A Final Year Project Report on:

“**Raspberry pi based voice operated personal assistant (neobot)**”

*Submitted by:*

Under guidance of:

**Mrs.**

**Department of Computer Engineering**

**XYZ OF ENGINEERING**

2016-2017

**CERTIFICATE**

This is to certify that the pre report on the project entitled

“**Raspberry pi based voice operated personal assistant(neobot)**”

*Submitted by:*

A partial fulfillment for BACHELOR OF COMPUTER ENGINEERING degree course of Mumbai University for year 2016-2017

INTERNAL GUIDE HOD

**( Prof. ) (Prof. )**

INTERNAL EXAMINER PRINCIPAL

EXTERNAL EXAMINER



# **ACKNOWLEDGEMENT**

No project is ever complete without the guidance of those expert how have already traded this past before and hence become master of it and as a result, our leader. So we would like to take this opportunity to take all those individuals how have helped us in visualizing this project.

We express out deep gratitude to our project guide Mrs. For providing timely assistant to our query and guidance that she gave owing to her experience in this field for past many year. She had indeed been a lighthouse for us in this journey.

We would also take this opportunity to thank our project co-ordinate Mr. For his guidance in selecting this project and also for providing us all this details on proper presentation of this project.

We extend our sincerity appreciation to our entire Professor from COLLEGE OF ENGINEERING for their valuable inside and tip during the designing of the project. Their contributions have been valuable in so many ways that we find it difficult to acknowledge of them individual.

We also great full to our HOD Mrs. For extending her help directly and indirectly through various channel in our project work.

.

Thanking You,

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ABSTRACT**

The goal of the project is to form a system which may act as serve the purpose of virtual personal assistant. It includes of components, as an example, IR sensors, DC Motors and Motor driver. Voice controlled intelligent personal assistant bot is a software agent or assistant that can perform tasks for an individual. It's a voice controlled personal assistant whose movements are controlled through voice directions and it's the capability to see the content from pictures then articulate the such as the client by utilizing the built-in speaker. It emphasizes the substitution of screen-based interaction by utilizing ambient technologies, robotics and IoT, means that the user interface is integrated with the physical gadget. It will help the externally disabled to attach with the world by giving them the access to informative sources like Wikipedia, Calculator, so on by using their voice because the command.

**Keywords:** Personal assistant, Text to speech, Speech to plain text, Raspberry pi, voice command system, Virtual assistant, Character Recognition

INDEX

|  |  |  |
| --- | --- | --- |
| SR.NO | TITLE | PG.NO |
| 1) | INTRODUCTION | 1 |
| 2) | LITERATURE SURVEY | 5 |
| 3) | PROBLEM DEFINITION | 8 |
| 4) | REQUIREMENT ANALYSIS | 11 |
| 5) | PLANNING AND ESTIMATION | 13 |
| 6) | ALGORITHM | 15 |
| 7) | IMPLEMENTATION | 22 |
| 8) | ADVANTAGES & DISADVANTAGES | 27 |
| 9) | FUTURE MODIFICATIONS | 29 |
| 10) | APPLICATION | 31 |
| 11) | BIBLIOGRAPHY | 33 |
| 12) | SCREENSHOTS | 48 |
| 13) | SOURCE CODE |  |

**Chapter 1**

**INTRODUCTION**

**INTRODUCTION**

A Voice Command System basically suggests that a system that processes voice as an input decodes or understands the meaning of that input processes it and generates an acceptable voice output. Any voice command system would like 3 basic parts that are speech to text convertor, query processor and a text to speech convertor. Voice has been a really integral a part of communication these days. Since, it's faster to process sound and voices than to process written communication, therefore voice command systems are present in computer devices. There are some superb innovations within the field of speech recognition.

A voice assistant is a software system agent that performs tasks or services appointed to that by the user through numerous commands. The software system agent that is accessed by online chat is referred as 'Chabot' in software terms that are a part of Virtual agent domain. Voice assistants of a similar domain are able to interpret human speech and answer it. On the opposite hand, information storage is major issue in day-after-day life. Each sensible device contains a limited information storage capability. When the storage is full, managing information is a tedious job. For this given issue, we make use of multiple hard drives that even have a limited capability. Moving information from completely different storage devices becomes difficult and is time intense. Once the surface of the disk is damaged by regular crashes, it results in the loss of the data stored in the disk.

These innovations have attributed to the technology industry using Google. It’s 2 step processes that are split into front end and back end. Initial part is responsible for converting numbers and abbreviations to a written word format. Google was able to reduce word error rate by 6% to 10% relative, for the system that had the word error rate of 17% to 52%. This can be additionally referred to as normalization of text. Second part involves the signal to be processed into an understandable one. Text to speech conversion is that the method of converting a machine recognized text into any language that may be identified by a speaker once the text is read out loud.

Our assistant is by artificial means intelligent and controlled through the preset voice directions. It will perform Arithmetic computations dependent on voice commands and giving back the processed solution through a voice and furthermore look web dependent on client's query and giving back the solution through a voice with more intuitive queries by the assistant. It makes the use of the Pi camera module for identifying handwritten or printed content from the image and articulates it to the client utilizing a inbuilt speaker. It gets an even signal from the IR sensing element thus as to find the constant manner for a run.

**Aim of Project**

To build personal assistant by using raspberry pi which assist to user whenever they need.

**Objectives of the Project**

The objectives of the systems development and event management are:

1. Once user send throw a question by using speech then assistant first record users voice and send it to the Google server to be analyzed additional with efficiency.
2. Google Server separates what users question into individual sounds. It counsels a info that containing completely different word’s to find that word mostly intently compare to the combination of individual sounds.
3. It then acknowledges keywords to grasp the tasks and relating functions.
4. For example if assistant notice word like “weather”, it might tell the weather forecasts without lifting a finger.

**Scope of the Project**

This makes the transition smooth to complete voice command system. in addition this also paves way for connected home using Internet of Things, voice command system and computer vision.

**Chapter 2**

**LITERATURE SURVEY**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Title** | **Authors** | **Advantages** | **Disadvantages** | **Result** |
| **Veton Këpuska,“Next-Generation of Virtual Personal**  **Assistants (Microsoft Cortana, Apple Siri, Amazon Alexa**  **and Google Home) “, Published in 2018,** | Veton Këpuska | we found that the  whole concept of this system is the best solution | Accuracy is less | This model will receive the final decision from the Interaction Model with an explanation, then it will choose the  perfect output device to show the result such data show, speakers or screen based on the result. |
| **Voice Controlled Personal Assistant Bot with Smart Storage** | Yash Pujari , Rohan Pawar , Pushkar Patil , Umar Peerzade , Ankita Gupta | storage system boosts the applications of the  assistant thereby increasing its targeted market and  customer base. | As the number of users increases, a need of higher processing power is required by the device, which  could lead to some users getting deprived of their  service | by analyzing  existing systems, we have came to conclusion that the  proposed system will not only be economical but will also  boost the application domain of current systems available  in the market. |
| **Survey on Virtual Assistant: Google Assistant,**  **Siri, Cortana, Alexa** | Amrita S. Tulshan and Sudhir Namdeorao Dhage | Its accuracy responded up to 28.42%. And  last assistant Alexa, responded up to 7.91% which shows very poor result as compared  to all the others. | Main challenge for voice recognition is that  people’s voices vary and they speak in different ways and in different number of  languages. | Its accuracy responded up to 28.42%. And  last assistant Alexa, responded up to 7.91% which shows very poor result as compared  to all the others. |
| **Conversational System, Intelligent Virtual Assistant (IVA) named- DIVA Using Raspberry Pi** | Divij Bajaj, Co-Author- Dr. Dhanya Pramod | , the device also provides much promise for the future as it is highly customizable | If the system is unable to match any of the commands with the keywords provided for each command, the system will apologize for not being able to perform the taskUltimately | Voice search to get anything from google.  Controlling home appliances using voice commands.  Controlling Tableau dashboard using voice commands. |

