Virtual Assistant

UCS-503 Software Engineering Project Report

Submitted by:

Ankush Gupta 3CO17 (101803384)

Hiten Gupta 3CO17 (101803382)

Ankush Sood 3CO17 (101803381)

BE Third Year, COE

Submitted to

Ms. Sawinder Kaur

Computer Science and Engineering Department,

TIET, Patiala

December 2020

TABLE OF CONTENTS

1. PROJECT OVERVIEW

As we know researches on AI system vastly increase since the last decade. Everyday new AI model/prototype has been introduced by the people around the globe. Today, everyone needs comfort and high-tech equipments. Everyone desires of more user friendly devices with which only by communication in natural language, they are able to make do the required tasks. With that point of view, google assistant, Alexa, IBM Watson, Sphinx, Cortana etc. come into play which provides a very user friendly interface or voice guided interface to their users. With the help of them, the user is able to do the tasks or operations in a desired way with great ease.

By following the same flow this project comes into play. This project is only one of its kinds since various features of different categories will be installed. We have seen that these types of system are dedicated to a specific task, but this system also function other tasks too. Our product is a Virtual Desktop Assistant to minimize the effort you need while using computer and to speed up so as to save time. The main USP is basically a voice-guided chat-based system which will include everything that you most likely do on your computer. We have used speech to text conversion technique to make this project. Providing answers to user about his/her queries from search engine over the internet and Personal assistant are some the functions of this project. Additional features could also be installed during the build.

The purpose of this project is to build a for web application which will facilitate the control over your computer with the help of Voice Recognition. This is an interesting concept and many people around the globe are working it. Today, time and conveniency are the two main things to which people are more sensitive, no one has the time to spoil and everyone wants to do things with minimal effort.

The scope of this project is to reduce the effort of human while using computer. The person just needs to give command through voice and the output will be visible to him. This Software will be a web application which will record the data input to the mic by voice commands. This software will be very useful for blind persons. Further this software has a huge advantage in healthcare sector widely where the spreading of infections could be there like in hospitals.

2. PROJECT REQUIREMENTS

2.1 Functional requirements

This subsection presents the identified functional requirements for the project. These requirements are categorized by use cases. For any use case there are specific requirements which are detailed below.

2.1.1 Authentication:

- Description and Priority:
  Description: The virtual assistant must have an authentication system as the assistant will be able to perform some privilege tasks also and for that user must not be a guest. So, a proper authentication system is there so that the guest might not enter in the system.

  Priority level: High.

- Stimulus/Response Sequences:
  - Firstly, the user will land on the login page where he/she needs to enter his /her email id.
  - With the help of otp-based authentication system, an otp will be send on the email of the user.
  - Next, the user needs to enter the otp received.
  - If the received otp is same as the system generated otp, then the user will be logged in. Otherwise it will show an error.
- Functional Requirements:

  For the authentication system to work properly, the below stated will be required:

  - Smtp lib must be installed in the system so that the mail functioning works.
  - Windows version must be 7 or higher to support the assistant.
  - An email account is required for the purpose to send the otp to the user.

2.1.2 Speech to Text (STT)

- Description and Priority:

  Description: Our virtual assistant is completely based on the voice recognition system. But the processing is done on the string or other data types by the system. For the processing of the command to be done, a mechanism or a functionality must be there that will be able to convert the voice to text so that the processing of command can be done.

Priority level: High.

- Stimulus/Response Sequences:
    - The user needs to send the command through voice only.
    - The key thing is that since we are using not using the NLP, therefore the user must send the voice in English language.
    - After the system recognizes the voice, this voice is fed to the google API where this voice is converted to text.
    - After this, the text is processed and functionality is performed.

- Functional Requirements:

  For the speech to text mechanism to work properly, the below stated will be required:

    - Pyaudio and Speech recognition must be installed on the system.
    - Windows version must be 7 or higher to support the assistant.
    - An internet connection is required for this functionality to work.\

2.1.3 Features and Tasks

- Description and Priority:

  Description: The virtual assistant is actually called an assistant only when it will be able to perform the necessary tasks on its own. There it is essential to have as many features as possible.

  Priority level: Low.

- Stimulus/Response Sequences:
    - Firstly, the user will speak up the task which he wants the virtual assistant to do.
    - Based on the task, various libraries will then be imported in the program.
    - These libraries will then perform the required action.
    - The output is then shown on the screen and will also be spoken up by the system speakers.

- Functional Requirements:

  For the various defined features to work properly, the below stated will be required:

    - Different libraries based on their features must be installed in the system.
    - Windows version must be 7 or higher to support the assistant.
    - Pyttsx3 must be installed on the system so that the feedback can also be conveyed through the speakers.

2.2 Non-Functional Requirements

2.2.1 Performance Requirements

Stating the fact that the system must perform as what every user expects, so in every action-response of the system, there are no immediate delays. Also, the performance hindrance can be showed during the user flow of instruction as large pauses may result in deactivation of recognizer and thus resulting in corrupted command. Also, when connecting to the server the delay is based on the distance of the 2 systems and the configuration between them so there is high probability that there will be or not a successful connection in less than 20 seconds.

2.2.2 Safety Requirements

Stating the fact that all user must be attachable to one server, so there would be appropriate control of the statistics and information. Also, in case of a potential loss of connection between the client and the server the clients will be logged off. When the client finishes its session (by pressing the finish button or by saying some keyword) then its progress is sent to the server and be logged. In case of a potential server breakdown, the web application can't be accessed.

2.2.3 Security Requirements

This program uses object-oriented mechanisms to protect its data passed using methods Also currently, there is no a security schema of this program. For authentication part, access to less secure applications must be there.

2.2.4 Portability

Testing across multiple platforms (Windows, OS X, Linux) and implementations of the python platform should ensure that code and external libraries are not platform or implementation-dependent.

2.2.5 Maintainability

Application code will be cohesive and have easily recognizable functionality. Classes will be abstract enough to facilitate changes in data structures. Class and function modularity should be implemented to avoid the need for major refactoring.

2.2.6 Portability

Testing across multiple platforms (Windows, OS X, Linux) and implementations of the python platform should ensure that code and external libraries are not platform or implementation-dependent.

3. STRUCTURED ANALYSIS

Structured analysis is a software engineering technique that uses graphical diagrams to develop and portray system specifications that are easily understood by users. These diagrams describe the steps that need to occur and the data required to meet the design function of a particular software.

3.1 Data Flow Diagrams

Data flow diagrams are used to graphically represent the flow of data in a business information system. DFD describes the processes that are involved in a system to transfer data from the input to the file storage and reports generation.

