credentials.Certificate(a)
firebase_admin.initialize_app(cred, {
  "databaseURL": "https://jarvisforwindows-default-rtdb.firebaseio.com/"
})

ref = db.reference('/App/')

def sendOtp(email):
  server = smtplib.SMTP('smtp.gmail.com', 587)
  server.starttls()

  # Login to the SMTP server
  server.login('systemcontrol910@gmail.com','mbqjhjyfgzizowaz')

  # Send the email
  otp = random.randint(1000,9999)
  msg = "Hello, Your otp is "+str(otp)
  server.sendmail('systemcontrol910@gmail.com', email, msg)

  # Disconnect from the SMTP server
  server.quit()
  return otp

def speak(text):
  pass
def setResponse(text):
  text = text.replace(" ", "_")
  ref.update({
    'resp': text
  })
def execute(cmd):
  if "type" in cmd:
    cmd = cmd.replace("type ", " ")
    pg.typewrite(cmd)
  elif "open youtube" in cmd:
    wb.open("https://www.youtube.com/")
    setResponse("Done!")
  elif "refresh" in cmd:
    pg.hotkey("ctrl","r")
  elif "skip" in cmd:
    pg.click(x=124,y=984)
    time.sleep(0.5)
    pg.click(x=605,y=626)
    time.sleep(0.5)
    pg.click(x=1257,y=760)
  elif "open google" in cmd:
    wb.open("https://www.google.com/")
    setResponse("Done!")
  elif "open facebook" in cmd:
    wb.open("https://www.facebook.com/")
    setResponse("Done!")
  elif "open instagram" in cmd:
    wb.open("https://www.instagram.com/")
    setResponse("Done!")
  elif "open twitter" in cmd:
    wb.open("https://www.twitter.com/")
    setResponse("Done!")
  elif "open github" in cmd:
    wb.open("https://www.github.com/")
    setResponse("Done!")
  elif "open stackoverflow" in cmd:
    wb.open("https://www.stackoverflow.com/")
    setResponse("Done!")
  elif "open linkedin" in cmd:
    wb.open("https://www.linkedin.com/")
    setResponse("Done!")
  elif "open gmail" in cmd:
    wb.open("https://mail.google.com/")
    setResponse("Done!")
  elif "open gdrive" in cmd:
    wb.open("https://drive.google.com/")
    setResponse("Done!")
```

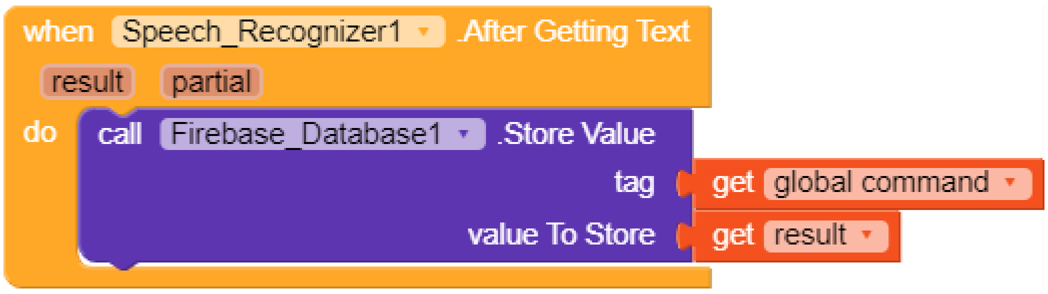
```
 elif "lock" in cmd:
   os.system("rundll32.exe user32.dll,LockWorkStation")
   setResponse("Done!")
 elif "open notepad" in cmd:
   subprocess.Popen(["notepad.exe"])
   setResponse("Done!")
 elif "open calculator" in cmd:
   os.system("calc")
   setResponse("Done!")
 elif "open paint" in cmd:
   os.system("mspaint")
   setResponse("Done!")
 elif "open word" in cmd:
   os.system("start winword")
   setResponse("Done!")
 elif "shutdown" in cmd:
   os.system("shutdown /s /t 1")
   setResponse("Done!")
 elif "restart" in cmd:
   os.system("shutdown /r /t 1")
   setResponse("Done!")
 elif "youtube" in cmd:
   cmd = cmd.replace("youtube ", "")
   kit.playonyt(cmd)
   setResponse("Done!")
 elif "sleep" in cmd:
   # python code to sleep the windows
   os.system("rundll32.exe powrprof.dll,SetSuspendState 0,1,0")
   setResponse("Done!")
 elif "battery level" in cmd:
   per = psutil.sensors_battery().percent
   resp = "Battery Level is " + str(per) + " percent"
   setResponse(resp)
   speak(resp)
 elif "take screenshot" in cmd:
   pg.screenshot("C:\\Users\\Ganesh\\OneDrive\\Pictures\\Screenshots\\"+str(randint(10000000,
99999999)) + ".png")
   setResponse("Done!")
 elif "mute" in cmd:
   pg.press("volumemute")
   setResponse("Done!")
 elif "increase volume" in cmd:
   pg.press("volumeup")
   setResponse("Done!")
 elif "decrease volume" in cmd:
   pg.press("volumedown")
```

