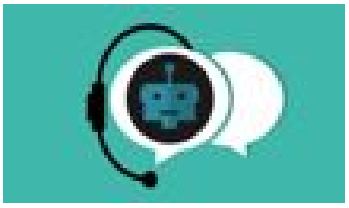
Project Report

SOFTWARE REQUIREMENT SPECIFICATION

ON

Virtual Voice Assistant



BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

Under the Guidance of

PROF. POOJA V. KAMAT

SYMBIOSIS INSTITUTE OF TECHNOLOGY

(A CONSTITUENT OF SYMBIOSIS INTERNATIONAL UNIVERSITY)

Pune - 412115

2019-20

SUBMITTED BY :

MANAN GANDHI (17070124022)

HARSHIL SHRIVASTAVA (17070124025)

YATHARTH JAIN (17070124028)

GAURAV JOSHI (17070124029)

Table of Contents

1. INTRODUCTION:

1.1 Purpose:

The purpose of this document is to delineate the uses/ functional and non-functional requirements, use case scenarios, constraints and the features of (Name of voice assistant)

1.2 Feasibility Study:

1.2.1 Product-

Alexa is a portable lightweight voice assistant for any and every device connected to the internet. It is an open source project and hence has a vast scope of improvements via feedback received from the user. Alexa aims to enable every individual to run his/ her device for most of their daily needs via voice control. We strive towards making it as light and inexpensive as possible. This would directly benefit the visually impaired among many other end users.

1.2.2 Technical Feasibility-

Once installed in the user's device, Alexa will require microphone access and internet connectivity and it will be good to go.

1.2.3 Social Feasibility-

Since Alexa is voice controlled, the only prerequisite that the user is expected to have is the ability to speak in English as Alexa can as of now only recognise voice input in English.

1.2.4 Economic Feasibility-

Alexa will be a free to download basis software. Internet connectivity will be the only expense that the user will be bearing.

1.3 Intended Audience:

Everyone stands to benefit from having a functional voice assistant, but there are some categories of people who are more active in their adoption of the technology than others. Alexa adds convenience of operating the device with sheer voice commands making queries quicker and more user-friendly to answer.

1.4 Product Scope:

With the shift in user demands, there is an increased overall awareness and a higher level of comfort demonstrated specifically by millennial consumers. The mass adoption of AI in everyday life can also be stated as a factor. WIth constant improvements, Alexa can potentially be made IoT enabled for smart homes and AI enabled to better understand user preferences to respond more interactively.

Applications of a voice assistant can be seen everywhere from refrigerators to automobiles to smart surroundings. Hence, the entire potential of Alexa remains untapped yet but provides a vast scope of improvement.

1.5 References:

https://www.python.org/
https://github.com/Ashutosh-kv/JARVIS/blob/master/assistant.py

2. OVERALL DESCRIPTION:

2.1 Product Perspective:

Alexa is built with the intent to empower its users with action driven assistance. For eg: “What will the weather be like today?” query by the user will return a short summary of today’s weather. When a user will talk to Alexa, they will usually try to get things done. In addition to making the device more conveniently operable for the user, Alexa aims at making real conversations with its users to provide a more comfortable experience.

2.2 Product Functions:

Following are the basic functions that Alexa will be able to perform:

1. Taking voice input from the user via the device’s in-built microphone
2. Processing the input command for certain keywords such as “weather”, “browser”, “close/ abort/ bye” etc.
3. Providing the output depending upon the processed command in both text and speech formats

2.3 User Classes:

The user is the person operating the device with Alexa installed in it. For this purpose no account creation is required. The user only needs to give voice commands for which he/ she expects a result. An environment with minimum background disturbances is preferred to achieve more accurate results.

2.4 Technology Environment:

MacOS, Linux UBUNTU 18.0.4 LTS and Windows 10 operating systems are used for the testing and deployment of Alexa. Future prospect is to bring an android app to enable Alexa in handheld devices as well.

Python 3.6.8 has been used for logical functioning and backend requirements along with Firebase for database connectivity.

2.5 Design and implementation constraints:

Following are the design and implementation constraints associated with using Alexa:-

- If the user is in a noisy surrounding, Alexa will not be able to properly process the commands and will hence throw exceptions or an unexpected result
- The user will have to give inputs in clear English as far as possible for most accurate output. Queries should be structured to avoid ambiguities

2.6 Process Model:

Evolutionary prototyping process model has been used for Alexa. Our main goal was to build a very robust prototype in a structured manner and constantly refine it.

