**Dedicated to**

**Our Parents**

**Acknowledgement**

First and foremost, we would like to express our gratitude to the All-Powerful, the most gracious and the most merciful, for allowing us to complete this work. And then we would like to thank to our supervisor **Rubya Shaharin**, Assistant Professor, Department of Computer Science & Engineering, **Jatiya Kabi Kazi Nazrul Islam University**, Trishal, Mymensingh, for her helpful suggestions in planning and implementing the work. Also, she greatly helped us in improving report writing and presentations, which, obviously, will guide us throughout our future life.

We wish to express thanks to **Dr. Mst. Jannatul Ferdous**, Professor and Head, Department of Computer Science & Engineering**, Jatiya Kabi Kazi Nazrul Islam University** for providing lab facilities with lab materials related to our project work.

Finally, we would like to thank to the Department of Computer Science and Engineering for giving the opportunity to study here and supporting us greatly through our graduate study.

**Abstract**

Chatbots of any kind are software programs that [interact via conversation](https://www.whoson.com/ecommerce/conversational-commerce-explained/). Usually, this works through text-based input and output. We have tried to do the work on Voice Controlled Chatbot.

Voice chatbots are chatbots that can communicate using vocal input and output. You can talk to the bot out loud, like you would a person. In turn, it will respond with a voice of its own.

Our voice chatbot will work through our voice command. If we want to get any information from internet; we will give the command to the voice chatbot named “Jarvis” and it will give us the desired information by using web scraping. This system is built with Python.

**Chapter-1**

**Introduction**

* 1. **Introduction**

The process of an online chat system would follow a client server approach which acquires the signal and streams it to a server. The input voice is then processed and a response is generated. This process places a large processing requirement on the server’s processor and memory resources. This limitation is even more evident when a large number of users are to be simultaneously accommodated on the system.

Our Voice Controlled Chatbot Jarvis is developed for providing information from various websites which we will want to know. This system will available in any windows, linux and mac operating systems.

* 1. **Purpose of the project**

There is some reason that’s why we want to do such kinds of project. Some objectives are:

* Faster responses and zero waiting time
* Better customer experience
* No human physical touch needed
* Increased productivity Using multitasking
* It will provide personalized response
  1. **Scope of the project**

The scope of our project is to give quick response of any query of user. One user can get proper information from this chatbot by giving the voice command here. The chatbot system modules will recognize our voice and will work on it and will give us the result within few seconds.

* 1. **Overview of this report**

The rest of this report is organized in the following ways.

**Chapter-2: Literature Review**

In this chapter we have discussed about the current situation of the current system and also discussed about our proposed system and benefits of proposed system.

**Chapter-3: Requirements Specification**

In this chapter we have discussed about the requirements of the user and the system where a system has functional, non-functional and domain requirements.

**Chapter-4: Methodology**

In this chapter we have described about the software model we used for the system, its benefits, the reason behind selecting the model and the software and hardware tools for the development.

**Chapter-5: Software Design and Implementation**

In this chapter we have described about the architectural design and its data flow and the actual development or implementation of the proposed system.

**Chapter-6: Testing**

In this section we used software testing process for executing a program with the intent of finding errors that is uncovering errors in a program makes it a feasible task and trying to find the errors (whose presence is assumed) in a program.

**Chapter-7: Conclusion**

In the conclusion part we have discussed and shown the working level of the project and also described about the scope and limitation, further development of the system.

**Chapter-2**

**Literature Review**

1. The process of an online chat system would
2. follow a client server approach which acquires the
3. signal and streams it to a server. The input voice is
4. then processed and a response is generated. This
5. process places a large processing requirement on the
6. server’s processor and memory resources. This
7. limitation is even more evident when a large number
8. of users are to be simultaneously accommodated on
9. the system.
10. The process of an online chat system would
11. follow a client server approach which acquires the
12. signal and streams it to a server. The input voice is
13. then processed and a response is generated. This
14. process places a large processing requirement on the
15. server’s processor and memory resources. This
16. limitation is even more evident when a large number
17. of users are to be simultaneously accommodated on
18. the system.
    1. **Background Study**

In today’s time, ChatBots have become extremely popular. Highly developed ChatBots like Siri, Cortana, and Alexa have left people surprised with their intelligence and capabilities. Our attempt is to see if we can take this a bit further, to modify the user comments to a broader basis and maybe provide the chatbot with other capabilities using a simple web request, automation, and scraping

**Speech Recognition**

There was a time when having robots help you with certain tasks was just something out of a movie. Yet, faster than we thought, chatbots appeared. And after a while, those same chatbots evolved and started talking to us and us with them.

