MEDICAL CHATBOT

A project report submitted in the partial fulfillment of the requirements for the award of degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

by

ROHIT CHAKRADHARA (Reg.No:317506408033)
ROHIT DUBEY (Reg.No:317506408034)
ROHIT REDDY BODUKONDU (Reg.No:317506408035)

Under the guidance of

PROF. SHASHI MOGALLA

DEPARTMENT OF COMPUTER SCIENCE AND SYSTEMS ENGINEERING
ANDHRA UNIVERSITY COLLEGE OF ENGINEERING (A)
ANDHRA UNIVERSITY, VISAKHAPATNAM-53000

2020-2021

DEPARTMENT OF COMPUTER SCIENCE AND SYSTEMS ENGINEERING

ANDHRA UNIVERSITY COLLEGE OF ENGINEERING (A)

ANDHRA UNIVERSITY, VISAKHAPATNAM

BONAFIDE CERTIFICATE

This is to certify that project work entitled "**MEDICAL CHATBOT**", done by

Rohit Chakradhara (Reg.No:317506408033),

Rohit Dubey (Reg.No:317506408034),

Rohit Reddy Bodukondu (Reg.No:317506408035),

students of 4/6 Computer Science & Networking, Department of Computer Science & Systems Engineering, Andhra University College of Engineering (A) during the period 2020 - 2021 in the partial fulfilment of the requirements for the award of degree of Bachelor of Technology in Computer Science and Engineering. This work has not been submitted to any other university for the award of Degree or Diploma.

Prof. Shashi Mogalla
Dept. of CSE
AUCE(A)

Prof K. Nageswara Rao
Head , Dept.of CSE
AUCE(A)

DECLARATION

WE **Rohit Chakradhara** (Reg.No:317506408033), **Rohit Dubey** (Reg.No:317506408034), **Rohit Reddy Bodukondu** (Reg.No:317506408035), hereby declare that the project entitled "**MEDICAL CHATBOT**" is an original work done at Andhra University College of Engineering (a), Visakhapatnam, submitted in partial fulfillment of the requirements for the award of degree of BACHELOR OF TECHNOLOGY. I assure that this project is not submitted in any university or college.

Rohit Chakradhara (Reg.No:317506408033),
Rohit Dubey (Reg.No:317506408034),
Rohit Reddy Bodukondu (Reg.No:317506408035),

Place: Visakhapatnam.
Date:

ACKNOWLEDGEMENT

It is with great sense of satisfaction that I present "**MEDICAL CHATBOT**" in the form of final project.

We have immense pleasure in expressing our earnest gratitude to our Project Guide **Prof. Shashi Mogalla**, Andhra University for his inspiring and scholarly guidance. Despite his pre-occupation with several assignments, he has been kind enough to spare his valuable time and gave us the necessary counsel and guidance at every stage of planning and constitution of this work. We express sincere gratitude for having accorded us permission to take up this project work and for helping us graciously throughout the execution of this work.

We express sincere Thanks to **Prof. K. Nageswara Rao**, Head of the Department Computer Science and Systems Engineering, Andhra University College of Engineering(A) for his keen interest and providing necessary facilities for this project study.

We express sincere gratitude to **Prof. P.V.G.D. Prasad Reddy**, Chairman, Board of Studies, Computer Science and Systems Engineering, Andhra University College of Engineering(A) for his keen interest and for providing necessary facilities for this project study.

We express sincere Thanks to **Prof. P. Srinivas Rao**, Principal, Computer Science and Systems Engineering, Andhra University College of Engineering(A) for his keen interest and for providing necessary facilities for this project study.

We extend our sincere thanks to our academic teaching staff and non-teaching staff for their help throughout our study.

ABSTRACT

To start a good life healthcare is very important. But it is very difficult to the consult the doctor if any health issues. The proposed idea is to create a healthcare chatbot using Natural Language Processing technique it is the part of Artificial Intelligence that can diagnose the disease and provide basic. To reduce the healthcare costs and improve accessibility to medical knowledge the Healthcare chatbot is built. Some chatbots acts as a medical reference books, which helps the patient know more about their disease and helps to improve their health. The user can achieve the benefit of a healthcare chatbot only when it can diagnose all kind of disease and provide necessary information. The system provides text or voice assistance, that means user can use his own convenient language, Bot will provide which type of disease based on the user symptoms, and provides doctor and also provides food suggestion that means which type of food you have to take. Thus, people will have an idea about their health and have the right protection. Chatbots are programs that work on Machine Learning (ML) as well as Artificial Intelligence (AI)Natural Language Processing (NLP) techniques such as NLTK for Python can be applied to analyses speech, and intelligent responses can be found by designing an engine to provide appropriate human like responses