3.1.1 DFD Level 0 :

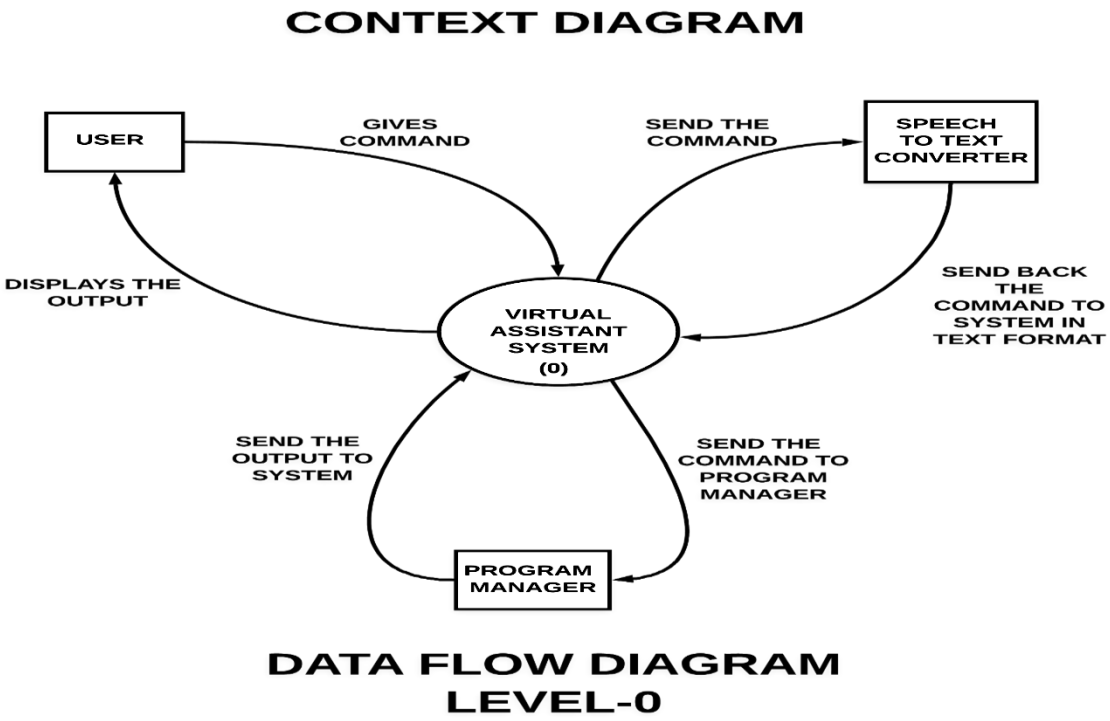
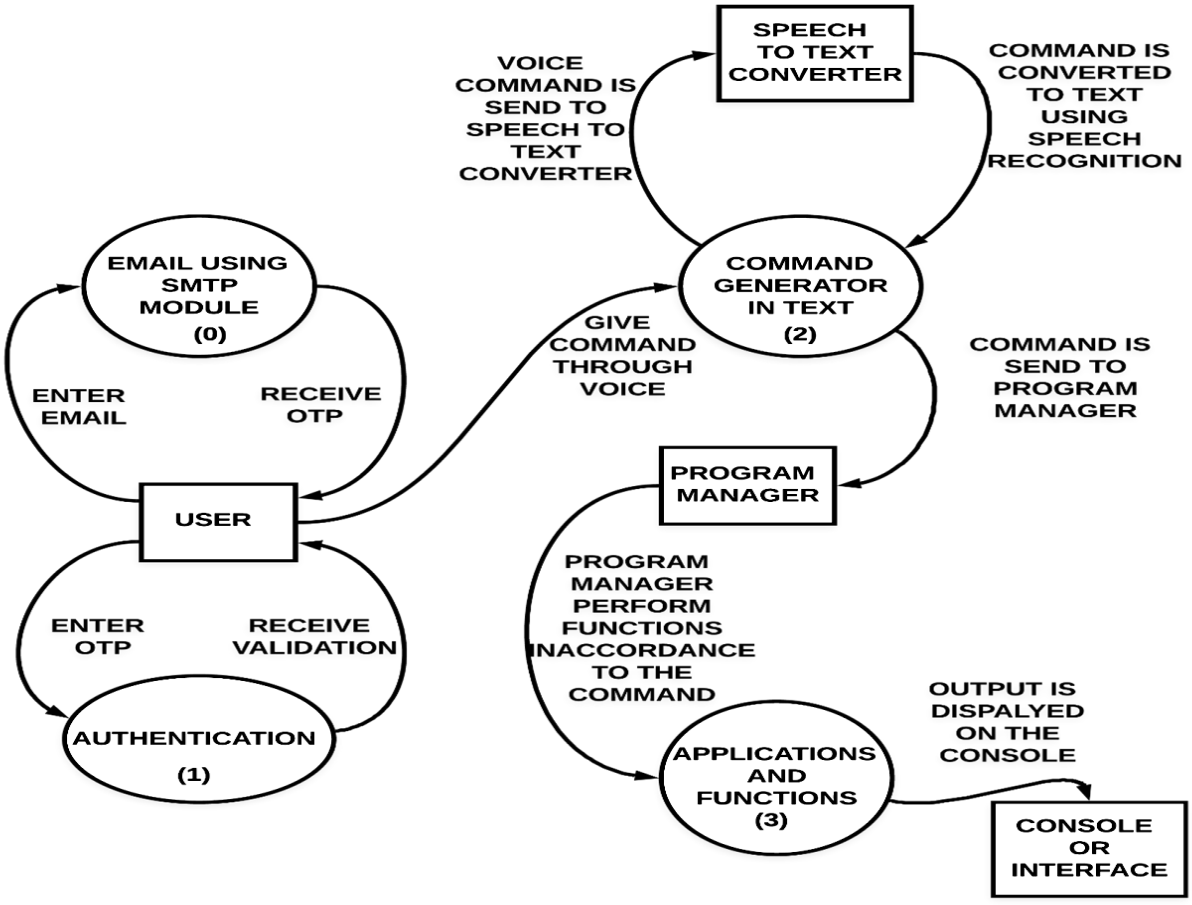


Fig 3.1.1: Context-Level DFD

- **Data Flow Diagram (Level -0) (Also known as Context Diagram)(fig 3.1.1):**
  - Data flow level 0 diagram includes only one but the main process behind the overall project. In our project, that main process is the actual system i.e. the virtual assistant system.
  - In our project, there are three main external entities i.e. user, speech to text converter and program manager.
  - Firstly, the user gives the command through voice to the virtual assistant.
  - This voice is then fed to the speech to text converter by virtual assistant.
  - Through the speech recognition library in the speech to text converter, this voice is converted to text and is fed to program manager by the virtual assistant.
  - In the program manager, the text instruction is processed and based on the instruction, various functionality is performed by the virtual assistant.
  - The output of the functionality is then fed to the console or the screen.

3.1.2 DFD Level 1:



DATA FLOW DIAGRAM
LEVEL-1

Fig 3.1.2: DFD Level 1

- **Data flow diagram (Level -1) (fig-3.1.2):**
  - In the data flow level 1 diagram, we have 4 processes and 4 external entities.
  - The four process are Email using smtp, authentication, command generator and application and functions respectively.
  - The four entities are the user, speech to text converter, program manager and the console or interface.
  - Firstly, the user needs to authenticate himself through an email based otp system.
  - The user needs to enter his/her email and through the smtp module, user receive the otp on his email.
  - The user now enters the otp and if the otp entered is same as the otp received, the user gets accessed to the virtual assistant. Otherwise, it shows an error.
  - Next**,** the user gives the command through voice and this voice is fed to speech to text converter.
  - Through the speech recognition module, this speech is converted to text and command is generated.
  - This command is then sent to the program manager where this command is processed and based on the functionality of the command, the output is displayed on the console or interface and the same will also be spoken up by the computer or desktop speakers.

4. OBJECT ORIENTED ANALYSIS

Object-oriented analysis is a technical approach for analyzing and designing an application, system, or business by applying object-oriented programming, as well as using visual modelling throughout the software development process to guide stakeholder communication and product quality.

