ATMIYA UNIVERSITY

RAJKOT

A

Report On

ALETTA

Under subject of

MINI PROJECT

B. TECH Semester – VII

(Computer Engineering)

Submitted by:

Sweta Nikul Patel (190002088)

Drashti Rakeshbhai Kothari (190002059)

Prof. Nirali Borad

(Faculty Guide)

Prof. Tosal M. Bhalodia

(Head of the Department)

Academic Year

(2022-23)

CANDIDATE'S DECLARATION

We hereby declare that the work presented in this project entitled "**ALETTA**" submitted towards completion of project in **7th Semester** of B. Tech. (Computer Engineering) is an authentic record of our original work carried out under the guidance of "**NIRALI BORAD**".

We have not submitted the matter embodied in this project for the award of any other degree.

Semester: 7th

Place: Rajkot

Signature:

Sweta Nikul Patel (190002088)

Drashti Rakeshbhai Kothari (190002059)

ATMIYA UNIVERSITY

RAJKOT

<u>CERTIFICATE</u>

Date: 30/03/2022

This is to certify that the "**ALETTA**" has been carried out by Ms. **Sweta Nikul Patel** under my guidance in fulfillment of the subject Mini Project in COMPUTER ENGINEERING (6th Semester) of Atmiya University, Rajkot during the academic year 2022.

Prof. Nirali Borad   Prof. Tosal M. Bhalodia
(Project Guide)   **(Head of the Department)**

ATMIYA UNIVERSITY
RAJKOT

CERTIFICATE

Date: 30/03/2022

This is to certify that the "**ALETTA**" has been carried out by Ms. **Drashti Rakeshbhai Kothari** under my guidance in fulfillment of the subject Mini Project in COMPUTER ENGINEERING (6th Semester) of Atmiya University, Rajkot duringthe academic year 2022.

Prof. Nirali Borad   Prof. Tosal M. Bhalodia
(Project Guide)   **(Head of the Department)**

<u>INDEX</u>

ACKNOWLEDGEMENT

We have taken many efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. We would like to extend our sincere thanks to all of them.

We are highly indebted to **Prof. Nirali Borad** for their guidance and constant supervision as well as for providing necessary information regarding the Mini Project titled **"Aletta".** We would like to express our gratitude towards staff members of ComputerEngineering Department, Atmiya University for their kind co- operation and encouragement which helped us in completion of this project.

We even thank and appreciate to our colleague in developing the project and people who have willingly helped us out with their abilities.

<div align="right">

Sweta Nikul Patel (190002088)

Drashti Rakeshbhai Kothari (190002059)

</div>

ABSTRACT

In this modern era, day to day life became smarter and interlinked with technology. We already know some voice assistance like google, Siri. etc. Now in our voice assistance is **ALETTA**, it can act as a basic medical prescriber, daily schedule reminder, note writer, calculator and a search tool. This project works on voice input and give output through voice and displays the text on the screen. The main agenda of our voice assistance makes people smart and give instant and computed results. The voice assistance takes the voice input through our microphone (Bluetooth and wired microphone) and it converts our voice into computer understandable language gives the required solutions and answers which are asked by the user. This assistance connects with the world wide web to provide results that the user has questioned. Natural Language Processing algorithm helps computer machines to engage in communication using natural human language in many forms.

LIST OF FIGURES

1. INTRODUCTION

Today the development of artificial intelligence (AI) systems that can organize a natural human-machine interaction (through voice, communication, gestures, facial expressions, etc.) are gaining in popularity. One of the most studied and popular was the direction of interaction, based on the understanding of the machine by the machine of the natural human language. It is no longer a human who learns to communicate with a machine, but a machine learns to communicate with a human, exploring his actions, habits, behaviour and trying to become his personalized assistant.