The reason for choosing this model was that we had a set of initial requirements to begin with and we kept adding in further requirements as the project progressed. To make sure the final model will be in a working condition, we had to make sure that after adding any significant function/ feature, the prototype was operational

Assumptions:

- The device has an active internet connection (preferably with a good speed for faster query resolution)
- The device has a functional microphone

Dependencies:

- Python 3.5+ required to run the program.
- The pyttsx3 (python text-to-speech engine) requires the user input to be only in English (English-US)

3. FUNCTIONAL REQUIREMENTS:

3.1 *Open Program-*

Purpose: Start Alexa

Input: Double click on the software file

Output: The user, irrespective of which OS he/ she is using, will be able to run the program from their terminal by simply double clicking on the software icon in its directory.

3.2 *Pass Query by voice command-*

Purpose: Take user input for processing

Input: User's query through his/ her voice.

Output: The user then passes his/ her query to the program by simply speaking it out loud into the device's microphone.

3.2.1 *Query Resolution-*

Purpose: Give the user what they expect from their query.

Input: Same as last function.

Output: Depending upon certain keywords in the query passed by the user, the corresponding function is called and subsequent query resolution is done.

Following are the functions included to resolve user queries:

3.2.1.1 - Greet user:

This function is called irrespective of user input as a means of creating a conversation between the program and the user. Depending upon the current hour, the user will be greeted with a

'Good Morning' if currentHour>=0 && currentHour<12

'Good Afternoon' if currentHour >= 12 && currentHour < 18

'Good Evening' if currentHour >= 18 and currentHour !=0

3.2.1.2 - Weather:

If the word ‘weather’ or ‘climate’ is mentioned in the input command, the weather function containing the weather API gets called. In this, using the IP address of the user’s device, the city is extracted and is passed as an input in the URL of the weather API to get the weather conditions of the user’s city.

3.2.1.3 - Browser:

If the phrase ‘open browser’ is mentioned in the input command, the default browser of the device is opened for the user.

3.2.1.4 - Humour:

If any of the following: joke/ entertain/ laugh/ smile is passed along with the input string, the joke function is invoked. This further promulgates the idea of a conversational voice assistant.

3.2.1.5 - Email:

If ‘email’ is mentioned in the input command, the email function is called. This function is still in the development phase and as of now the user can only send a mail to himself/ herself by hardcoding the username, password and the recipient in the program itself.

3.2.1.6 - WolfRamAlpha:

For any query that is not resolved by any of the functions mentioned above, it gets passed to wra() (WolfRamAlpha function).
This API is majorly used for calculations/ business queries (Eg: What is the price per share for google today? etc.).

3.4 Output in speech format-

Purpose: Speaking the result out loud.
Input: The result obtained upon query resolution.
Output: Once the query has been resolved, the output is read out loud by Alexa.

4.0 NON-FUNCTIONAL REQUIREMENTS:

4.1 Performance Requirements

The only major constraint while testing for the performance of this application will be the connectivity and the quality of the voice input provided. There are no immediate delays within the application since it runs as a standalone application.

4.2 Software Quality Attributes

Usability:

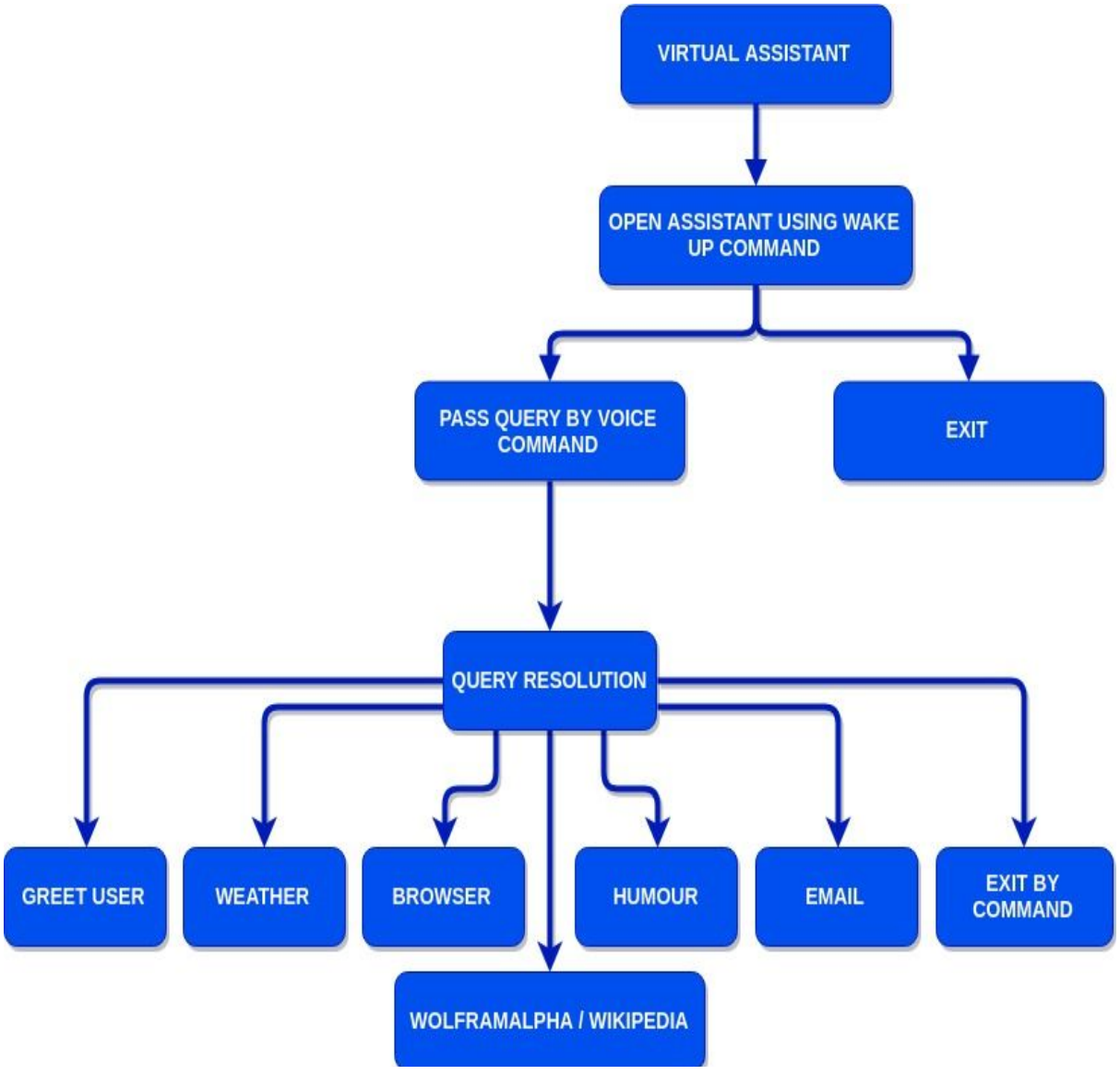
Since Alexa is a cross-platform application, it is supported across MacOS, Windows and Linux hence providing the user with flexibility of use on any machine.
Alexa will get activated by a wake up call and hence it may be used as an assistive technology for the blind.

Reliability:

The predefined conditions for the operation of Alexa are fairly minimal and once met, ensure a smooth flow. The user needs to have the required modules installed and a steady connection for uninterrupted services.