Background of the Project :

Chatbots have played a prominent role as human-computer interfaces. Chatbot as a program that attempts to simulate typed conversation, with the aim of at least temporarily feeling the human into thinking they were talking to another person. Chatbot is a conversational agent that can interact with user in a given subject using the natural language. Many chatbots have been deployed on the internet for the purpose of education, customer service site, guidance, entertainment. The Hospital Chatbot (HC), is used to provide quick and smart replies to queries provided by patients as well as Doctors. Users can chat using any format there is no specific format the user has to follow. The System uses artificial algorithms to match the patterns to answer the query

CONTENTS

Introduction

People get addicted to the internet in obtaining information for every problem they face. This not only yet people to seek knowledge about general topics but also their health concerns. However, people are afraid of misinterpretation when they googled their symptoms since most search end up with creating unnecessary paranoid to the users and may sometimes inaccurate. Based on those needs, people start to develop several technologies to help people get the most accurate results on their disease. One of them is by creating a yes-no answer questionnaire system. It certainly helps, however, due to some diseases have almost the same symptoms as the other, we can't rely on this yes-no system since more information need to be elaborated to obtain accuracy. Another one is creating website whereas according to Aswini, a medical website plays a vital role in today's digital world and a lot of fora is available for answering the queries provided by the user. The need for a reliable and accurate diagnosis wakes the rise of a new generation of healthcare technology called the Medical Chatbot. The main idea of creating this chatbot is to replicate a person's discussion. This helps people to learn more about their symptoms and give them the most accurate diagnosis possible. The chatbot is also drawing upon the ever-growing medical question range, to broaden its already significant wealth of medical expertise. Many seemingly static scenes contain subtle changes that are invisible to the naked human eye. However, it is possible to pull out these small changes from videos through the use of algorithms via motion magnification. Motion magnification gives a way to visualize these small changes by amplifying them and to pull out interesting signals from these videos, such as the human pulse.

They are not involved regarding their health. So, they avoid to travel in hospitals for little issues.it may become a significant drawback. So, we will offer a thought is to make a health care chatbot system using AI that may identification the illness and supply basic information regarding the illness before consulting a doctor. Which helps the patients apprehend additional regarding their illness and improves their health. User can do the all-reasonably illness information. The system application uses question and answer protocol within the style of chatbot to answer user queries. The response to the question is replied supported the user question. The significant keywords are fetched from the sentence and answer to those sentences. If match is discovered or vital answer are given or similar answers are displayed can identification which sort of illness you have got supported user symptoms and additionally offers doctor details of explicit illness. It may cut back their health

problems by victimization this application system. The system is developed to scale back the tending price and time of the users because it isn't potential for the users to go to the doctors or consultants once in real time required.

1. INTRODUCTION

1.1 WHAT IS SOFTWARE?

Software, generally sense, is understood as a group of instructions or programs that instructs to a computer to perform specific tasks. Software could be a general term that's want to describe computer programs. Scripts, applications, programs and a group of instructions are all different terms want to describe software.

The theory of software was first proposed by Alan Mathison Turing in 1935 in his essay "Computable numbers with an application to the Entscheidungs Problem." However, the word software was proposed by statistician and mathematician John Tukey in a very 1958 issue of yank Mathematical Monthly within which he discussed the electronic calculators' programs.

Software is typically divided into three categories:

• System software could be a base for application software. System software generally includes operating systems, device drivers, text editors, compilers, disk formatters and utilities helping the pc to regulate more efficiently. it's responsible in providing basic non-specific-task functionalities and management of hardware components. The system software is typically written within the language of C programming.

• Programming software could also be a group of tools to help developers in writing programs. the numerous tools available are linkers, compilers, interpreters' debuggers and text editors.

• Application software is typically used to perform certain tasks and also the samples of the applying software includes educational software, database managing systems, office suites, application on gaming. the applying software can either be one program or a gaggle of portable programs.

Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms.

The Python interpreter and the extensive standard library are freely available in source or binary form for all major platforms from the Python Web site, https://www.python.org/, and may be freely distributed. The same site also contains distributions of and pointers to many free third-party Python modules, programs and tools, and additional documentation.