Virtual assistants are software programs that help you ease your day to day tasks, such as showing weather reports, creating remainders, making shopping lists etc. They can take commands via text (online chatbots) or by voice. Voice-based intelligent assistants need an invoking word or wake word to activate the listener, followed by the command. We have so many virtual assistants, such as Apple's Siri, Amazon's **ALETTA** and Microsoft's Cortana.

This system is designed to be used efficiently on desktops. Personal assistants software improves user productivity by managing routine tasks of the user and by providing information from an online source to the user.

This project was started on the premise that there is a sufficient amount of openly available data and information on the web that can be utilized to build a virtual assistant that has access to making intelligent decisions for routine user activities.

Keywords: Virtual Assistant Using Python, AI, Digital assistance, Virtual Assistance, Python

1.1. Purpose

Purpose of **ALETTA** is to being capable of voice interaction, music playback, making to-do lists, setting alarms, streaming podcasts, playing audiobooks, and providing weather, traffic, sports, and other real-time information, such as news. Virtual assistant sense able users to speak natural language voice commands in order to operate the device and its apps. There is an increased overall awareness and a higher level of

comfort demonstrated specifically by millennial consumers. In this ever-evolving digital world where speed, efficiency, and convenience are constantly being optimized, it's clear that we are moving towards less screen interaction.

1.2. Scope

Voice assistants will continue to offer more individualized experiences as they get better at differentiating between voices. However, it's not just developers that need to address the complexity of developing for voice as brands also need to understand the capabilities of each device and integration and if it makes sense for them specific brand. They will also need to focus on maintaining a user experience that is consistent within the coming years as complexity becomes more of a concern. This is because the visual interface with voice assistants is missing. Users simply cannot see or touch a voice interface.

1.3. Technology and tools

1. PYTHON:

- Python is a high-level, general-purpose programming language.
- Its design philosophy emphasizes code readability with the use of significant indentation.
- Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small- and large-scale projects.
- Python is dynamically-typed and garbage-collected.
- It supports multiple programming paradigms, including structured (particularly procedural), object-oriented and functional programming.
- It is often described as a "batteries included" language due to its comprehensive standard library.
- It is a robust, highly useful language focused on rapid application development (RAD).
- Python helps in easy writing and execution of codes.
- Python can implement the same logic with as much as 1/5code as compared to other OOPs languages.
- Python provides a huge list of benefits to all.
- The usage of Python is such that it cannot be limited to only one activity.

- Its growing popularity has allowed it to enter into some of the most popular and complex processes like Artificial Intelligence (AI), Machine Learning (ML), natural language processing, data science etc.

2. Speech Recognition:

- This is a library for performing speech recognition, with support for several engines and APIs, online and offline.
- It supports APIs like Google Cloud Speech API, IBM Speech to Text, Microsoft Bing Voice Recognition etc.

3. Pyttsx3

- Pyttsx stands for Python Text to Speech. It is a cross-platform Python wrapper for text-to speech synthesis.
- It is a Python package supporting common text-to- speech engines on MacOS X, Windows, and Linux. It works for both Python2.x and 3.x versions. Its main advantages that it works offline.

2. PROJECT MANAGEMENT

2.1 Project Planning:

Project Planning is concerned with identifying and measuring the activities, milestones and deliverables produced by the project. Project planning is undertaken and completed sometimes even before any development activity starts. Project planning consists of following essential activities:

- Scheduling manpower and other resources needed to develop the system.
- Staff organization and staffing plans.
- Risk identification, analysis, and accurate planning.
- Estimating some of the basic attributes of the project like cost, duration and efforts.
- The effectiveness of the subsequent planning activities is based on the accuracy of these estimations.
- Miscellaneous plans like quality assurance plan, configuration management plan, etc.
- Project management involves planning, monitoring and control of the people, process, and the events that occurs as the software evolves from a preliminary concept to an operational implementation. Cost estimation is a relative activity that is concerned with the resources required to accomplish the project plan.