Speech Recognition is a branch of Computer Science and Computational Linguistics. [It’s a field that develops methodologies that allow computers to translate speech inputs into written text.](https://www.ibm.com/cloud/learn/speech-recognition#:~:text=Speech%20recognition%2C%20also%20known%20as,speech%20into%20a%20written%20format.&text=IBM%20has%20had%20a%20prominent,of%20%E2%80%9CShoebox%E2%80%9D%20in%201962.)In addition to Speech Recognition, this technology is also known as Speech Recognition, ASR (Automatic Speech Recognition), Computer Speech Recognition, or Speech to Text (STT).

**How does Speech Recognition work?**

As the name implies, this technology allows recognizing the user’s voice commands and their conversion into text to be processed by the system.

There are two types of Speech Recognition:

#### 1. Speaker-dependent

These are speech recognition systems that require training. In this training, a speaker reads texts or just loose vocabulary to feed the computer’s knowledge base.

This type of Voice Recognition is more accurate since it analyzes the person’s voice in question and tunes the system to recognize that particular voice.

#### 2. Speaker-independent

On the other hand, these speech recognition systems do not require training. They do not recognize specific voices but analyze and translate into text what has been said.

An example of the use of this software occurs in telephone applications.

## The Correlation between Speech Recognition and Artificial Intelligence

[Artificial Intelligence or AI](https://futureoflife.org/background/benefits-risks-of-artificial-intelligence/) is the science that enables a machine to perform tasks, once performed by humans, and to mimic their reasoning process. In other words, it provides, so to speak, some level of intelligence to a system.

AI is currently used in many different areas, and depending on the technologies implemented, the software can perform certain actions.

Sometimes confused, AI and [Machine Learning (ML)](https://mitsloan.mit.edu/ideas-made-to-matter/machine-learning-explained) are distinct yet related concepts. Simply put, Machine Learning is a branch of Artificial Intelligence and is the technology that allows a machine to learn without the need for human intervention.

For a system to learn, it needs to be fed large volumes of data to recognize patterns and thus characterize them.

**Speech Recognition and AI are intertwined by learning processes and data processing.**

As we mentioned earlier, Speech Recognition software needs large volumes of inputs to provide the best possible performance. That’s where AI comes in with the automation of these processes.

In addition, there is also [NLP (Natural Language Processing)](https://nlp.stanford.edu/), another branch of AI, which enables the processing of human languages. For example, English, Portuguese, Spanish, etc.

## Chatbots vs. Voice bots

Chatbots are possibly one of the most accessible ways to see Artificial Intelligence in action.

These conversational agents use areas of AI, namely ML and NLP, to communicate with humans.

Through these technologies, they can understand and extract the information given to them by written messages and respond.

Chatbots are service automation solutions often found in various digital communication channels, such as company websites, [WhatsApp](https://www.visor.ai/whatsapp-how-to-win-with-this-innovative-channel/), Facebook Messenger, etc.

[Learn here how digital channels can transform your communication with your customers!](https://www.visor.ai/digital-contact-channels-the-advantages-for-companies/)

On the other hand, there are voice bots or, more often known as virtual assistants.

These, just like chatbots, have the goal of automating certain processes but with a more personal role.

They work as a kind of personal assistant that responds to voice inputs. The most common examples are [Siri](https://www.apple.com/siri/)and [Alexa](https://developer.amazon.com/alexa). But every day, more companies launch their own voice bots.

So, to summarize, companies mostly use chatbots to automate their services (internal or customer support), and their inputs and outputs are written messages. As the name might suggest, voice bots only receive voice commands and respond in the same way. Moreover, they have a more individual support role.

## Speech Recognition: The Ears of Chatbot

Knowing the difference between chatbots and voice bots, it’s also possible to add a third option involving both solutions – speech recognition chatbots or voice chatbots.

As we mentioned earlier, Speech Recognition technology is what allows a system to convert voice commands and translate them into text.

Once translated, that information goes through the same processing that a written text message would have in a “normal” chatbot.

Just like giving OCR the ability to read the text contained in images, you can also give your chatbot the ability to listen.

If you were curious about what OCR is, you might be interested in the following article: [**OCR: Give Eyes to Your Chatbot**](https://www.visor.ai/ocr-in-chatbot/)**.**

## The Benefits of Speech Recognition Chatbots

Users are becoming more and more interested and demanding regarding these technological solutions.

If it’s to be as similar to human dialogues, then both chatbots and voice chatbots have to meet the demands and characteristics of those same conversations.

Eliminating waiting time, getting answers that correctly answer our questions are two examples of what your customers expect. Be it by message or by voice.

Nobody likes to be talking and get an answer that has nothing to do with the topic. In such cases, you might as well send the user to human assistants via Live Chat.