**Chapter 3**

**PROBLEM**

**DEFINITION**

**Problem Statement**

Today it's become very rare to search out a human being while not interacting with a screen, no matter whether or not it's a personal computer or mobile. We are going to enter the period of screen-less cooperation. Soon with the increasing addition of the internet of Things. A screen that is a post card size surface has somehow become a barrier and escapes the route in social things, gripping our gaze and taking us somewhere else.

**Existing System**

The existing system suffers from the drawback that only predefined voices. Hence the user can’t get the full information and there is a need of convenient storage device which can be accessed by multiple users simultaneously. They store just constrained commands.

**Disadvantages of Existing System:**

1. Voice recognition is not accurate.
2. There is predefining voice so user can’t get full information.

**Proposed System**

The proposed system is with the end goal that it will defeat from the disadvantage of current system by creating it a standalone personal assistant which will be associated completely through the client’s voice. We are developing an improved and efficient system which might offer storage and access the correct information from Google dictionary at the same time to the end users in numerous numbers. Furthermore that performs completely different errands like perusing content from an image, controlling movement through voice based assistant being used in our day to day life.

**Advantages of Proposed System:**

1. Capture photos and convert it into plain text by using OCR.
2. Checking similarity between two phrases.

**Methodology**

The raspberry pi primarily based personal assistant includes of 3 basic modules: Voice control, Character recognition, and Virtual Assistant.

**Voice control:-**

This assistant is controlled by the customer by giving specific voice directions. Right off the bat, the speech is reworked into text by a microphone. At that time, the content is processed and once the order given to the assistant is perceived, the assistant can react by moving in a provided specific guidance.

The steps of Voice control are as follows:

1. It'll take the speech as an input through the Mic

2. Convert the speech into plain text.

3. Then the query is processed supported the plain text generated in step2.

4. Assistant can attempt to move within the provided direction if the path is detected otherwise Assistant can stop.

**Character Recognition:-**

This assistant can have the capability to examine manually written or printed content whether or not from a checked record, a photograph of an archive or from caption content superimposed on a picture. The Pi camera module is used to capture the picture. The content record is then modified over to a document that is given as contribution for interpretation. The captured image is modified over to a .txt record. The picture is caught by the camera module and put away in a .jpg document organize.

**Virtual Assistant:-**

Virtual Assistant depends on natural language process, a system of adjusting over discourse into text. During this module, we utilized the Google Assistant API since Google Assistant is that the best-adjusted remote helper. It got a little-preferred viewpoint over others for precedents Alexa, Siri. It answers the foremost inquiries effectively and furthermore conversational informal and setting mindful. With Alexa and Siri, it's critical to get the command without flaw thus on conjure the specified reaction but in correlation; it's actually adept at understanding the natural language.

**Chapter 4**

**HARDWARE & SOFTWARE REQUIREMENT**

**Hardware and Software requirements**

**Hardware:**

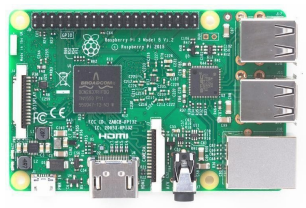
1. Raspberry pi
2. Motor driver IC
3. IR sensor
4. Pi camera

**Software:**

1. Python.
2. openCV.
3. OCR
4. Google assistant SDK
5. Google assistant library
6. Google assistant service

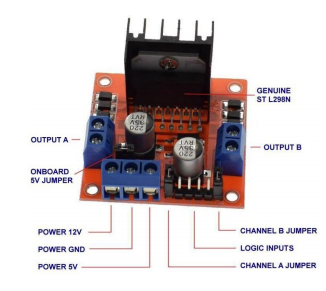
**Hardware Requirement:**

1. **Raspberry pi:**

****

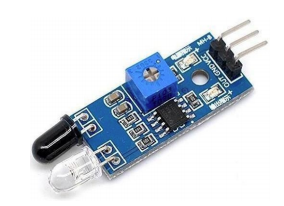
The Raspberry Pi could be a token effort, Visa calculable computer that connects to a computer or TV. it's an able very little device that gives power to individuals of any age to explore computing and to work out the way to program in dialects like Scratch and Python. It will do everything you'd expect a personal computer to try and do, from browsing the web and enjoying the top-notch video to spreadsheets, data processing, and playback.

1. **Motor driver IC:**



The L298N could be a dual H-Bridge motor driver that within the meantime permits speed and management of 2 DC motors. It uses the quality logic level management signal. The module will drive DC motors with voltages of 5 and 35V with a peak current of up to 2A.

1. **IR sensor:**

****

This module incorporates a few infrared transmitters and also the receiver tube, the infrared emission tube that emits a particular frequency, experiences a snag discovery course (reflecting surface), mirrored infrared back to the receiver tube.

1. **Pi camera:**



It tends to be used to require fine quality video, and additionally, stills photos. It underpins 1080p30, 720p60, and VGA90 video modes, and still capture. It appends by means that of a 15cm lace link to the CSI port on the Raspberry Pi.

**Technologies Used:-**

**Python:**

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

**OpenCV:-**

OpenCV (Open source computer Vision Library) could be a BSD-licensed open-source library with many hundred computer view algorithms. It’s a standard structure which means that the package includes some common or static libraries. It presently supports a good range of programming dialects like C++, Python, and Java and then on and is accessible on varied platforms as well as, Windows, Linux, OS X, Android, iOS then forth.

**OCR:-**

OCR stands for "Optical Character Recognition." OCR is a technology that perceives message within a computerized image. it's typically used to understand the message Speech to Text conversion query processing on text Assistant move in given specific direction. yes Assistant move in same direction. in examined records. Its innovation will be utilized to vary over a printed copy of a record into an electronic rendition.

**Google Assistant SDK:-**

It provides us an opportunity to include hot word detection, voice control, normal dialect comprehension and Google's smarts to our gadget. Our device catches an auditory communication, sends it to the Google Assistant and gets spoken audio however the crude content of the articulation.

