

15SE496L – MAJOR PROJECT

Virtual Assistant Machine for Visually impaired People

A Project Report

By

Aman Srivastava (RA1511020010188)

Dhananjai Gupta (RA1511020010182)

Under the guidance of

Mrs. S. Aruna (Sr. G)

(Assistant Professor, Department of Software Engineering)

In partial fulfillment for the award of the degree

Of

BACHELOR OF TECHNOLOGY

In

SOFTWARE ENGINEERING



FACULTY OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

Kattankulathur, Kancheepuram

May 2019

BONAFIDE CERTIFICATE

Certified that this project report “Virtual Assistant Machine for Visually impaired People” is the bonafide work of Aman Srivastava (RA1511020010188) and Dhananjai (RA1511020010182) who carried out the project work under my supervision.

Mrs. S. Aruna (Guide)

Assistant Professor (Sr. G)

Department of Software Engineering

SRM Institute of Science and Technology

Dr C. Lakshmi

Head of the Department

Department of Software Engineering

SRM Institute of Science and Technology

INTERNAL EXAMINER

EXTERNAL EXAMINER

ABSTRACT

The inspiration driving this application gives direction based Chatbot to Visually impaired people. It will guidance-based inquiries from people who are prevented by the outside world. There is no uncertainty that they can send applications with the assistance of the Google Voice look and feel. At the point when the application opens, it gives voice direction to utilize the application. Likewise, with the substance structure, income will be given in the phonetic structure. In this manner, an average individual can likewise utilize the application. Bot can be described as software that can chat with people using artificial intelligence. This software are used to perform tasks such as quickly responding to users, informing them, helping to purchase products and providing better service to customers. In this paper, we present the general working principle and the basic concepts of artificial intelligence based chatbots and related concepts as well as their applications in various sectors such as telecommunication, banking, health, customer call centres and e-commerce. Additionally, the results of an example chatbot for donation service developed for telecommunication service provider are presented using the proposed architecture.

ACKNOWLEDGEMENT

We take this opportunity to thank our **Director (E&T), Dr. C. MUTHAMIZHCHELVAN**, SRM Institute of Technology for providing the infrastructure that is required for the development of the project.

We would also like to thank **Dr C. LAKSHMI, Head of Department of Software Engineering**, SRM Institute of Technology for providing us conducive ambience for developing the project.

We would also like to thank our Project Coordinators, **Mrs J. JEYASUDHA** and **Mrs M. UMA**, Department of Software Engineering, SRM Institute of Technology for continuous support.

We would also like to express our deepest gratitude to my guide **Mrs. S. Aruna, Assistant Professor (Sr. G)**, Department of Software Engineering, SRM Institute of Technology, for her assistance, timely suggestion and guidance throughout the duration of this project.

We also convey our gratitude and sincere thanks to all those who have contributed to this project.

Finally, we thank our parents and friends near and dear ones who directly and indirectly contributed to the successful completion of our project.

TABLE OF CONTENTS

CHAPTERS	CONTENTS	PAGE NO.
1.	INTRODUCTION	1
2.	PROJECT OVERVIEW	5
	2.1. LITERATURE SURVEY	5
	2.2. PROBLEM DESCRIPTION	8
	2.3. REQUIREMENTS GATHERING	9
	2.4. REQUIREMENT ANALYSIS	10
	2.4.1. FUNCTIONAL REQUIREMENTS	10
	2.4.2. NON- FUNCTIONAL REQUIREMENTS	10
	2.5. DATA SOURCE	11
	2.6. COST ESTIMATION	12
	2.7. PROJECT SCHEDULE	14
	2.8. RISK ANALYSIS	15
	2.9. SRS	16
3.	ARCHITECTURE & DESIGN	21
	3.1. SYSTEM ARCHITECTURE	21
	3.2. DATA FLOW DESIGN	22
	3.3. CLASS DIAGRAM	23
	3.4. SEQUENCE DIAGRAM	25
	3.5. USE CASE DIAGRAM	26
	3.6. INTERACTION DIAGRAM	27
	3.7. STATE DIAGRAM	28
	3.8. COMPONENT & DEPLOYMENT DIAGRAM	29
4.	IMPLEMENTATION	30
	4.1. DATABASE DESIGN	30
	4.1.1. ER DIAGRAM	30
	4.1.2. RELATIONAL MODEL	30
	4.2. USER INTERFACE	31
	4.3. MIDDLEWARE	32
	4.4. CODE	33
5.	VERIFICATION & VALIDATION	37
	5.1. UNIT TESTING	37
	5.2. INTEGRATION TESTING	39

	5.3. USER TESTING	40
	5.4. SIZE - LOC	42
	5.5. COST ANALYSIS	43
	5.6. DEFECT ANALYSIS	45
	5.7. McCALL'S QUALITY FACTORS	46
6.	EXPERIMENT RESULTS & ANALYSIS	47
	6.1. RESULTS	47
	6.2. RESULT ANALYSIS	47
	6.3. CONCLUSION & FUTURE WORK	49
7.	PLAGIARISM REPORT	50
8.	CONFERENCE CERTIFICATE	52
9.	REFERENCES	53

LIST OF FIGURES AND GRAPHS

S. NO.	HEADING	PAGE NO.
1.	Dataset	7
2.	System Architecture	17
3.	UI Prototype	18
4.	Data Flow Diagram	19
5.	Class Diagram	20
6.	Sequence Diagram	21
7.	Use case Diagram	22
8.	Interaction Diagram	23
9.	State Diagram	24
10.	Component Diagram	25
11.	Deployment Diagram	25
12.	ER Diagram	26
13.	Relational Model	26
14.	User Interface	27
15.	Middleware	28
16.	Unit Test cases	33
17.	Unit Test result	34
18.	Integration Test result	35
19.	User Test result	36
20.	LOC Metrics	38
21.	Cause-Effect Diagram	41
22.	McCall's Quality Factors	42
23.	Setup of the hardware components.	43

CHAPTER 1: INTRODUCTION

ChatBot combines the creative advancement of artificial intelligence (AI) in blended learning (as a combination of personalized e-learning and traditional learning room technology). After the separation of learning phases, the multiplication of community-oriented laboratories and online testing (e-learning), opportunities for machine learning (m-Learning) have arrived to help with versatile and self-coordinating training. Our exploration is to use a versatile “exquisite customer” (tablet or mobile phone) to develop a learning/communication process in the study hall with the help of Chatbots. The personal portable terminals of these schools are linked to the archives of the college's mentoring cloud, in a dynamic relationship, in which Chatbots are implemented as teacher content - mechanization experts between learners, enabling continuous criticism to improve learning outcomes - Keep a lot of information, multi-faceted, and adapt to the level of each member through customized workouts. Useful use cases Chatbots are everywhere in the tech zone. Comparative Exam - About the advantages of using a chat bot in a coaching framework - Investigate the interpersonal relationship with a conversational expert and the specific issues of using a chat bot in teaching. Amazing promises about chatbot planning and improvement, and their hybrid in AI applications - through XML extensions like XML

AIML (Artificial Intelligence Markup Language). Our answer is to coordinate the joint access phase in the communication between educators and students to help them stay engaged and motivated. Our goal is to naturally adjust the multifaceted nature of personalized practice to each of our own understanding dimensions. This useful use case is assuming a study hall where the instructor introduces his practice; each person associates their tablet/phone with a savvy right hand to provide personalized help.

Engineering arrangement

The arrangements created by the creators include: many Java-based miniaturization economies of scale for sending and promoting these smaller-scale cloud domains; Chatbot vendors - we use the Google Stage. From a mechanical point of view, the correspondence between Chatbots and client applications is to leverage REST (REpresentational State Transfer) through HTTP conventions. The letter is supervised and verified by the Chatbot supplier. The basic principles of miniaturized managed application code libraries and associated subsystems are created in Java SE. For informational inventory, we used an open source PostgreSQL database that provides a half-variety approach to gathering information (based on social and archiving).

Communication and entertainment trends change at a breakneck pace. Unfortunately, we can't say the same for educational methods and practices. Thirty years ago, when students wanted a

break from study, they would listen to music on cassette players. Alternatively, they would use landline telephones and pagers to arrange dates.

Today, they are communicating through messenger apps and video calls. Their favorite music is being streamed from distant servers, directly to their smart device. Unfortunately, in many public schools in the United States and internationally, printed textbooks, and lecturing to large groups of students are the only available teaching methods.

Education is much less pragmatic and tech-obsessed than the corporate market. It is a strict and heavily regulated system. Although many ground breaking technologies were patented by universities, educational institutions are not good at using their own inventions for improving the learning process.

Cooperation Between Schools And Corporations

One of the biggest breakthroughs in the development of artificial intelligence and natural language procession happened when Georgetown University and IBM joined their forces and presented the first demonstration of machine translation.

Today, there are many similar partnerships between corporations and educational institutions that try to make the institutional learning transparent and more efficient. In 2016, Bill Gates has announced that the Bill and Melissa Gates Foundation will invest more than \$240 million dollars in a tech project. It personalizes institutional learning. Facebook has also followed the Bill Gates's example and joined the world-famous Summit Learning project.

How Will Artificial Intelligence Personalize Education?

Students who attend the same class have different skills, interests, and abilities. That is why they need personal tutors, who can provide one-on-one lectures. Unfortunately, even some of the most expensive schools and colleges in the world are not able to provide this type of service. That is why chatbots are the most logical and affordable alternative for personal learning.

The project that involves Bill and Melissa Gates Foundation, Facebook and Summit public school, uses bots for basic lecturing. The idea is that chatbots can serve as the virtual advisers, who will adapt their work and curriculum to the students' abilities. This way, the students can follow their own learning pace.

In the Summit Learning project, chatbots serve as vertical tutors. They engage in a dialogue with each student and determine the areas where they are falling behind. Then, chatbots use this

data to compose an entirely personalized learning program that focuses on troubling subjects. Their job is also to follow the students' advancement from the first to the last lesson, check their assumptions, and guide them through the curriculum.

Will Chatbots Take Teachers' Jobs?

When we talk about educational chatbots, this is probably the biggest concern of teachers and trade union organizations. The answer is no, chatbots won't take anyone's job. The truth is that they will take over the repetitive tasks and make a teacher's work more meaningful.

Today, many teachers are solely focused on memorizing lessons and grading tests. By taking over these tasks, chatbots will allow teachers to concentrate on establishing a stronger relationship with students. They will have the opportunity to provide them with personal guidance and enhance the curriculum with their own research interests. This way, teachers will also be able to provide better-quality mentorship. Consequently, this will be especially helpful for students with learning disabilities.

Ashok Goel, a computer science professor at Georgia Tech, is one of the first teachers to simplify his work in this way, with the help of artificial intelligence. He built a chatbot using the IBM's Watson platform and named it Jill Watson. The bot answers students' questions on an online forum and provides technical information about courses and lectures.

Students believed that Jill is was one of the assistants, and they hadn't noticed any difference before the final exam when the professor told them they were talking to a machine. The reactions were positive. Students praised Jill's abilities, and some of them even wanted to nominate her for the prestigious Teaching Assistant award.

How Else Can Artificial Intelligence Influence Education?

The Summit Learning project and Jill Watson are ideal examples how chatbots can bring constructive change to the learning process and make it more efficient. There are also dozens of simpler bots and Artificial Intelligence apps, used in various schools and colleges.

These programs have one or a few functionalities that tackle specific problems. This article on Chatbots Magazine, written by the creators of Hubert, has pointed out six ways how Artificial Intelligence and chatbots can improve education, and we will list the three most important ones.

1. Essay Scoring

Educators often use the multiple-choice tests to make their job easier. Essays offer much better insight into a student's level of knowledge, methodology, and problem-solving skill, but they

are much harder to grade and assess. In the future, educational chatbots will have essay scoring functionality.

In the same way, as word processing tools tell us that our texts are too wordy, complex machine-learning algorithms will be able to assess and grade students' writing on a particular subject. Although this technology is currently in the prototype phase, the Hewitt's Foundation has organized a competition between the most famous essay scorers. According to the report written by Huyen Nguyen and Lucio Dery, from the Department of Computer Science at Stanford University, the winning app had 81% correlation with the human grader.

2. Spaced Interval Learning

This learning concept involves repeating the old lessons, just before you forget them. The spaced interval learning was used as a basis for developing an app that helps people to track the learning process and reminds them to repeat the lessons they are about to forget. The app was created by the Polish inventor Piotr Wozniak and promoted by the SuperMemo company.

3. Student Feedback And Professor Assessment

In modern educational institutions, student feedback is the most important factor for assessing a teacher's work. Most schools and universities have upgraded their feedback collection process by shifting from print to online forms. They can make it even more efficient by using chatbots for this task. While chatting with bots, students will have the chance to explain their claims. On the other hand, the bot can be trained to ask additional questions based on their previous answers.

How Can Schools And Teachers Create Their Chatbots?

Modern chatbots are trained to conduct very complex tasks, yet they can be easily built without coding. Most bots provide specific answers depending on the words and phrases people use, so the building process usually involves asking questions and generating possible outcomes.

CHAPTER 2: PROJECT OVERVIEW

2.1 LITERATURE SURVEY

1. Title: Virtual Assistant for Hotel Reservation System.

Author: Mohamed Rawidean Mohd Kassim, Ibrahim Mat, Ahmad Nizar Harun.

Year: 2014.

Theoretical Model: In this method, the user has sent specific text to Chatbot, which will prompt the user to enter the details of the subscription. Once detailed, we will make a reservation. a virtual assistant (VA) is a great way to lighten your work load without having to spend too much on salaries. Basically, a virtual assistant is someone who works for you, but is not physically present in your office. Sites such as OnlineJobs provides a deep pool of talented VA's and because of the difference in currency, you get more for your dollar (you'll probably save 75% or more in salary if you hire a VA from the Philippines versus hiring a VA from the United States).

2. **Title: Web Based Chatbot .**

Author: Imran Ahmed and Shikha Singh

Year: 2014.