2.2 Project Scheduling:

The scheduling is the peak of a planning activity, a primary component of software project management. When combined with estimation methods and risk analysis, scheduling establishes a roadmap for project management. The characteristics of the project are used to adapt an appropriate task set for doing work.

2.3 Risk Management:

Risk management consists of a series of steps that help a software development team to understood and manage uncertain problems that may arise during the course of software development and can plague a software project.

Risks are the dangerous conditions or potential problems for the system which may damage the system functionalities to very high level which would not be acceptable at any cost. So in order to make our system stable and give its 100% performance we must have identify those risks, analyze their occurrences and effects on our system and must prevent them to occur.

2.3.1 Risk Identification:

Risk identification is a first systematic attempt to specify risks to project plan, Scheduling resources, project development. It may be carried out as a team process using brainstorming approach.

Technology risk:

Technical risks concern implementation, potential design, Interfacing, testing, and maintenance problems

- Database Corruptness
- Garbage Collection

People Risks:

These risks are concerns with the team and its members who are taking part in developing the system.

- Leaking an important data
- Failure of the administration
- Lack of knowledge.
- Lack of clear product vision.
- Technical staff conflict
- Poor communication between people.

Tools Risks:

These are more concerned with tools used to develop the system.

- Tools containing virus.

General Risks:

General Risks are the risks, which are concerned with the mentality and resources.

- Lack of resources can cause great harm to efficiency and timely productivity.

- Rapidly changing requirements.
- Changes in requirements can cause a great harm to implementation, designing and schedule of developing the system.
- Insufficient planning and task identification.

2.3.2 Risk Analysis

"Risk analysis = risk assessment + risk management + risk communication." Risk analysis is employed in its broadest sense to include:

Risk assessment:

Involves identifying sources of potential harm, assessing the likelihood that harm will occur and the consequences if harm does occur.

For this project It might be: -
- System Crash.

Risk management

Evaluates which risks identified in the risk assessment process require management and selects and implements the plans or actions that are required to ensure that those risks are controlled.

Precautions taken to make risks minimal are as under:-
- Periodical backups are taken to avoid major loss in case of system crash.

Risk communication

Involves an interactive dialogue between stakeholders and risk assessors and risk managers which actively informs the other processes.

Steps taken for risk communication is as under: -
- Probability of certain risks is negotiated with client.
- All the possible risks are listed out during communication and project is developed taking care of that risks.

3. <u>SYSTEM REQUIREMENT STUDIES</u>

3.1 Hardware and Software Requirement

This shows minimum requirements to carry on to run this system efficiently.

3.1.1 Hardware Requirements

Server-side Hardware Requirements:

Devices	Description
Processor	Intel Core Duo 2.0 GHz or more
RAM	512 MB or more
Hard Disk	2 GB or more

Table 3.1.1.1 Server-side Hardware Requirement

3.1.2 Software Requirements

For which	Software
Operating System	Windows 7/8/10
Coding Language	Python

Table 3.1.2.1 Software Requirements

3.2 Constraints

3.2.1 Hardware Limitations

The major hardware limitations faced by the system are as follows:

- If the appropriate hardware is not there like processor, RAM, hard disks
- The problem in processing requests of client
- If appropriate storage is not there our whole database will crash due to less storage because our main requirement is large storage.

3.2.2 Reliability Requirements

Since many users can access the server simultaneously, load on the server becomes very high. Hence, the server should be of enough high configurations. There should be high back up storage and management of huge data for overall ideas, videos, images, multiple countries, multiple user profile.

The Reliability requirements are the validations used to protect the system against one or more incorrect activities. Without proper validation of the system, the failure possibilities of it grow higher so it is must to understand the proper validation of the system and must implement them. All the required validator controls spend very good role to keep the system secure from any unauthorized or incorrect information. In all these validation actions if system found one or more entries violating validation rules then user will be warned by proper error messages and the details or the record is not going to be saved until corrections are made to them.