**Google Assistant Library:-**

It provides us a keeper resolution for incorporating the voice assistant to our device. This library is written in python and is bolstered on devices with Linuxarmv7l and Linux-x86\_64 structures.

**Google Assistant Service:-**

It’s the simplest alternative for adaptability and wide stage bolster. It uncovered a low-level API that specifically manipulates the sound bytes of an Assistant ask for and reaction.

**Chapter 5**

**PLANNING AND ESTIMATION**

**Software development Life Cycle**

The entire project spanned for duration of 6 months. In order to effectively design and develop a cost-effective model the Waterfall model was practiced.

****

**Requirement gathering and Analysis phase:**

This phase started at the beginning of our project, we had formed groups and modularized the project. Important points of consideration were

1. Define and visualize all the objectives clearly.

2.Gather requirements and evaluate them

Consider the technical requirements needed and then collect technical specifications of various peripheral components (Hardware) required.

3. Analyze the coding languages needed for the project.

4. Define coding strategies.

5. Analyze future risks / problems.

6. Define strategies to avoid this risks else define alternate solutions to this risks.

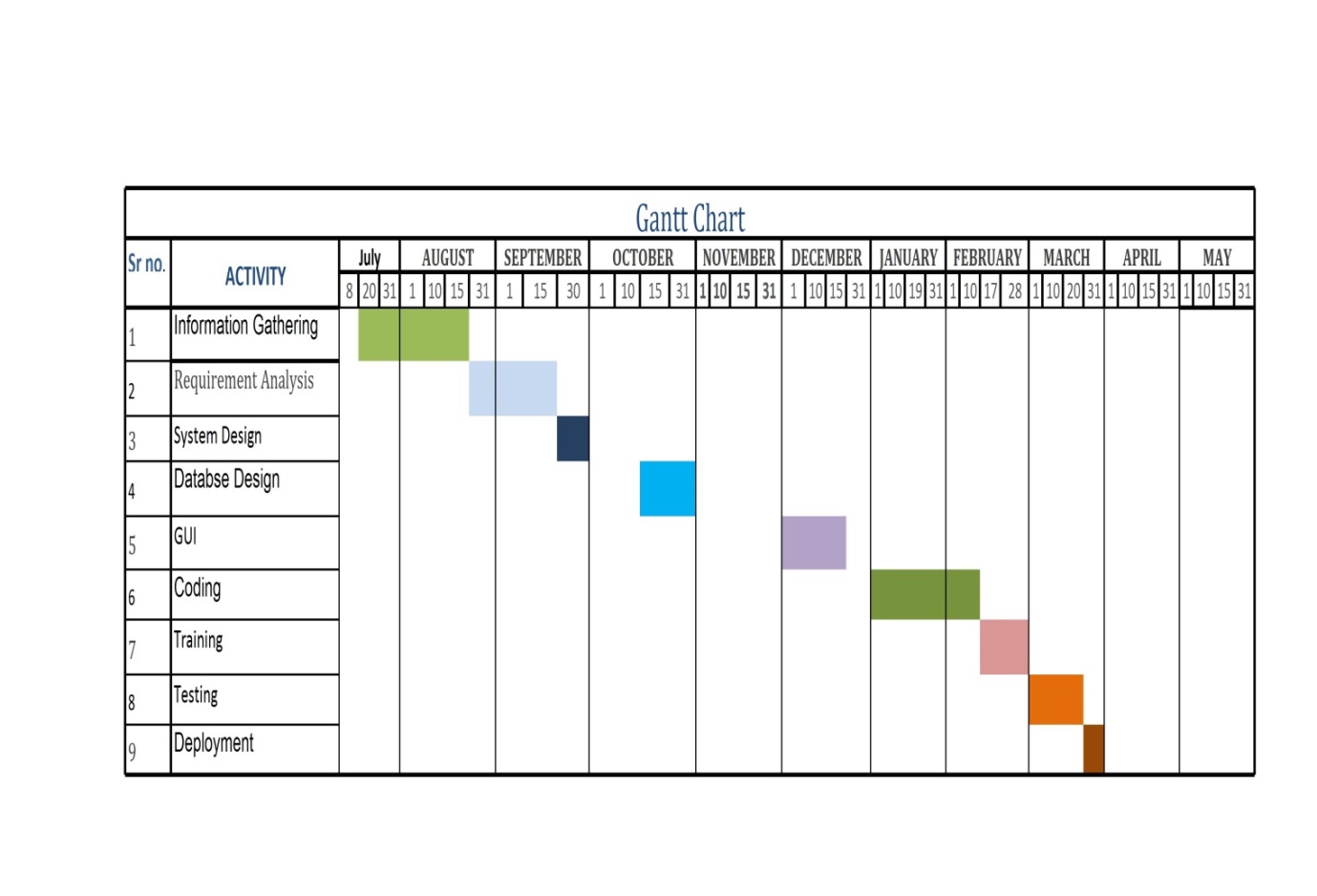
7. Check financial feasibility.

8. Define Gantt charts and assign time span for each phase.

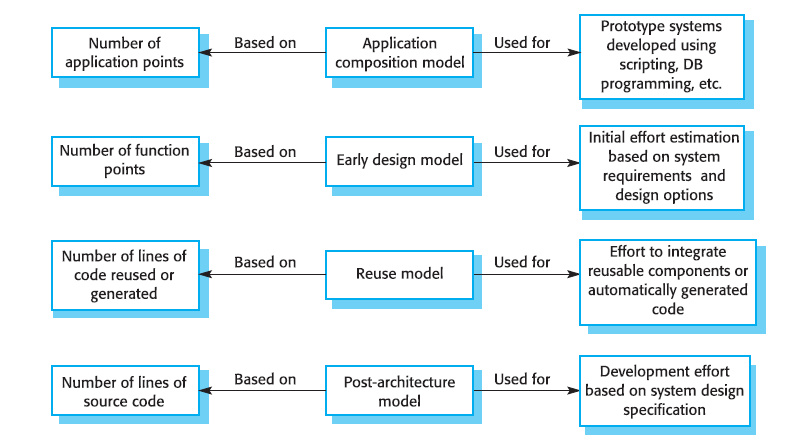
By studying the project extensively we developed a Gantt chart to track and schedule the project. Below is the Gantt chart of our project.