Theoretical Model: Imran Ahmed and Shikha Singh designed a Web-based Chatbot that is implemented in Python using the AIML language. It includes NLP and speech recognition technology. The output will be received in speech using TTS (Textspeak). This paper focuses on automating the process of communication by use of chat-bot and it also focuses on providing customized results to the user which makes the process of hotel booking convenient and user friendly for him. An extensive research done on existing systems gave us an insight into their shortcomings which this system attempts to overcome by creating a chat-bot using Artificial Intelligence Mark-up Language and using various algorithms such as Keyword Matching, String Similarity , Spell Checker and Natural language parser. The implementation of this system has resulted in better resource utilization and increased responsiveness of user behaviour. This system has been implemented to integrate with any Hotel Management Android Application to ease the process of hotel booking.

.

3. **Title: Braille Education System.**

Author: Louis Braille.

Year: 2016

Theoretical Model: The current blind education system is used by the Braille education system invented by Louis Blair. In this method, each alphabet is represented by a raised point in the embossed paper. The blind can read the letters by touching the marked points. Braille is vital to all visually impaired individuals and it's the only system through which visually impaired children can learn to read and write, yet the rate of Braille literacy among visually impaired people belonging to developing countries including Pakistan is alarming low. Today in developing countries less than 3% of visually impaired children are learning to read Braille in school. This continues despite the fact that studies have shown that 80% of all employed visually impaired people read and write Braille fluently. Thus, Braille literacy is the key to employment and full participation in society. This research paper presents the design of a low-cost, low-power, portable, self-learning, and user friendly Braille system.

4. Title: Instant Messaging Chatbot

Author: Slam

Year: 2017

Theoretical Model: After that National Federation of the Blind Youth Slam proposed the Instant Messaging Chatbot for visually impaired people. It provides information about weather, news and dictionary. User have to type the query via keyboard for the input. The output content will be given as voice for Blind people. Instant messaging is an incredible tool, so close to users nowadays, but has one flaw: it is not scalable. Once you reach certain volumes, one-to-one communication becomes too expensive. And it is precisely here that the chatbots take over: they allow automation and scalability.

Chatbots can be an important ally for lead generation, for converting, or simply for stimulating user engagement or brand awareness.

So when should you develop a chatbot, and what kind is most suitable for your needs? To paraphrase a famous slogan of Apple, soon there will be a bot for everything. Developing a chatbot to communicate with customers has many advantages, but it is not always necessary and above all, it must be based of the goal we need to achieve.

5. Title: Android application work as assistant for the student

Author: Mikic Fonte, Martín Llamas Nistal, Juan C. Burguillo Rial and Manuel Caeiro Rodríguez

Year: 2015

Theoretical Model: Mikic Fonte, Martín Llamas Nistal, Juan C. Burguillo Rial and Manuel Caeiro Rodríguez in which android application work as assistant for the student in their learning process. Students can search the topics related to their subjects and the system will provide a result about the topic. Student can search for particular topics in subjects, exam questions and answers.

6. Title: Natural Language Processing

Author: Unnati Dhavare and Umesh Kulkarni

Year: 2014

Theoretical Model: The concept of Natural Language Processing using AI (Artificial Intelligence) has proposed by Unnati Dhavare and Umesh Kulkarni. In this concept input and output of then system can be in speech/text/images. The input is handle by NLP (Natural language Processing) techniques. Abstract Artificial Intelligence (AI) is the study of how to make computers (machines) do things which, at the moment, people do better. There are many applications of the artificial intelligence. NATURAL LANGUAGE PROCESSING (NLP) is one of the upcoming applications of AI. The goal of the Natural Language Processing is to design and build software that will analyze, understand, and generate languages that humans use naturally, so that eventually you will be able to address your computer as though you were addressing another person. Natural Language Processing is the use of computers to process written and spoken language for some practical, useful, purpose: to translate languages, to get information from the web on text data banks so as to answer questions, to carry on conversations with machines, and to get advice about, say, pensions and so on. Practical applications of natural language processing are machine translation, extracting data from text etc.

2.2 PROBLEM DESCRIPTION

In the course of the most recent decade, the innovation headways were immediately consumed and incorporated into instruction procedures and eLearning which empowered the improvement of proficient and creative arrangements ready to go about as client custom fitted learning assets. A standout amongst the most noteworthy advancements in the eLearning field is the combination of Chatbots innovation into showing background, which goes about as an astute scaffold among training and innovation.

Chatbots innovation can be utilized in a wide range of fields because of its adaptability and capacity to adjust to the pace at which every individual feels good, without causing overpowering or weariness. This bot plans to mix e-Assisted learning and correspondence into study halls, presenting Chatbots as ongoing instructive partners for the instructor - they are intended to disseminate intelligence with understudies and to oversee it at individualized dimension.

This methodology includes a very expanded level of intelligence and personalization to the learning knowledge, angles which are basic from the understudy perspective as every individual has its very own learning and getting capacities. The creators consolidated present day programming advances, for example, Cloud based improvement, smaller scale administrations, Java advances and Chatbots to make a mechanized shut circle for the showing pace and versatile unpredictability in a study hall. The proposed arrangement abstains from putting the instructor in an optional position, shadowed by innovation - it rather engages him with e-Assistants that give him access to a continuous and pertinent measurement of individual understudies' getting level - an important input for customized training which prompts an improved mentorship quality.

2.3 REQUIREMENTS GATHERING

The project focuses on the characteristics of precision answering of questioning asked by the user. Since most of them are only provided theoretically in the form of research papers, we intend to use them for practical applications. We use three methods for demand collection analysis:

[1] Brainstorming: This is a way to gather ideas about a specific issue or problem from a team member. It helps identify all possibilities, simplifying the solution and prioritizing the best ideas originally requested. Therefore, after completing the documentation and observations, the team members discussed all the advantages and disadvantages of the existing system, as well as what should be strengthened and improved, in order to provide a better automation system. It also determines which methods can be implemented.

[2] Observation: Determine the next improvement by studying the user, the system process and its shortcomings. It can be active or passive. When we want to understand each step of a process, we take the initiative to observe, and passively observe where we record and / or questionnaires, and wait for the work to be completed in order to complete the work without interruption. In this project, we choose passive observation, because the best interests of farmers are to let them complete the work without any interference, and then clear the doubts in order to make a better analysis of the improvements needed.

[3] Literature analysis: This is one of the most important technologies. It relates to the written documentation of the system, the work and procedural methods it uses. In addition, these can help determine the requirements for the current project start. It also helps to validate and validate the method and determine which key points have been missed in the previous system, and should consider using the upcoming technology to create a new system.

2.4 REQUIREMENT ANALYSIS

Functional Requirements

1.API Request

Client Responsibilities

1.1 The client will send a GET request to the web API and use the question as a URL parameter.

Priority 1

1.2 The client will specify the header Content-Type: application/json in their requests as convention. Priority 1

1.3 A valid API query is a single URL parameter containing one sentence that is a question in standard English. Priority 1

1.4 The client will be able to parse the JSON and determine if there was an error. Priority 1.

2.Server Responsibilities

2.1 The server will send all API data in the JSON response document using the header Content-Type: application / json. Priority 1

2.2 The server will respond with a 200 OK status code if a request has the header Content-Type: application/json and is a valid API query. Priority 1

2.3 The server will respond with a 400 Bad Request status code if a request does not specify the header Content-Type: application/json OR is a malformed API query. Priority 1

Non-Functional Requirements

The network can adapt its pattern according to the data provided.

2.5 DATA SOURCE

The essential information wellspring of the venture is from the SRM Website. There are a number of information base accessible for structure chatbots however, relies on the need and reason you are going to utilize it for. So as to assemble a chatbot two things are required a learning base and innovative structure squares.

Learning base: A rich information base empowers a chatbot to accomplish something beyond essentially going about as a door to FAQs. It conveys a consistent client experience by giving moment goals to make associations more easy to use.

Man-made consciousness (AI): AI controlled Chatbot helps in making client administration progressively proficient, beneficial and savvy.

Regular Language Processing (NLP): Chatbots fueled with NLP perceives client purpose by parsing their messages and comprehend sentences to take activities.

SRM Website offers a rich learning base improved with intuitive visual aides, choice trees, and then some; that helps in structuring a keen visit bot

2.6 COST ESTIMATION

COCOMO is the most widely used programming estimation model on Earth. The COCOMO model predicts the effort and length of companies that rely on information source identification based on the scope of the subsequent framework and the various "cost drivers" that affect efficiency.

The key parameters of the nature of any product that characterizes Cocomo results are basically efforts and arrangements:

Force: The amount of work required to complete the errand. Estimated to be face-to-face monthly units.

Timeline: A brief description of the amount of time required to complete an activity, which obviously corresponds to the effort imposed. It is estimated in terms of time, for example, weeks, months.

The meaning of Boehm's natural, semi-automatic and installation framework:

- **Organic** - If the required group measures are cracked, then the product company is considered a natural product, the problem is certainly known and previously understood, and colleagues have a clear encounter on this issue.

- **Semi-Isolation** - If critical qualities (eg, group estimates, understanding, information about different programming conditions) are located between nature and embedded, then the product company is considered semi-closed. A semi-independent company is relatively unnatural, difficult to contrast with natural companies, and requires more participation and better direction and creativity.

Embedded - A product enterprise that requires the most versatile nature of the anomalous number of imaginative and empirical prerequisites fall into this category. Such programming requires a larger set of measurements than the other two models, and designers should have enough experience and innovation to develop this complex model.

All of the above system types use different estimates of the constants used in Effort Calculations.

Estimation of Effort: Calculations –

Basic Model –

$$\text{Effort} = (a \times (\text{KLOC})^b) * \text{EAF}$$

SOFTWARE PROJECTS	A	B
Organic	2.4	1.05
Semi Detached	3.0	1.12
Embedded	3.6	1.20

Development cost – Rs. 1000 (approx)

npm api integration – Rs.35 per 1000 loads.

PC miler api integration – Rs. 6000

Overall cost estimate – Rs. 7000 (approx)

2.7 PROJECT SCHEDULE

TIME PERIOD		TASK	OUTCOME
FROM DATE	TO DATE		
12-07-2019	30-07-2019	Literature Survey	A thorough study of the related research papers was done.
01-01-2019	01-01-2019	Requirements Gathering	An analysis of all the functional and non-functional requirements was done.
02-01-2019	03-01-2019	Cost Estimation and Risk Analysis	The estimated cost and all the possible risks were analyzed.
03-01-2019	06-01-2019	System Architecture, UI Prototype and Data Flow Design	A basic layout of the whole system is developed to understand the data flow and make it user-friendly.
01-02-2019	03-02-2019	Behavioural Design	Use-case, Sequence, Interaction and State diagrams were made.
05-02-2019	07-02-2019	Structural Design	Class, Component and Deployment diagrams were made.
10-02-2019	17-02-2019	Database Design	ER and Relational models were made.
03-03-2019	19-03-2019	Implementation	The code for the system was implemented.
20-03-2019	22-03-2019	Testing	The implemented code is tested for possible errors and bugs.
23-03-2019	24-03-2019	Sizing	An estimate of the overall project size is done using metrics.
25-03-2019	26-03-2019	Cost and Defect Analysis	The overall cost after the building of the project and all the defects were analyzed.
29-03-2019	29-03-2019	Result and Analysis	A quality analysis of the results obtained and a check for the possibility of any future work is done.

TABLE 1. Timeline for the project.

2.8 RISK ANALYSIS

The risk analysis for the project is determined using SWOT(Strength, Weakness, Opportunity and Threats) analysis, which is a qualitative risk analysis because there are no such consequences which are required to be expressed in terms of magnitude or numerical units. It gives a broad, relative classification for the risk.

- **Strengths:** Advantageous characteristics of the project.
- **Weaknesses:** Characteristics which are a disadvantage to the project.
- **Opportunities:** Elements which can be for the betterment of the project.
- **Threats:** Elements which can cause trouble for the project.

It can be done using a risk matrix which can be done by cross-referencing the internal and external factors with the helpful and harmful objectives of the project.

- **Helpful Objectives:** The elements which are valuable for the project and hence are used in the project, or can be used in future.
- **Harmful Objectives:** The elements which are not good for the project and hence are not used in the project, or is mostly avoided being used in any instance.
- **Internal Factors:** The elements used in the project.
- **External Factors:** The elements not used in the project but are stored anyways.

The risk analysis for the project is given in the below:-

	HELPFUL	HARMFUL
INTERNAL	Strength Low overall cost	Weakness No complete analysis
EXTERNAL	Opportunity Pattern-recognised prediction	Threat No security

TABLE 2. Risk Analysis for the project.

- [1] **Strengths:** — Reducing the overall cost of analysis tool
 - Giving required insights from real-time data to the bot.
- [2] **Weaknesses:** — Not full analysis of the data generated
 - Cannot computation heavy algorithms.
- [3] **Opportunities:** — The important and required patterns are analysed by the real-time data which will help in prediction.
- [4] **Threats:** — The security of the system is not considered.

2.9 SRS

2.9.1 Introduction

2.9.1.1 Document purpose

The purpose of this document is to explain the various requirements and the objective of the proposed system.

2.9.1.2 Document conventions

This SRS's format is simple. Bold letters are used for headings and the normal one for specifications. The entire document is written using the standard font, Times New Roman. Main Headings of any page are denoted by Times-16 and Times-14, sub-headings by Times-12.

2.9.1.3 Intended audience and reading suggestions

This particular document is expected to be read by project managers, developers, documentation writers, users, and testers. Being a technical document, terms are intended to be understood by the customer clearly. This SRS should be read starting with Introduction. The main purpose of the document is to identify if the developer is using the correct methods, requirements and other fulfilled methods. This document is used by both the developer and tester. The tester will get to know about functional and non-functional requirements.

2.9.1.4 Product scope

Is an AI chatbot that receives questions from users, tries to understand the question, and provides

appropriate answers. It does this by converting an English sentence into a machine-friendly query, then going through relevant data to find the necessary information, and finally returning the answer in a natural language sentence. In other words, it answers your questions like a human does, instead of giving you the list of websites that may contain the answer.

2.9.2 Overall Description

2.9.2.1 Product perspective

Today, a large part of the web index is similar to Google, using a framework (The PageRank Calculate) Ranking special site pages. When the customer enters a question, the query is Interpreted as a slogan, the framework restored the most notable website page's rundown May respond to the survey. At this point, the client must experience the destruction of the page. Find the appropriate response they are looking for. Anyway, Chatbot will try to get it Ask questions and give authoritative answers.