4. <u>SYSTEM ANALYSIS</u>

4.1 Study Current System

Implementation is the stage where the theoretical design is turned into a working system. The most crucial stage in achieving a new successful system and in giving confidence on the new system for the users that it will work efficiently and effectively.

The system can be implemented only after thorough testing is done and if it is found to work according to the specification.

It involves careful planning, investigation of the current system and its constraints on implementation, design of methods to achieve the change over and an evaluation of change over methods a part from planning. Two major tasks of preparing the implementation are education and training of the users and testing of the system.

The more complex the system being implemented, the more involved will be the systems analysis and design effort required just for implementation.

The implementation phase comprises of several activities. The required hardware and software acquisition is carried out. The system may require some software to be developed. For this, programs are written and tested. The user then changes over to his new fully tested system and the old system is discontinued

4.2 Problem and weakness of current system

- Inconsistency in data entry and generate errors
- System is fully dependent on skilled individuals
- Time consuming and costly to produce reports
- Entry of false information
- Lack of security
- Unnecessary details

4.3 Requirements of New System

4.3.1 User Requirements:

The user requirement for this system is to make the system fast, flexible, less prone to error, reduce expenses and save the time.

4.3.2 System Requirements:

- **Functional System Requirement:**

  This section gives a functional requirement that applicable to the Relier 4. There are three sub modules in this phase.

  1. Numpy module.
  2. sys module.
  3. Pygame module
  4. Math module

- **The functionality of each module is as follows:**

1. **Speech recognition module**: The system uses Google's online speech recognition system for converting speech input to text. The speech input Users can obtain texts from the special corpora organized on the computer network server at the information centre from the microphone is temporarily stored in the system which is then sent to Google cloud for speech recognition. The equivalent text is then received and fed to the central processor.

2. **Pyttsx3 module** : Pyttsx stands for Python Text to Speech. It is a cross-platform Python wrapper for text-to-speech synthesis. It is a Python package supporting common text-to-speech engines on MacOS X, Windows, and Linux. It works for both Python2.x and 3.x versions. Its main advantages that it works offline.

3. **Date time module** : Python Datetime module supplies classes to work with date and time. These classes provide a number of functions to deal with dates, times and time intervals. Date and datetime are an object in Python, so when you manipulate them, you are actually manipulating objects and not string or timestamps.

4. **Wikipedia module**: **Wikipedia** is a Python library that makes it easy to access and parse data from Wikipedia. Search Wikipedia, get article summaries, get data like links and images from

a page, and more. Wikipedia wraps the MediaWiki API so you can focus on using Wikipedia data, not getting it.

5. **PyWhatkit module:** PyWhatKit is a Python library with various helpful features. It's easy-to-use and does not require you to do any additional setup. Currently, it is one of the most popular library for WhatsApp and YouTube automation. New updates are released frequently with new features and bug fixes.

6. **Pyjokes module :** One line jokes for programmers (jokes as a service).

- Non-Functional System Requirements:

i. EFFICIENCY REQUIREMENT:

When a python game implemented user can play in an efficient manner.

ii. RELIABILITY REQUIREMENT:

The system should provide a reliable environment to both customers and owner.

iii. USABILITY REQUIREMENT:

The game is designed for user friendly environment and ease of use.

iv. IMPLEMENTATION REQUIREMENT:

Implementation of the system using python.

v. DELIVERY REQUIREMENT:

The whole system is expected to be delivered in four months of time with a weekly evaluation by the project guide.

4.4 Feasibility Study

The feasibility study of any system is mainly intended to study and analyze the proposed system and to decide whether the system under consideration will be viable or not after implementation. That is it determines the usability of the project after deployment. To come to result a set of query is answered keeping the efficiency of the software and its impact on the domain for which it was developed.