**Timeline**

**Please make changes as per your requirement**



**Cost Estimation**



Cost estimation is done using cocomo model

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Cost Drivers | **Ratings** | | | | | |
| Very Low | Low | Nominal | High | Very High | Extra High |
| **Product attributes** |  |  |  |  |  |  |
| Required software reliability | 0.75 | 0.88 | 1.00 | 1.15 | 1.40 |  |
| Size of application database |  | 0.94 | 1.00 | 1.08 | 1.16 |  |
| Complexity of the product | 0.70 | 0.85 | 1.00 | 1.15 | 1.30 | 1.65 |
| **Hardware attributes** |  |  |  |  |  |  |
| Run-time performance constraints |  |  | 1.00 | 1.11 | 1.30 | 1.66 |
| Memory constraints |  |  | 1.00 | 1.06 | 1.21 | 1.56 |
| Volatility of the virtual machine environment |  | 0.87 | 1.00 | 1.15 | 1.30 |  |
| Required turnabout time |  | 0.87 | 1.00 | 1.07 | 1.15 |  |
| **Personnel attributes** |  |  |  |  |  |  |
| Analyst capability | 1.46 | 1.19 | 1.00 | 0.86 | 0.71 |  |
| Applications experience | 1.29 | 1.13 | 1.00 | 0.91 | 0.82 |  |
| Software engineer capability | 1.42 | 1.17 | 1.00 | 0.86 | 0.70 |  |
| Virtual machine experience | 1.21 | 1.10 | 1.00 | 0.90 |  |  |
| Programming language experience | 1.14 | 1.07 | 1.00 | 0.95 |  |  |
| **Project attributes** |  |  |  |  |  |  |
| Use of software tools | 1.24 | 1.10 | 1.00 | 0.91 | 0.82 |  |
| Application of software engineering methods | 1.24 | 1.10 | 1.00 | 0.91 | 0.83 |  |
| Required development schedule | 1.23 | 1.08 | 1.00 | 1.04 | 1.10 |  |

The Intermediate Cocomo formula now takes the form:

**E=*ai*(kloc)*(bi)*.EAF**

Using above calculation we found that the total time period of the project is around 6 months, the per month cost comes out to be Rs.12, 000/- so the total comes to be Rs.72, 000/-

**FEASIBILITY STUDY**

This system is feasible for all hotels, home, offices, labs, this system will use without experts in that field anyone will use who have knowledge about using online services which will help to use this system. Any generation people will use this system in pc.

**TECHNICAL FEASIBILITY**

The system must be evaluated from the technical point of view first. The assessment of this feasibility must be based on an outline design of the system requirement in the terms of input, output, programs and procedures. Having identified an outline system, the investigation must go on to suggest the type of equipment, required method developing the system, of running the system once it has been designed.

* Technical issues raised during the investigation are:
* Does the existing technology sufficient for the suggested one?
* Can the system expand if developed?

The project should be developed such that the necessary functions and performance are achieved within the constraints. The project is developed within latest technology. Through the technology may become obsolete after some period of time, due to the fact that never version of same software supports older versions, the system may still be used. So there are minimal constraints involved with this project. The system has been developed using Java the project is technically feasible for development.

**ECONOMIC FEASIBILITY**

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminary investigation:

* The costs conduct a full system investigation.
* The cost of the hardware and software.
* The benefits in the form of reduced costs or fewer costly errors.

Since the system is developed as part of project work, there is no manual cost to spend for the proposed system. Also all the resources are already available, it give an indication of the system is economically possible for development.

**BEHAVIORAL FEASIBILITY**

This includes the following questions:

* Is there sufficient support for the users?
* Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible.

RISK ANALYSISPROCESS

Regardless of the prevention techniques employed, possible threats that could arise inside or outside the organization need to be assessed. Although the exact nature of potential disasters or their resulting consequences are difficult to determine, it is beneficial to perform a comprehensive risk assessment of all threats that can realistically occur to the organization. Regardless of the type of threat, the goals of business recovery planning are to ensure the safety of customers, employees and other personnel during and following a disaster.

The relative probability of a disaster occurring should be determined. Items to consider in determining the probability of a specific disaster should include, but not be limited to: geographic location, topography of the area, proximity to major sources of power, bodies of water and airports, degree of accessibility to facilities within the organization, history of local utility companies in providing uninterrupted services, history of the area’s susceptibility to natural threats, proximity to major highways which transport hazardous waste and combustible products. Potential exposures may be classified as natural, technical, or human threats. Examples include:

* Natural Threats: internal flooding, external flooding, internal fire, external fire, seismic activity, high winds, snow and ice storms, volcanic eruption, tornado, hurricane, epidemic, tidal wave, typhoon.
* **Technical Threats:** power failure/fluctuation, heating, ventilation or air conditioning failure, malfunction or failure of CPU, failure of system software, failure of application software, telecommunications failure, gas leaks, communications failure, nuclear fallout.
* Human Threats: robbery, bomb threats, embezzlement, extortion, burglary, vandalism, terrorism, civil disorder, chemical spill, sabotage, explosion, war, biological contamination, radiation contamination, hazardous waste, vehicle crash, airport proximity, work stoppage (Internal/External), computer crime.

All locations and facilities should be included in the risk analysis. Rather than attempting to determine exact probabilities of each disaster, a general relational rating system of high, medium and low can be used initially to identify the probability of the threat occurring. The risk analysis also should determine the impact of each type of potential threat on various functions or departments within the organization. A Risk Analysis Form, found here(PDF Format), can facilitate the process. The functions or departments will vary by type of organization. The planning process should identify and measure the likelihood of all potential risks and the impact on the organization if that threat occurred.

To do this, each department should be analyzed separately. Although the main computer system may be the single greatest risk, it is not the only important concern. Even in the most automated organizations, some departments may not be computerized or automated at all. In fully automated departments, important records remain outside the system, such as legal files, PC data, software stored on diskettes, or supporting documentation for data entry. The impact can be rated as: 0= No impact or interruption in operations, 1= Noticeable impact, interruption in operations for up to 8 hours, 2= Damage to equipment and/or facilities, interruption in operations for 8 - 48 hours, 3= Major damage to the equipment and/or facilities, interruption in operations for more than 48 hours. All main office and/or computer center functions must be relocated. Certain assumptions may be necessary to uniformly apply ratings to each potential threat.

Following are typical assumptions that can be used during the risk assessment process:

1. Although impact ratings could range between 1 and 3 for any facility given a specific set of circumstances, ratings applied should reflect anticipated, likely or expected impact on each area.

2. Each potential threat should be assumed to be “localized” to the facility being rated.

3. Although one potential threat could lead to another potential threat (e.g., a hurricane could spawn tornados), no domino effect should be assumed.

4. If the result of the threat would not warrant movement to an alternate site(s), the impact should be rated no higher than a “2.”

5. The risk assessment should be performed by facility. To measure the potential risks, a weighted point rating system can be used.

**Functional requirement**

In software engineering, a functional requirement defines a function of a software system or its component. A function is described as a set of inputs, the behavior, and outputs (see also software). Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define *what* a system is supposed to accomplish. Behavioral requirements describing all the cases where the system uses the functional requirements are captured in use cases.

Functional requirements are supported by non-functional requirements (also known as *quality requirements*), which impose constraints on the design or implementation (such as performance requirements, security, or reliability). Generally, functional requirements are expressed in the form "system must do <requirement>", while non-functional requirements are "system shall be <requirement>". The plan for implementing *functional* requirements is detailed in the system *design*. The plan for implementing *non-functional* requirements is detailed in the system *architecture*.