The framework will have four major units working together to understand

Question and return a suitable answer:

- Universal query development - equipped with featured languages. The problem and gradually make it traditional.
- Universal answer development - equipped with non-exclusive queries. Format and give a traditional answer layout.
- Common answer crowd - with traditional answer layout. More importantly, populate it with data from the database to build the answer.
- Information extraction - equipped to discover data in an organized way Or on the other hand, an unstructured site and put the data in the database

2.9.2.2 Product functions

The two sorts of clients of the framework are as per the following:

- API clients

Programming interface clients incorporate application designers who need to consolidate the Chatbot API into other programming applications.

- Mobile application/web application

These clients are comprised of non-specialized clients who need to find solutions to their inquiries. These clients pose inquiries and find solutions by means of a versatile, web or instant

message interface. Such clients incorporate present and future understudies, showing staff and personnel.

2.9.2.3 User classes and characteristics

Real-time data, Administrator, User, and using System.

2.9.2.4 Operating environment

In this product, we have two to set up web servers since it is web-based. This product is view on the web server which has been set up on UBUNTU or Windows operating system.

2.9.2.5 Design and implementation constraints

No constraints at this point in time.

2.9.2.6 User documentation

- Software Requirement Specification.
- Required software.
- User manual.

2.9.2.7 Assumptions and dependencies

The software depends on one thing that is mics. In the future, we introduce new technologies in the software such as typing.

2.9.3 External Interface Requirements

2.9.3.1 User interface

The software will provide a way to the user to ask Questions which can be answered in different ways text and speech. the user can manually ask questions and learn new keywords during the session.

2.9.3.2 Hardware interface

The hardware interface in the project is mic, and keyword.

2.9.3.3 Software interface

The system makes direct use of the database, it also uses Virtual studio and IDLE.

2.9.3.4 Communication interfaces

The system uses a report to communicate.

2.9.4 System Features

2.9.4.1 Description and Priority:

The software has a simple login type method. The user can interact with software with the help of keyword and mic . The person can interact with the help of mics and keyword.

2.9.4.2 Functional Requirements:

Functional means providing a particular service to the user and also provide specific behaviour of the system. The functional output of the given service can be split into three main segments. The first and the primary segment is the Create Database segment whereby the database is created using the API call to YouTube, which in turn keeps updating itself from time to time in order give results real time. There is a search term field which is required to take the user input. There is a visualization functionality which is used to obtain the graphs and the charts. There is a final results tab which helps us showcase the results.

2.9.5 Other Nonfunctional Requirements

2.9.5.1 Performance requirements

Our project is considered to be based on client-server architecture so that if the client sends a request to the server, the server should respond. Also, as the number of clients are going to be larger, the server gets overloaded. In order to serve all the requests if the clients there should be proper hardware and software networking capability. The architecture of the network should be such that the request/response time is measured. And the time between request and the response should be as minimum as possible.

2.9.5.2 Software quality attributes: -

1. Adaptability:

The software must be able to incur changes within itself as well as changes in the machine architecture.

2. Correctness:

The system should produce correct and exact output so as to evaluate the results.

2.9.5.3 Safety Requirements

Other safety requirements should be that of power supply which should be uninterrupted. The networking devices should be properly connected. And faulty networking devices as the router, switch and the hub should be removed as early as possible.

2.9.5.4 Security Requirements

IDs and Passwords should be regulated to only certain lengths and the password must be strong, containing non-alphanumeric characters as well. The database of the user should be kept safe so that no one misses it.

4 ARCHITECTURE & DESIGN

3.1 SYSTEM ARCHITECTURE

System architecture tells about how the application is connected to the user interface. Our application contains an audio segment which is connected to different databases, which are templates, grammar keywords, and pattern. The engineering of the framework has appeared in Figure 3.1 . To start with, the client accepts the contribution as a voice, which is perceived by the Speech Recognizer class in android. Next, convert the discourse to content to get the yield from the asset. Clients can change assets at runtime

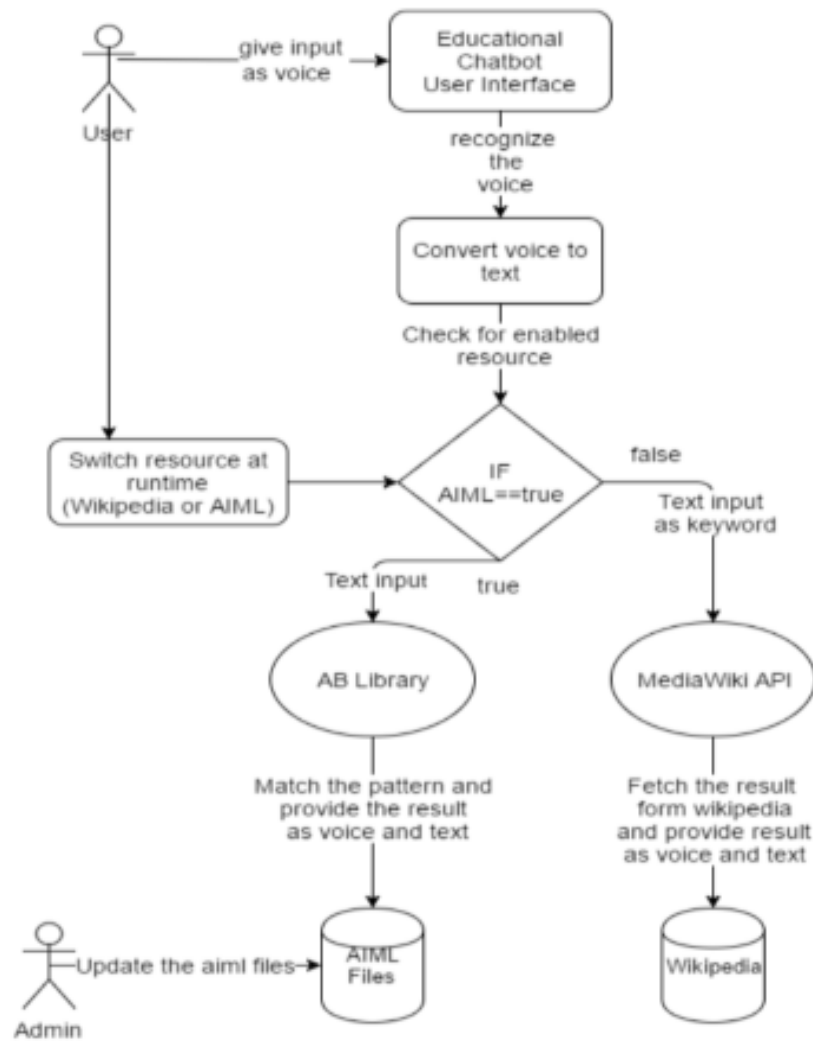


Fig 3.1 System Architecture of BOT

3.2 DATA FLOW DESIGN

A data-flow diagram (DFD) is a way of representing a flow of data in the system. The DFD also provides information about the outputs and inputs of each entity. A data-flow diagram has no decision rules and no loops. DFD contain different entity such as videos, image frame and speed recognition, which perform different functions. Data Flow Design (DFD) maps out the graphical representation of the flow of data in a system. DFDs are used to visualize the steps involved in a system to process the inputs and produce the required outputs for a certain functionality of the system.

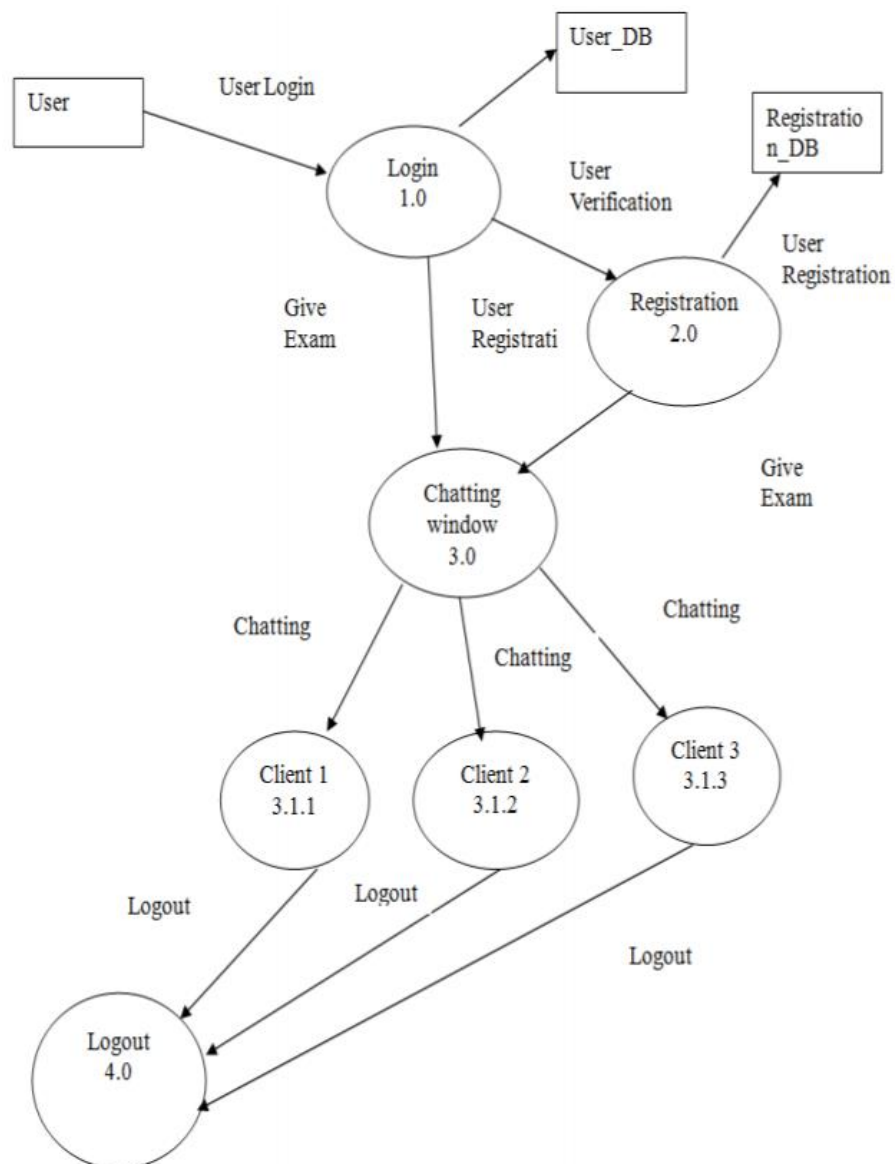


Fig 3.2 Data flow Diagram for BOT

3.3 CLASS DIAGRAM

Class Diagram shows the static structure of the system by describing the classes and their attributes, operations and objects. Various components of a class diagram can be used to create a class in an actual program of the system too.

- **For Server:**

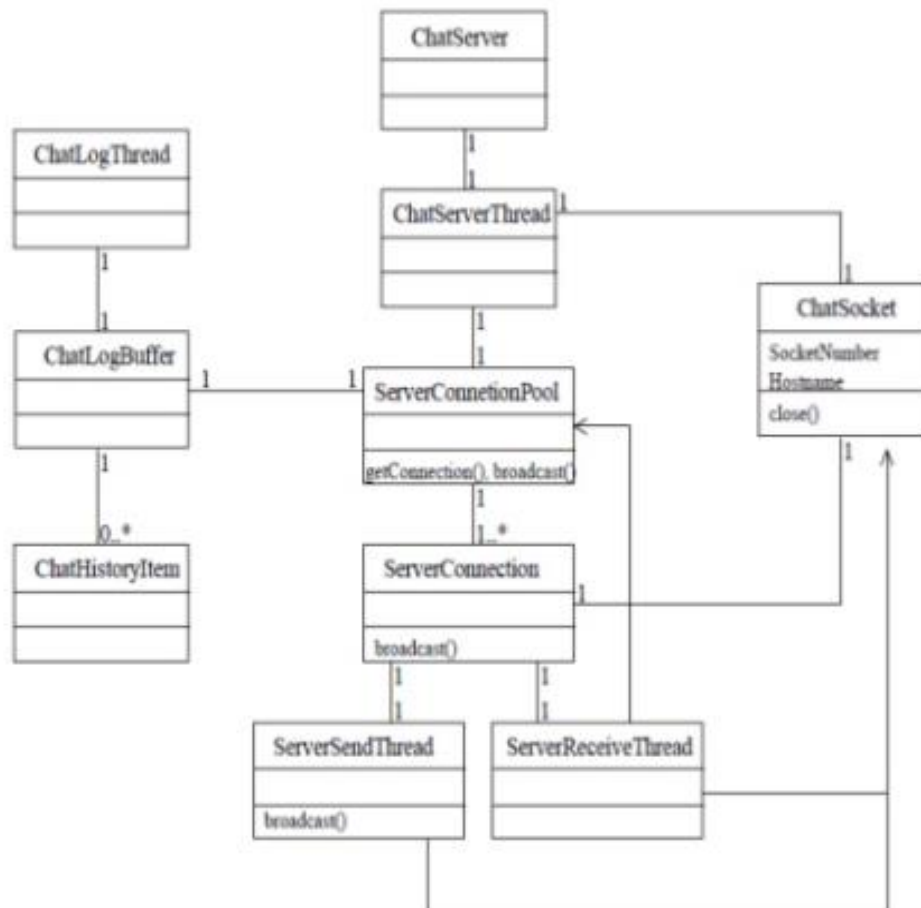


Fig 3.3.1 Class Diagram for BOT.