4.1 Use Case Diagram

- Fig 4.1 represents the use case diagram of our virtual assistant.
- Here, the primary actor is the user as he/she is using the system.
- The various tasks and functionality performed by the virtual assistant will be tracked and will be done through this use case diagram.
- The use case diagram is of high priority because of the most important and key functionalities of the project it contains.
- One of the preconditions is that the user must be a valid authenticated entity.
- One of the postcondition is that the user must have to provide the input in the form of voice in English language.
- The flow of event will be as follows:
  - Firstly, the user will authenticate himself through an otp based authentication system.
  - As soon as the user will be provided access, microphone starts and speech recognition is allocated.
  - Next, the user has to give the input in the form of voice in English language.

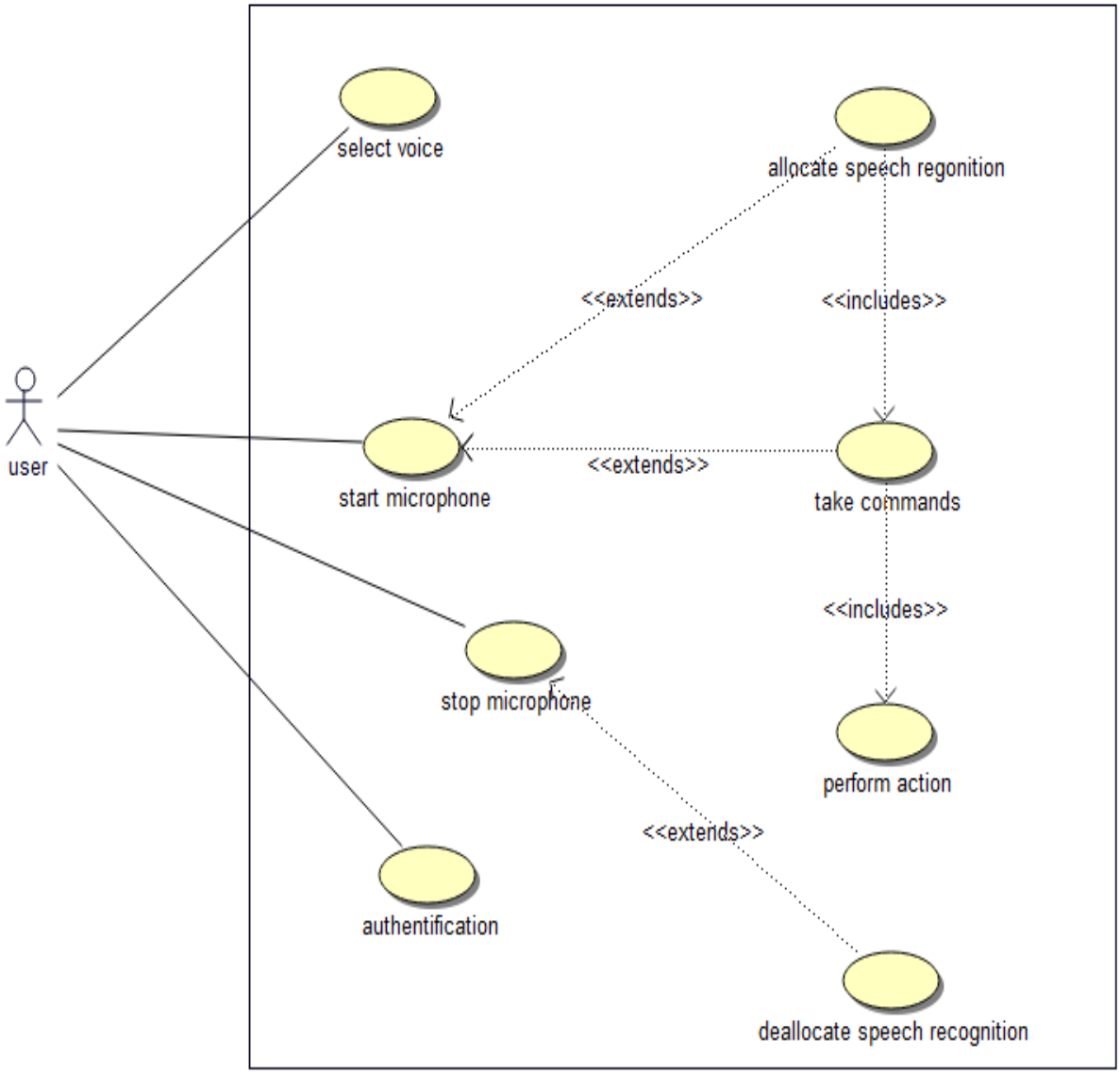


Fig 4.1 Use case Diagram

4.2 Use Case Template

Use Case Diagram for Virtual Desktop Assistant

Use Case:	Speech to text Converter and Task Performance
Primary Actor:	User
Goal in Context:	To convert the input voice into text and to execute the required task
Preconditions:	The system has login facility. The users must be logged in.
Trigger:	The user decides the kind of function he wants the system to perform.
Scenario:	1. The user logs onto the Desktop Assistant app. 2. The user enters his/her username. 3. The user enters the password. 4. The app displays all the major functionality. 5. The user inputs the instruction in natural language through his/her voice by pressing the speak button. 6. The input voice is processed on the backend and the desired output is displayed to the user through the windows notification panel.
Alternate flow:	1. The user selects stop is he thinks e might have given the wrong instruction.
Exceptions:	1. Username or password is incorrect: User re- enters the correct username and password. 2. Invalid functionality: Since there can be a lot of functionality and at this stage, the desired application supports limited feature. So might be the functionality that user wants is not there. 3. API not available. There are chances that output is not visible to user or the application terminated because of server maintainance or non availablility of API's.
Post conditions:	The desired output of the respective instruction is visible to the user through windows notification panel.
Supporting actor:	The trained deep learning model and some API's.

4.3 Activity Diagram

- Fig 4.3 depicts the activity diagram for the virtual assistant.
- At the initial state, the user needs to firstly provide his email and during the course as the mail is received by the user regarding the otp, he needs to enter that otp at the login page.
- If the otp is valid, he will be accessed to the virtual assistant. Else, he will be restricted.
- As soon as the access has been provided, the user can speak the command and the command will be processed after converting to text by the speech to text converter.
- If the command exists, the output will be received and the user again can continue the session but if the command doesn't exist, then an error message will be showed up.