As defined in requirements engineering, functional requirements specify particular results of a system. This should be contrasted with non-functional requirements which specify overall characteristics such as cost and reliability. Functional requirements drive the *application architecture* of a system, while non-functional requirements drive the *technical architecture* of a system.

**Non-functional requirement**

In systems engineering and requirements engineering, a non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. This should be contrasted with functional requirements that define specific behavior or functions. The plan for implementing *functional* requirements is detailed in the system *design*. The plan for implementing *non-functional* requirements is detailed in the system *architecture*.

In general, functional requirements define what a system is supposed to *do* whereas non-functional requirements define how a system is supposed to *be*. Functional requirements are usually in the form of "system shall do <requirement>", while non-functional requirements are "system shall be <requirement>".

Non-functional requirements are often called qualities of a system. Other terms for non-functional requirements are "constraints", "quality attributes", "quality goals", "quality of service requirements" and "non-behavioral requirements".

**Chapter 6**

**TESTING**

**Testing**

Software testing methods are traditionally divided into black box testing and white box testing. These two approaches are used to describe the point of view that a test engineer takes when designing test cases.

#### Black box testing

Black box testing treats the software as a "black box"—without any knowledge of internal implementation. Black box testing methods include: equivalence partitioning, boundary value analysis, all-pairs testing, fuzz testing, model-based testing, traceability matrix, exploratory testing and specification-based testing.

* **Specification-based testing**:

Specification-based testing aims to test the functionality of software according to the applicable requirements. Thus, the tester inputs data into, and only sees the output from, the test object. This level of testing usually requires thorough test cases to be provided to the tester, who then can simply verify that for a given input, the output value (or behavior), either "is" or "is not" the same as the expected value specified in the test case.Specification-based testing is necessary, but it is insufficient to guard against certain risks.

* **Advantages and disadvantages**:

The black box tester has no "bonds" with the code, and a tester's perception is very simple: a code *must* have bugs. Using the principle, "Ask and you shall receive," black box testers find bugs where programmers do not. *But,* on the other hand, black box testing has been said to be "like a walk in a dark labyrinth without a flashlight," because the tester doesn't know how the software being tested was actually constructed. As a result, there are situations when (1) a tester writes many test cases to check something that could have been tested by only one test case, and/or (2) some parts of the back-end are not tested at all.

Therefore, black box testing has the advantage of "an unaffiliated opinion," on the one hand, and the disadvantage of "blind exploring," on the other.

#### White box testing

White box testing is when the tester has access to the internal data structures and algorithms including the code that implement these.

**Types of white box testing**

The following types of white box testing exist:

* API testing (application programming interface) - Testing of the application using Public and Private apis
* Code coverage - creating tests to satisfy some criteria of code coverage (e.g., the test designer can create tests to cause all statements in the program to be executed at least once)
* Fault injection methods - improving the coverage of a test by introducing faults to test code paths
* Mutation testing methods
* Static testing - White box testing includes all static testing

**Code completeness evaluation**

White box testing methods can also be used to evaluate the completeness of a test suite that was created with black box testing methods. This allows the software team to examine parts of a system that are rarely tested and ensures that the most important function points have been tested.

Two common forms of code coverage are:

* *Function coverage*, which reports on functions executed
* *Statement coverage*, which reports on the number of lines executed to complete the test

They both return code coverage metric, measured as a percentage.

**Integration testing**

Integration testing is any type of software testing, which seeks to verify the interfaces between components against a software design. Software components may be integrated in an iterative way or all together ("big bang"). Normally the former is considered a better practice since it allows interface issues to be localized more quickly and fixed.

### Acceptance testing

Acceptance testing can mean one of two things:

1. A smoke test is used as an acceptance test prior to introducing a new build to the main testing process, i.e. Before integration or regression.
2. Acceptance testing performed by the customer, often in their lab environment on their own HW, is known as user acceptance testing (UAT).

**Chapter 7**

**Design & Implementation**

System flowchart:

A **flowchart** is a type of diagram that represents an algorithm or process, showing the steps as boxes of various kinds, and their order by connecting them with arrows. This diagrammatic representation illustrates a solution to a given problem. Process operations are represented in these boxes, and arrows; rather, they are implied by the sequencing of operations. Flowcharts are used in analyzing, designing, documenting or managing a process or program in various fields.

**Start and end symbols**

-Represented as circles, ovals or rounded (fillet) rectangles, usually containing the word “Start” or “End”, or another phrase signaling the start or end of a process, such as “submit inquiry” or “receive product”.

**Arrows**

Showing “[flow of control](http://en.wikipedia.org/wiki/Control_flow)” An arrow coming from one symbol and ending at another symbol represents that control passes to the symbol the arrow points to. The line for the arrow can be solid or dashed. The meaning of the arrow with dashed line may differ from one flowchart to another and can be defined in the legend.

**Generic processing steps**

Represented as [rectangles](http://en.wikipedia.org/wiki/Rectangles) Examples: “Add 1 to X”; “replace identified part”; “save changes” or similar.

**Subroutines**

Represented as rectangles with double-struck vertical edges; these are used to show complex processing steps which may be detailed in a separate flowchart. Example: process-files. One subroutine may have multiple distinct entry points or exit flows (see [co routine](http://en.wikipedia.org/wiki/Coroutine)); if so, these are shown as labeled ‘wells’ in the rectangle, and control arrows connect to these ‘wells’.

**Input/output**

Represented as a [parallelogram](http://en.wikipedia.org/wiki/Parallelogram) Examples: Get X from the user; display X **Prepare conditional** Represented as a [hexagon](http://en.wikipedia.org/wiki/Hexagon) Shows operations which have no effect other than preparing a value for a subsequent conditional or decision step (see below).

**Conditional or decision**

Represented as a diamond ([rhombus](http://en.wikipedia.org/wiki/Rhombus)) showing where a decision is necessary, commonly a Yes/No question or True/False test. The conditional symbol is peculiar in that it has two arrows coming out of it, usually from the bottom point and right point, one corresponding to Yes or True, and one corresponding to No or False. (The arrows should always be labeled.) More than two arrows can be used, but this is normally a clear indicator that a complex decision is being taken, in which case it may need to be broken-down further or replaced with the “pre-defined process” symbol.

**Junction symbol**

Generally represented with a black blob, showing where multiple control flows converge in a single exit flow. A junction symbol will have more than one arrow coming into it, but only one going out. In simple cases, one may simply have an arrow point to another arrow instead. These are useful to represent an [iterative](http://en.wikipedia.org/wiki/Iteration) process (what in Computer Science is called a [loop](http://en.wikipedia.org/wiki/Control_flow#Loops)). A loop may, for example, consist of a connector where control first enters, processing steps, a conditional with one arrow exiting the loop, and one going back to the connector. For additional clarity, wherever two lines accidentally cross in the drawing, one of them may be drawn with a small semicircle over the other, showing that no junction is intended.