The Python interpreter is easily extended with new functions and data types implemented in C or C++ (or other languages callable from C). Python is also suitable as an extension language for customizable applications.

This tutorial introduces the reader informally to the basic concepts and features of the Python language and system. It helps to have a Python interpreter handy for hands-on experience, but all examples are self-contained, so the tutorial can be read off-line as well.

For a description of standard objects and modules, see The Python Standard Library. The Python Language Reference gives a more formal definition of the language. To write extensions in C or C++, read Extending and Embedding the Python Interpreter and Python/C API Reference Manual. There are also several books covering Python in depth.

This tutorial does not attempt to be comprehensive and cover every single feature, or even every commonly used feature. Instead, it introduces many of Python's most noteworthy features, and will give you a good idea of the language's flavor and style. After reading it, you will be able to read and write Python modules and programs, and

you will be ready to learn more about the various Python library modules described in The Python Standard Library.

The Python Standard Library

While The Python Language Reference describes the exact syntax and semantics of the Python language, this library reference manual describes the standard library that is distributed with Python. It also describes some of the optional components that are commonly included in Python distributions.

Python's standard library is very extensive, offering a wide range of facilities as indicated by the long table of contents listed below. The library contains built-in modules (written in C) that provide access to system functionality such as file I/O that would otherwise be inaccessible to Python programmers, as well as modules written in Python that provide standardized solutions for many problems that occur in everyday programming. Some of these modules are explicitly designed to encourage and enhance the portability of Python programs by abstracting away platform-specifics into platform-neutral APIs.

The Python installers for the Windows platform usually include the entire standard library and often also include many additional components. For Unix-like operating systems Python is normally provided as a collection of packages, so it may be necessary to use the packaging tools provided with the operating system to obtain some or all of the optional components

WHAT IS SOFTWARE DEVELOPMENT LIFE CYCLE(SDLC)?

The software development life cycle can be a framework that defines the tasks performed at each step within the software development process. SDLC could be a structure followed by a development team within the software organization. It consists of an in-depth plan describing the way to develop, maintain and replace specific

software. The life cycle defines a technique for improving the standard of software and therefore the overall development process. The software development life cycle is additionally called the software development process. SDLC consists of following activities:

1.Planning: the foremost important parts of software development, requirement gathering or requirement analysis are usually done by the foremost skilled and experienced software engineers within the organization. After the necessities are gathered from the client, a scope document is made during which the scope of the project is decided and documented.

2.Implementation: The software engineers start writing the code in keeping with the client's requirements.

3.Testing: this is often the method of finding defects or bugs within the created software.

4.Documentation: Every step within the project is documented for future reference and for the development of the software within the development process. the planning documentation may include writing the appliance programming interface (API).

5.Deployment and maintenance: The software is deployed after it's been approved for release.

6.Maintaining: Software maintenance is completed for future reference. Software improvement and new requirements (change requests) can take longer than the time needed to form the initial development of the software.

SDLC is nothing but Software Development Life Cycle. It is a standard which is used by software industry to develop good software.

SDLC (Spiral Model):

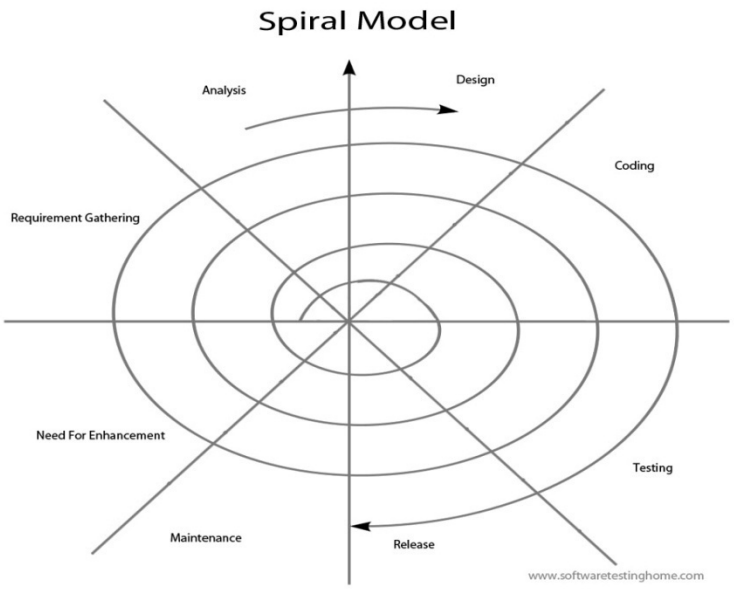


Fig 1: Spiral Model

Stages of SDLC:

- Requirement Gathering and Analysis
- Designing
- Coding
- Testing
- Deployment

Requirements Definition Stage and Analysis:

The requirements gathering process takes as its input the goals identified in the high-level requirements section of the project plan. Each goal will be refined into a set of one or more requirements. These requirements define the major functions of the intended application, define operational data areas and reference data areas, and define the initial data entities. Major functions include critical processes to be managed, as well as mission critical inputs, outputs and reports. A user class hierarchy is developed and associated with these major functions, data areas, and data entities. Each of these definitions is termed a Requirement. Requirements are identified by unique

requirement identifiers and, at minimum, contain a requirement title and textual description.

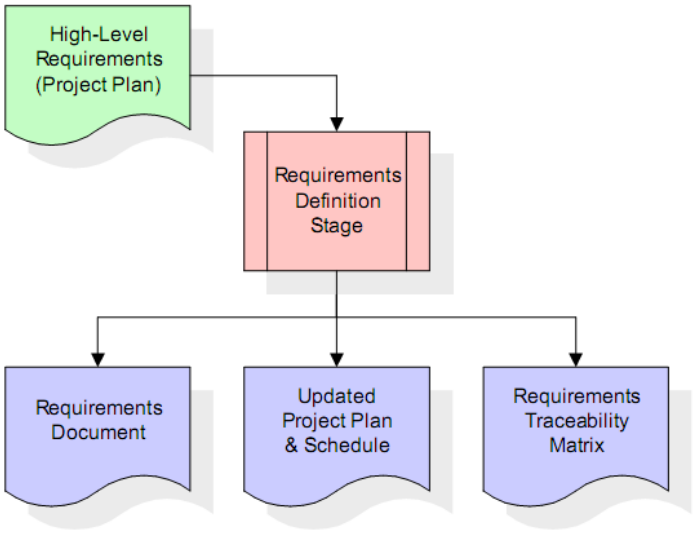


Fig 2: Requirement Stage

These requirements are fully described in the primary deliverables for this stage: the Requirements Document and the Requirements Traceability Matrix (RTM). the requirements document contains complete descriptions of each requirement, including diagrams and references to external documents as necessary. Note that detailed listings of database tables and fields are *not* included in the requirements document. The title of each requirement is also placed into the first version of the RTM, along with the title of each goal from the project plan. The purpose of the RTM is to show that the product components developed during each stage of the software development lifecycle are formally connected to the components developed in prior stages.

In the requirements stage, the RTM consists of a list of high-level requirements, or goals, by title, with a listing of associated requirements for each goal, listed by requirement title. In this hierarchical listing, the RTM shows that each requirement developed during this stage is formally linked to a specific product goal. In this format, each requirement can be traced to a specific product goal, hence the term *requirements traceability*. The outputs of the requirements definition stage include the requirements document, the RTM, and an updated project plan.

Design Stage:

The design stage takes as its initial input the requirements identified in the approved requirements document. For each requirement, a set of one or more design elements will be produced as a result of interviews, workshops, and/or prototype efforts. Design elements describe the desired software features in detail, and generally include functional hierarchy diagrams, screen layout diagrams, tables of business rules, business process diagrams, pseudo code, and a complete entity-relationship diagram with a full data dictionary. These design elements are intended to describe the software in sufficient detail that skilled programmers may develop the software with minimal additional input.