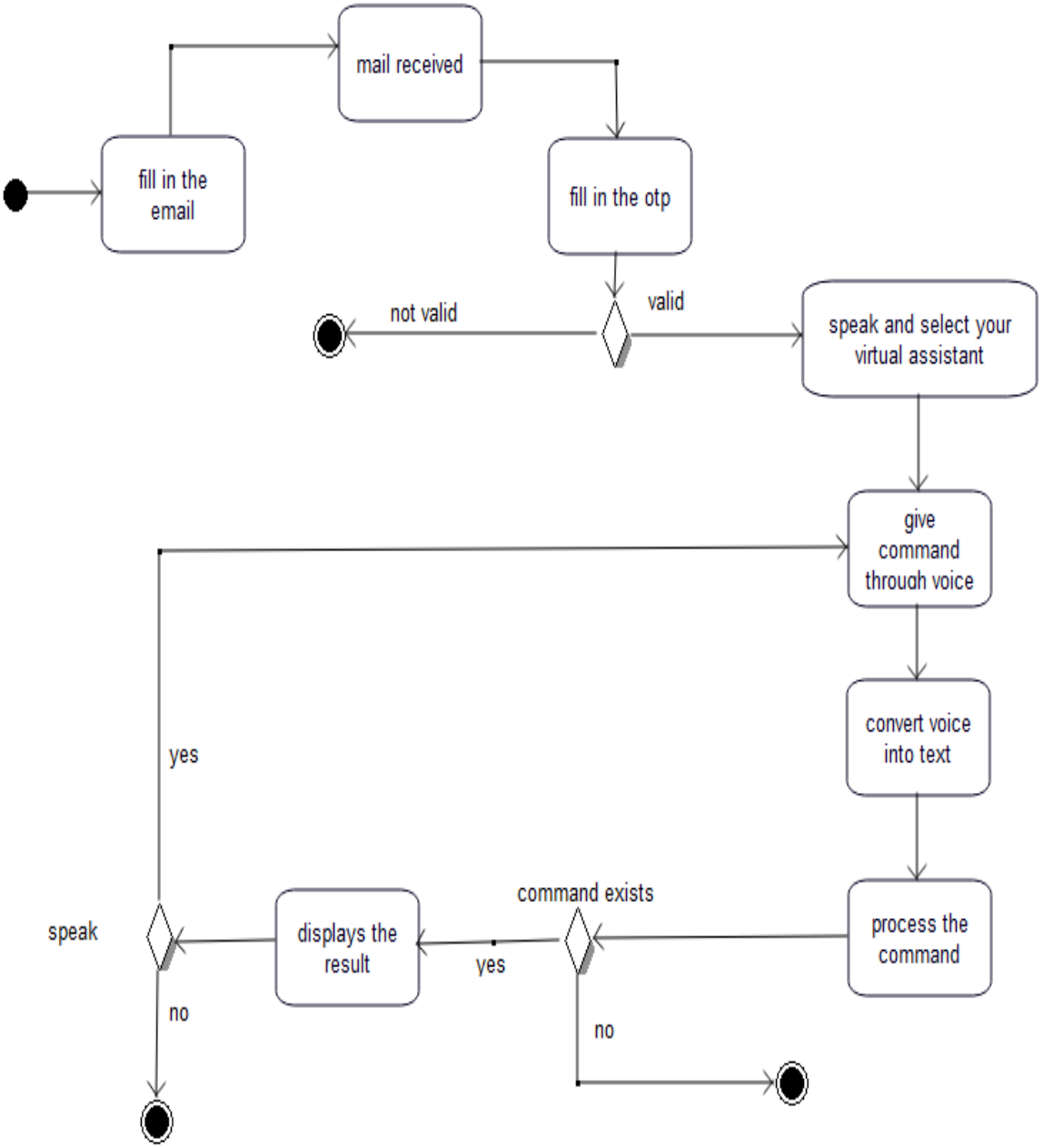


Fig 4.3: Activity diagram for Virtual Assistant

4.4 Class Diagram

- Fig 4.4 depicts the class diagram for the virtual assistant.
- The given class diagram has six classes.
- The relationship between the classes is as follows:
  - There is a one to many relationship between the user and voice class.
  - There is one to one relationship between the voice and the receiver class.
  - There is one to many relationship between the voice and tasks class.
  - There is zero to many relationship between the tasks and open_file class.
  - There is zero to many relationship between the tasks and site class.
- The flow of event will be as follows:
  - Firstly, the user will authenticate himself through an otp based authentication system.
  - As soon as the user will be provided access, microphone starts and speech recognition is allocated.
  - Next, the user has to give the input in the form of voice in English language.
  - The voice will be converted to text through speech recognition by google api.
  - The converted text will be then processed and the functionality that the command offers is performed.
  - The output will be spoken up by the system speakers as well as will be displayed at the output console window.

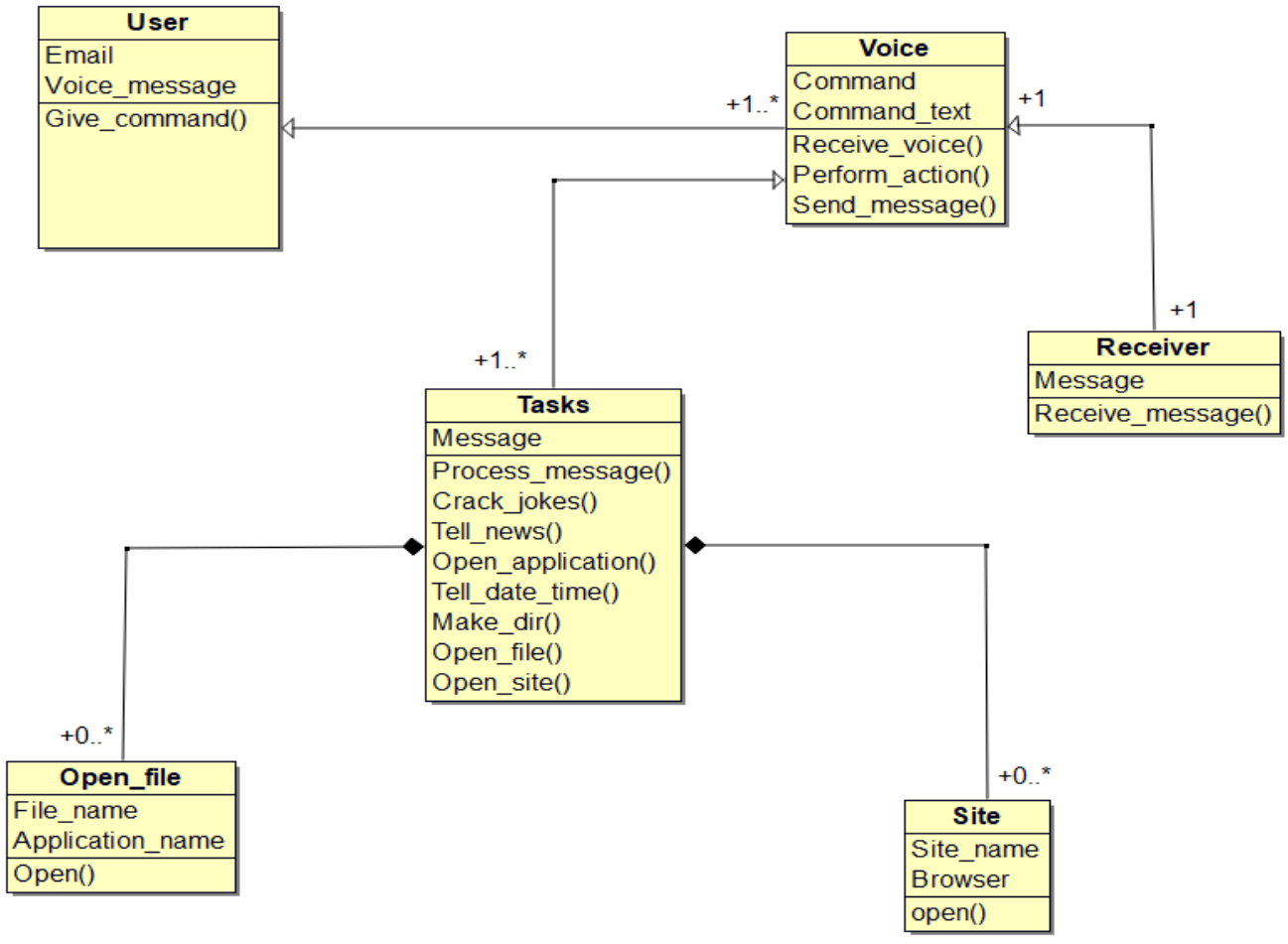


Fig 4.4: Class diagram for Virtual Assistant

4.5 Collaboration Diagram

A collaboration diagram, also known as a communication diagram, is an illustration of the relationships and interactions among software objects in the UML. These diagrams can be used to portray the dynamic behavior of a particular use case and define the role of each object.