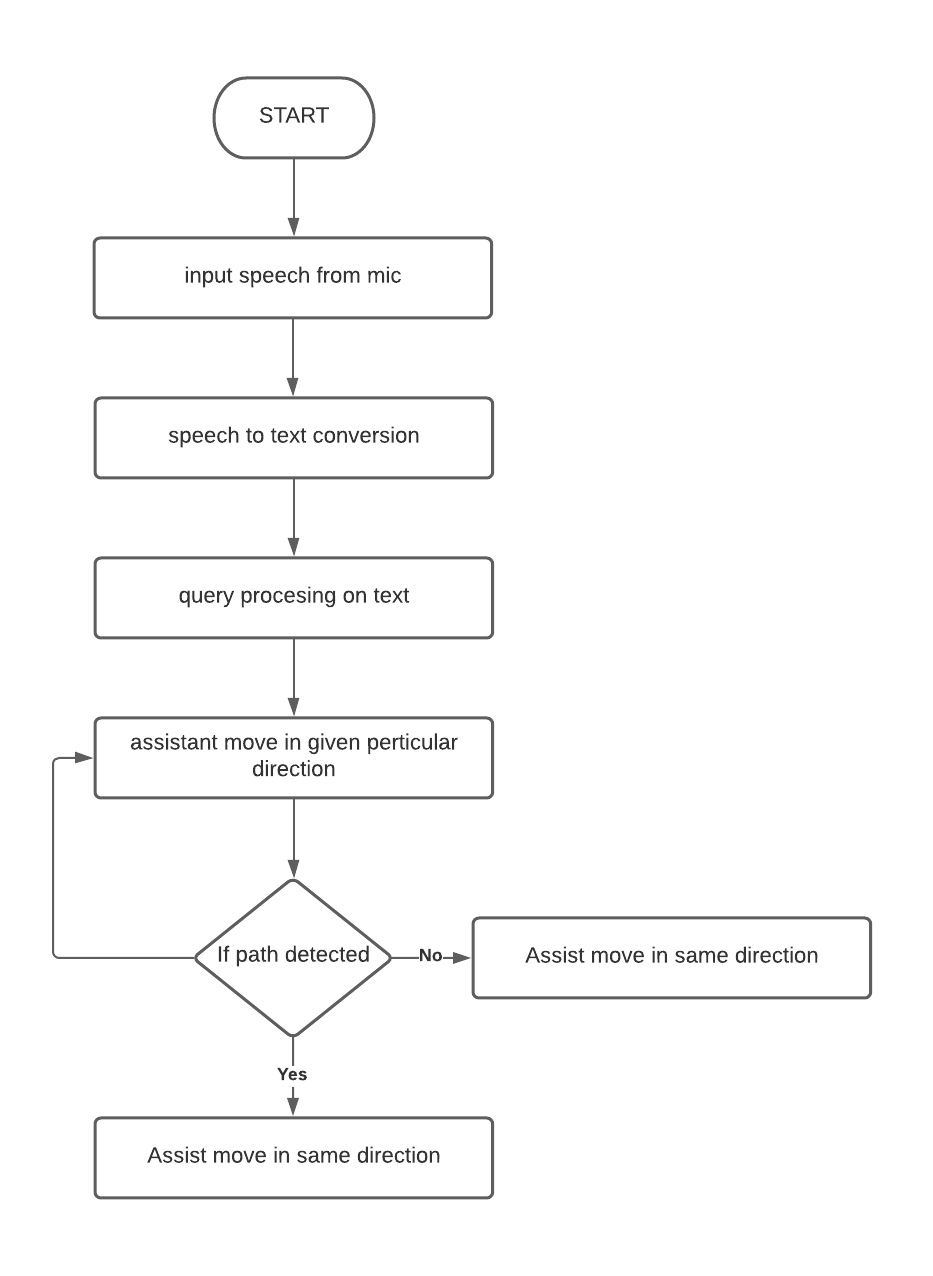
**Labeled connectors**

Represented by an identifying label inside a circle. Labeled connectors are used in complex or multi-sheet diagrams to substitute for arrows. For each label, the “outflow” connector must always be unique, but there may be any number of “inflow” connectors. In this case, a junction in control flow is implied.

**Concurrency symbol**

Represented by a double transverse line with any number of entry and exit arrows These symbols are used whenever two or more control flows must operate simultaneously. The exit flows are activated concurrently when all of the entry flows have reached the concurrency symbol. A concurrency symbol with a single entry flow is a *fork*; one with a single exit flow is a *join*. It is important to remember to keep these connections logical in order. All processes should flow from top to bottom and left to right.

**Flow chart**



**SYSTEM IMPLEMENTATION**

Implementation includes all those activities that take place to convert from the old system to the new. The old system consists of manual operations, which is operated in a very different manner from the proposed new system. A proper implementation is essential to provide a reliable system to meet the requirements of the organizations. An improper installation may affect the success of the computerized system.

**IMPLEMENTATION METHODS:**

There are several methods for handling the implementation and the consequent conversion from the old to the new computerized system.

The most secure method for conversion from the old system to the new system is to run the old and new system in parallel. In this approach, a person may operate in the manual older processing system as well as start operating the new computerized system. This method offers high security, because even if there is a flaw in the computerized system, we can depend upon the manual system. However, the cost for maintaining two systems in parallel is very high. This outweighs its benefits.

Another commonly method is a direct cut over from the existing manual system to the computerized system. The change may be within a week or within a day. There are no parallel activities. However, there is no remedy in case of a problem. This strategy requires careful planning.

A working version of the system can also be implemented in one part of the organization and the personnel will be piloting the system and changes can be made as and when required. But this method is less preferable due to the loss of entirety of the system.

**IMPLEMENTATION PLAN:**

The implementation plan includes a description of all the activities that must occur to implement the new system and to put it into operation. It identifies the personnel responsible for the activities and prepares a time chart for implementing the system. The implementation plan consists of the following steps.

* List all files required for implementation.
* Identify all data required to build new files during the implementation.
* List all new documents and procedures that go into the new system.

The implementation plan should anticipate possible problems and must be able to deal with them. The usual problems may be missing documents; mixed data formats between current and files, errors in data translation, missing data etc.

**DFD**

A data flow diagram (DFD) is a graphical representation of the flow of data through an information system. A data flow diagram can also be used for the visualization of data processing (structured design). It is common practice for a designer to draw a context-level DFD first which shows the interaction between the system and outside entities. This context-level DFD is then exploded to show more detail of the system being modeled.