5. PROJECT SCHEDULING AND ESTIMATION:

5.1 Functional Decomposition (WBS):



5.2 Roles and Responsibility:

 **MANAN GANDHI**

 **HARSHIL SHRIVASTAV**

 **YATHARTH JAIN**

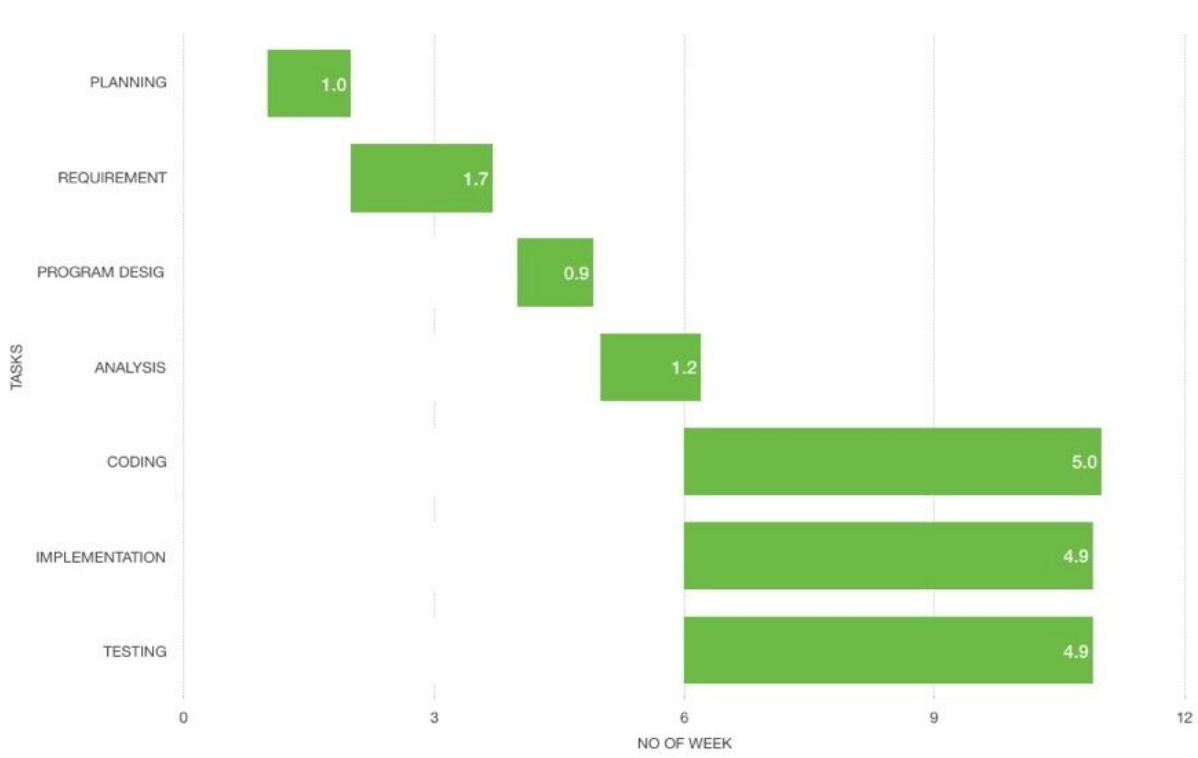
 **GAURAV JOSHI**

Aspects	**Tasks**	**Subtasks**	**Responsibilities**
Planning	**1.Research** **2.Survey** **3.Intended audience**	**1. Market study** **2. Customer consideration** **3. Technology analysis**	
Requirement	**1.Requirement Analysis** **2.Requirement Gathering** **3.Collating results of Analysis**		

Program Design	**1.Deciding functions to be included** **2.Determining key programming freedom & constraints**		
Analysis	**1.Use case Diagram** **2.Class Diagram** **3.Data Flow Diagram**		
Coding	**Modules**	**1.Speak command & speech recognition** **2.Weather & Humour** **3.Greet user** **4.Wikipedia**	

Testing	**1.Unit Testing** **2.System Testing**		
Implementation	**1.Building final release** **2.Maintenance**		

5.3 Gantt chart



5.4 Size and Budget Estimation:

S. No.	Name	Optimistic	Most likely	Pessimistic
1.	Speak output	3	6	12
2.	Listen/ accept input	10	18	30
3.	Weather	8	10	22
4.	Humour	4	6	12
5.	WolfRamAlpha	3	5	11
6.	Wikipedia	6	12	23
7.	Greet User	35	55	70

The count is in units of LOC (Lines of Code)

Estimated Value (Ev) = $\frac{(\text{Optimistic} + 4*\text{Most Likely} + \text{Pessimistic})}{6}$

Ev(1) = 7,
Ev(2) = 19,
Ev(3) = 12,
Ev(4) = 7,
Ev(5) = 6,
Ev(6) = 13,
Ev(7) = 54.

Total Ev = 7 + 19 + 12 + 7 + 6 + 13 + 54 = 118.

Assume cost/ LOC = ₹ 70

Therefore, total cost = 118 * 70 = ₹ 8260

5.5 RISK MITIGATION, MONITORING AND MANAGEMENT (RMMM):

5.5.1 Technical Risks

1. Misinterpretation : Our software won't always put your words on the screen 100% accurately. Programs cannot understand the context of language the way that humans can, leading to errors that are often due to misinterpretation. When you talk to people, they decode what you say and give it a meaning. Voice recognition software can do this but may not be capable of choosing the correct meaning. For example, it cannot always differentiate between homonyms, such as "their" and "there." It may also have problems with slang, technical words and acronyms.

2. Accents and Speech Recognition: Voice recognition systems can have problems with accents. Even though some may learn to decode your speech over time, you have to learn to talk consistently and clearly at all times to minimize errors.

If you mumble, talk too fast or run words into each other, the software will not always be able to cope. Programs may also have problems recognizing speech as normal if your voice changes, say when you have a cold, cough, sinus or throat problem.

3. Network error: Our software requires an active internet connection to request the data so the user needs to be connected for the entire duration of interaction with the program (preferably a high speed connection)

5.5.2 Business Risk

1. Sales Risk: There may be a risk of project not being sold by the sales team that can be due to poor marketing strategies or may be the product is not latest by the market standards.