```
    setResponse("Done!")
  elif "open code" in cmd:
    os.system("code")
    setResponse("Done!")
  elif "open command prompt" in cmd:
    os.system("cmd")
    setResponse("Done!")
  elif "search" in cmd:
    cmd = cmd.replace("search ", "")
    kit.search(cmd)
    setResponse("Done!")
  elif "close" in cmd:
    pg.hotkey("ctrl", "w")
    setResponse("Done!")
  elif "minimize" in cmd:
    pg.hotkey("win", "m")
    setResponse("Done!")
  elif "maximize" in cmd:
    pg.hotkey("win", "m")
    setResponse("Done!")
  elif "open file manager" in cmd or "open explorer" in cmd:
    pg.hotkey("win", "e")
    setResponse("Done!")
  elif "type my speech" in cmd:
    pg.hotkey("win", "h")
    setResponse("Done!")
  elif "select all" in cmd:
    pg.hotkey("ctrl", "a")
    setResponse("Done!")
  elif "copy" in cmd:
    pg.hotkey("ctrl", "c")
    setResponse("Done!")
  elif "paste" in cmd:
    pg.hotkey("ctrl", "v")
    setResponse("Done!")
  elif "cut" in cmd:
    pg.hotkey("ctrl", "x")
    setResponse("Done!")
  elif "undo" in cmd:
    pg.hotkey("ctrl", "z")
    setResponse("Done!")
  elif "redo" in cmd:
    pg.hotkey("ctrl", "y")
    setResponse("Done!")
  elif "show desktop" in cmd:
    pg.hotkey("win", "d")
    setResponse("Done!")
  elif "switch window" in cmd:
    pg.hotkey("alt", "tab")
    setResponse("Done!")
  elif "go back" in cmd:
    pg.hotkey("alt", "left")
    setResponse("Done!")
  elif "go forward" in cmd:
    pg.hotkey("alt", "right")
    setResponse("Done!")

def main():
  x = ref.get()
  if mac + "-e" in x:
    print("User: "+x[mac+"-e"])
    otp = sendOtp(x[mac+"-e"])
    print("Otp has been sent. It should be entered in your phone.")
    ref.update({
      mac+"-o": str(otp)
    })
  else:
    email = input("Enter your email: ")
    otp = sendOtp(email)
    userOtp = int(input("Enter Otp: "))
    if otp == userOtp:
      print("OTP Verification Success")
      ref.update({
        mac+"-e" : email,
        mac+"-o" : "null",
        mac+"-c" : "null",
      })
      print("Update success")
    else:
      print("Incorrect OTP")
    main()
  while True:
    k = ref.get()[mac+"-c"].lower()
    if k!="null":
      ref.update({
        mac+"-c": "null"
      })
      k = k[1:-1]
      print(k)
      execute(k)

main()
```

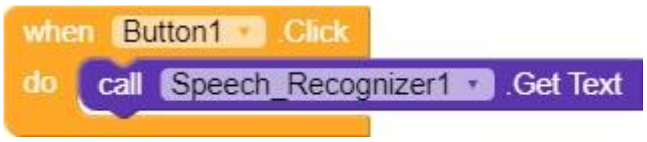
9.2 Scratch Code:



```
if   get tag = get global macOtp
then set start_screen . Visible to true
     set main_screen . Visible to false
call Firebase_Database1 .Get Value
                    tag  " resp "
     value If Tag Not There  " null "
call Firebase_Database1 .Get Value
                    tag  " voice "
     value If Tag Not There  " null "

if   get value ≠ " null "
then call Speech_Recognizer1 .Get Text
     call Firebase_Database1 .Store Value
                         tag  " voice "
              value To Store  " null "

set start_screen . Visible to false
set main_screen . Visible to true
set global command to  join  get global mac
                             " -c "

call Text_To_Speech1 .Speak
               message  replace all text  get value
                                 segment  "  "
                             replacement  "  "
call Firebase_Database1 .Store Value
                    tag  " resp "
         value To Store  " null "