**Symbols:**

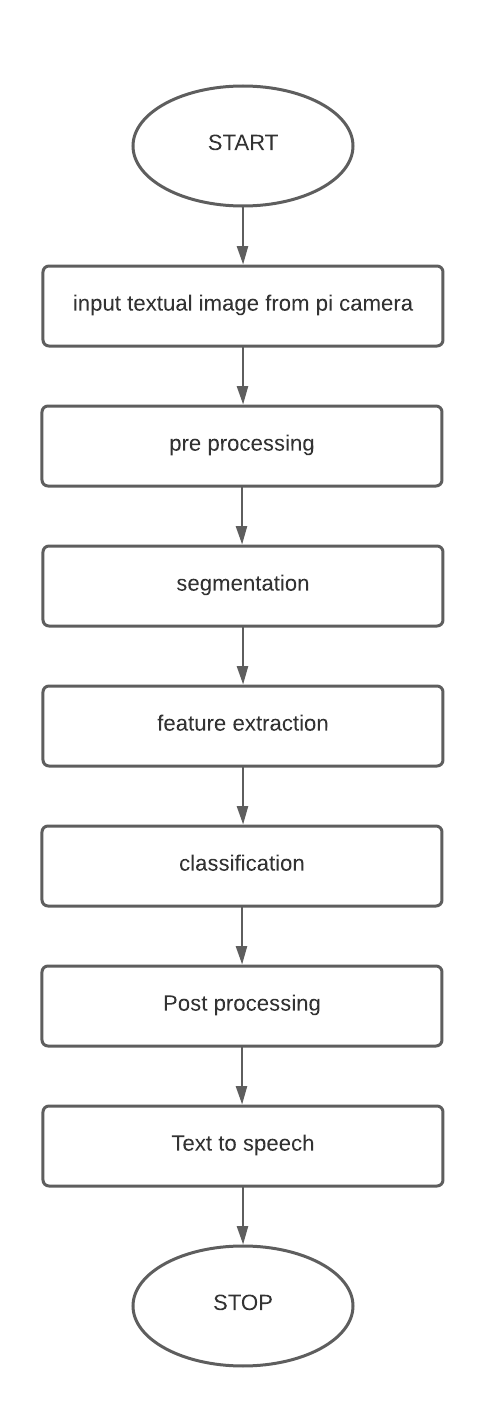
**The four components of a data flow diagram (DFD) are:**

*\_* External Entities/Terminators are outside of the system being modeled. Terminators represent where information comes from and where it goes. In designing a system, we have no idea about what these terminators do or how they do it.

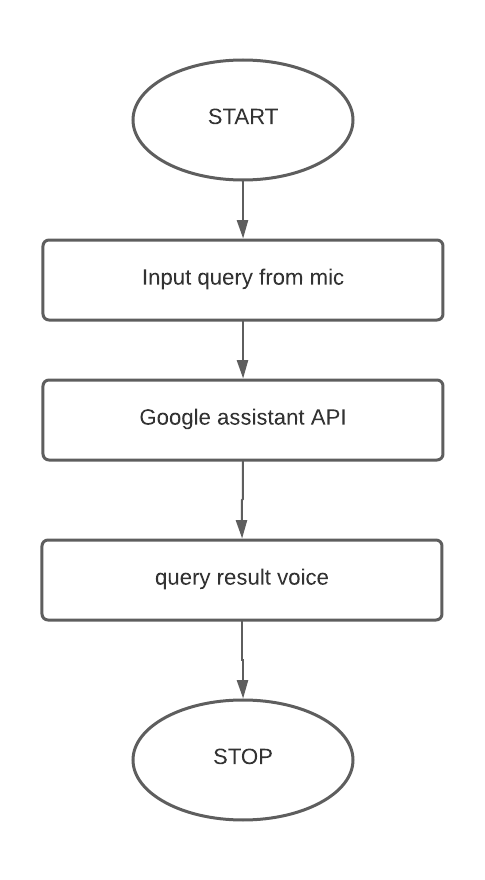
*\_* Processes modify the inputs in the process of generating the outputs

*\_* Data Stores represent a place in the process where data comes to rest. A DFD does not say anything about the relative timing of the processes, so a data store might be a place to accumulate data over a year for the annual accounting process.