- For Client

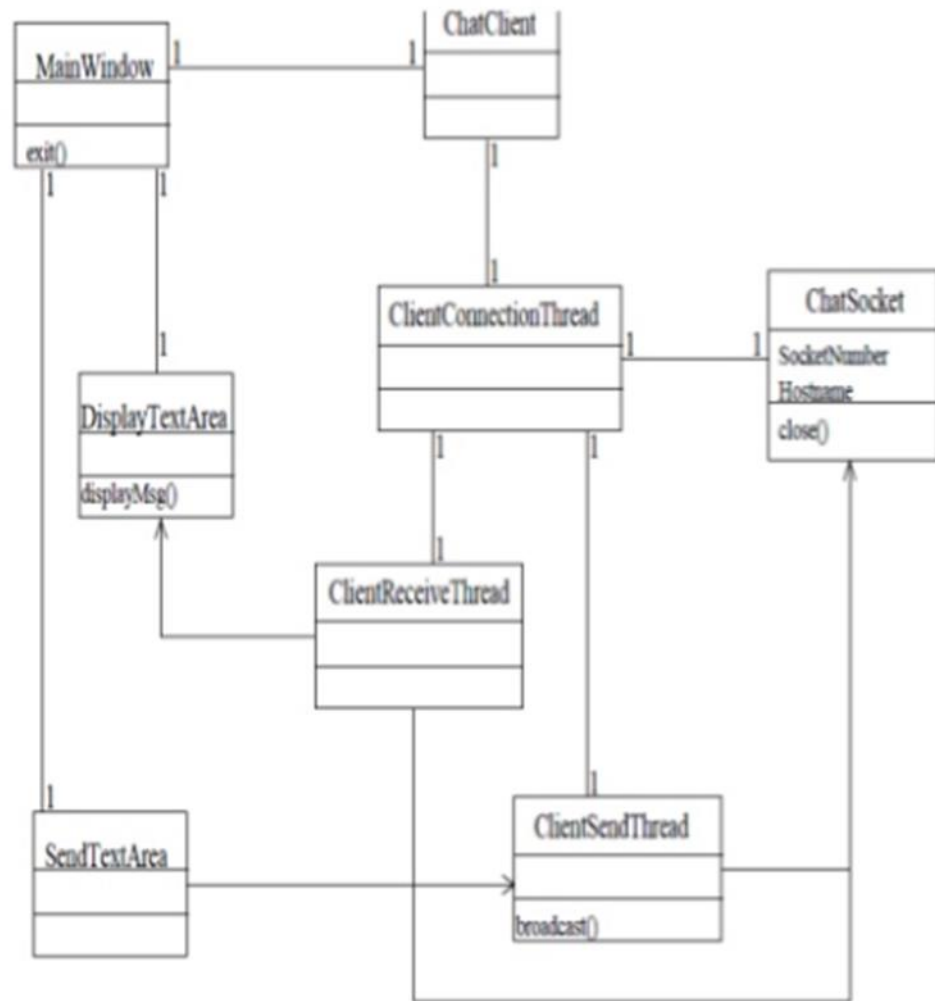


Fig 3.3.2 Class Diagram for BOT

3.4 SEQUENCE DIAGRAM

Sequence Diagram shows the interaction between the objects arranged in a time sequence of event calling. It depicts the sequence of messages exchanged between the objects, in form of horizontal arrows, to process a functionality of the system in a chronological order, which is shown using vertical parallel lines(lifelines).

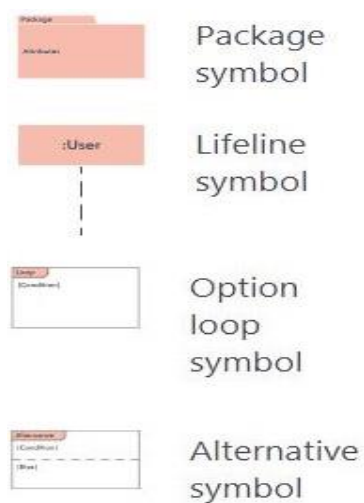
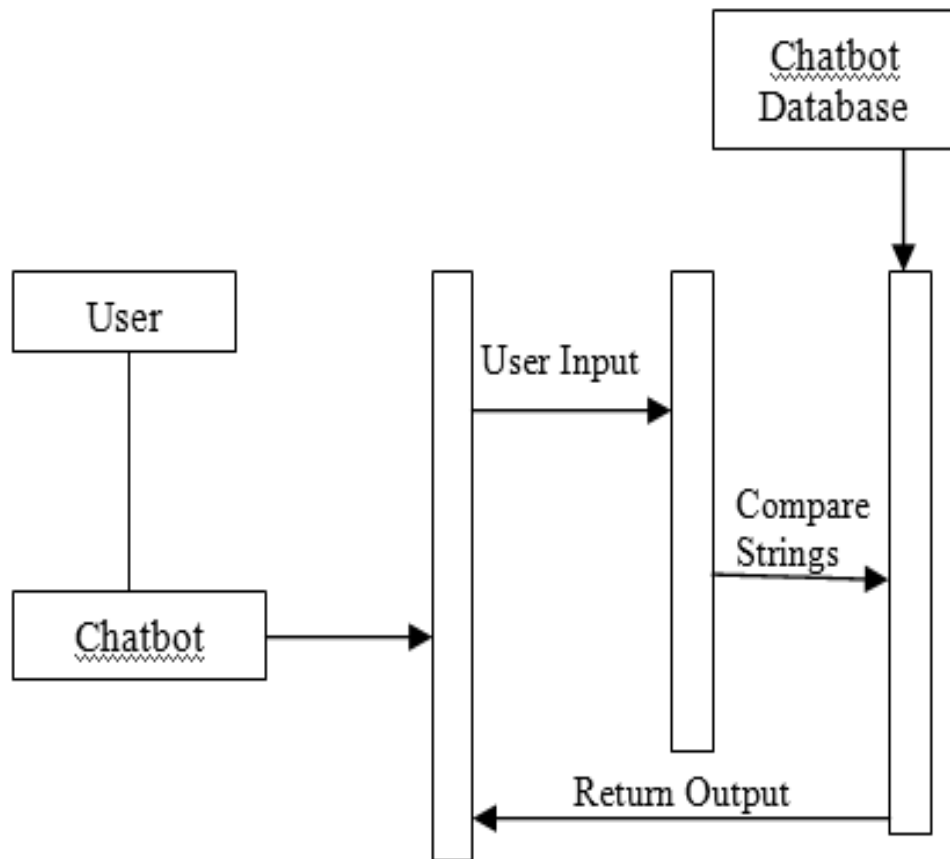


Fig 3.4 Sequence Diagram for BOT

3.5 USE CASE DIAGRAM

A Use-case Diagram summarizes the list of steps that define an interaction between the users(actors) and system

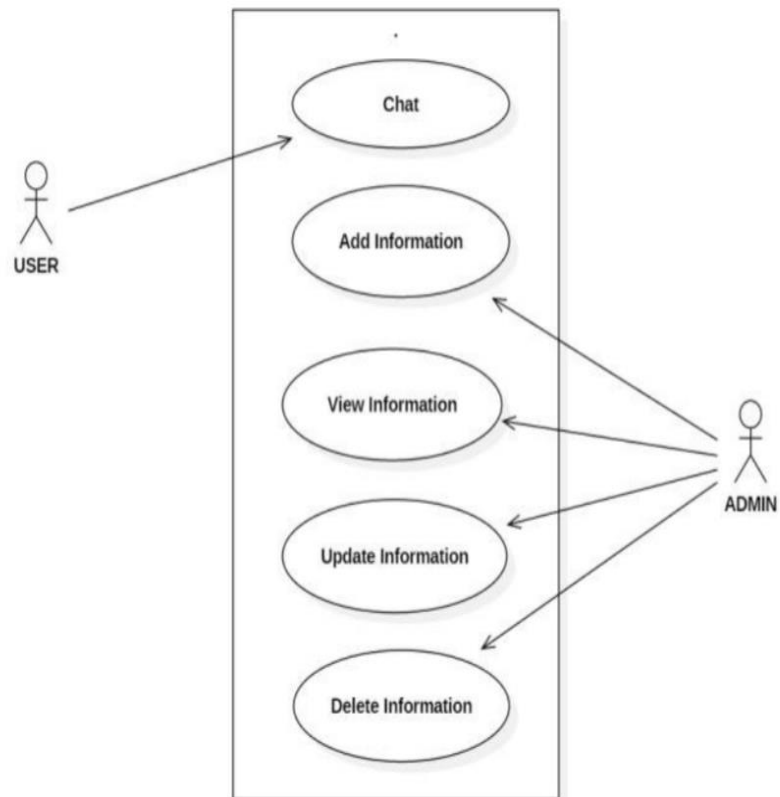


Fig 3.5 USE CASE Diagram for Bot.

3.6 INTERACTION DIAGRAM

Interaction Diagram shows the collaboration of a group of objects in a particular sequence. The sequence of activities is numbered. **System integration** is defined in engineering as the process of bringing together the component sub-systems into one system (an aggregation of subsystems cooperating so that the system is able to deliver the overarching functionality) and ensuring that the subsystems function together as a system, and in information technology as the process of linking together different computing systems and software applications physically or functionally, to act as a coordinated whole.

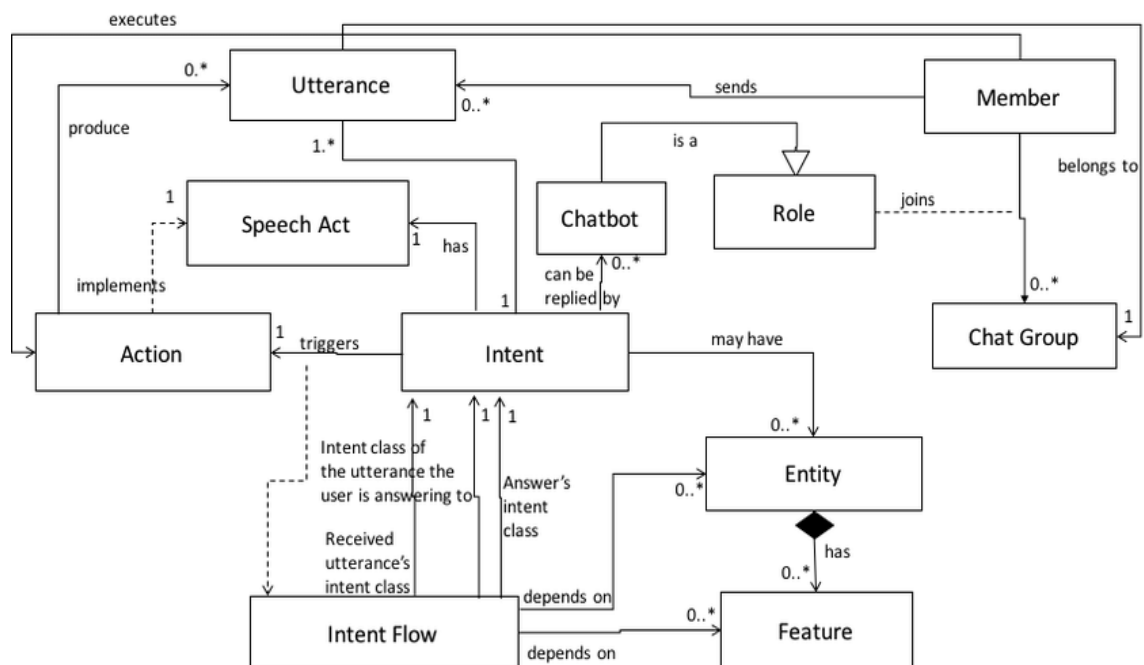


Fig 3.6 Interaction Diagram

3.7 STATE DIAGRAM

State Diagram describes the state of an object in a system at a given time. It shows the behaviour of objects in response to external stimuli. The elements that constitute a state diagram are rounded boxes representing the states and arrows showing transitions to the next state. The activity section depicts the activities the object performs while it is in that state. Every state diagram starts with an initial state, which is the state where the object is created. Right after the initial state, objects change their states, and the next state is determined by conditions based on activities.

A transition in a state diagram is a progression from one state to another and is triggered by an event that is internal or external to the entity modelled. An action is an operation that is invoked by an entity that is modelled. A very traditional form of state diagram for a finite machine is a directed graph.

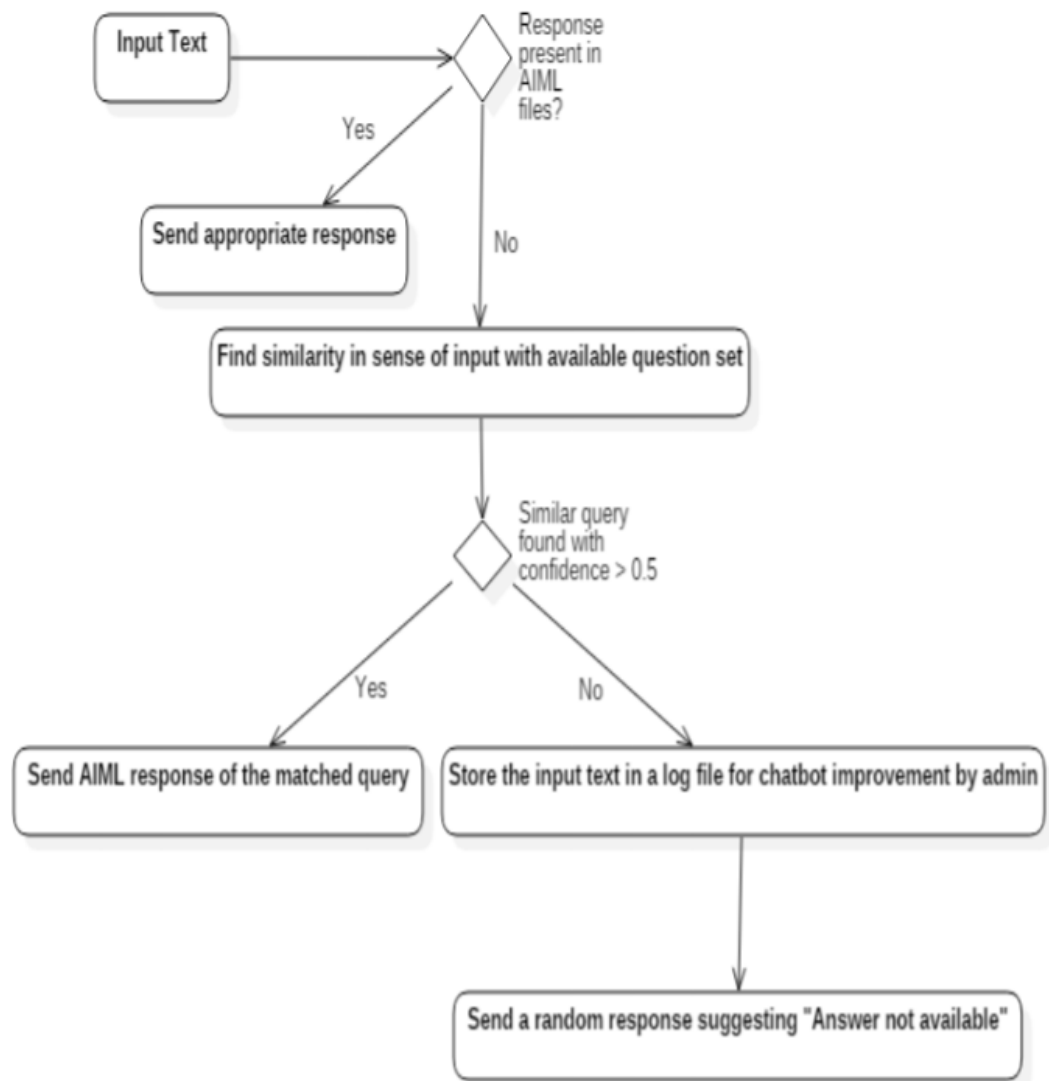


Fig: 3.7.1 State Diagram for BOT.