set global mac to  mac . Text
call Firebase_Database1 .Get Value
                    tag  get global macOtp
     value If Tag Not There  " notfound "
```

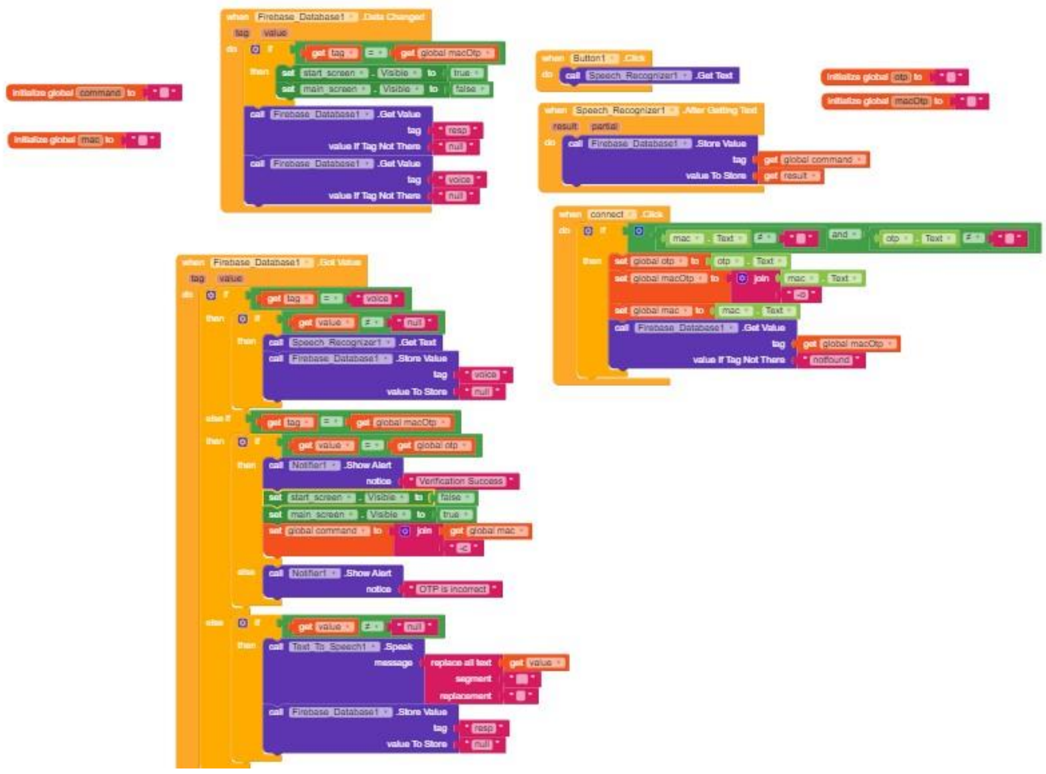


```
when Speech_Recognizer1 .After Getting Text
  result  partial
do  call Firebase_Database1 .Store Value
                        tag  get global command
             value To Store  get result
```



```
when Button1 .Click
do  call Speech_Recognizer1 .Get Text
```

Complete Code:



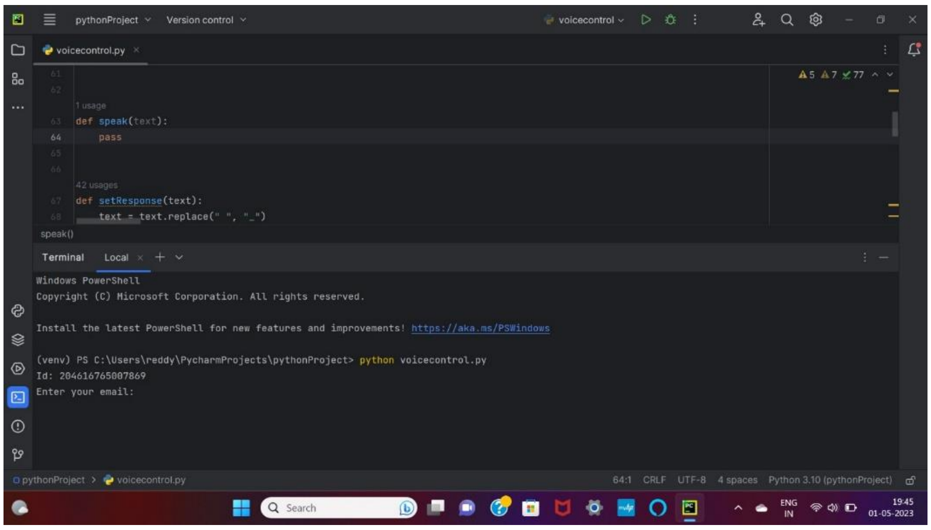
```
initialize global command to " "

initialize global mac to " "

when Firebase_Database1 .Data Changed
  tag  value
do  if  get tag = get global macOtp
    then  set start_screen . Visible to true
          set main_screen . Visible to false
    call Firebase_Database1 .Get Value
                         tag  "resp"
        value If Tag Not There  "null"
    call Firebase_Database1 .Get Value
                         tag  "voice"
        value If Tag Not There  "null"