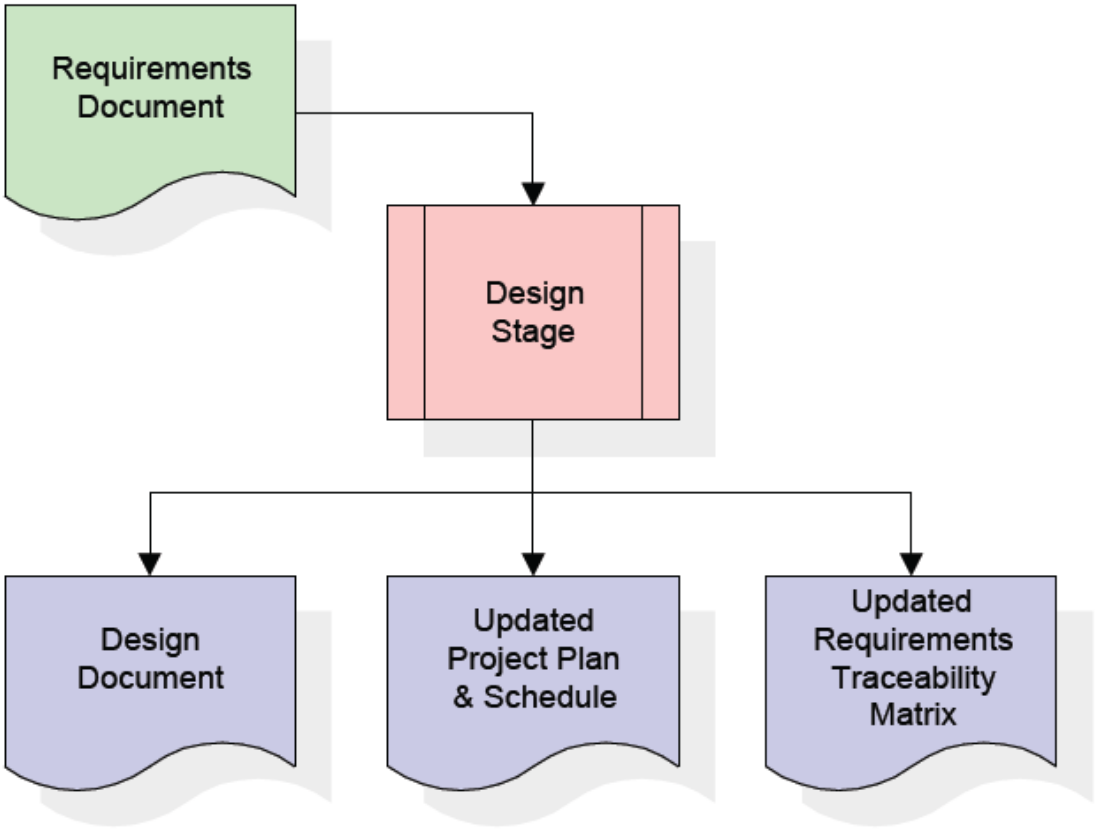


Fig 3: Design Stage

When the design document is finalized and accepted, the RTM is updated to show that each design element is formally associated with a specific requirement. The

outputs of the design stage are the design document, an updated RTM, and an updated project plan.

Development Stage:

The development stage takes as its primary input the design elements described in the approved design document. For each design element, a set of one or more software artifacts will be produced. Software artifacts include but are not limited to menus, dialogs, data management forms, data reporting formats, and specialized procedures and functions. Appropriate test cases will be developed for each set of functionally related software artifacts, and an online help system will be developed to guide users in their interactions with the software.