3.8 COMPONENT & DEPLOYMENT DIAGRAM

Component Diagram shows how the different subsystems are wired together to form a larger subsystem or a complete system. It describes the dependencies of each module in a system.

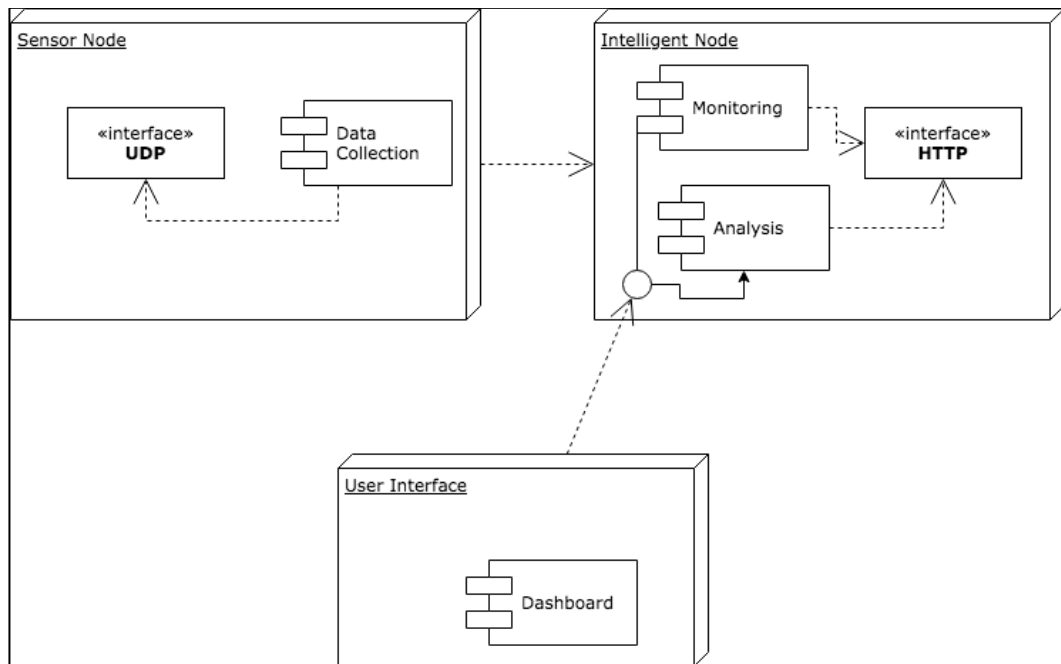


Fig 3.8 Component Diagram of BOT.

Deployment Diagram models the physical deployment of information processed by the software artefact's on web servers, database servers using connectors like JDBC, REST.

4 IMPLEMENTATION

4.1 DATABASE DESIGN

4.1.1 ER Diagram

An Entity-Relationship (ER) Diagram is a conceptual database model in the form of a flowchart which shows how different entities of a system are related to one another, using specific relationships which can exist between instances.

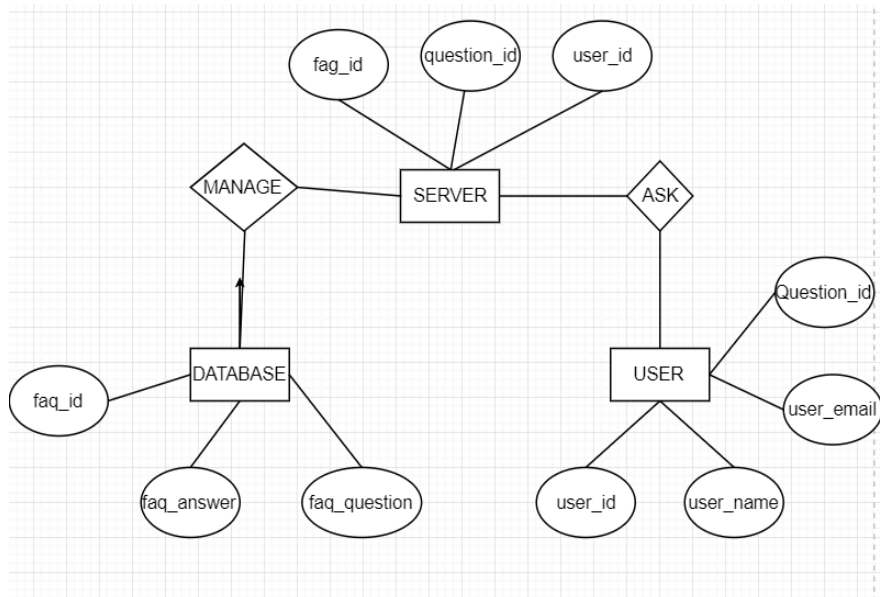


Fig 4.1.1 ER Diagram for BOT.

4.1.2 Relational Model

Relational Model is the essential information model utilized for information stockpiling and preparing as tables utilizing organized language.

Relational Model

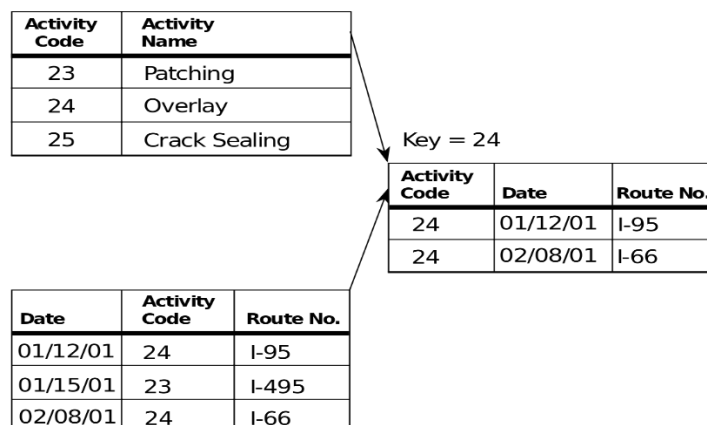


Fig 4.1.2 Relational Model Diagram for BOT.

4.2 USER INTERFACE

User Interface is the visual layout of the system made for human-computer interaction. It allows the user to control and interact with the system, while the system provides back simultaneous outputs to the user's commands. the user interface (UI) is everything designed into an information device with which a person may interact. This can include display screens, keyboards, a mouse and the appearance of a desktop. It is also the way through which a user interacts with an application or a website. The growing dependence of many companies on web applications and mobile applications has led many companies to place increased priority on UI in an effort to improve the user's overall experience.

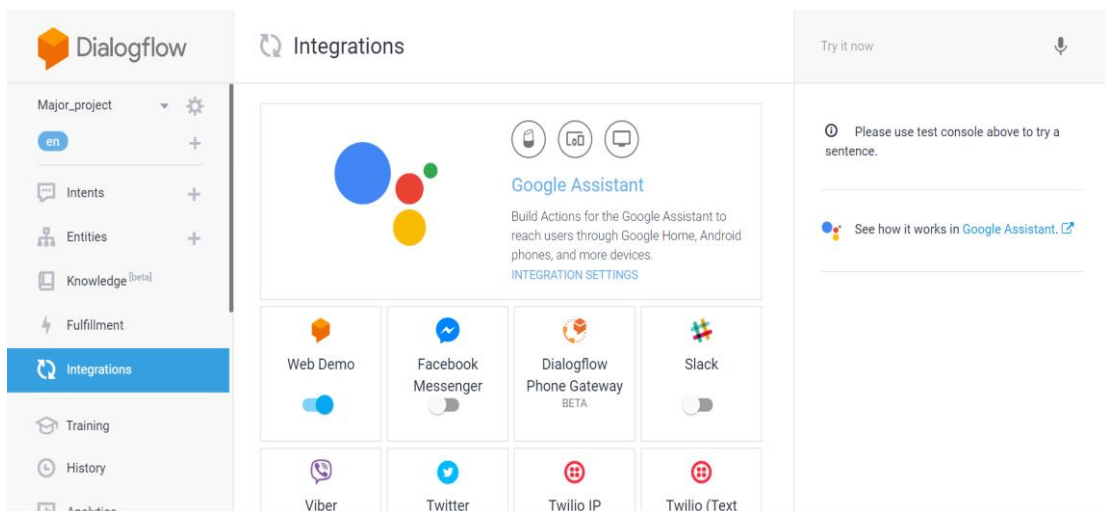


Fig 4.2.1 UI for BOT

4.3 MIDDLEWARE

Firebase is used as a middleware to store the data for the analysis purposes. The data is stored for every information and collective analysis. Firebase is used as it is the best real-time database which serves the purpose of the project.

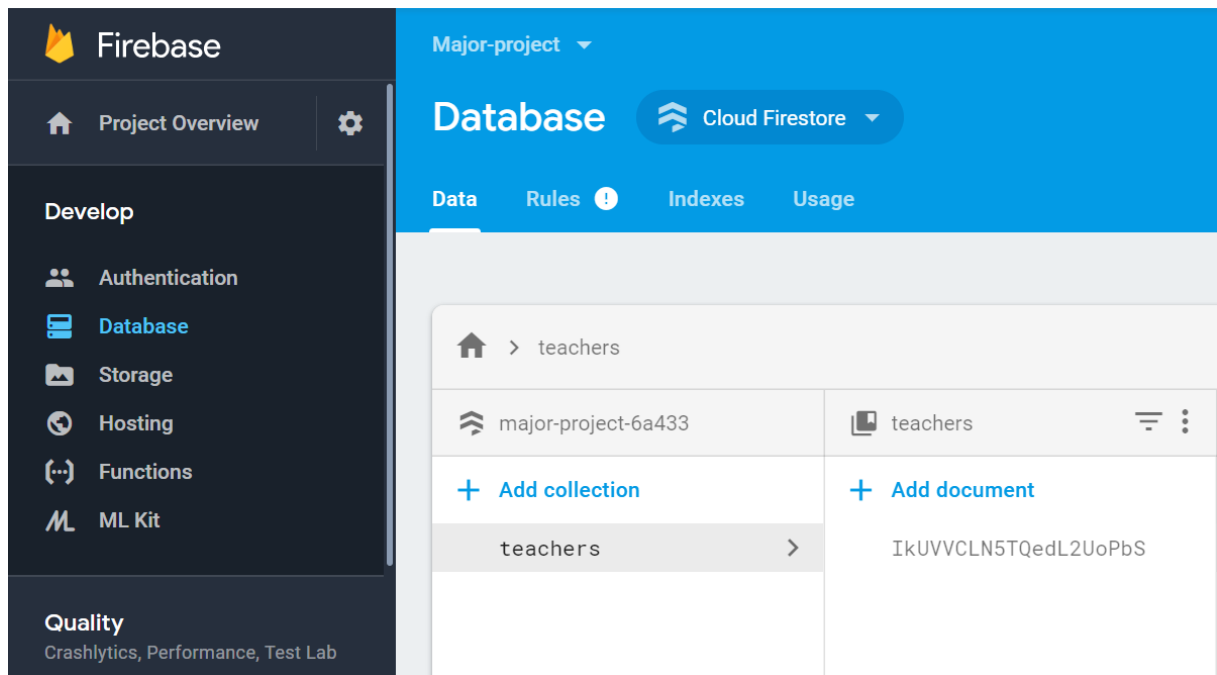


Fig 4.3.1 middleware for BOT

4.4 CODE

Main.java

```
package com;

import java.io.File;

public class Chatbot {
    private static final boolean TRACE_MODE = false;
    static String botName = "super";

    public static void main(String[] args) {
        try {

            String resourcesPath = getResourcesPath();
            System.out.println(resourcesPath);
            MagicBooleans.trace_mode = TRACE_MODE;
            Bot bot = new Bot("super", resourcesPath);
            Chat chatSession = new Chat(bot);
            bot.brain.nodeStats();
            String textLine = "";

            while(true) {
                System.out.print("Human : ");
                textLine = IOUtils.readInputTextLine();
                if ((textLine == null) || (textLine.length() < 1))
                    textLine = MagicStrings.null_input;
                if (textLine.equals("q")) {
                    System.exit(0);
                } else if (textLine.equals("wq")) {
                    bot.writeQuit();
                    System.exit(0);
                } else {
                    String request = textLine;
                    if (MagicBooleans.trace_mode)
                        System.out.println("STATE=" + request + ":THAT=" + ((History) chatSession.thatHistory.get(0)).get(0) +
                        String response = chatSession.multisentenceRespond(request);
                    while (response.contains("&lt;"))
                        response = response.replace("&lt;", "<");

                    String request = textLine;
                    if (MagicBooleans.trace_mode)
                        System.out.println("STATE=" + request + ":THAT=" + ((History) chatSession.thatHistory.get(0)).get(0) +
                        String response = chatSession.multisentenceRespond(request);
                    while (response.contains("&lt;"))
                        response = response.replace("&lt;", "<");
                    while (response.contains("&gt;"))
                        response = response.replace("&gt;", ">");
                    System.out.println("bot : " + response);
                }
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    private static String getResourcesPath() {
        File currDir = new File(".");
        String path = currDir.getAbsolutePath();
        path = path.substring(0, path.length() - 2);
        System.out.println(path);
        String resourcesPath = path + File.separator + "src" + File.separator + "main" + File.separator + "resources";
        return resourcesPath;
    }
}
```