when Firebase_Database1 .Got Value
  tag  value
do  if  get tag = "voice"
    then  if  get value ≠ "null"
          then  call Speech_Recognizer1 .Get Text
                call Firebase_Database1 .Store Value
                                     tag  "voice"
                          value To Store  "null"
    else if  get tag = get global macOtp
    then  if  get value = get global otp
          then  call Notifier1 .Show Alert
                               notice  "Verification Success"
                set start_screen . Visible to false
                set main_screen . Visible to true
                set global command to  join  get global mac
                                             "-c"
          else  call Notifier1 .Show Alert
                               notice  "OTP is incorrect"
    else  if  get value ≠ "null"
          then  call Text_To_Speech1 .Speak
                    message  replace all text  get value
                                      segment  " "
                                  replacement  " "
                call Firebase_Database1 .Store Value
                                     tag  "resp"
                          value To Store  "null"

when Button1 .Click
do  call Speech_Recognizer1 .Get Text

when Speech_Recognizer1 .After Getting Text
  result  partial
do  call Firebase_Database1 .Store Value
                        tag  get global command
             value To Store  get result

initialize global otp to " "

initialize global macOtp to " "

when connect .Click
do  if  mac . Text ≠ " "  and  otp . Text ≠ " "
    then  set global otp to otp . Text
          set global macOtp to  join  mac . Text
                                      "-o"
          set global mac to mac . Text
          call Firebase_Database1 .Get Value
                               tag  get global macOtp
              value If Tag Not There  "notfound"
```

Fig 9.2.1 Scratch Code

CHAPTER 10

SCREEN LAYOUTS

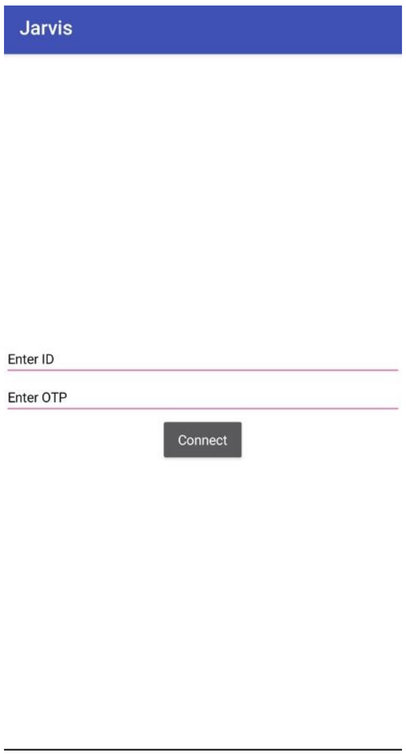
1. Registration



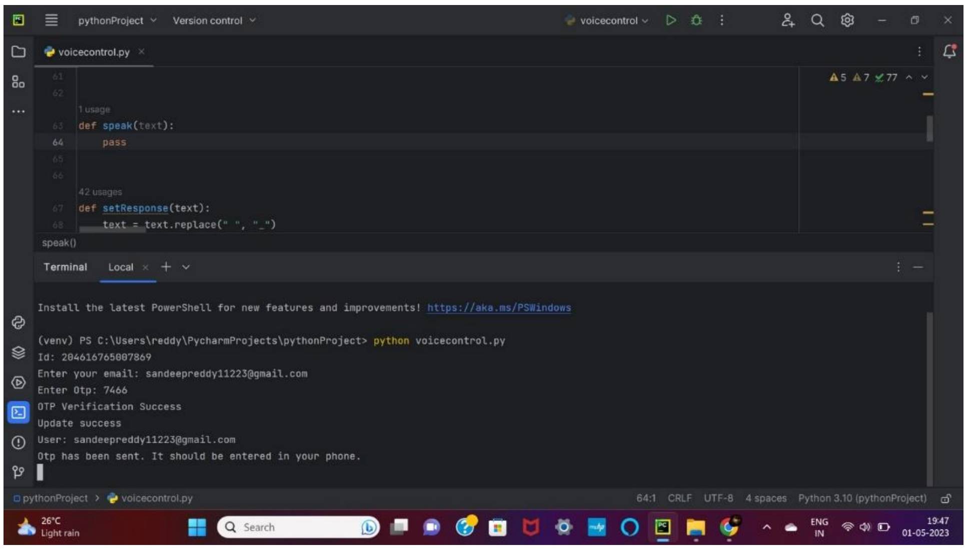
```
1 usage
def speak(text):
    pass


42 usages
def setResponse(text):
    text = text.replace(" ", "_")
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