Technical Feasibility:

In technical feasibility, we study all technical issues regarding the proposed system. It is mainly concerned with the specifications of the equipment and the software, which successfully satisfies the end-user's requirement. The technical needs of the system may vary accordingly but include:

- The feasibility to produce outputs in a given time.
- Response time under certain conditions.
- Ability to process a certain volume of the transaction at a particular speed.
- Facility to communicate data.

5. System Design

5.1 Input/Output interface

> **Aletta Platform**

```
1    import speech_recognition as sr
2    import pyttsx3
3    import pywhatkit
4    import datetime
5    import wikipedia
6    import pyjokes
7
8    listener = sr.Recognizer()
9    engine = pyttsx3.init()
10   voices = engine.getProperty('voices')
11   engine.setProperty('voice', voices[1].id)
12   engine.runAndWait()
13
14   def talk(text):
15       engine.say(text)
16       engine.runAndWait()
```

Figure 5.1.1 Aletta Platform

```
PROBLEMS   OUTPUT   TERMINAL   JUPYTER   COMMENTS   DEBUG CONSOLE
listening...

[Done] exited with code=1 in 51.454 seconds

[Running] python -u "c:\Users\Admin\Desktop\Alexa.py"
listening...
current time
07:58 PM
listening...
```

Figure 5.1.2 Output

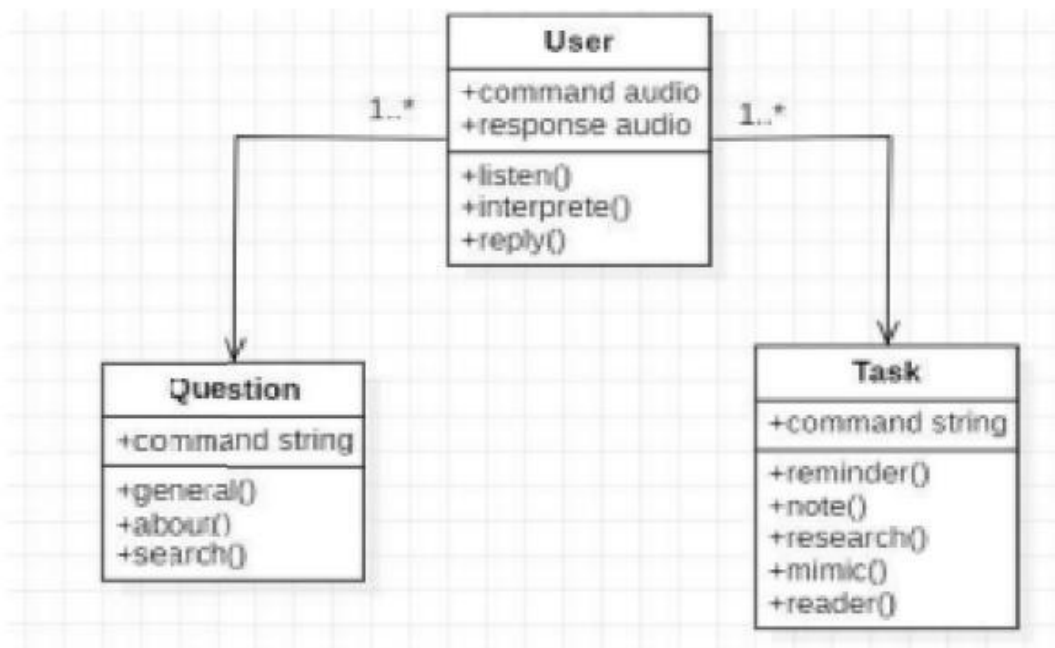## 5.2   Interface Design

5.2.1 Class Diagram

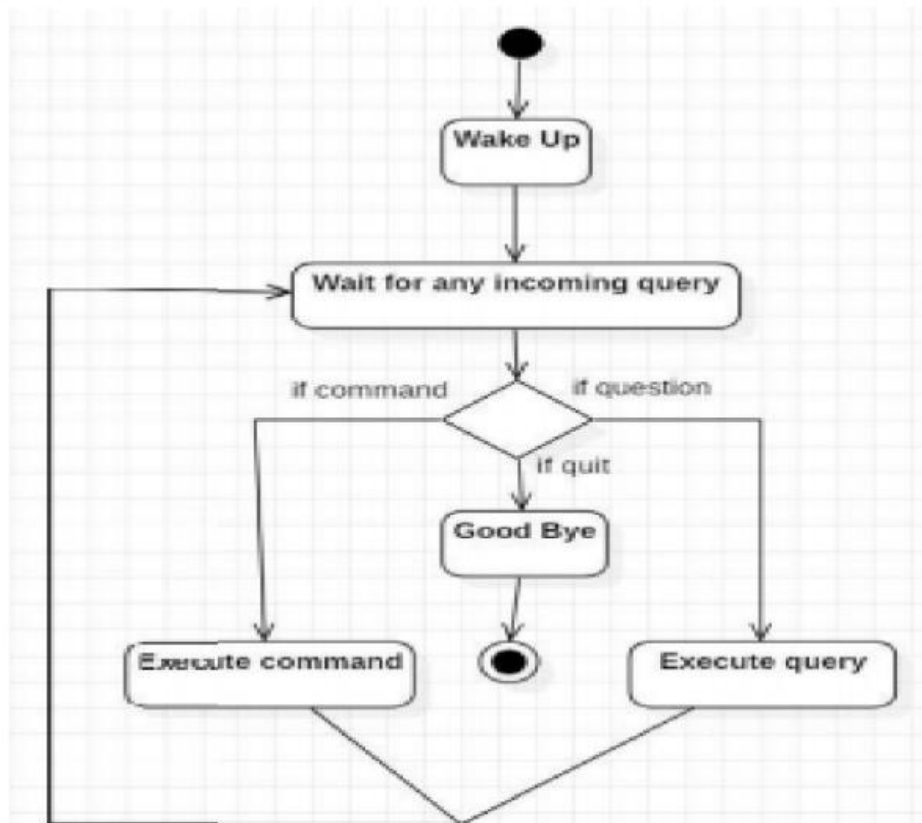Figure 5.2.1. Class Diagram

5.2.2 Activity Diagram

Figure 5.2.2 Activity Diagram

5.2.3 Data Flow Diagrams
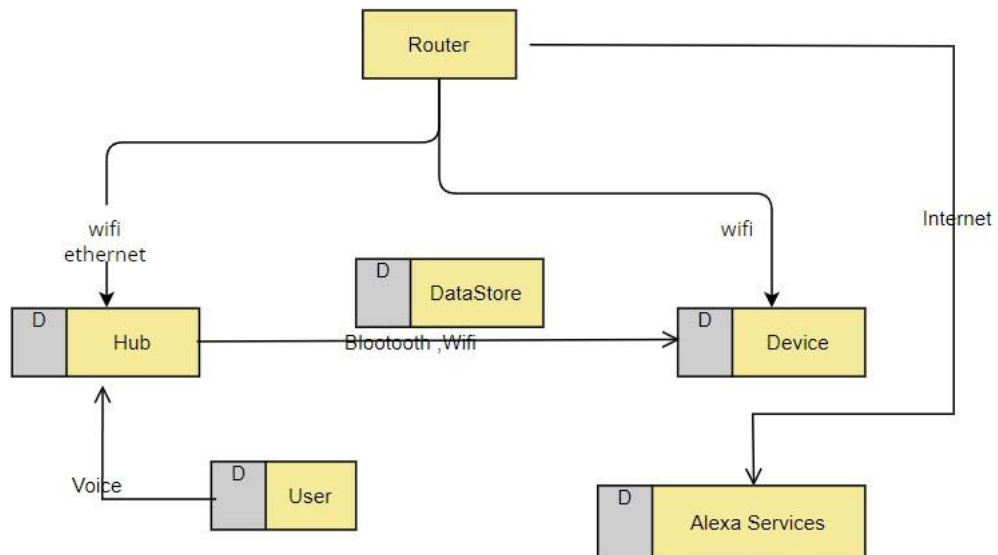
5.2.3.1 Context Level DFD

Figure 5.2.4.1 Context level DFD

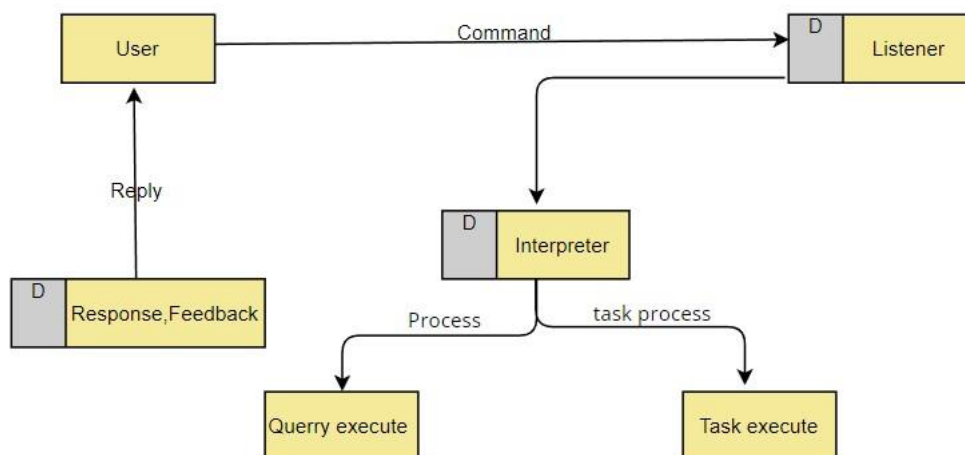## 5.2.3.2 First Level DFD

Figure 5.2.4.2 First level DFD
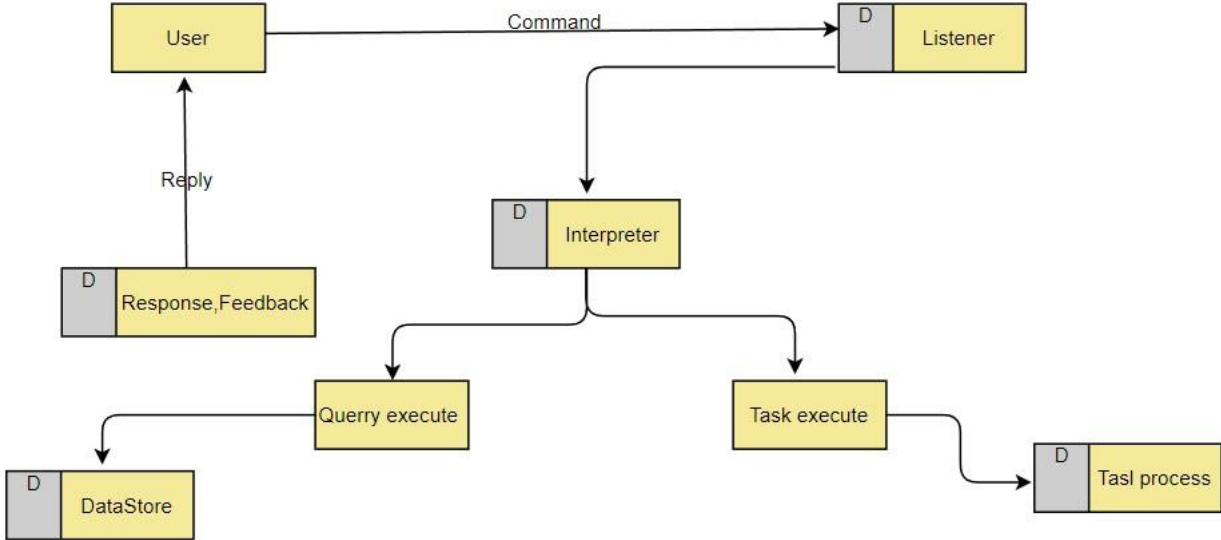
5.2.3.3   Second level DFD