Aiml.java

```
package com;

import java.io.File;

public class AddAiml {

    private static final boolean TRACE_MODE = false;
    static String botName = "super";

    public static void main(String[] args) {
        try {

            String resourcesPath = getResourcesPath();
            System.out.println(resourcesPath);
            MagicBooleans.trace_mode = TRACE_MODE;
            Bot bot = new Bot("super", resourcesPath);

            bot.writeAIMLFiles();

        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    private static String getResourcesPath() {
        File currDir = new File(".");
        String path = currDir.getAbsolutePath();
        path = path.substring(0, path.length() - 2);
        System.out.println(path);
        String resourcesPath = path + File.separator + "src" + File.separator + "main" + File.separator + "resources";
        return resourcesPath;
    }
}
```

Pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>com.example.howtodojava</groupId>
    <artifactId>cmdChatbot</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <dependencies>
        <dependency>
            <groupId>com.google</groupId>
            <artifactId>Ab</artifactId>
            <version>0.0.4.3</version>
            <scope>system</scope>
            <systemPath>${basedir}/lib/Ab.jar</systemPath>
        </dependency>
    </dependencies>
</project>
```

Client.aiml

```

<?xml version="1.0" encoding="UTF-8"?>
<aiml>
<category><pattern>MY FAVORITE COLOR IS <SET>COLOR</SET></pattern>
<template><set name="favoritecolor"><formal><star/></formal></set> is a nice color. <think><set name="it"><set name="topic"><star/></set></set>
</think></template>
</category>
<category><pattern>MY FAVORITE COLOR IS *</pattern>
<template>I didn't recognize <star/> as a color.</template>
</category>
<category><pattern>MY FAVORITE COLOR IS GREEN</pattern>
<template><set name="favoritecolor">Green</set> is my favorite color too! <think><set name="it"><set name="topic">Green</set></set></think></template>
</category>
<category><pattern>I LIKE <SET>COLOR</SET></pattern>
<template><srai>MY FAVORITE COLOR IS <star/></srai></template>
</category>
<category><pattern>*</pattern><that>HOW MANY YEARS OLD ARE YOU</that>
<template><srai>MY AGE IS <star/></srai></template>
</category>
<category><pattern>*</pattern><that>WHAT WOULD * CHANGE MY NAME TO</that>
<template><srai>YOUR NAME IS <star/></srai></template>
</category>
<category><pattern>*</pattern><that>WHAT IS YOUR SIGN</that>
<template><think><set name="sign"><star/></set></think>
<srai>HOROSCOPE FOR <get name="sign"/></srai></template>
</category>
<category><pattern>HIS NAME IS *</pattern>
<template>How well do you know him?<think><set name="he"><set name="topic"><star/></set></set></think></template>
</category>
<category><pattern>I DO NOT LIKE *</pattern>
<template>I will add <set name="dislikes"><star/></set> to your list of dislikes. <think><set name="it"><set name="topic"><star/></set></set>
</think></template>
</category>
<category><pattern>I HAVE A *</pattern>
<template><random><li>How long have you had it?</li><li>How does it make you feel?</li><li>When did you get it?</li></random><think><set name="it"><set name="topic"><star/></set></set></think></template>
</category>
<category><pattern>I HAVE *</pattern>
<template>Is that one of your favorite possessions? <think><set name="has"><star/></set></think> <think><set name="it"><set name="topic"><star/></set></set></think></template>
</category>
<category><pattern>I AM MARRIED</pattern>
<template>How long have you been <set name="status">married</set>?</template>
</category>
<category><pattern>I AM A DOCTOR</pattern>
<template><think><set name="job">Doctor</set></think> Where did you go to medical school?</template>
</category>
<category><pattern>I AM FROM *</pattern>
<template>Is that where you live now? <think><set name="birthplace"><star/></set></think></template>
</category>
<category><pattern>I LIKE *</pattern>
<template>What do you like most about it? <think><set name="it"><set name="topic"><star/></set></set></think></template>
</category>
<category><pattern>I LIKE TO *</pattern>
<template><think><set name="likes"><star/></set></think> <random><li>How long have you been doing it?</li><li>Do you get to do it very often?</li><li>Do you think I would like to?</li></random><think><set name="it"><set name="topic"><star/></set></set></think></template>
</category>
<category><pattern>I WOULD RATHER BE IN *</pattern>
<template>What would be better about it? <think><set name="it"><set name="topic"><star/></set></set></think></template>
</category>
<category><pattern>HE</pattern>
<template><get name="he"/></template>
</category>
<category><pattern>MY SIGN IS *</pattern>
<template>I know a lot of <formal><star/></formal>s.
<think>
<set name="it"><star/></set>
<set name="sign"><star/></set>
<set name="topic"><star/></set>
</think></template>
</category>
<category><pattern>MY ORIENTATION</pattern>

```

```

<template><get name="orientation"/></template>
</category>
<category><pattern>MY ORIENTATION IS *</pattern>
<template>Thanks for confiding in me. <think><set name="orientation"><star></set></think></template>
</category>
<category><pattern>MY DAUGHTER</pattern>
<template><get name="daughter"/></template>
</category>
<category><pattern>MY DAUGHTER S NAME IS *</pattern>
<template>I will remember <star> is your daughter. <think><set name="daughter"><star></set></think><think><set name="she"><set name="topic">
<star></set></set></think></template>
</category>
<category><pattern>MY LATITUDE</pattern>
<template><get name="latitude"/></template>
</category>
<category><pattern>MY BOYFRIEND</pattern>
<template><get name="boyfriend"/>?</template>
</category>
<category><pattern>MY NAME</pattern>
<template><get name="name"/></template>
</category>
<category><pattern>MY FULL NAME</pattern>
<template><condition name="firstname">
<li value="what"/>
<li><get name="firstname"/></li>
</condition>
<condition name="middlename">
<li value="what"/>
<li><get name="middlename"/></li>
</condition>
<condition name="lastname">
<li value="what"/>
<li><get name="lastname"/></li>
</condition>.
I call you <get name="name"/>.</template>
</category>

```

```

<condition name="has"><li value="what"/><li>has: <get name="has"/>.</li></condition>
<condition name="he"><li value="who"/><li>he: <get name="he"/>.</li></condition>
<condition name="husband"><li value="who"/><li>husband: <get name="husband"/>.</li></condition>
<condition name="it"><li value="what"/><li>it: <get name="it"/>.</li></condition>
<condition name="lastname"><li value="what"/><li>lastname: <get name="lastname"/>.</li></condition>
<condition name="latitude"><li value="which"/><li>latitude: <get name="latitude"/>.</li></condition>
<condition name="likes"><li value="what"/><li>likes: <get name="likes"/>.</li></condition>
<condition name="location"><li value="where"/><li>location: <get name="location"/>.</li></condition>
<condition name="longitute"><li value="which"/><li>longitute: <get name="longitute"/>.</li></condition>
<condition name="middlename"><li value="what"/><li>middlename: <get name="middlename"/>.</li></condition>
<condition name="mother"><li value="who"/><li>mother: <get name="mother"/>.</li></condition>
<condition name="name"><li value="Friend"/><li>name: <get name="name"/>.</li></condition>
<condition name="orientation"><li value="unknown"/><li>orientation: <get name="orientation"/>.</li></condition>
<condition name="phonenumber"><li value="unknown"/><li>phonenumber: <get name="phonenumber"/>.</li></condition>
<condition name="province"><li value="which"/><li>province: <get name="province"/>.</li></condition>
<condition name="religion"><li value="unknown"/><li>religion: <get name="religion"/>.</li></condition>
<condition name="residence"><li value="where"/><li>residence: <get name="residence"/>.</li></condition>
<condition name="road"><li value="which"/><li>road: <get name="road"/>.</li></condition>
<condition name="she"><li value="who"/><li>she: <get name="she"/>.</li></condition>
<condition name="sign"><li value="which"/><li>sign: <get name="sign"/>.</li></condition>
<condition name="sister"><li value="who"/><li>sister: <get name="sister"/>.</li></condition>
<condition name="state"><li value="which"/><li>state: <get name="state"/>.</li></condition>
<condition name="status"><li value="what"/><li>status: <get name="status"/>.</li></condition>
<condition name="street"><li value="which"/><li>street: <get name="street"/>.</li></condition>
<condition name="there"><li value="where"/><li>there: <get name="there"/>.</li></condition>
<condition name="them"><li value="who"/><li>them: <get name="them"/>.</li></condition>
<condition name="topic"><li value="unknown"/><li>topic: <get name="topic"/>.</li></condition>
<condition name="town"><li value="which"/><li>town: <get name="town"/>.</li></condition>
<condition name="voicemailnumber"><li value="unknown"/><li>voicemailnumber: <get name="voicemailnumber"/>.</li></condition>
<condition name="we"><li value="who"/><li>we: <get name="we"/>.</li></condition>
<condition name="wife"><li value="who"/><li>wife: <get name="wife"/>.</li></condition></template>
</category>
<category><pattern>YOUR NAME IS *</pattern>
<template>OK, from now on you can call me <star>.
<think><set name="customname"><star></set></think></template>
</category>
<category><pattern>HOW OLD AM I</pattern>
<template>You said you were <get name="age"/> years old?</template>
</category>
<category><pattern>I AM GAY</pattern>
<template>Thank you for telling me. <think><set name="orientation">gay</set></think></template>
</category>
</aiml>

```

5 VERIFICATION & VALIDATION

5.1 UNIT TESTING

Unit Testing is a period of programming testing where singular segments of a framework, with controlled information and handling use, are tried. It checks for the littlest testable piece of the framework. For our task, we have utilized Junit

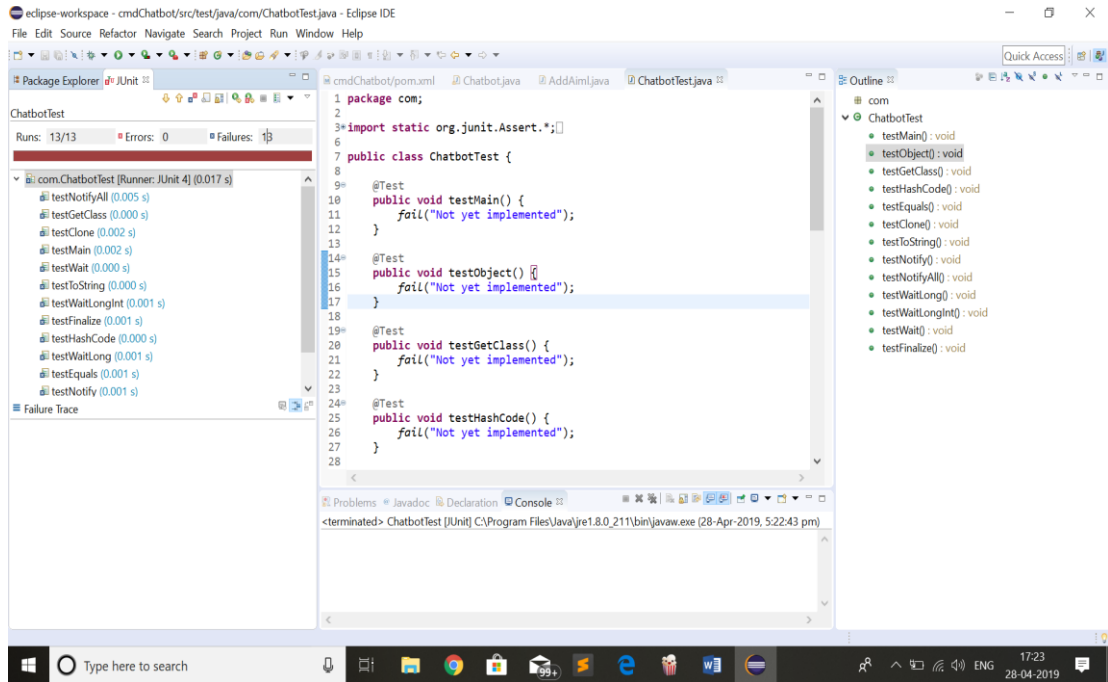


Fig 5.1.1 Unit Test cases for BOT.

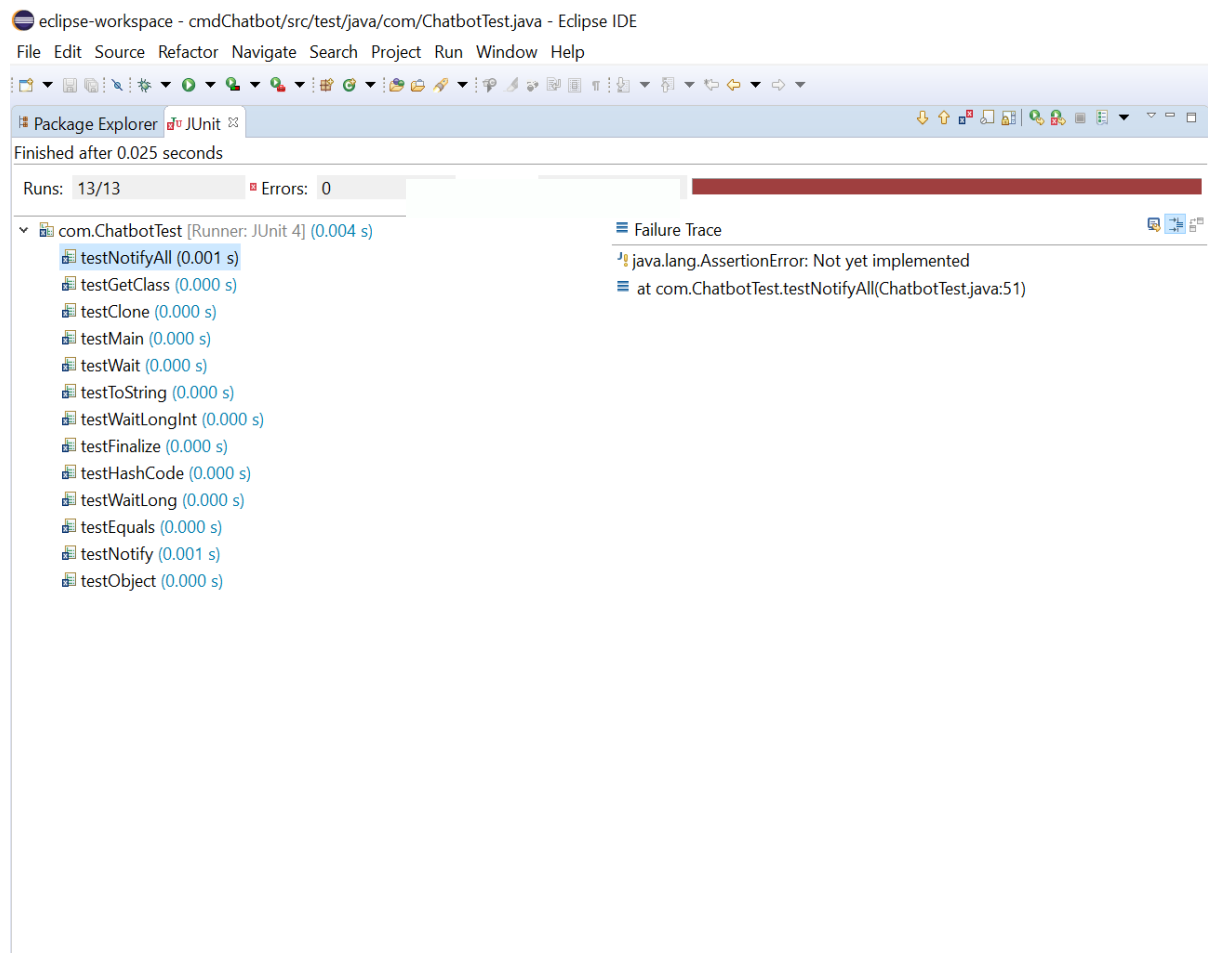


Fig 5.1.2 Unit Test result for BOT.

5.2 INTEGRATION TESTING

Integration Testing is the next phase of software testing in which two or more modules of a system are integrated and tested as a group. It is done to either test the integration between components of a system or between different parts of a system. In this project, it is done using a test suite created in *unit test*. Integration testing is the second dimension of programming testing, comes after unit testing. The little units are joined or coordinate with different units to frame a gathering. Integration testing is done to discover the flaw that happens amid joining or blend of units.

Passage Criteria:

- Completion of Unit testing

Sources of info:

- Requirements Data
- Design Document
- Verification Plan
- Integration Documents