(venv) PS C:\Users\reddy\PycharmProjects\pythonProject> python voicecontrol.py
Id: 204616765007869
Enter your email:
```

2. Login



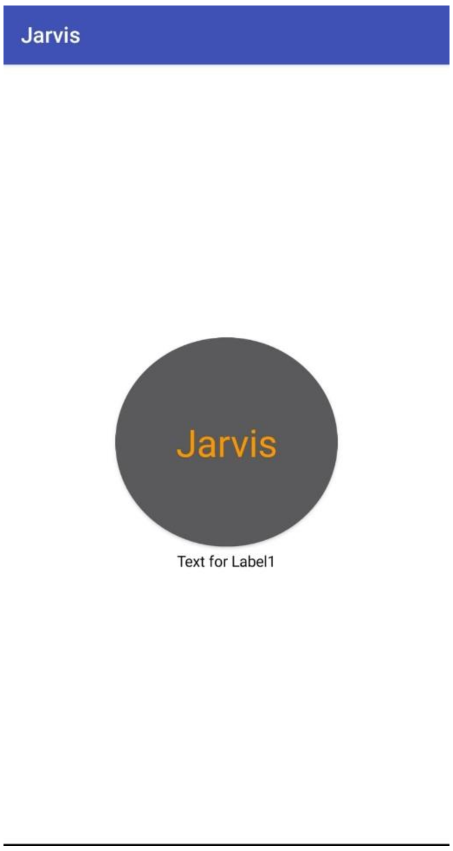
3. Authentication



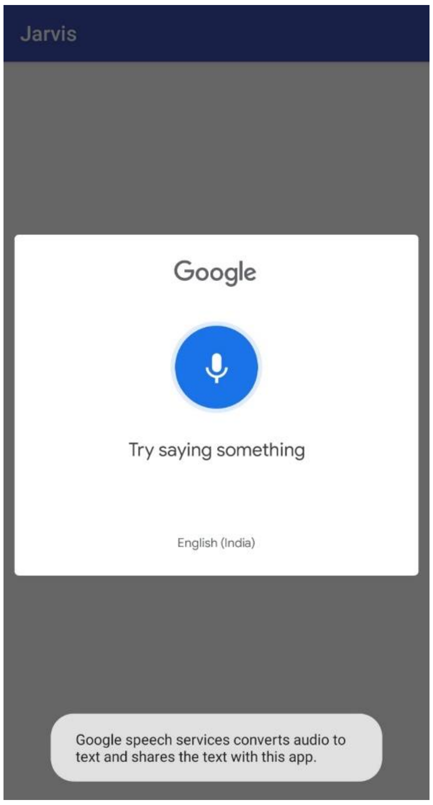
```
def speak(text):
    pass


def setResponse(text):
    text = text.replace(" ", "_")
```

```
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

(venv) PS C:\Users\reddy\PycharmProjects\pythonProject> python voicecontrol.py
Id: 204616765007869
Enter your email: sandeepreddy11223@gmail.com
Enter Otp: 7466
OTP Verification Success
Update success
User: sandeepreddy11223@gmail.com
Otp has been sent. It should be entered in your phone.
```

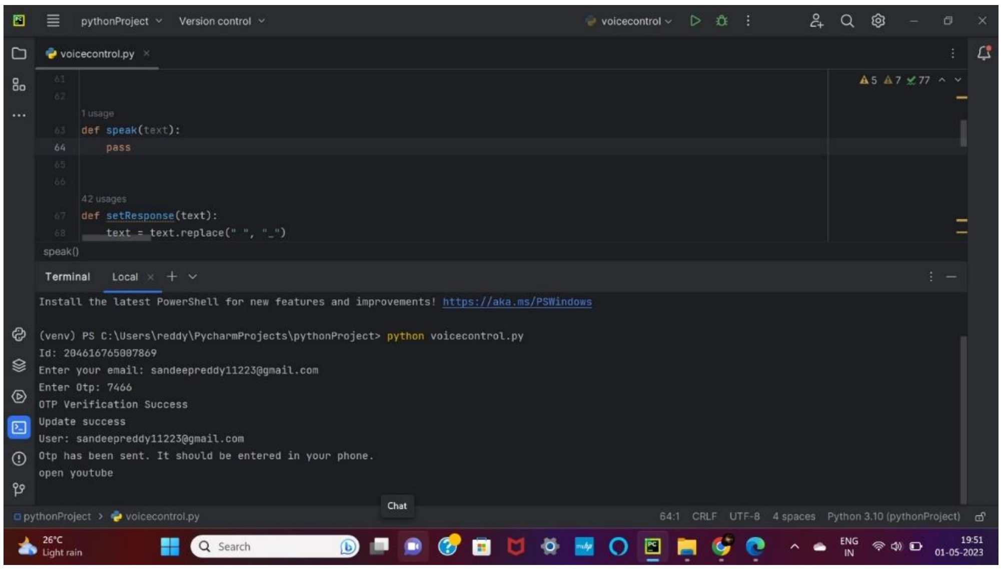
4. Home



5. Taking Voice command as input


Jarvis
Google
Try saying something
English (India)
Google speech services converts audio to text and shares the text with this app.

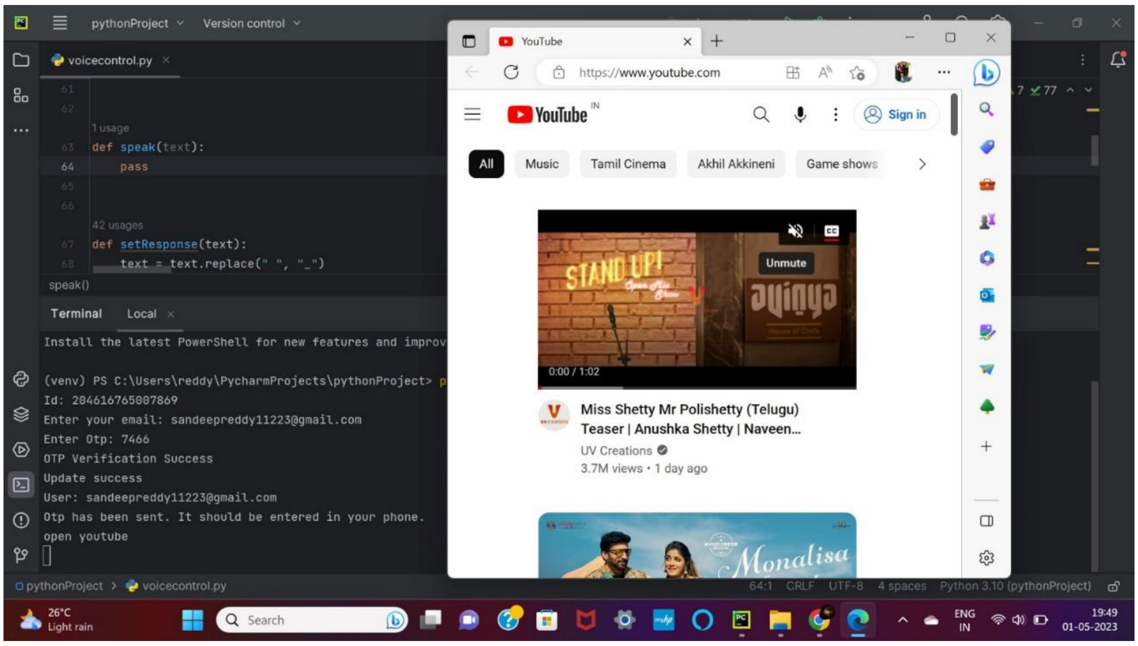
6. Input 1