Figure 5.2.4.3 Second level DFD

5.2.4 ER Diagram

Figure 5.2.4. ER Diagram DFD

6. Code Implementation

6.1 Implementation Environment

Challenges identified for successful design and implementation of this project are dominated by:

- Complexity, reliability/availability, transparent data access. The project was a result of a group consensus. The team was having two members. The team was guided by project manager. The team structure depends on the management style of the organization, the no. of people in the team, their skill levels and the problem difficulty.

6.2 Program/Module Specification

- System GUI must be as simple and user friendly as anyone can use it. At front side we implemented login form to access the system.
- A Session is maintained throughout the system when a particular user enters into the system. The Session is regularly checked whenever it is required.
- Proper validation is placed as and when it is required.

6.3 Coding Standards

- Normally, good software development organization requires their programmers to maintain some well-defined and standard style of coding called coding standard.

6.3.1 Comment Standards:

- The comment should describe what is happening, how it is being done, what parameters mean, which global are used and which are modified, and any registration or bugs.

    The standards I have followed are:

- Comment may also be used in the body of the Cascading style sheets to explain individual sections or lines of codes to easily get access and easily review or manage the classes or properties for the pages.
- Inline comments should be made with the //. Comment style and should be indented at the same level as the code described.
- For multiple line comments we write between /* ..... */.

7. Testing

7.1 Testing Strategy

A strategy for software testing integrates software test case design method into a well-planned series of steps that result in the successful construction of the software. The strategy provides the roadmap that describes the steps to be conducted as a part of testing, then these steps are planned and then undertaken, and how much effort, time and resource will be required.

7.2 Testing Method

7.2.1 Unit Testing

The unit testing is meant for testing smallest unit of software. There are two approaches namely bottom-up and top-down. In bottom-up approach the last module is tested and then moving towards the first module while top-down approach reverses the action. In present work we opt for the first one. The bottom-up approach for the current project is carried out as shown in.

7.2.2 Validation Testing

After the integration testing software is completely assembled as a package, interfacing error have been uncovered and corrected, and then validation testing may begin. Validation can be defined in many ways but a simple definition is what a validation succeeds when software functions in a manner that can be reasonably accepted by the user.

7.2.3 Integration Testing

The integration testing is meant to test all the modules simultaneously because it is possible that all the modules may function correctly when tested individually. But they may not work altogether and may lead to unexpected outcome.

8. Limitations and Future Enhancement

8.1 LIMITATIONS:

Though we tried our best in developing this system but as limitations are mere parts of any System so are of our system. Some limitations of **ALETTA** are as under:

- Limited game features.
- Easy Designing.

8.2 FUTURE ENHANCEMENT:

There is always a scope for enhancements in any developed system, especially when our nature of the project is iterative waterfall which allows us to rethink on the method of development to adopt changes in the project. Below mentioned are some of the changes possible in the future to increase the adaptability, and efficiency of the system.

- More attractive GUI (Graphical user interface).
- More menu and Settings
- More Levels

9. Conclusion

In this paper "Virtual Assistant **ALETTA** Using Python" we discussed the design and implementation of Digital Assistance. The project is built using open source software modules with PyCharm community backing which can accommodate any updates shortly. The modular nature of this project makes it more flexible and easy to add additional features without disturbing current system functionalities.

It not only works on human commands but also give responses to the user based on the query being asked or the words spoken by the user such as opening tasks and operations. It is greeting the user the way the user feels more comfortable and feels free to interact with the voice assistant. The application should also eliminate any kind of unnecessary manual work required in the user life of performing every task. The entire system works on the verbal input rather than the next one.