2. Market Risk: A product is nothing if we don't create a value from it. So there will be no use of project if it didn't solve someone's problem in efficient way. So, we need to update it from the market perspective.

5.5.3 Project Risk

1. Schedule risk: The risk that activities will take longer than expected. Slippages in schedule typically increase costs and delay the receipt of project benefits, with a possible loss of competitive advantage.

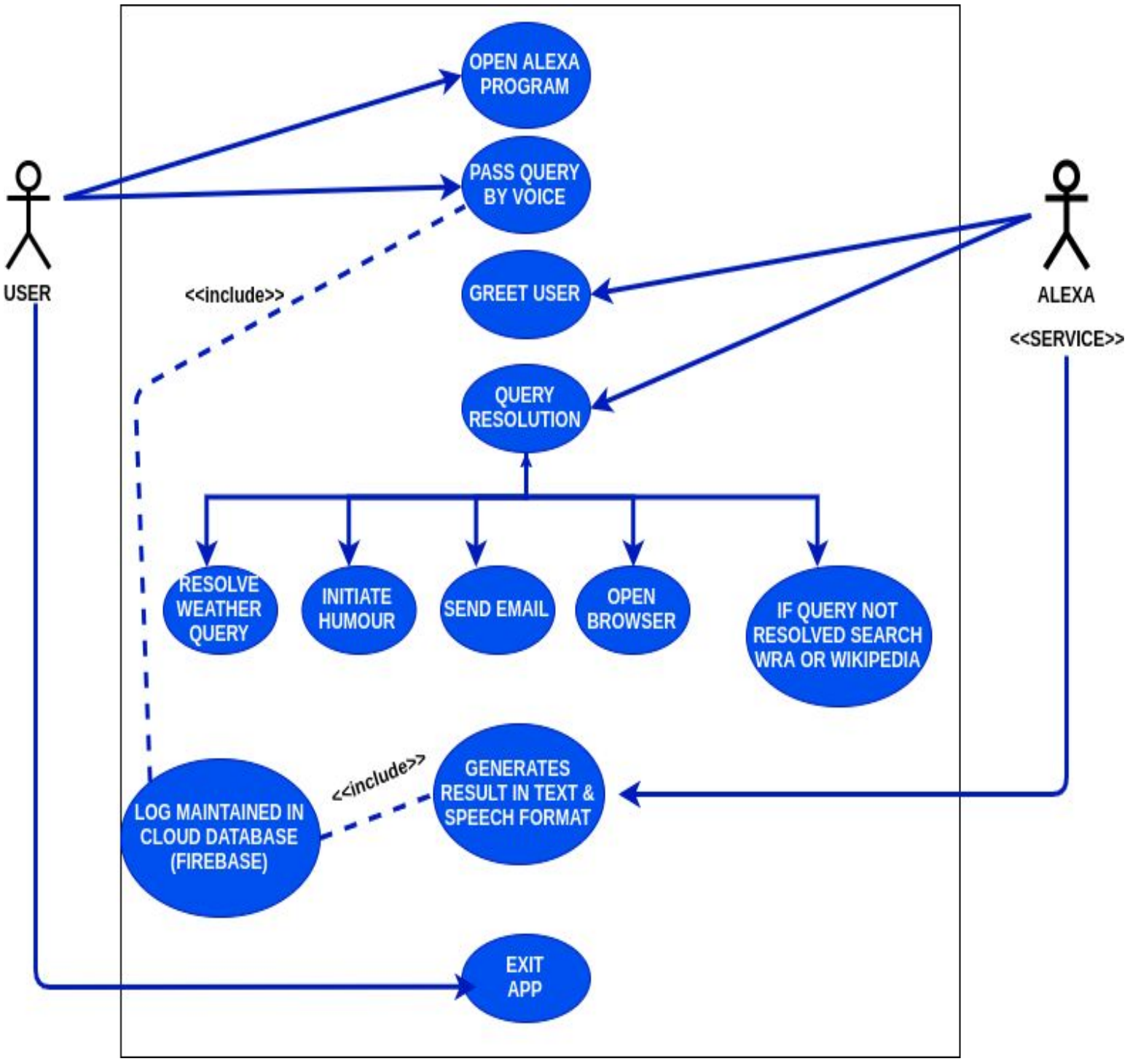
2. Performance risk: The risk that the project will fail to produce results consistent with project specifications and user demands.