```
/Users/vaibhav3301/Desktop/projects/major/agrowell/db.py:58: RankWarning: The fit may be poorly conditioned
  coefs = poly.polyfit(x, y, 1)
Head: test_api_data (__main__.FlaskBookshelfTests)
Result: <Response streamed [200 OK]>
Status: Passed
.Head: test_api_status_code (__main__.FlaskBookshelfTests)
Result: <Response streamed [200 OK]>
Status: Passed
.Head: test_home_data (__main__.FlaskBookshelfTests)
Result: <Response streamed [200 OK]>
Status: Passed
.Head: test_home_status_code (__main__.FlaskBookshelfTests)
Result: <Response streamed [200 OK]>
Status: Passed
.
-----
Ran 4 tests in 3.404s
OK
```

Fig 5.2.1 Integration Test result of BOT.

5.3 USER TESTING

User Testing is the last phase of software testing done by a set of selected clients to validate the end-to-end business flow. It is also called beta testing as it helps determine if the requirements are met and whether it is ready for delivery to the market. Client Testing is utilized to locate the constant issue looked by the client. The client testing is finished by the client itself in its condition. At the point when the client is doing trying, the analyser keeps the watch on the issues or disarray looked by the client while utilizing the software. Some of the issues looked by the client are:

1. Mistake amid introducing the product in light of the fact that the product is as of now introduced.

2. During organization, a blunder can happen if the records or organizer identified with the product is opened.

3. The mistake happens amid the keep running as due to the PC unfit to discover the area of the document.

4. PC explicit issues:

- Less space is left on the PC to introduce the product.
- If the temporary documents are not erased from the PC.
- Locked records in the PC lead to client perplexity and transports.

5. Application explicit blunders.

- Unable to discover the introduce application.
- File missing amid establishment.

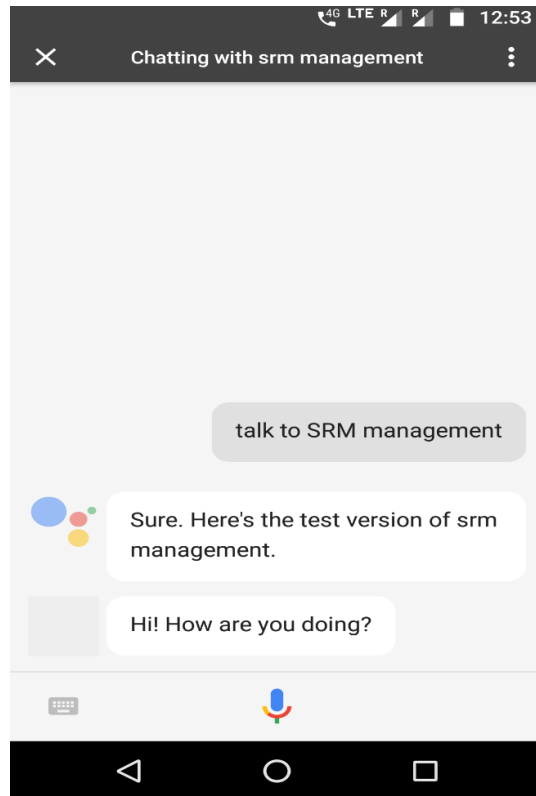


Fig 5.3.1 User Test result for BOT.

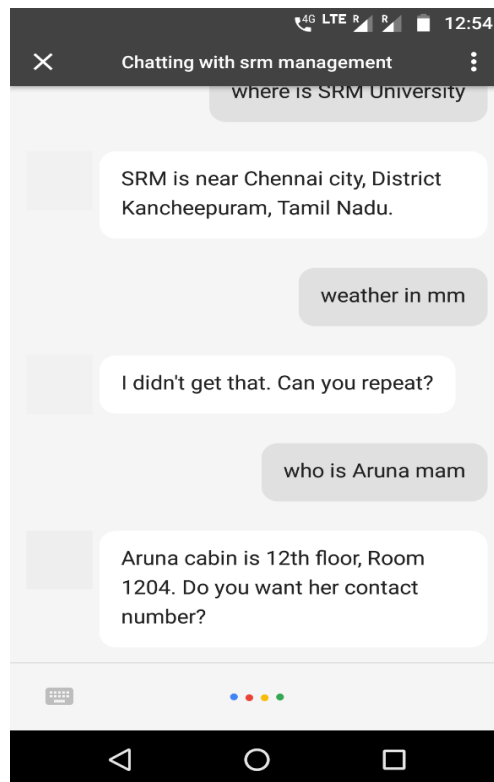


Fig 5.3.2 User Test result.

5.4 SIZE - LOC

Lines of Code(LOC) metrics is one of the techniques for software size estimation. It is done to measure the length of the whole program for the system. In this project, LOC metrics is done using *LineCount* extension for python language.

```
1  =====
2  EXTENSION NAME : linecount
3  EXTENSION VERSION : 0.1.7
4  -----
5  count time : 2018-03-12 13:07:34
6  count workspace : /Users/vaibhav3301/Desktop/projects/major/agrowell
7  total files : 73
8  total code lines : 371392
9  total comment lines : 1830
10 total blank lines : 2407
11
12 __pycache__/app.cpython-36.pyc, it is a binary file.
13 __pycache__/db.cpython-36.pyc, it is a binary file.
14 app.py, code is 19, comment is 0, blank is 5.
15 data.csv, code is 58586, comment is 0, blank is 0.
16 db.py, code is 55, comment is 0, blank is 7.
17 listner.py, code is 39, comment is 2, blank is 9.
18 mytest.py, code is 37, comment is 23, blank is 13.
19 rethinkdb data/117a620d-6f62-46e0-bac1-3132ff943357. it is a binary file.
```

Fig 5.4.1 LOC Metrics.

5.5 COST ANALYSIS

Cost Analysis is done by accumulating, examining and manipulating the cost summary into its constituents for comparative study and report on each factor. This is done after the completion of the project. It will be done using IFPUG function point analysis, as the cost estimation was done using the same which gave an estimation of 165 fp.

After the project completion, we see that there 5 external inputs of average complexity, 4 external outputs of high complexity, 1 external inquiries of high complexity, 1 internal logical file of average complexity and 2 external interfaces files of high complexity.

Component	No. of Components	Corresponding Weighting Factor	Weighting Factor Value
EIs	5	average	4
EOs	4	high	7
EQs	1	high	6
ILFs	1	average	10
EIFs	2	high	10

Fig 5.5 Weighting Factor according to the project.

Their degree of impact for GSCs are mentioned below in Table VIII:-

General System Characteristics (GSCs)	Degree of Influence
Data Communications	5
Distributed Data Processing	5
Performance	3
Heavily Used Configuration	4
Transaction Rate	3
Online Data Entry	1
End-User Efficiency	4
Online Update	5
Complex Processing	5
Reusability	5

Installation Ease	2
Operational Ease	4
Multiple Sites	1
Facilitate Changes	5

Fig 5.5.2 GSCs and Degree of influence according to the project.

The calculations are as follows:-

$$\mathbf{UFP} = [(5 * 4) + (4 * 7) + (1 * 6) + (1 * 10) + (2 * 10)] \text{ fp}$$

$$\mathbf{= 84 \text{ fp}}$$

$$\mathbf{TDI = 52}$$

$$\mathbf{VAF = (52 * 0.01) + 0.65}$$

$$\mathbf{=1.17}$$

$$\mathbf{FP = (84 * 1.17) \text{ fp}}$$

$$\mathbf{= 98.28 \text{ fp}}$$

$$\mathbf{\approx 98 \text{ fp}}$$

We see that the analyzed cost is lesser than the estimated cost. This happened because of the introduction of database server into our system which was not included previously during planning and estimation. It increased our logical file system and reduced our interface files hence leading to reduced overall cost.

5.6 DEFECT ANALYSIS

Defect Analysis is a technique used to identify and prevent the problems from occurring and ensure continuous quality improvement in the system. It can be done using Pareto Chart or Cause-Effect Analysis (using Fishbone Diagram). We have used Cause-Effect Analysis for our project as it will help improve the quality and detect the defects in an efficient way.

- Data vulnerability: If the information put away in volumes have an infection and have consents which are executable.
- Container vulnerability: If the compartment picture is defenceless which can commandeer the host framework.
- OS Vulnerability: If the framework rendition isn't steady which prompts an accident or uncover a defencelessness which can be damaged.
- Dependency Vulnerability: It's an issue in a reliance code that could be abused to harm the classification, respectability, or accessibility of the venture

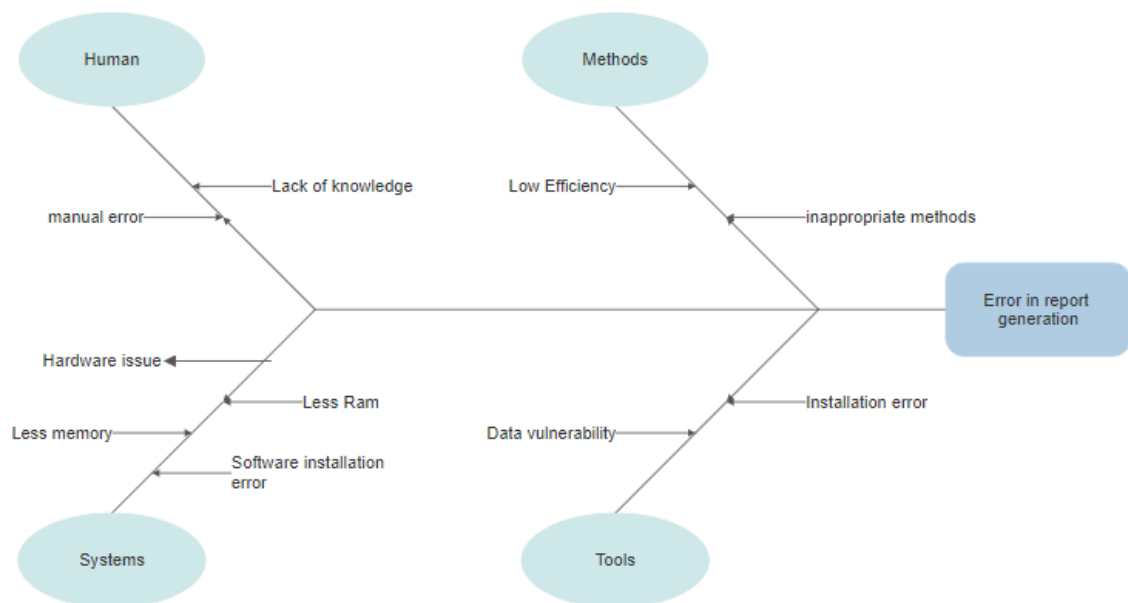


Fig 5.6 Cause-Effect for BOT

5.7 McCALL'S QUALITY FACTORS

McCall's Quality Factors focus on three main perspectives which further consist of a set of quality attributes. A software system built on these factors will demonstrate a high quality, even during changing technologies.

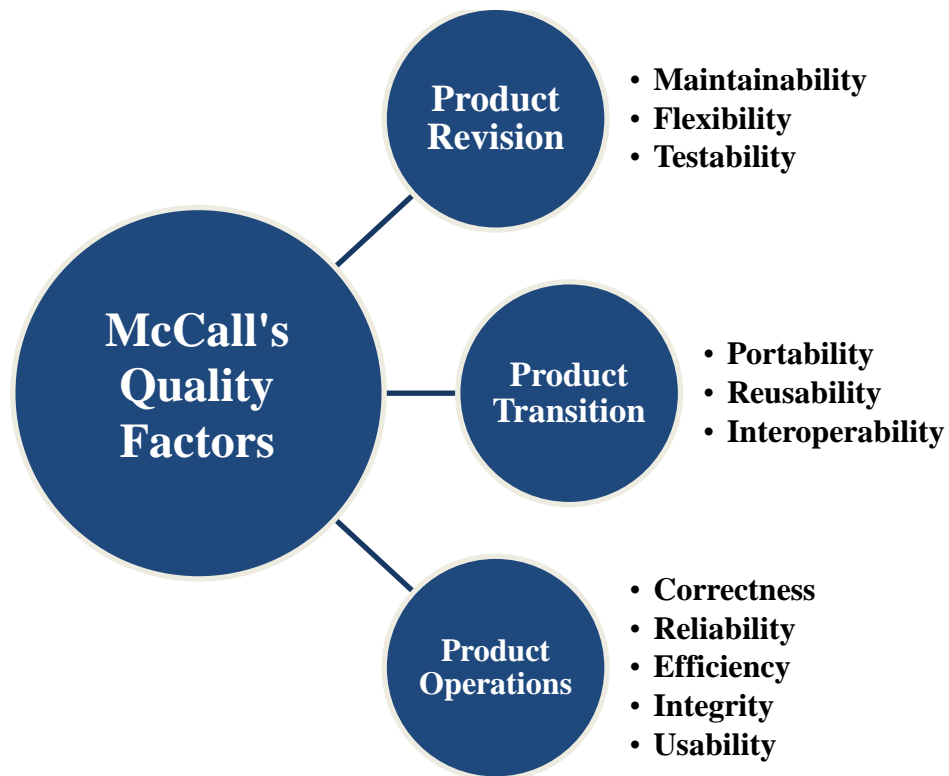


Fig 2: McCall's Quality Factors

McCall's Quality Factors-

Correctness: The system works to gather information about place

Reliability: System is reliable in terms of results produced.

Efficiency: The system is efficient and works well for visually imapaired people.

Flexibility: Extra modules can be added over the
functionality of the server.

Portability: Could be run on multi platform Operating
system with the access of Internet.

Usability: Person should have the smart phone with internet.

Reusability: System can be made to work for accessing different universities data if necessary.

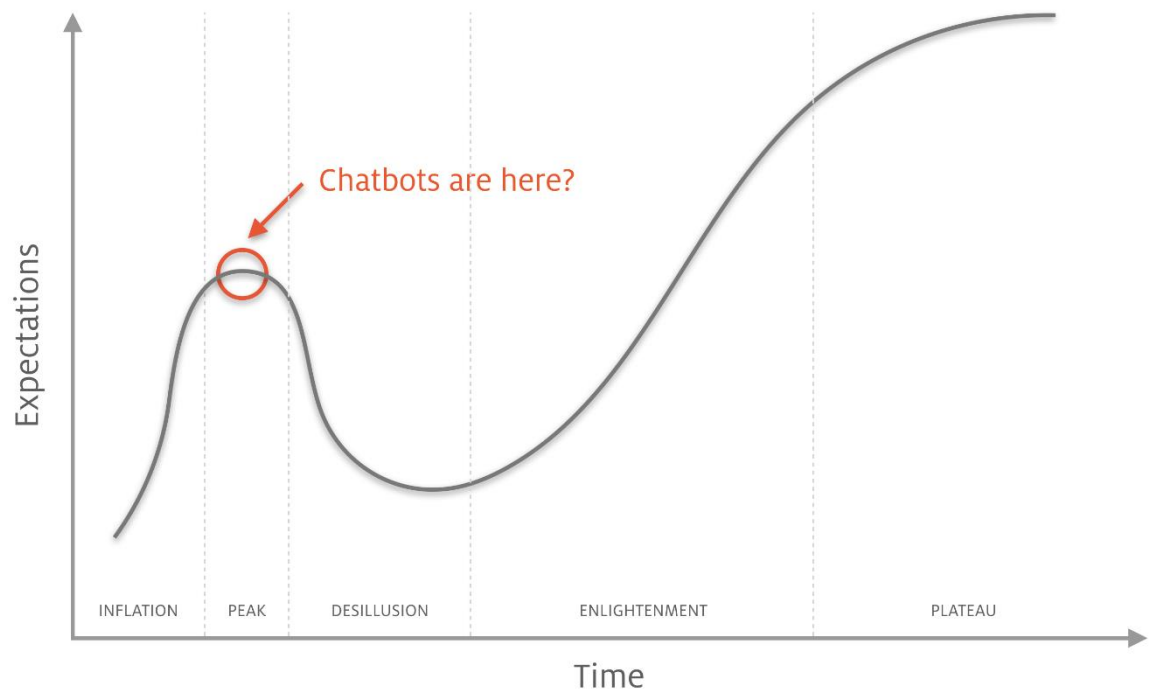
6 EXPERIMENT RESULTS & ANALYSIS

6.1 RESULTS

We present an education based chatbot for visually impaired people to help them gather information of different universities. The application was developed with web server platform. Java programming language was used for development. Dataflow was also used for dataset. Based on user testing it was found that the system is not too visually attractive and needs more User interface work to be done on it.

This application is useful for people who want to gather university information. Overall this system was found very useful and efficient by the users. The testing revealed a few weak points in the system that may cause the user some inconvenience.

6.2 RESULT ANALYSIS



RESULT GENERATION:

The user is able to get answer for all there queries . Once the query is in place, the user can get the results after getting processed through the collaborative filtering procedure.

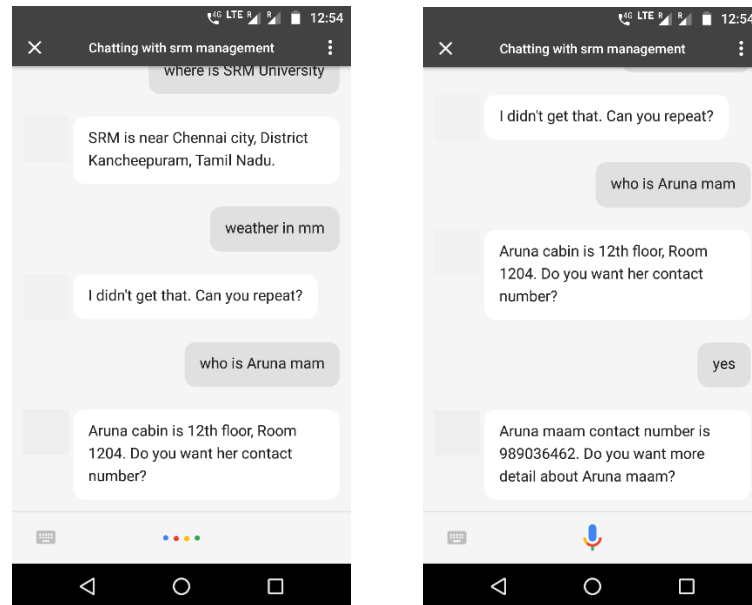


Fig 6.2 Result on mobile application of BOT