#### A chatbot already brings many advantages to your company, such as:

* 24/7 service
* Decreased response time
* Automation of repetitive processes
* A way of capturing commercial leads
* Increased satisfaction of customers and their teams
* Means of supporting customers and/or [employees](https://www.visor.ai/what-can-internal-chatbots-do-for-your-company/)

Still, between writing and speaking, no doubt speaking is much more natural for a human being. For this reason, the inclusion of voice commands takes your chatbot to another level.

* 1. **Proposed System**
* It will be pretty cool, if our chatbot can help us book a hotel, play a song for us, tell us  
  about the weather reports, and so on.
* b) We will also try that, our chatbot may help us in gaining knowledge related to science, arts, general knowledge etc.
  1. **Benefits of the Proposed System**

#### A voice chatbot gives you the following benefits:

##### 1) Provides more “human-like” interactions

As we already mentioned, speech is much more natural than writing. For this reason, you get communications that are more spontaneous and not as thought out, as is the case with written messages.

This spontaneity is very useful for your chatbot, as it increases its knowledge base and improves its AI with the various commands it receives.

##### 2) Improves user experience

Companies are very concerned about the usability and user experience on their website and communication channels.

If a chatbot already greatly improved the user experience when navigating to your website, having the ability to talk to the chatbot improves it even more.

##### 3) Enables multitasking

Also related to the consumer experience, the voice option allows the consumer to do other tasks while asking your chatbot for information.

This factor, as well as the others, greatly increases customer satisfaction in that they don’t have to stop doing what they are doing to communicate with the bot.

##### 4) More inclusive solution

Despite being an almost non-existent factor (thankfully!), there are still illiterate people who, for some reason, have not learned to write.

Other than that, there are people with motor disabilities that do not allow them to use keyboards or other types of tactile devices.

Voice can be the ideal solution to include and show that all your customers are taken into consideration.

**Chapter-3**

**Requirements Specification**

Requirements specification is the process of writing down the user and system requirements in a requirements document. Ideally, the user and system requirements should be clear, unambiguous, easy to understand, complete, and consistent.

In practice, this is difficult to achieve as stakeholders interpret the requirements in different ways and there are often inherent conflicts and inconsistencies in the requirements [1]. For a system the functional and nonfunctional requirements should be described so that they are understandable by system users who don’t have detailed technical knowledge. Ideally, they should specify only the external behavior of the system. The requirements document should not include details of the system architecture or design. System requirements are the user requirements expanded versions of the user requirements that are used by software engineers as the starting point for the system design. They add detail and explain how the user requirements should be provided by the system. They may be used as part of the contract for the implementation of the system and should therefore be a complete and detailed specification of the whole system.

**3.1 Functional Requirements**

The functional requirements for a system describe what the system should do. These requirements depend on the type of software being developed, the expected users of the software, and the general approach taken by the organization when writing requirements. When expressed as user requirements, functional requirements are usually described in an abstract way that can be understood by system users. However, more specific functional system requirements describe the system functions, its inputs and outputs, exceptions, etc., in detail.

Functional system requirements vary from general requirements covering what the system should do to very specific requirements reflecting local ways of working or an organization’s existing systems.

**3.2 Non-Functional Requirements**

Non-functional requirements, as the name suggests, are requirements that are not directly concerned with the specific services delivered by the system to its users. They may relate to emergent system properties such as reliability, response time, and store occupancy. Alternatively, they may define constraints on the system implementation such as the capabilities of I/O devices or the data representations used in interfaces with other systems. Non-functional requirements are often more critical than individual functional requirements.

Non-functional requirements arise through user needs, because of budget constraints, organizational policies, the need for interoperability with other software or hardware systems, or external factors such as safety regulations or privacy legislation.

The implementation of the requirements may be diffused throughout the system. There are two reasons for this:

1. Non-functional requirements may affect the overall architecture of a system rather than the individual components.
2. A single non-functional requirement, such as a security requirement, may generate a number of related functional requirements that define new system services that are required.

**3.2.1 Types of Non-Functional Requirements**

Non-functional requirements are three types.

**3.2.1.1 Product Requirements**

These requirements specify or constrain the behavior of the software. Examples include performance requirements on how fast the system must execute and how much memory it requires, reliability requirements that set out the acceptable failure rate, security requirements, and usability requirements.

**Product requirements of the “Voice Controlled Chatbot” are:**

To design this **“Voice Controlled Chatbot”** those modules are required as product requirements.

**3.2.1.2 Organizational Requirements**

These requirements are broad system requirements derived from policies and procedures in the customer’s and developer’s organization. Examples include operational process requirements that define how the system will be used, development process requirements that specify the programming language, the development environment or process standards to be used, and environmental requirements that specify the operating environment of the system.