S. No.	Risk Name	Risk Type	Probability	RMMM
1.	**Misinterpretation due to accents and or speech recognition**	**Technical**	**30%**	**Mitigation:** Input commands given only in less noisy backgrounds **Monitoring:** Adequate filters for input speech has been applied to filter background noises **Management:** User shall try and give commands in general english and in a preferably quieter place
2.	**Connectivity error**	**Technical**	**40%**	**Mitigation:** Prompt a message in startup function to make sure that the device is connected to the internet **Monitoring:** Instead of crashing the program entirely, temporarily halt execution until connectivity is restored

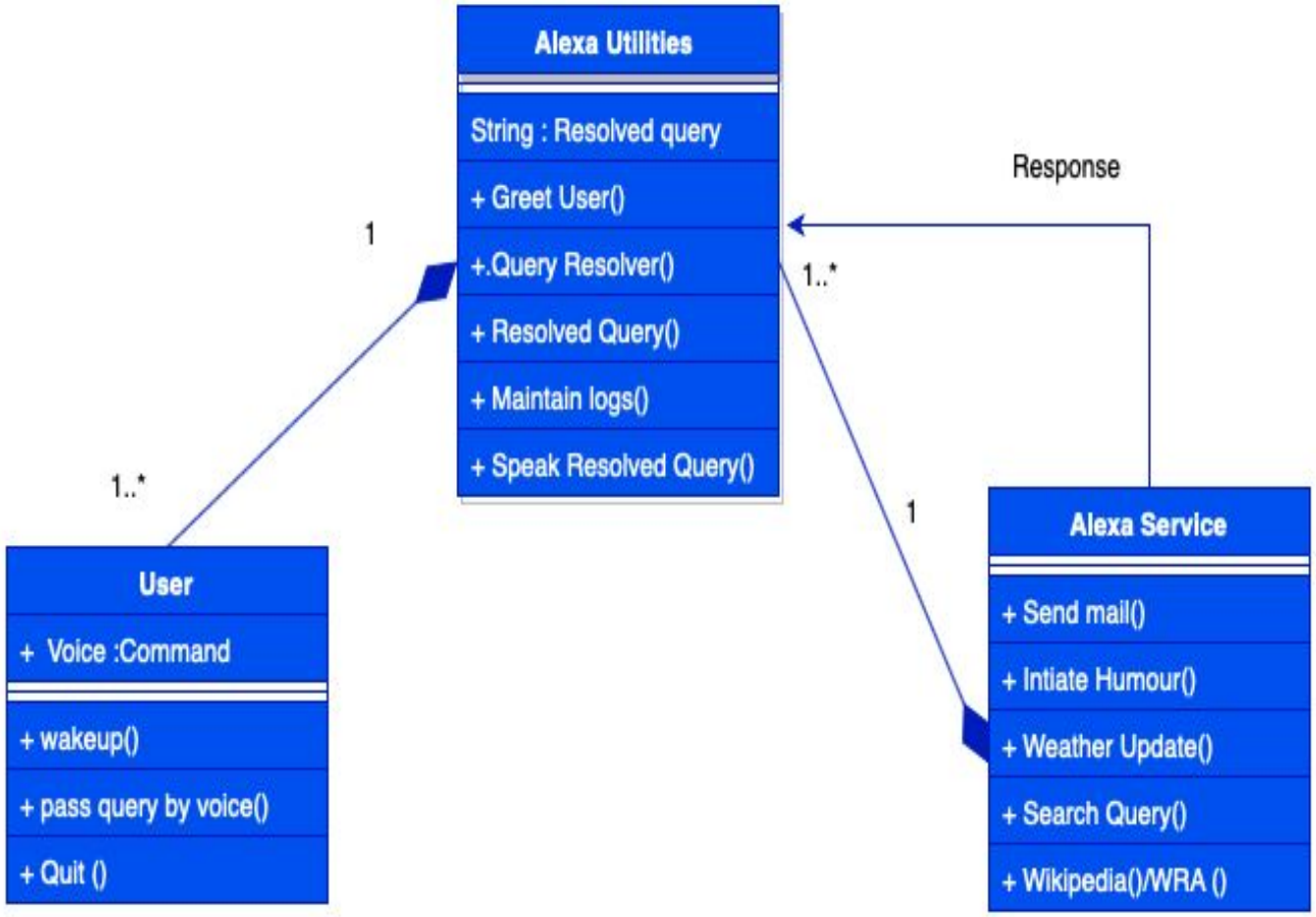
				Management: Display appropriate user-friendly messages to check for connectivity issues
3.	**Performance Risk**	**Non-technical (Project)**	**60%**	**Mitigation:** Ensure the product lives up to the current standards of the market of virtual assistant **Monitoring:** Release system updates before the technology used starts to feel obsolete due to a better competition **Management:** Enter the market only after careful analysis of existing technologies and try to address the residing gaps
4.	**Not enough sales**	**Non-technical (Business)**	**70%**	**Mitigation:** Ensure maximum level of marketing possible within the budget **Monitoring:** Marketing team can take user feedback to make them feel better involved in product development. **Management:** Build a strong marketing team to boost sales which will in turn help improve the technology used.