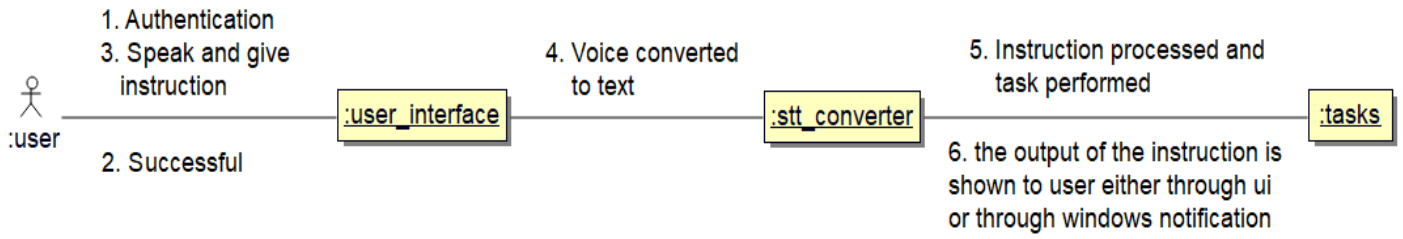


Fig 4.5: Collaboration diagram for Virtual Assistant

4.6 Sequence Diagram

A sequence diagram is a type of interaction diagram because it describes how—and in what order—a group of objects works together. These diagrams are used by software developers and business professionals to understand requirements for a new system or to document an existing process.

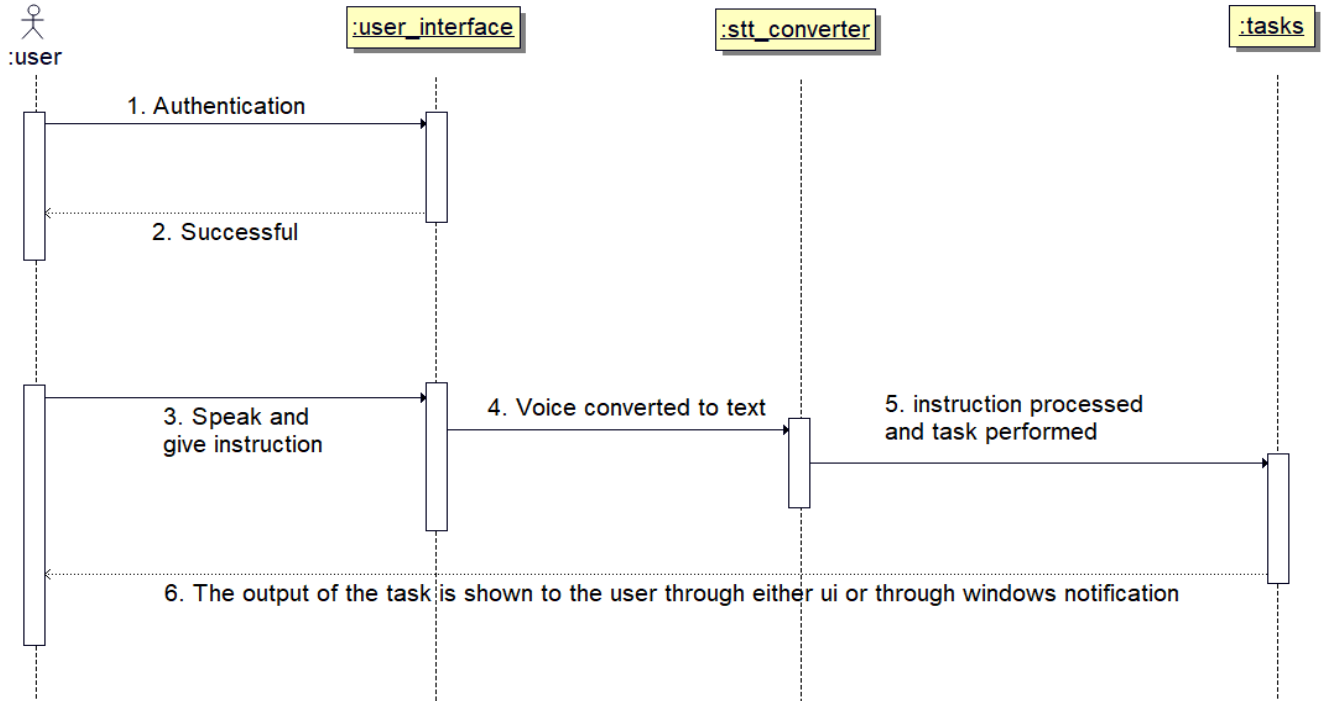


Fig 4.6: Sequence diagram for Virtual Assistant

4.7 State Chart Diagram

Statechart diagram is one of the five UML diagrams used to model the dynamic nature of a system. They define different states of an object during its lifetime and these states are changed by events.