**Organizational Requirements of “Voice Controlled Chatbot” are:**

**Software Requirements:**

|  |  |
| --- | --- |
| **Specifications** | **Description** |
| Platform: | Python |
| Software: | Pycharm |
| Browsers: | Firefox, Google Chrome, Opera Mini. |

**Hardware Requirements:**

This system is online based, but the system is not uploaded to online. When the system will be uploaded to online, any device able to connect to internet & browse pages can be used to access this system. To access and use the system properly, a computing device like a computer, mobile, a laptop is needed. There is no specific configuration or platform for this proposed system. Any computer or laptop which is able to connect to the internet is able to use this system.

**3.2.1.3 External Requirements**

This broad heading covers all requirements that are derived from factors external to the system and its development process. These may include regulatory requirements that set out what must be done for the system to be approved for use by a regulator, such as a central bank; legislative requirements that must be followed to ensure that the system operates within the law, and ethical requirements that ensure that the system will be acceptable to its users and the general public.

**Chapter-4**

**Methodology**

**Objectives of this chapter:**

In this chapter, we will be discussing about defining a system model. It also discusses about software and hardware tools that needs for system development. It also discusses the reason behind selecting the model.

**4.1 Software Model**

A system model is the conceptual model as a result of system modeling that describes and represents a system. A system comprises multiple views such as planning, requirement (analysis), design, implementation, deployment, structure, behavior, input data and output data views. A system model is required to describe and represent all these multiple views [2].

There are various software development models or methodologies [3]. They are as follows:

* [Waterfall model](http://istqbexamcertification.com/what-is-waterfall-model-advantages-disadvantages-and-when-to-use-it/)
* [V model](http://istqbexamcertification.com/what-is-v-model-advantages-disadvantages-and-when-to-use-it/)
* [Incremental model](http://istqbexamcertification.com/what-is-incremental-model-advantages-disadvantages-and-when-to-use-it/)
* [RAD model](http://istqbexamcertification.com/what-is-rad-model-advantages-disadvantages-and-when-to-use-it/)
* [Agile model](http://istqbexamcertification.com/what-is-agile-model-advantages-disadvantages-and-when-to-use-it/)
* [Iterative model](http://istqbexamcertification.com/what-is-iterative-model-advantages-disadvantages-and-when-to-use-it/)
* [Spiral model](http://istqbexamcertification.com/what-is-spiral-model-advantages-disadvantages-and-when-to-use-it/)
* [Prototype model](http://istqbexamcertification.com/what-is-prototype-model-advantages-disadvantages-and-when-to-use-it/)

In this project we have used the Incremental model for the processing of **“Voice Controlled Chatbot”**.

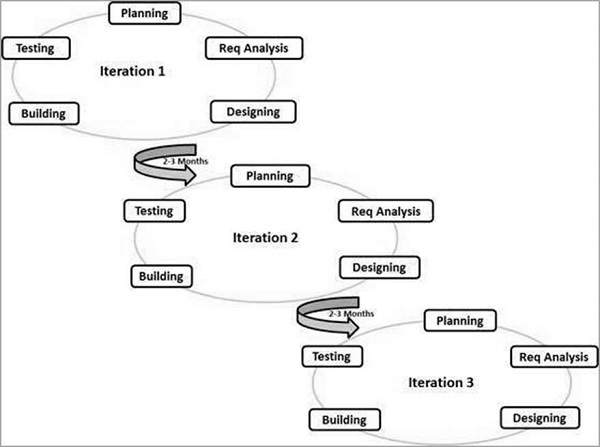
**4.1.1 The Agile Model**

Agile methods universally rely on an incremental approach to software specification, development, and delivery. They are best suited to application development where the system requirements usually change rapidly during the development process. They are intended to deliver working software quickly to customers, who can then propose new and changed requirements to be included in later iterations of the system. They aim to cut down on process bureaucracy by avoiding work that has dubious long-term value and eliminating documentation that will probably never be used. The philosophy behind agile methods is reflected in the agile manifesto that was agreed on by many of the leading developers of these methods. This manifesto states:

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

* Individuals and interactions over processes and tools
* Working software over comprehensive documentation
* Customer collaboration over contract negotiation
* Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.