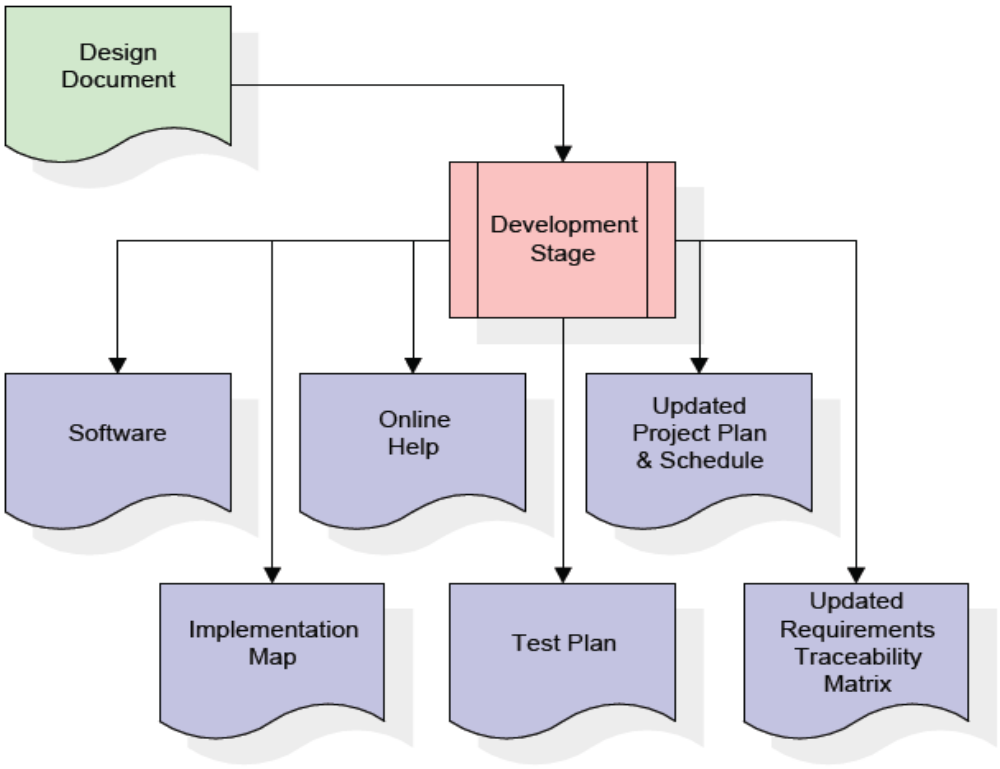


Fig 4: Development Stage

The RTM will be updated to show that each developed artefact is linked to a specific design element, and that each developed artefact has one or more corresponding test case items. At this point, the RTM is in its final configuration. The outputs of the development stage include a fully functional set of software that satisfies the requirements and design elements previously documented, an online help system that

describes the operation of the software, an implementation map that identifies the primary code entry points for all major system functions, a test plan that describes the test cases to be used to validate the correctness and completeness of the software, an updated RTM, and an updated project plan.

Integration & Test Stage:

During the integration and test stage, the software artefacts, online help, and test data are migrated from the development environment to a separate test environment. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite confirms a robust and complete migration capability.

During this stage, reference data is finalized for production use and production users are identified and linked to their appropriate roles. The final reference data (or links to reference data source files) and production user list are compiled into the Production Initiation Plan.

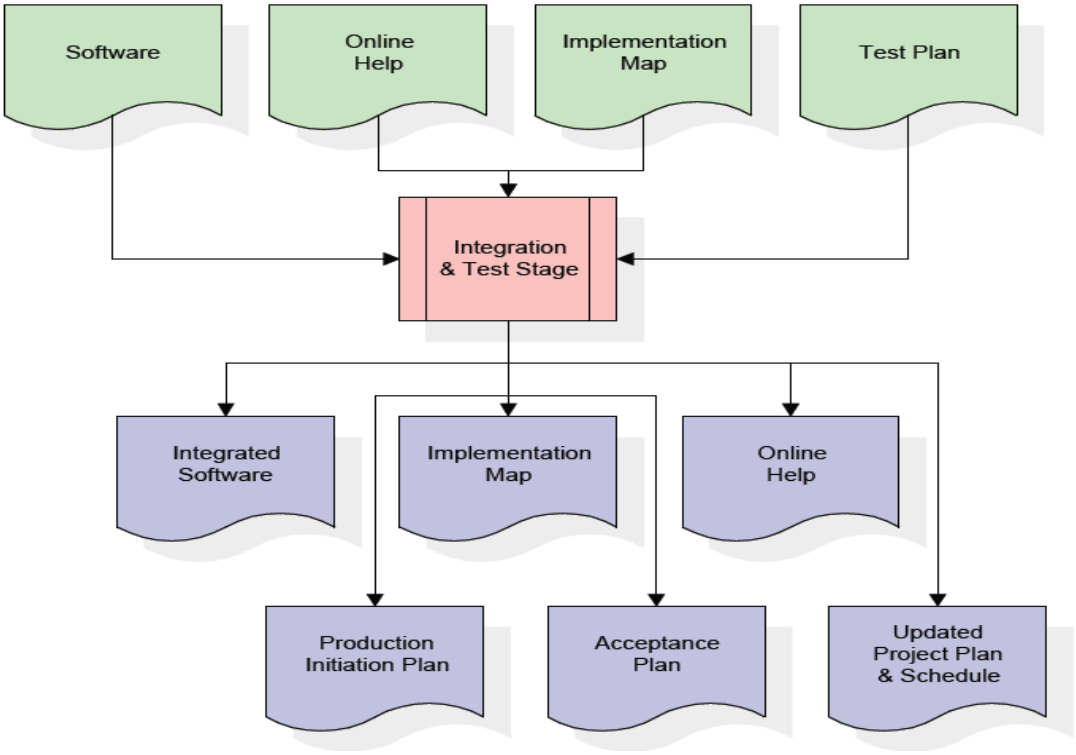


Fig 5: Integration and Test stage

The outputs of the integration and test stage include an integrated set of software, an online help system, an implementation map, a production initiation plan

that describes reference data and production users, an acceptance plan which contains the final suite of test cases, and an updated project plan.