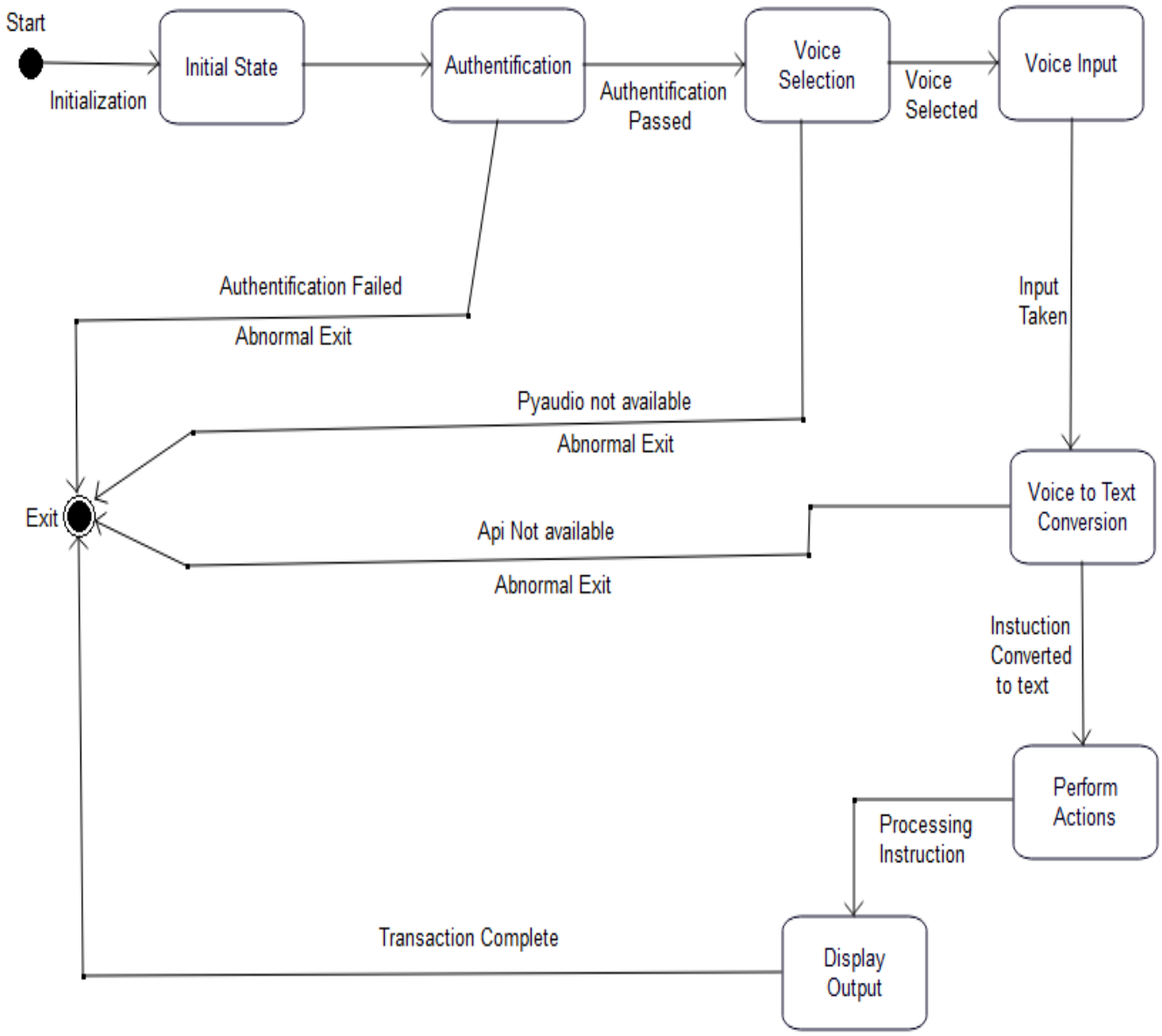


Fig 4.7: State diagram for Virtual Assistant

4.8 Component Diagram

A component diagram, describes the organization and wiring of the physical components in a system. Component diagrams are often drawn to help model implementation details and double-check that every aspect of the system's required functions is covered by planned development.

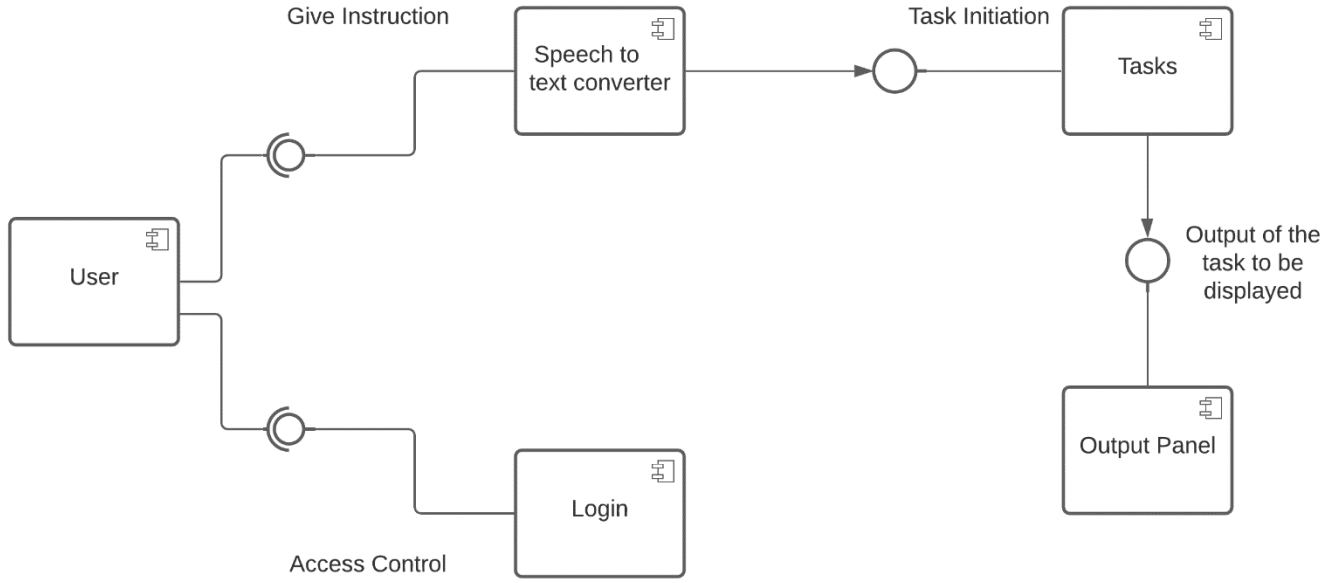


Fig 4.8: Component diagram for Virtual Assistant

4.9 Deployment Diagram

A deployment diagram is a UML diagram type that shows the execution architecture of a system, including nodes such as hardware or software execution environments, and the middleware connecting them. Deployment diagrams are typically used to visualize the physical hardware and software of a system. Figure 4.9 depicts a deployment diagram.

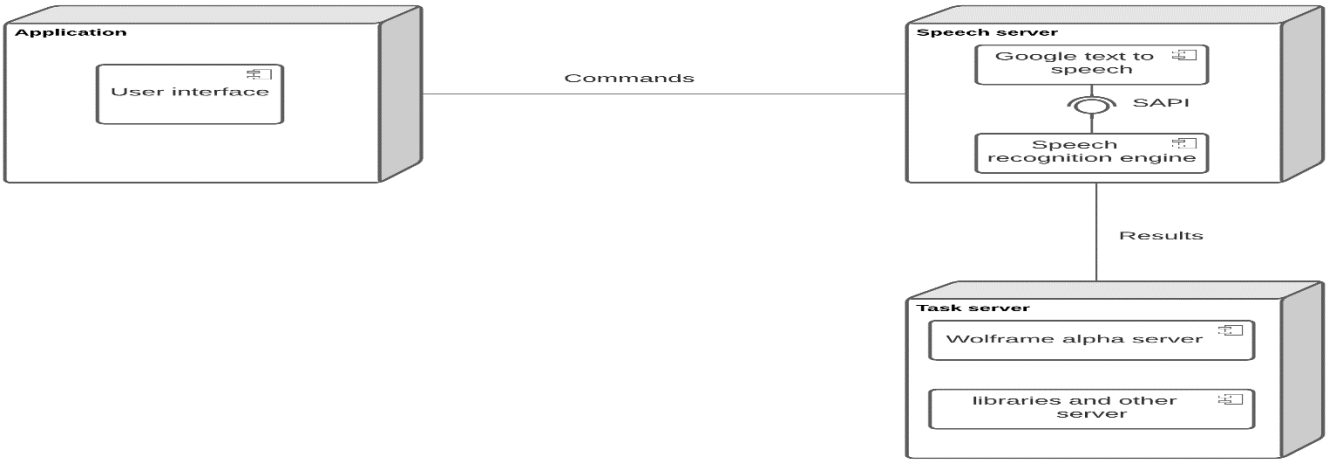


Fig 4.9: Deployment diagram for Virtual Assistant

5. TESTING

5.1 Test Plan

The overall goal is to test that all the components work fine alone and together with other components. We will perform testing on the Login Form and the functionality page.

Parameter	Description
Identifier: Version Number	**V 1.0.0**
Test Items	subsystems to be tested are • Login is successfully done • Password incorrect • Username incorrect
Features to be tested	• User is able to login • User is able to create file successfully • User is able to get the news accurately • User is able to fetch the weather details • User is able to get his system info • User is able to solve mathematical problems • User is able to operate the system efficiently through voice • User is able to solve some system related functions such as getting system info, emptying the recycle bin, lock window, take a screenshot, snap a photo • User will be able to get the info about anything available on the internet • User will be able to listen to jokes
Approach	Generate all types of dummy data and test each component.