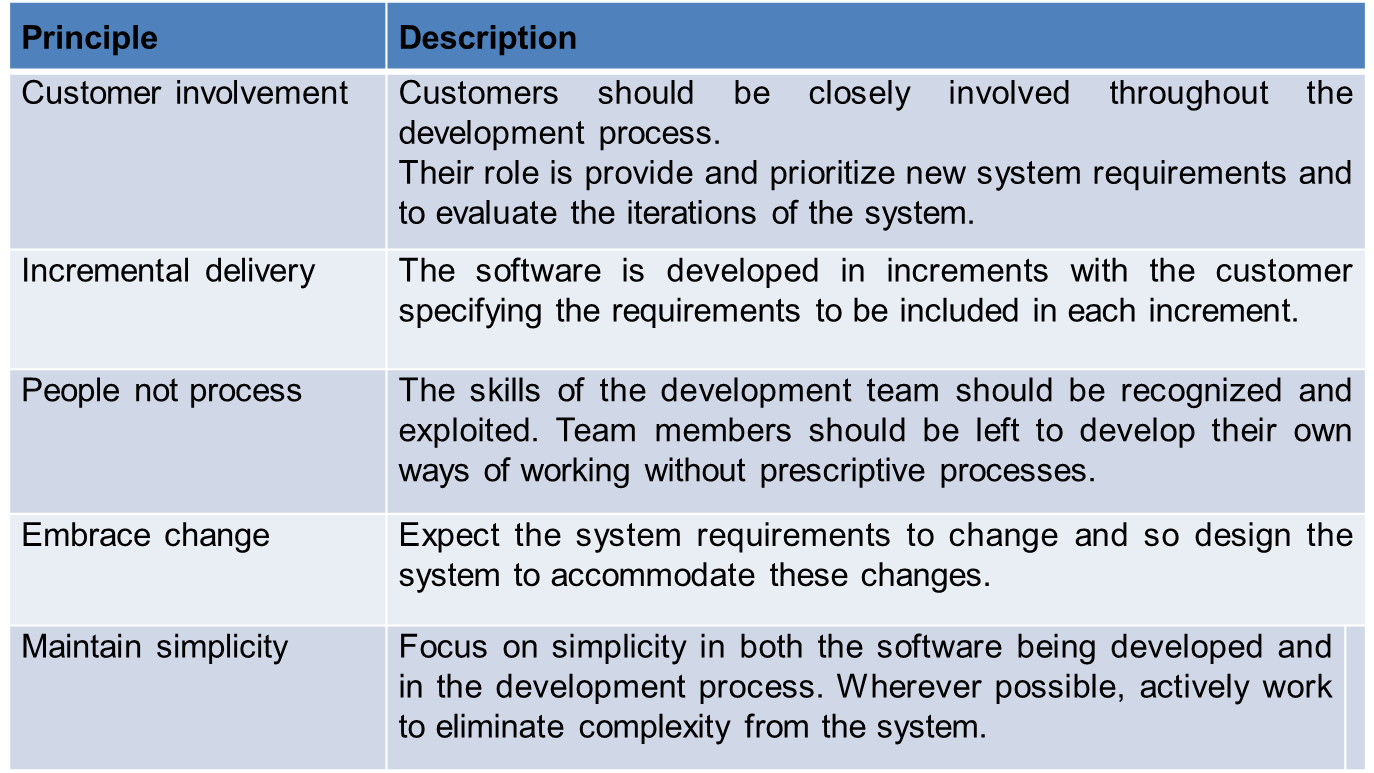
**Agile methods have been very successful for some types of system development:**

* Product development where a software company is developing a small or medium-sized product for sale.
* Custom system development within an organization, where there is a clear commitment from the customer to become involved in the development process and where there are not a lot of external rules and regulations that affect the software.

**4.1.2 Benefits of Agile Model**

* We can deploy software quicker, so your customer can get value sooner rather than later
* We waste fewer resources because you always work on up-to-date tasks
* We can better adapt to change and respond faster
* Faster turnaround times
* We can detect and fix issues and defects faster
* We spend less time on bureaucracy and busywork
* There's a big community of Agile practitioners with whom we can share knowledge
* We can get immediate feedback (which also improves team morale)
* Developers can improve their skills based on QA feedback
* We don't have to worry about premature optimization
* We can experiment and test ideas because its costs are low

**4.1.3 Selecting the Agile Method**



**4.2 Software Development Tools**

A software development tool is a computer program that software developers use to create, debug, maintain, or otherwise support other programs and applications.

The term usually refers to relatively simple programs, that can be combined together to accomplish a task, such as one might use multiple hand tools to fix a physical object. The ability to use a variety of tools productively is one hallmark of a skilled software engineer.

The most basic tools are a source code editor and a compiler or interpreter, which are used ubiquitously and continuously. Other tools are used more or less depending on the language, development methodology, and individual engineer, and are often used for a discrete task, like a debugger or profiler. Tools may be discrete programs, executed separately – often from the command line – or may be parts of a single large program, called an integrated development environment (IDE).

So, to develop a software tools will be hardware, software, programming language etc.

**4.2.1 Hardware Support**

* Processor –Core-i3 or i5
* Hard Disk – Minimum 5 GB Space
* Memory – Minimum 1GB RAM

**4.2.2 Software Support**

* Pycharm
* Anaconda
* Visual Studio code
* Google Chrome/Mozilla Firefox
* Recognizer

**4.2.3 Programming Language**

In the project we use python programming language as python has huge library platform.Weimport tkinter, pyttsx3, speech recognition, web browser for this purpose.