```

Problems @ Javadoc Declaration Console
Chatbot [Java Application] C:\Program Files\Java\jre1.8.0_211\bin\javaw.exe (29-Apr-2019, 3:14:11 pm)
AIML modified Sun Apr 14 22:59:55 IST 2019 AIMLIF modified Sun Apr 28 17:01:19 IST 2019
readCertainIFCategories 0 categories from deleted.aiml.csv
readCertainIFCategories 0 categories from unfinished.aiml.csv
Loading AIML files from C:\Users\Aman\Downloads\cmdChatbot\cmdChatbot\src\main\resources\bots/s
Reading Learnf file
AIML Set birdfeature not found.
AIML Set birdfeature not found.
Loaded 4849 categories in 0.243 sec
--> Bot super 4849 completed 0 deleted 0 unfinished
28550 nodes 22307 singletons 4849 leaves 0 shortcuts 1394 n-ary 28549 branches 0.99996495 average
Human : hello
bot : Hi it's great to see you!
Human : how are you
bot : Excited to talk with you.
Human :

```

Fig 6.2.2 Result on window for BOT

6.3 CONCLUSION & FUTURE WORK

Without the complexity of multiple forms and windows, it is often impossible to get all the data on a single interface. Chatbots are designed to eliminate this difficulty by providing a versatile and user-friendly interface to address college and teacher queries. The purpose of the chat robot system is to simulate human conversation. Its architecture integrates language models and computational algorithms to simulate the online communication of information between people and computers using natural language. College students and employees are free to upload their enquiries. Chat bots can quickly and efficiently search for answers to queries and get relevant links to their questions. Conducted a background study that included an overview of the conversation process and tried to find relevant keywords related to the query

Provide the appropriate link. The database store includes information about questions, answers, keywords, and logs. We have also developed an interface. The developed interface will consist of two parts, one for the user and one for the administrator. Other algorithms can be implemented instead of AIML-based robots. We can include speech-based queries. The user must provide voice input and the system will provide text output. In addition, after the successful implementation of the chat bot in the university domain name, we can implement it in other fields, such as medical, forensic, sports and so on. This will be beneficial in all areas, because without spending too much time, we also access related information, which is also without any sorting

7 PLAGIARISM REPORT

aman 29.4 345pm

ORIGINALITY REPORT

6%

SIMILARITY INDEX

3%

INTERNET SOURCES

3%

PUBLICATIONS

3%

STUDENT PAPERS

PRIMARY SOURCES

1

M Naveen Kumar, P C Linga Chandar, A Venkatesh Prasad, K Sumangali. "Android based educational Chatbot for visually impaired people", 2016 IEEE International Conference on Computational Intelligence and Computing Research (ICCIIC), 2016

Publication

2%

2

Submitted to North South University

Student Paper

1%

3

www.geeksforgeeks.org

Internet Source

1%

4

Submitted to Carnegie Mellon University

Student Paper

1%

5

Submitted to De Montfort University

Student Paper

<1%

6

www2.southeastern.edu

Internet Source

<1%

7

Submitted to University of Westminster

Student Paper

<1%

8	Mourad Debbabi, Fawzi Hassaïne, Yosr Jarraya, Andrei Soeanu, Luay Alawneh. "Verification and Validation in Systems Engineering", Springer Nature, 2010 <small>Publication</small>	<1%
9	shwetashwetakhatri.blogspot.com <small>Internet Source</small>	<1%
10	www.dulcian.com <small>Internet Source</small>	<1%
11	maxwellsci.com <small>Internet Source</small>	<1%

Exclude quotes ☐ On

Exclude matches

< 10 words

Exclude bibliography ☐ On

8 CONFERENCE CERTIFICATE

Virtual Assistant Machine for Visually impaired People

By-Aman Srivastava , Dhananjai Gupta

Guide: Mrs. S. Aruna

Abstract

The inspiration driving this application gives direction based Chatbot to Visually impaired people. It will guidance-based inquiries from people who are prevented by the outside world. There is no uncertainty that they can send applications with the assistance of the Google Voice look and feel. At the point when the application opens, it gives voice direction to utilize the application. Likewise, with the substance structure, income will be given in the phonetic structure. In this manner, an average individual can likewise utilize the application.

I. Introduction

ChatBot combines the creative advancement of artificial intelligence (AI) in blended learning (as a combination of personalized e-learning and traditional learning room technology). After the separation of learning phases, the multiplication of community-oriented laboratories and online testing (e-learning), opportunities for machine learning (m-Learning) have arrived to help with versatile and self-coordinating training. Our exploration is to use a versatile "exquisite customer" (tablet or mobile phone) to develop a learning/communication process in the study hall with the help of Chatbots. The personal portable terminals of these schools are linked to the archives of the college's mentoring cloud, in a dynamic relationship, in which Chatbots are implemented as teacher content - mechanization experts between learners, enabling continuous criticism to improve learning outcomes - Keep a lot of information, multi-faceted, and adapt to the level of each member through customized workouts. Useful use cases Chatbots are everywhere in the tech zone. Comparative Exam - About the advantages of using a chat bot in a coaching framework - Investigate the interpersonal relationship with a conversational expert and the specific issues of using a chat bot in teaching. Amazing promises about chatbot planning and improvement, and their hybrid in AI applications - through XML extensions like XML

AIML (Artificial Intelligence Markup Language). Our answer is to coordinate the joint access phase in the communication between educators and students to help them stay engaged and

9 REFERENCES

- [1] Yuhua Li, David McLean, Zuhair A. Bandar, James D. O'Shea, Keeley Crockett, "Sentence Similarity Based on Semantic Nets and Corpus Statistics", IEEE Transactions on Knowledge and Data Engineering, Volume 18 - No. 8, August 2006.
- [2] Emanuela Haller, Traian Rebedea, "Designing a Chat-bot that Simulates an Historical Figure", IEEE Conference Publications, July 2013.
- [3] Pratik Slave, Vishruta Patil, Vyankatesh Gaikwad, Girish Wadhwa, "College Enquiry Chat Bot", International Journal on Recent and Innovation Trends in Computing and Communication, Volume 5, Issue 3, March 2015.
- [4] "AIML Based Voice Enabled Artificial Intelligent Chatterbot", International Journal of u- and e- Service, Science and Technology Volume 8 - No. 2, 2015.
- [5] Amey Tiwari, Rahul Talekar, Prof. S. M. Patil, "College Information Chatbot System", International Journal of Engineering Research and General Science, Volume 2, Issue 2, April 2017.
- [6] Rachit Kulkarni, Ankit Methwani, Nakul Pawar, Charmi Valecha, Pooja Shetty, "College Chatbot", International Journal of Advanced Research in Computer Engineering & Technology, Volume 6, Issue 4, April 2017.
- [7] Chaitrali S. Kulkarni, Amruta U. Bhavsar, Savita R. Pingale, Prof. Satish S. Kumbhar, "BANK CHATBOT - An Intelligent Assistant System Using NLP and Machine Learning", International Research Journal of Engineering and Technology, Volume 4, Issue 5, May 2017.
- [8] Yash Mehta, Shreya Sawkar, "The college chatbot", International Journal of Computer Applications, Volume 173 - No. 7, September 2017.
- [9] Prof. K. Bala, Mukesh Kumar, Sayali Hulawale, Sahil Pandita, "Chat-Bot For College Management System Using A.I.", International Research Journal of Engineering and Technology, Volume 4, Issue 11, Nov 2017.