```
def speak(text):
    pass


def setResponse(text):
    text = text.replace(" ", "_")
```

```
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

(venv) PS C:\Users\reddy\PycharmProjects\pythonProject> python voicecontrol.py
Id: 204616765007869
Enter your email: sandeepreddy11223@gmail.com
Enter Otp: 7466
OTP Verification Success
Update success
User: sandeepreddy11223@gmail.com
Otp has been sent. It should be entered in your phone.
open youtube
```

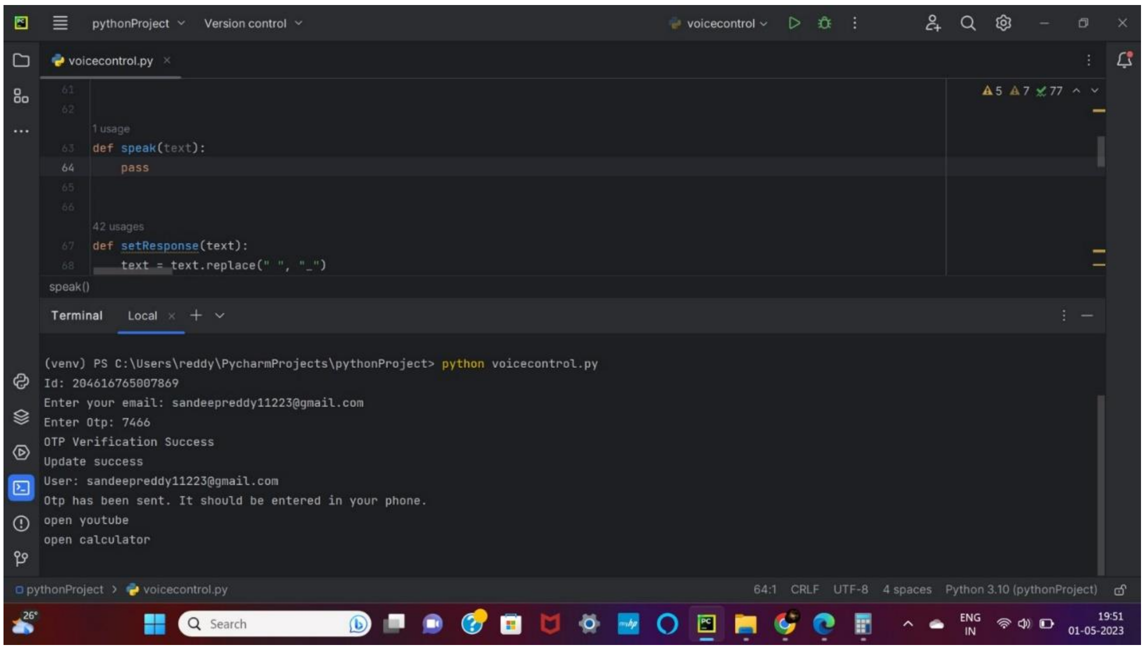
7. Output 1