**Chapter-5**

**Software Design and Implementation**

**Software design** is the process by which an [agent](https://en.wikipedia.org/wiki/Agency_(philosophy)) creates a specification of a [software artifact](https://en.wikipedia.org/wiki/Artifact_(software_development)), intended to accomplish [goals](https://en.wikipedia.org/wiki/Goal), using a set of primitive components and subject to [constraints](https://en.wikipedia.org/wiki/Constraint_(mathematics)). Software design may refer to either "all the activity involved in conceptualizing, framing, implementing, commissioning, and ultimately modifying complex systems" or "the activity following [requirements](https://en.wikipedia.org/wiki/Software_requirements) specification and before [programming](https://en.wikipedia.org/wiki/Computer_programming)."

**5.1 Architectural Design**

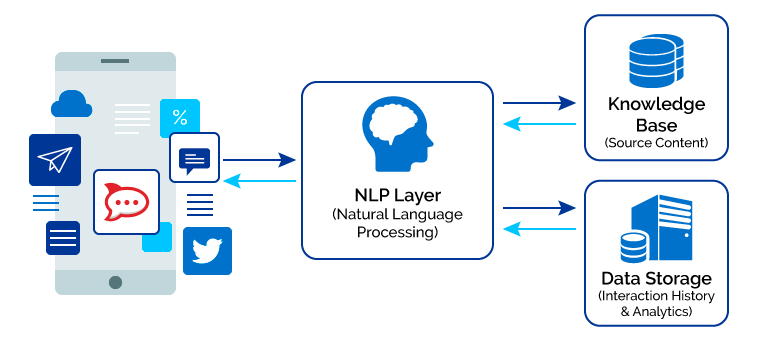
Architectural design is concerned with understanding how a system should be organized and designing the overall structure of that system. In the model of the software development process, architectural design is the first stage in the software design process. It is the critical link between design and requirements engineering, as it identifies the main structural components in a system and the relationships between them. The output of the architectural design process is an architectural model that describes how the system is organized as a set of communicating components. Architectural decomposition is usually necessary to structure and organize the specification.

In practice, conceptual views are almost always developed during the design process and are used to support architectural decision making. They are a way of communicating the essence of a system to different stakeholders. During the design process, some of the other views may also be developed when different aspects of the system are discussed, but there is no need for a complete description from all perspectives.

**5.2 Development of the System**

In order to achieve deliverable of acceptance and meeting of objectives, the new system being built must be tested. The construction phase does two things: builds and tests a functional system that fulfills business or organizational design requirements and implements the interface between the new system and the existing production system. The project team must construct the database, application programs, user and system interfaces, and networks. Developing a system includes programs and structured databases.

Stages of development of the **“Voice Controlled Chatbot”** are described below: -

****

**Chapter-6**

**Software Testing**

**Testing** is intended to show that a program does what it is intended to do and to discover program defects before it is put into use. When we test software, we execute a program using artificial data. We check the results of the test run for errors, anomalies, or information about the programs non-functional attributes. The testing process has two distinct goals:

* To demonstrate to the developer and the customer that the software meets its requirements.
* To discover situations in which the behavior of the software is incorrect, undesirable, or does not conform to its specification.

**6.1 Test Plan**

A Software Test Plan is a document describing the testing scope and activities. It is the basis for formally testing any software/product in a project [7]. It’s a document which provides a central artifact to govern the planning and control of the test effort. It defines the general approach that will be employed to test the software and to evaluate the results of that testing, and is the top-level plan that will be used by stakeholders to govern and direct the detailed testing work.

**6.1.1 Test Plan Types**

One can have the following types of test plans:

* **Master Test Plan:** A single high-level test plan for a project/product that unifies all other test plans.
* **Testing Level Specific Test Plans:** Plans for each level of testing unit testing, module testing, sub-system testing, system testing etc.

**6.2 Types of Testing**

Here we just mentioned that the types of testing that we use in the project.

* **Unit Testing**

In unit testing individual program units or object classes are tested. Here by using this testing we have focused on testing the functionality of methods of **“Voice Controlled Chatbot”** system.

* **Module Testing**

The combination of unit program is called module. Here we tested the modules program have dependency such as all project information are stored in the database and admin can find all information by their records.

* **Subsystem Testing**

Then we combined some modules such as admin modules, user modules for the preliminary system testing in this project.

* **System Testing**

It is the combination of two or more sub-system and then it is tested. Here we tested the entire system as per the requirements.