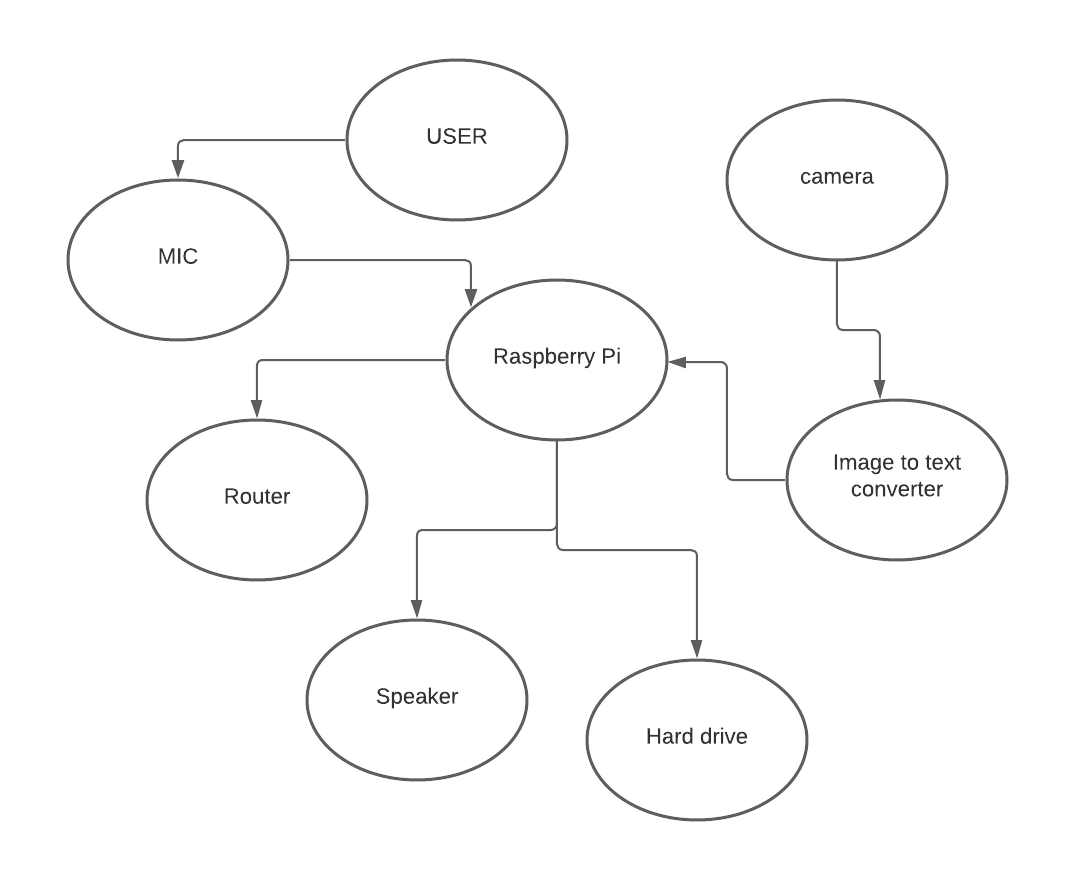
**0-Level DFD**



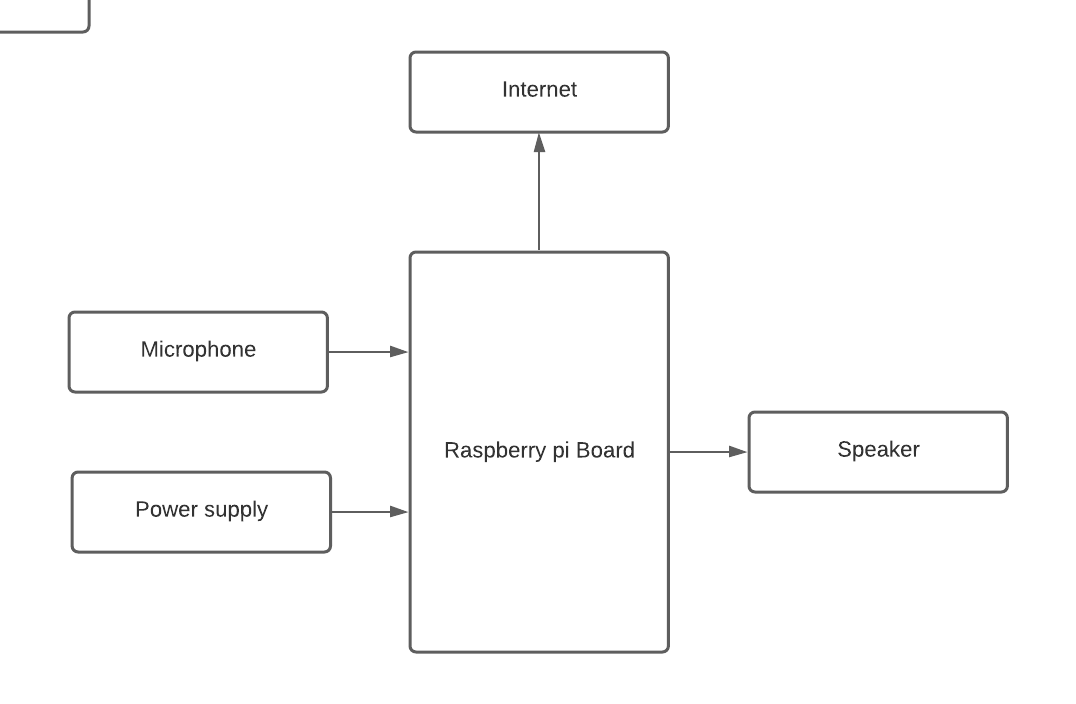
**DFD level 1**



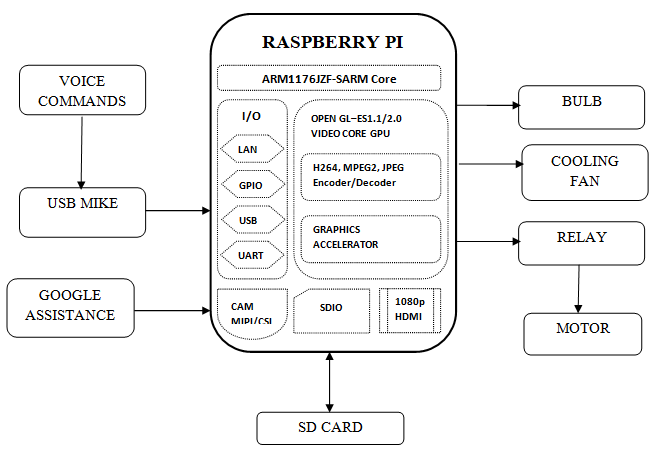
**DFD level 2:**



**Block Diagram :**



System Architecture:



**Chapter 8**

**Advantages**

**Advantages:**

* 1. Capturing photos and recognizing faces in the captured photo.
  2. Checking similarity between two faces.
  3. Performing Arithmetic calculations based on voice commands and giving back the computed solution through a robotic voice.
  4. Searching Google based on user's voice input and giving back the reply through a robotic voice with further interactive questions by the machine.

**Chapter 09**

**FUTURE MODIFICATIONS**

**&**

**CONCLUSION**

**Future Modification**

The Voice Command System has a huge scope within the future. Like Siri, Google currently and Cortana become well-liked in the mobile industry. This makes the transition smooth to an entire voice command system. in addition, this additionally paves manner for a Connected Home using internet of Things, voice command system and computer vision.

**Conclusion**

On the idea of literature survey and by analyzing existing systems, we've came to conclusion that the projected system won't solely be economical however also will boost the application domain of current systems available within the market., introduced the concept and explanation behind the Voice Command System, the issues within the current system and therefore the manner of resolution those flaws and arranged out the system design of the given Voice Command System. Several modules are of open source systems and have customized those modules per the given system. This helps get the most effective performance from the system in terms of space time quality.

**Chapter 10**

**BIBILIOGRAPHY**

**References**

[1] G. Daniels, Human Blood Groups, 2nd ed. Blackwell Science, 2002.Available: download.bioon.com/view/upload/201106/17202758\_9385.pdf

[2] Suthathira Vanitha N., Professor, Department of EEE, Knowledge Institute of Technology, Tamil Nadu, India, A novel approach in identification of blood group using laser technology, International Journal of Research in Engineering and Technology Available: esatjournals.net/ijret/2014v03/i23/IJRET20140323005.pdf

[3] CallumJ.L, Kaplan.H.S, MerkleyL.L, (2001), Reporting of near- miss events for transfusion medicine: improving transfusion safety Transfusion, vol. 41, pp. 1204-1211.Available: www.ncbi.nlm.nih.gov/pubmed/1160681

[4] Jose Fernandes, Sara Pimenta, Student Member, IEEE, Filomena O. Soares, Senior Member, IEEE and Graca Minas, Senior Member, IEEE,(2012), A Complete Blood Typing Device for Automatic Agglutination Detection Based on Absorption Spectrophotometry, IEEE Transactions On Instrumentation And Measurement.

[5]. B. Jyothi, Y. M. Latha, and V. S. K. Reddy, “Medical image retrieval using multiple features,” Research India Publications, vol. 3, pp. 387–396, 2010. Available: searchdl.org/public/book\_series/AETS/7/546.pdf

[6] Z. Q. Song, H. Cui, and Y. A. Hu, “Research and development of support vector machine theory,” Journal of Naval Aeronautical Engineering Institute, vol. 23, no. 2, pp.143-148, 2008.

[ 7] Nazia Fathima S.M (2013) Classification of blood type by microscopic color images, International Journal of Machine Learning and Computing. Vol. 3, No. 4 Available: www.ijmlc.org/papers/342-L472.pdf

[ 8] Priyadharshini, Ramya, Kalaiyarasi, Kalpana Devi., SuthathiraVanitha.

A NOVEL APPROACH IN IDENTIFICATION OF BLOOD GROUP USING LASER TECHNOLOGY.

International Journal of Research in Engineering and Technology Volume: 03 Special Issue: 11 Available: www.ijret.org

[ 9] Mäenpää, Topi The local binary pattern approach to texture analysis-Extension and application

Available:http://urn.fi/urn:isbn:9514270762

**Chapter 11**

**SCREENSHOTS**

**Chapter 12**

**SOURCE CODE**