Installation & Acceptance Stage:

During the installation and acceptance stage, the software artifacts, online help, and initial production data are loaded onto the production server. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite is a prerequisite to acceptance of the software by the customer.

After customer personnel have verified that the initial production data load is correct and the test suite has been executed with satisfactory results, the customer formally accepts the delivery of the software.

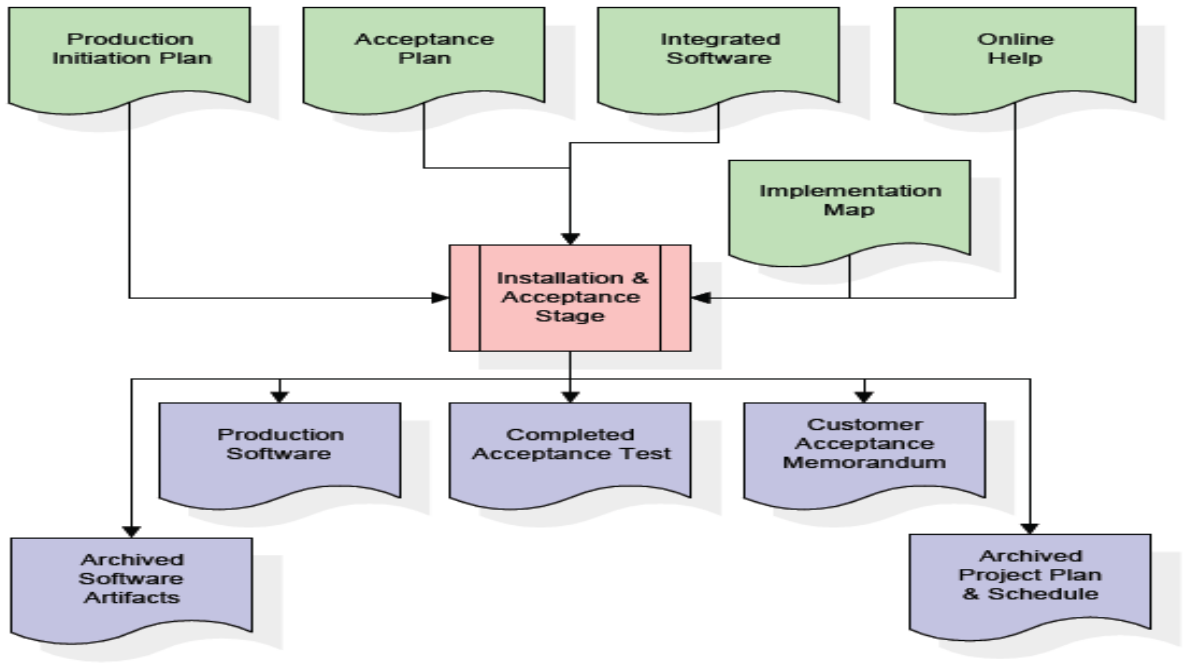


Fig 6: Installation and Acceptance Stage

The primary outputs of the installation and acceptance stage include a production application, a completed acceptance test suite, and a memorandum of customer acceptance of the software. Finally, the PDR enters the last of the actual labor data into the project schedule and locks the project as a permanent project record. At this point the PDR "locks" the project by archiving all software items, the implementation map, the source code, and the documentation for future reference.

SYSTEM ARCHITECTURE

Architecture Flow:

Below architecture diagram represents mainly flow of request from the users to database through servers. In this scenario overall system is designed in three tiers separately using three layers called presentation layer, business layer, data link layer. This project was developed using 3-tier architecture.

3-Tier Architecture:

The three-tier software architecture (three-layer architecture) emerged in the 1990s to overcome the limitations of the two-tier architecture. The third tier (middle tier server) is between the user interface (client) and the data management (server) components. This middle tier provides process management where business logic and rules are executed and can accommodate hundreds of users (as compared to only 100 users with the two-tier architecture) by providing functions such as queuing, application execution, and database staging.

The three-tier architecture is used when an effective distributed client/server design is needed that provides (when compared to the two tier) increased performance, flexibility, maintainability, reusability, and scalability, while hiding the complexity of distributed processing from the user. These characteristics have made three-layer architectures a popular choice for Internet applications and net-centric information systems.