* **Acceptance Testing**

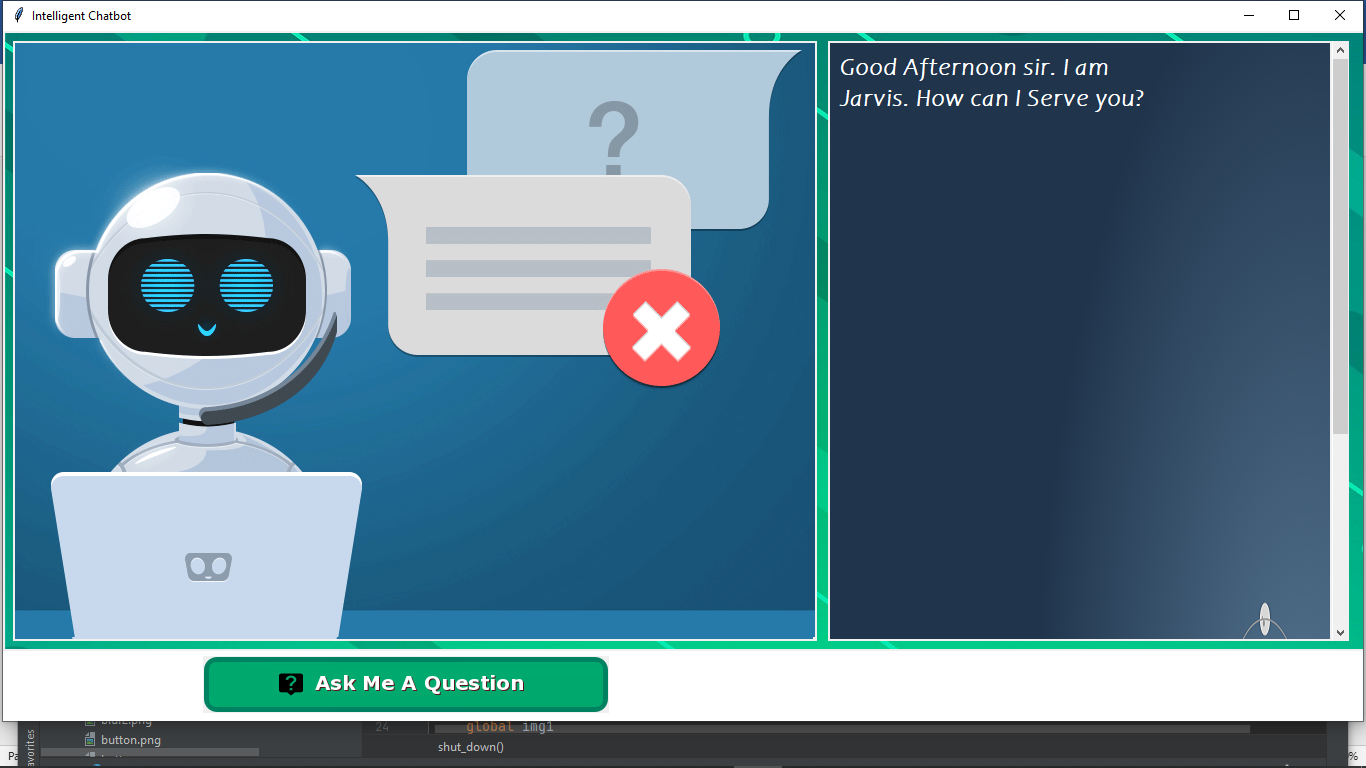
Normally this type of testing is done to verify if system meets the customer specified requirements. After submitting the project to user then they tested it and to determine whether to accept application.