```
1 usage
def speak(text):
    pass


42 usages
def setResponse(text):
    text = text.replace(" ", "_")
speak()
```

```
Install the latest PowerShell for new features and improv

(venv) PS C:\Users\reddy\PycharmProjects\pythonProject> p
Id: 204616765007869
Enter your email: sandeepreddy11223@gmail.com
Enter Otp: 7466
OTP Verification Success
Update success
User: sandeepreddy11223@gmail.com
Otp has been sent. It should be entered in your phone.
open youtube
```

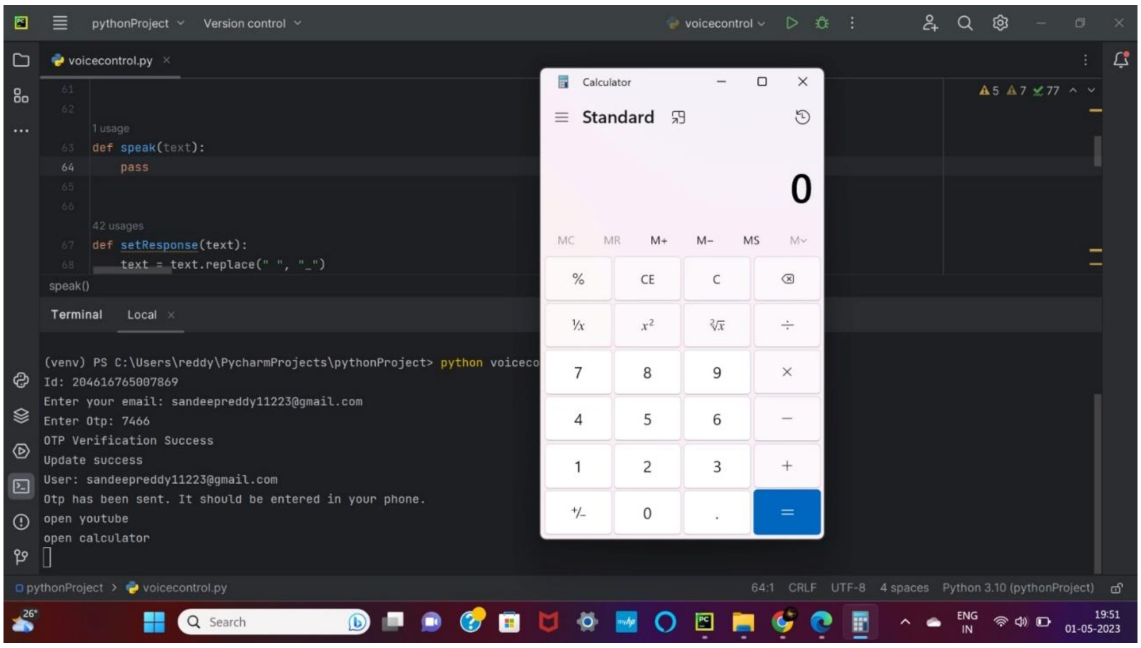
8. Input 2



```
1 usage
def speak(text):
    pass


42 usages
def setResponse(text):
    text = text.replace(" ", "_")
speak()
```

```
(venv) PS C:\Users\reddy\PycharmProjects\pythonProject> python voicecontrol.py
Id: 204616765007869
Enter your email: sandeepreddy11223@gmail.com
Enter Otp: 7466
OTP Verification Success
Update success
User: sandeepreddy11223@gmail.com
Otp has been sent. It should be entered in your phone.
open youtube
open calculator
```

9. Output 2