5.2 Test Case Report

Test Case: 5.2.1	Test Case Name: Login
System: Virtual Assistant	Subsystem: Login
Designed By: Ankush Gupta	Design Date: 2/12/2020
Executed By: Hiten Gupta	Execution Date: 2/12/2020
Short Description: to successfully login in the application	

Preconditions:

1. The user has accessed the Virtual Assistant application by opening the exe file.
2. The user has a valid email address

Step	Action	Expected System Response	Pass/ Fail	Comment
1	Enter the username	The system will save the username internally. If the username is empty, it will display an error stating that 'username cannot be empty'.	Pass	
2	Enter a valid email address	The system will save this email address. Ig the email address is empty, it will display an error stating that email address cannot be empty.	Pass	
3	**Check post-condition 1**	A system generated otp must be sent on the provided email address.	Pass	
4	Enter the otp correctly.	The system would have send an otp to the provided email. This otp will be the basis of authentication of the user. Also if the otp entered is incorrect or is empty, it will show an error.	Pass	
5	Enter the user interface	If the entered otp is correct, then the user will be able to access the user interface and desktop assistant.	Pass	

Post-conditions:

1. The user will receive an otp on the provided email.

Test Case: 5.2.2	Test Case Name: Give command and generate output
System: Virtual Assistant	Subsystem: Generate Output
Designed By: Ankush Gupta	Design Date: 2/12/2020
Executed By: Ankush Sood	Execution Date: 2/12/2020
Short Description: To successfully process the given input command and obtain the output as via a push notification or through voice.	

Preconditions:

1. The user has accessed the Virtual Assistant application by opening the exe file.
2. The user has entered the otp correctly and is able to access the user interface.
3. The system displays the 'Home' page.

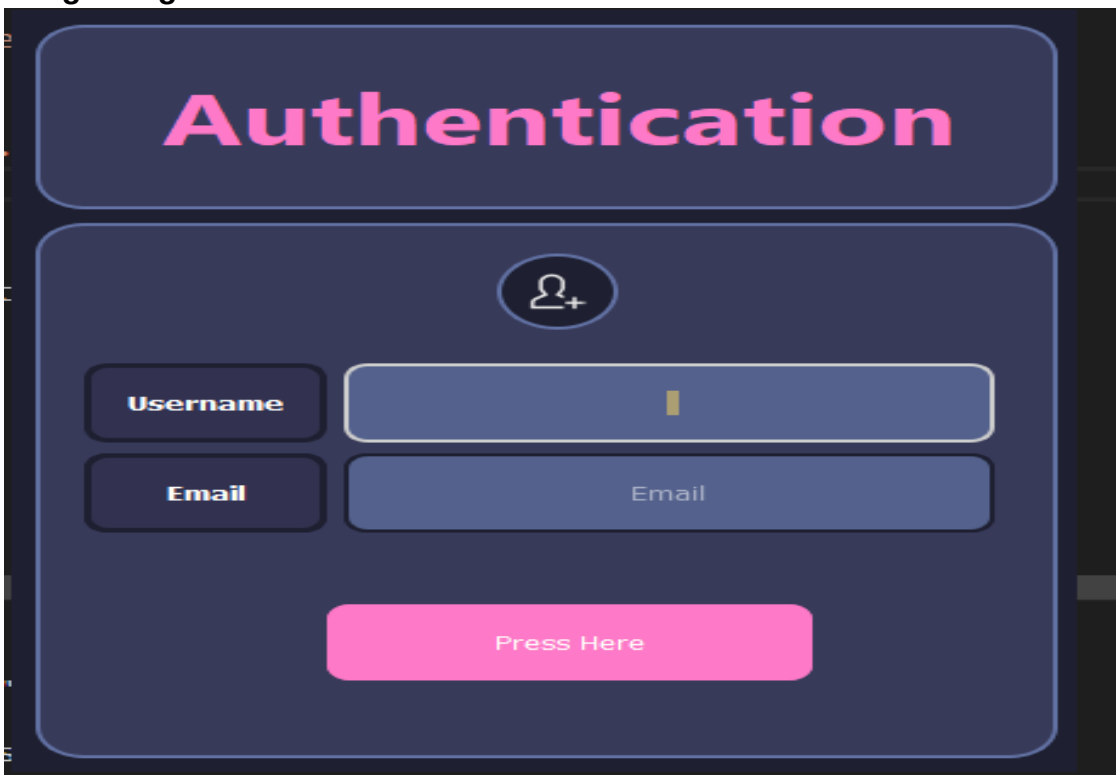
Step	Action	Expected System Response	Pass/ Fail	Comment
1	Click the 'Account' button	The system must displays the login information of the logged-in user.	Pass	
2	Click the cancel button.	The application must be terminated successfully without any error.	Pass	
3	Click the speak button.	Voice recognition must be started along with the loading of an animation which shows that the process is working fine.	Pass	
4	As soon as the voice recognition start, the user need to give the command through voice. The input command will be gone through some processes.	The input command must being properly processing internally and the corresponding api according to command will be fetched and the output must be shown.	Pass	
5	**Check post-condition 1** **Check post-condition 2**	The output of the command after being properly processed must be shown . For some command the output is shown as through the windows push notification and for some commands, through voice.	Pass	

Post-conditions:

1. A windows push notification will be seen showing the result of the given instruction.
2. A voice can be heared stating that operation is successful.