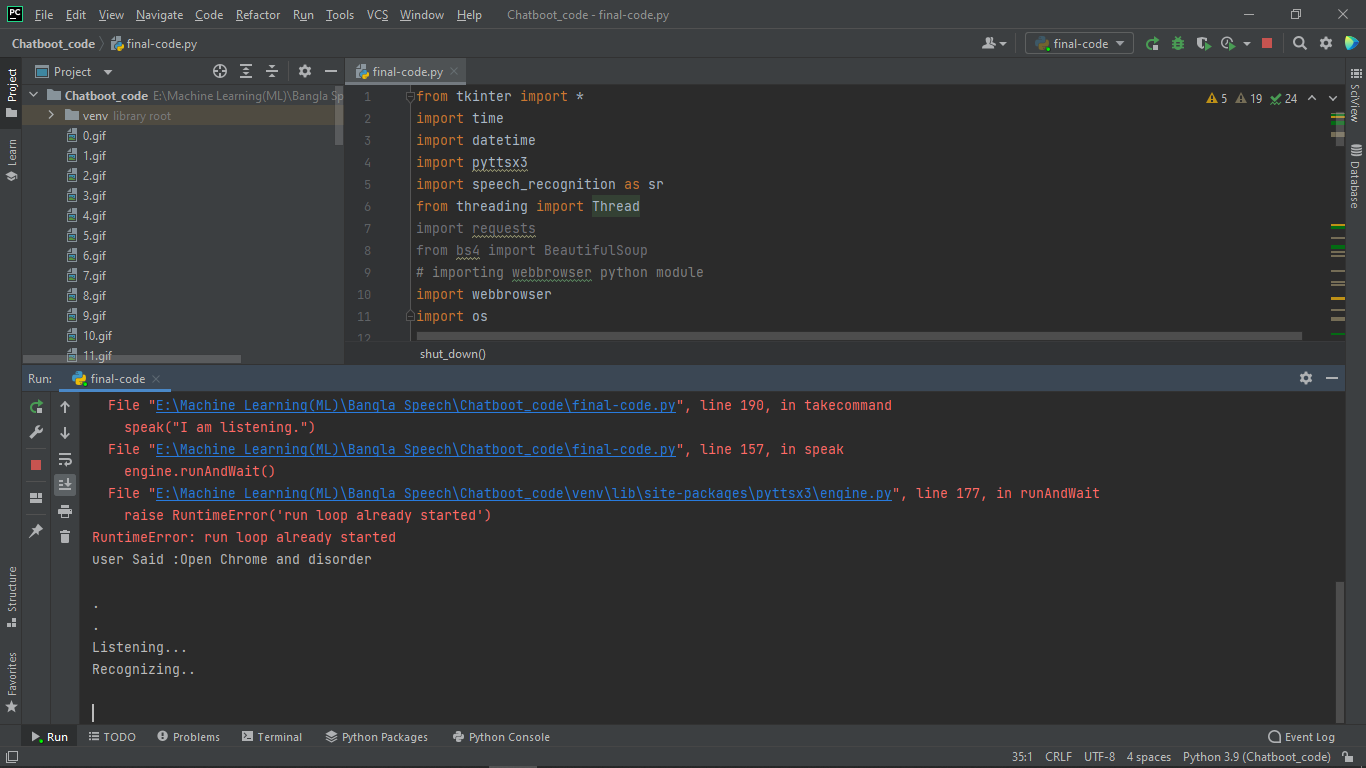
**Chapter-7**

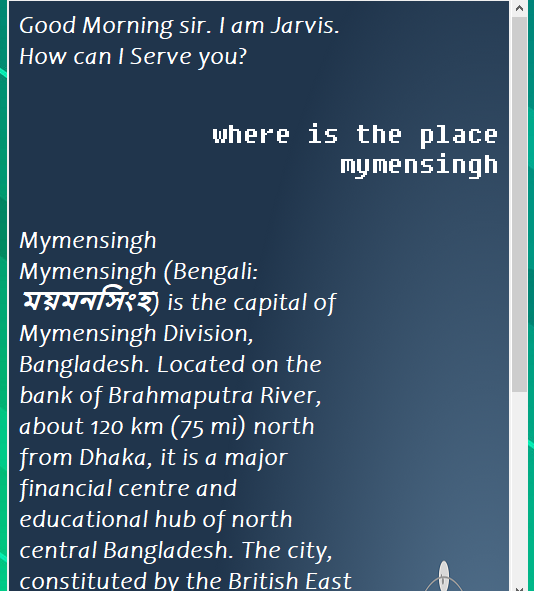
**Conclusion**

**Result and Discussion:**

Screenshot of the project









In this report, we have discussed the development process of **“Voice Controlled Chatbot”**. We have implemented a complete system, but we find some errors which we will recover them in future.

**7.1 Limitations of the System**

The system has the following limitations:

* 1.can’t get command correctly
* 2.unable to recognize sometimes
* 3.API doesn’t response correctly

**7.2 Future development of the System**

Our model is ready to chat, so now let’s create a nice graphical user interface for our chatbot in a new file .In future we shall adding web scrapping with this interface to more interactive our “Voice Controlled Chatbot”.

**References  
I. https://realpython.com/pythonspeechrecognition/?fbclid=IwAR2n96pjA8WBUWmiTy8eSERaGl7FRoQBU  
rZ8TtRO4XdjudwFVWi7iEiT3s  
II. https://www.geeksforgeeks.org/pytho  
n-convert-speech-to-text-and-text-tospeech/?fbclid=IwAR1QoXzGPH2K  
HLtQ41ekOHRqtuE9pngo8U17pcYE  
8\_TywaJcOve2T6Ceec**

**III. https://towardsdatascience.com/howto-build-a-speech-recognition-botwith-python-  
81d0fe3cea9a?fbclid=IwAR3SbV9K7e7B6CFVu5U3\_HnSe6Smpa2  
6gKd\_r56mXZD2S6KS-UlGS4-ZFM  
IV. https://www.youtube.com/watch?v=lp  
SShvLJSnc&fbclid=IwAR2prXXXy3  
-  
IV. 63YdrUAvdAmc1hSf63OM6htaWxn  
pGfSbecKikOCvuf6KiNg4**