```
def speak(text):
    pass


def setResponse(text):
    text = text.replace(" ", "_")
```

```
(venv) PS C:\Users\reddy\PycharmProjects\pythonProject> python voiceco
Id: 204616765007869
Enter your email: sandeepreddy11223@gmail.com
Enter Otp: 7466
OTP Verification Success
Update success
User: sandeepreddy11223@gmail.com
Otp has been sent. It should be entered in your phone.
open youtube
open calculator
```

CHAPTER 11

CONCLUSION

The proposed system presents an intuitive and easy way of controlling a computer system remotely using voice commands. The system uses a mobile interface built on Scratch, which captures the voice commands and transmits them to the remote computer system using Firebase. The remote computer system executes the commands received from the mobile interface. The system performs well under various network conditions and is an improvement over traditional socket-based connection.

CHAPTER 12

FUTURE ENHAMCEMENTS

Explore additional functionalities that can be added to the system, such as integrating with other platforms, adding support for additional voice commands, and making the system more user-friendly. Continuously improve the system to meet the evolving needs of the users.

CHAPTER 13

BIBILOGRAPHY

Good Teachers are worth more than a thousand books. We have them in Our Department.

References Made From:

1. Sarikaya, R.: The technology behind personal digital assistants. IEEE Signal Process. Mag.34,67–81 (2017).
2. Tsiao, J.C.-S., Tong, P.P., Chao, D.Y.: Natural-Language Voice-Activated Personal Assistant, United States Patent (10), Patent No.: US 7,216,080 B2 (45), 8 May 2007.
3. Sirbi, K., Patankar, A.J.: Personal assistant with voice recognition intelligence. Int. J. Eng.Res. Technol. 10(1), 416–419 (2017). ISSN 0974-3154.
4. Cowan, B.R.: What can i help you with?: infrequent users experiences of intelligent personal assistants. In: 2015 IEEE 10th International Conference on Industrial and Information Systems, ICIIS 2015, Sri Lanka (2015).
5. Weeratunga, A.M., Jayawardana, S.A.U., Hasindu, P.M.A.K, Prashan, W.P.M., Thelijjagoda, S.: Project Nethra - an intelligent assistant for the visually disabled to interact with internet services. In: 2015 IEEE 10th International Conference on Industrial and Information System (2015).
6. Lingyan Bi, Weining Wang, Haobin Zhong, Wenxuan Liu, "Design and Application of Remote-Control System Using Mobile Phone with JNI Interface", The 2008 International Conference of Embedded Software and Systems Symposia (ICESS2008), pp.416-419.2008.
7. Michael Spreitzenbarth, "Tools and processes for Forensic Analyses of smartphones and Mobile Malware", 6. GI FG SIDAR Graduierten- Workshop ueber Reaktive Sicherheit (SPRING), March 22nd, 2011.
8. Xinfang Lee, Chunghuang Yang, Shihjen Chen, Jainshing Wu, "Design and Implementation of Forensic System in Android Smart Phone", the 5th Joint Workshop on Information Security, 2009.
9. Enck, W., Ongtang, M., McDaniel, P.,"Understanding Android Security", Security & Privacy, IEEE, Jan.-Feb. 2009, Volume 7, Issue 1, pp.50-57.
10. T. Richardson, Q. Staford-Fraser, K. Wood and A. Hooper, \Virtual networking computing", Internet Computing, Vol. 2, No. 1, pp.33-38, 1998.
11. Sarikaya, R.: The technology behind personal digital assistants. IEEE Signal Process. Mag.34,67– 81 (2017).

12. Tsiao, J.C.-S., Tong, P.P., Chao, D.Y.: Natural-Language Voice-Activated Personal Assistant, United States Patent (10), Patent No.: US 7,216,080 B2 (45), 8 May 2007.
13. Sirbi, K., Patankar, A.J.: Personal assistant with voice recognition intelligence. Int. J. Eng.Res. Technol. 10(1), 416–419 (2017). ISSN 0974-3154.
14. Cowan, B.R.: What can i help you with?: infrequent users experiences of intelligent personal assistants. In: 2015 IEEE 10th International Conference on Industrial and Information Systems, ICIIS 2015, Sri Lanka (2015).
15. Weeratunga, A.M., Jayawardana, S.A.U., Hasindu, P.M.A.K, Prashan, W.P.M., Thelijjagoda, S.: Project Nethra - an intelligent assistant for the visually disabled to interact with internet services. In: 2015 IEEE 10th International Conference on Industrial and Information System (2015).
16. Lingyan Bi, Weining Wang, Haobin Zhong, Wenxuan Liu, "Design and Application of Remote-Control System Using Mobile Phone with JNI Interface", The 2008 International Conference of Embedded Software and Systems Symposia (ICESS2008), pp.416-419.2008.
17. Michael Spreitzenbarth, "Tools and processes for Forensic Analyses of smartphones and Mobile Malware", 6. GI FG SIDAR Graduierten-Workshop ueber Reaktive Sicherheit (SPRING), March 22th, 2011.
18. Enck, W., Ongtang, M., McDaniel, P.,"Understanding Android Security", Security & Privacy,IEEE, Jan.-Feb. 2009, Volume 7, Issue 1, pp.50-57.
19. T. Richardson, Q. Staford-Fraser, K. Wood and A. Hooper, \Virtual networking computing", Internet Computing, Vol. 2, No. 1, pp.33-38, 1998.

Sites Referred:

1. https://doi.org/10.1109/msp.2016.2617341
2. https://www.ijert.org/controlling-pclaptop-via-android-phone-android-remote-control
3. https://www.rroij.com/open-access/remote-access-to-pc-using-android-phone