6. ANALYSIS MODELS

6.1 USE-CASE DIAGRAM:



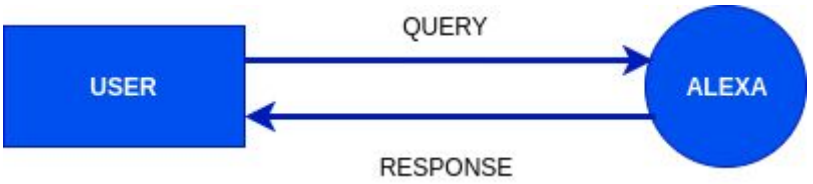
6.2 CLASS DIAGRAM:



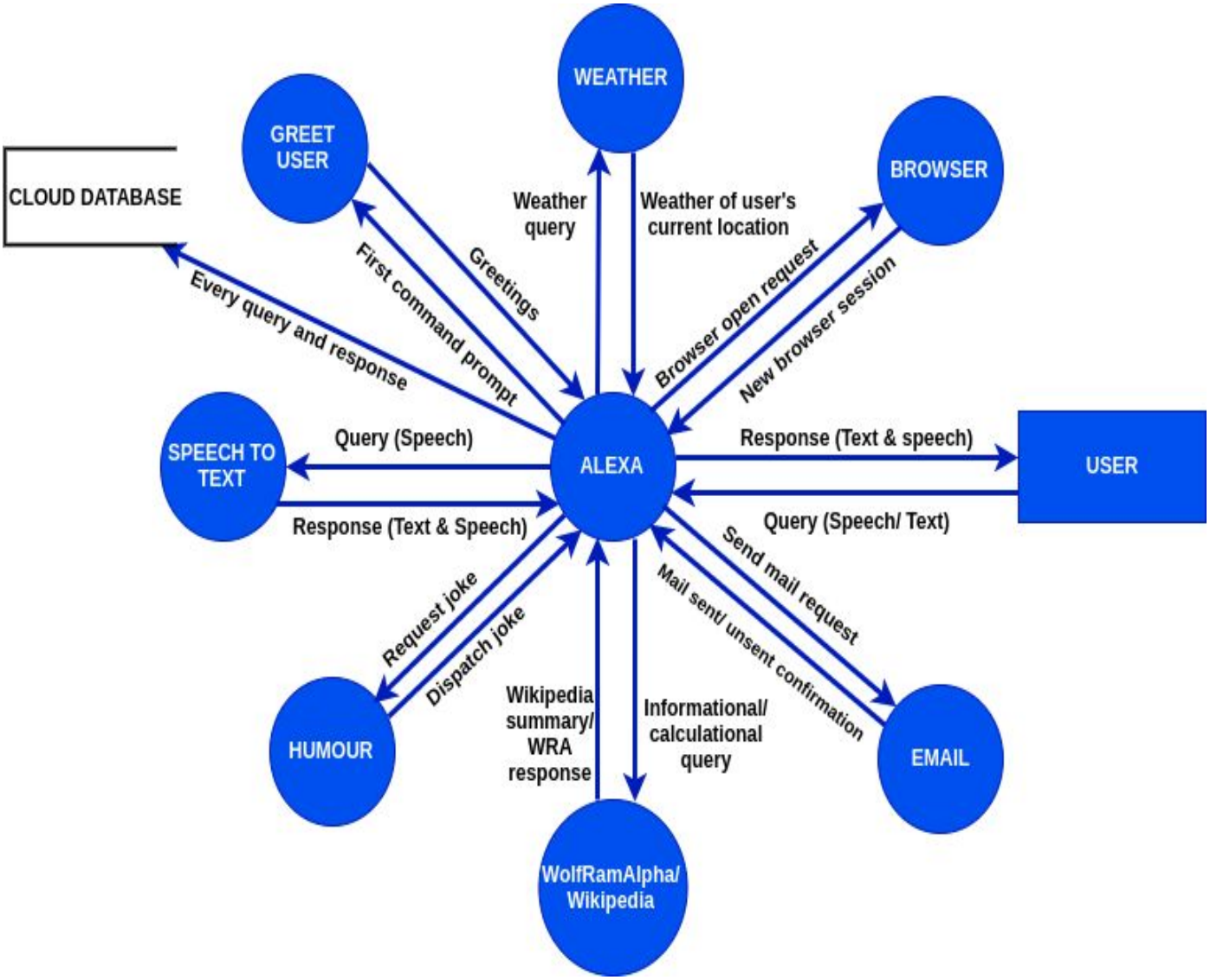
6.3 DATA FLOW DIAGRAM:

6.3.1 LEVEL 0-Birds eye view of what Alexa is capable of doing and how it interacts with the outside world (its users).

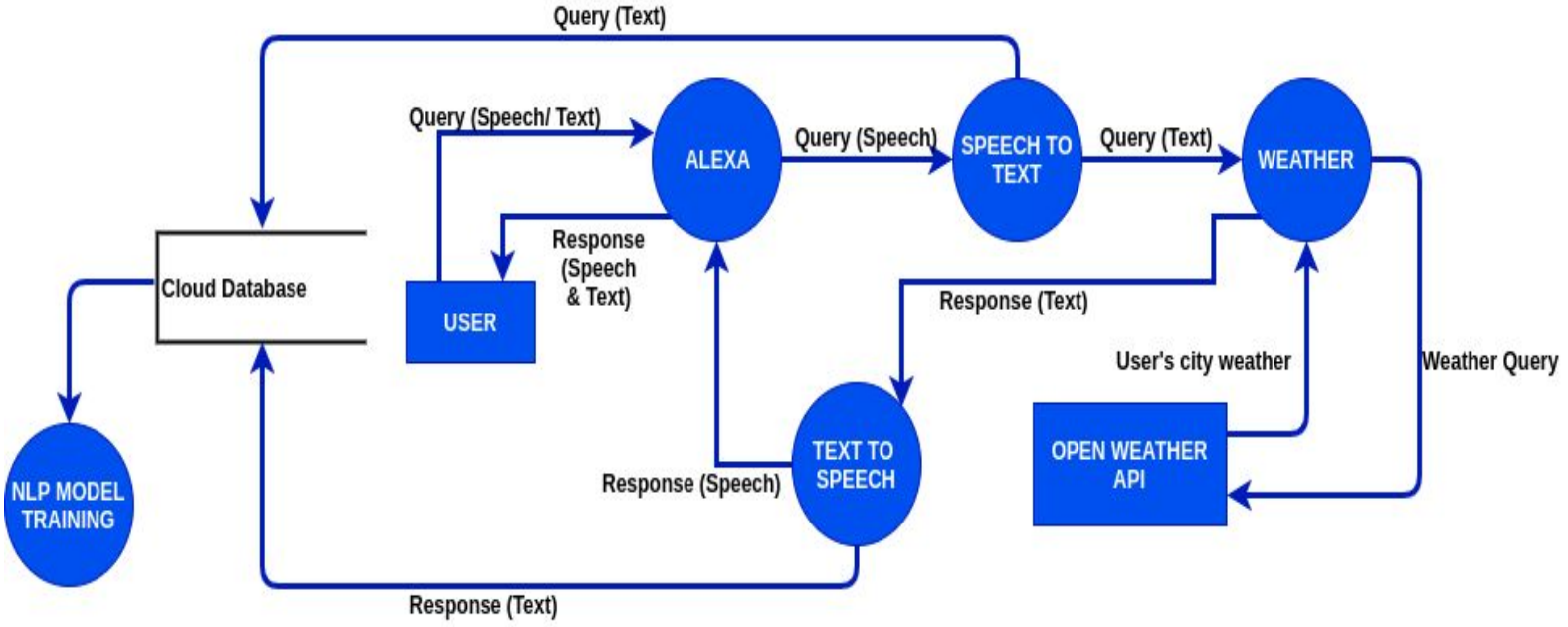
Following diagram depicts the input and the output of the entire system.



6.3.2 LEVEL 1- The basic modules and data flow between them is shown in this diagram. It represents the high level processes and the data store.



6.3.3 LEVEL 2.1- This diagram is a further expansion of Level 1 to depict the sub processes of each module. For eg: Here we see how the weather module works with its corresponding queries and their resolution



6.3.4 LEVEL 2.2- Similarly here we see how the WolfRamAlpha API works with calculational/ business/ general queries and their resolution