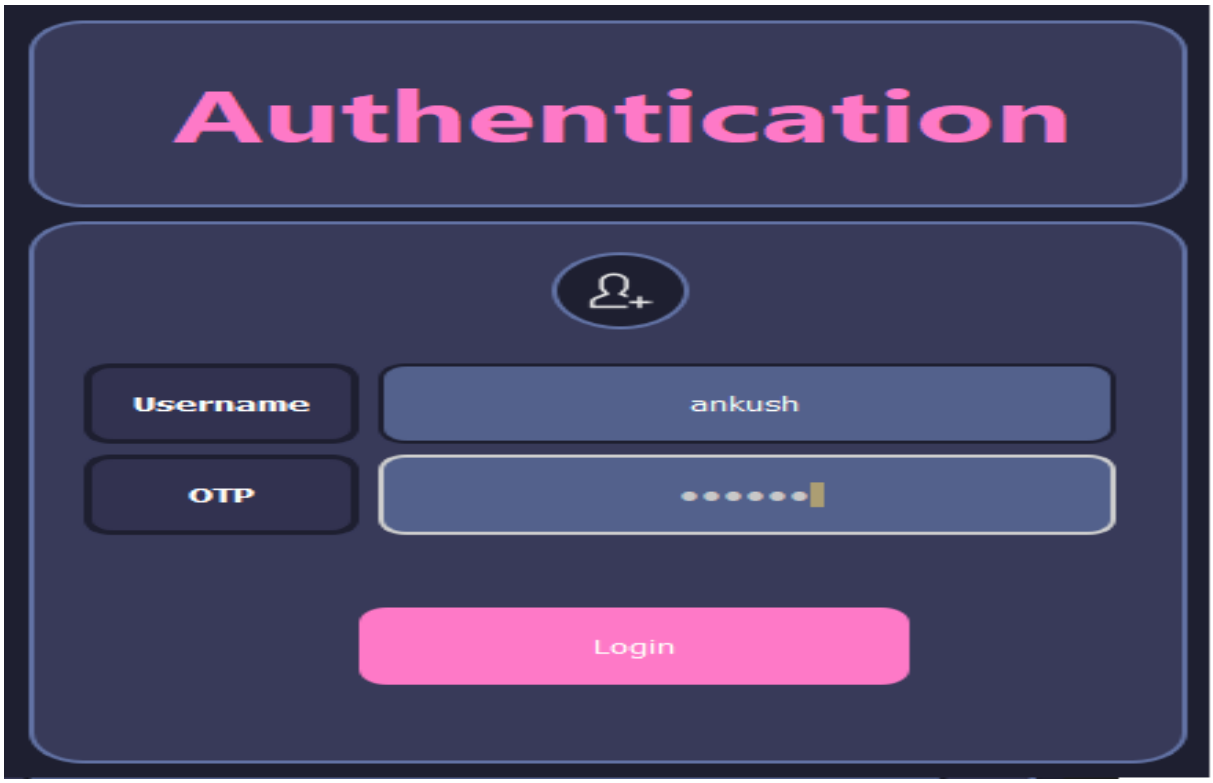
5.3 Screenshots

#Login Page

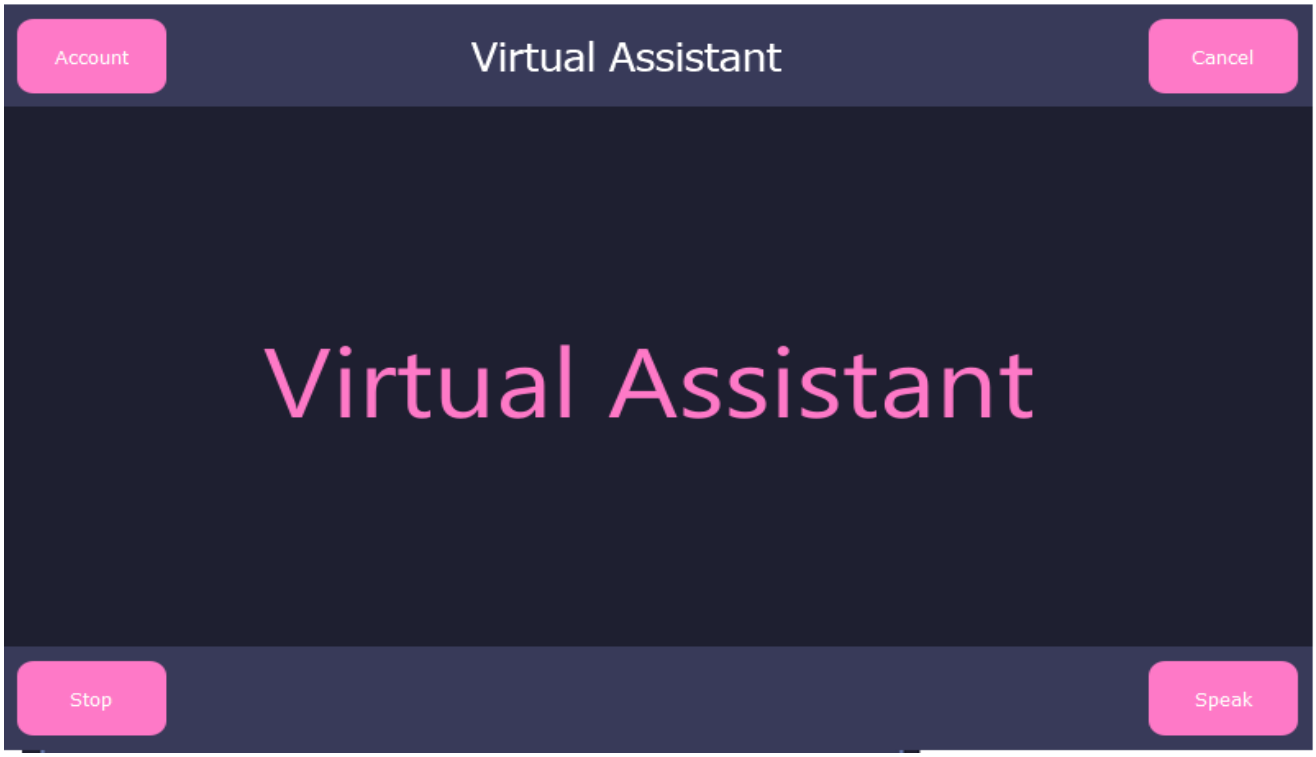


#mail received

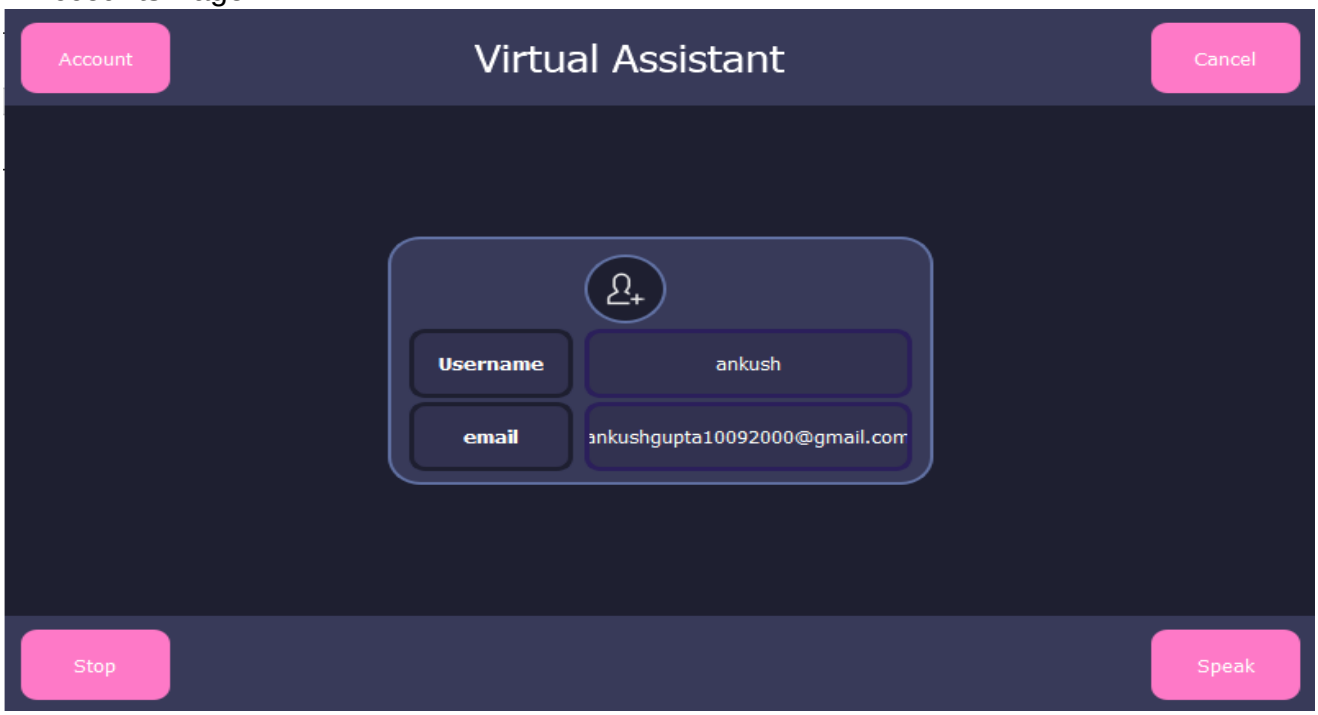
 # OTP page:



 # user interface



Accounts Page



Voice Input Page