Advantages of Three-Tier:

- Separates functionality from presentation.
- Clear separation - better understanding.
- Changes limited to well define components.

2.LITERATURE REVIEW

Simon Hoermann [1]

discuss this proof for the practicability and effectiveness of online one-on-one psychological state interventions that use text-based synchronous chat. Synchronous written conversations are getting well-liked as Web-based psychological state interventions. This review is predicated on associate analysis of individual synchronous Web-based chat technologies. Several of the prevailing systems have live chats through texts and a few limitations like there's no instant response given to the patients they need to attend for consultant's acknowledgement for an extended time. A number of the processes could charge quantity to measure chat or telecom communication. However, the difficulty of those technologies is cost effective in clinical practice remains a thought for future analysis studies.

Saurav Kumar Mishra [2]

says that the chatbot will act as a virtual doctor and makes possible for the patient to interact with virtual doctor. Natural language processing and pattern matching algorithm for the development of this chatbot. It is developed using the python Language. Based on the survey given it is found that the no of correct answer given by the chatbot is 80% and incorrect/ambiguous answer given is 20%. From this survey of chatbot and analysis of result suggested that this software can be used for teaching and as a virtual doctor for awareness and primary care.

Divya Madhu [3]

proposed an idea in which the AI can predict the diseases based on the symptoms and give the list of available treatments If a person 's body is analyzed periodically, it is possible to predict any possible problem even before they start to cause any damage to the body. Some Challenges are research and implementation costs, and government regulations for the successful implementation of personalized medicine, they are not mentioned in the paper.

Hameed Ullah Kazi [4]

describes the development of a chatbot for medical students, that is based on the open source AIML based Chatter bean. The AIML based chatbot is customized to convert natural language queries into relevant SQL queries. A total of 97 question samples were collected and then those questions were divided into categories depending on the type of question. According to the number of questions in each category the resultant categories were ranked. Questions were based on quires, where 47% are of posed questions

3.PROBLEM IDENTIFICATION AND OBJECTIVES

Existing System:

Many of the existing systems have live chats through texts and some limitation such as there is no instant response given to the patients, they have to wait for expert's acknowledgement for a long time. Some of the processes may charge amount to live chat or telephony communication. However, the issue of these technologies is cost effective in clinical practice remains a consideration for future research studies. Here the studies are based on to recognize emotions classification using AI methods. The studies train emotions classification models from a lot of labelled data based on RNN, deep learning, convolutional neural network. Linguistic interaction is most important in counselling using NLP and NLG to understand dialogues of users. Here the multi-modal approach is used of emotion-recognition. They have collected corpuses to learn semantic information of words and represent as vector using the word vector, synonym knowledge of lexical are collected. [1] In this paper a voice recognition chat-bot is developed, if the questions are not understood asked to the bot is further processed using the third-party expert-system. The web-bots are created as text-based web-friends, an entertainer for the user. Here they focused on the improved system if the system is not only text-based but also voice-based trained. Here the voice recognition requires a 2-part process of capturing and analysis of an input signal. Server response recognition data retrieval and information output. The server used here is SOAP based on black box approach. The use of expert system allows unlimited and autonomous intelligence improvements. [2] This chatbot aims to make a conversation between human and machine. Here the system stores the knowledge database to identify the sentence and making a decision to answer the question. The input sentence will get the similarity score of input sentences using bigram. The chatbot knowledge is stored in RDBMS. [3] The chatbot implemented using pattern comparison in which the order of the sentence is recognized and saved response pattern. Here the author describes the implementation of the chatbot Operating system,

software programming language, and database. How results input and output is stored. Here the input is taken using text () function and other punctuation is removed using trim () function and random () function is used to choose a response from the database. The chatbot is used for an entertainment purpose. [4] Here they use n-gram technique for extracting the words from the sentences. Here n-gram is used for comparison and deduction of the input with case data using Moro phonemes and phonemes as the deciding parameter. Probability analysis for the closest match is performed. The final expression is redirected through an expert system. [5] The chatbot developed here for healthcare purposes for the android application. The user sends the text message or voice message using Google API. Here the user gets only related answer from the chatbot. SVM algorithm is used to classify the dataset. Here the Porter algorithm is used to discard unwanted words like suffixes or prefixes. [6] The different documents served in web, the content is checked by tagging the dataset using n-gram based low dimensional demonstration, TF-IDF matrix that generates S, U, and V and finally multiplying the 3 matrices cosine similarity is calculated. [7] Here the chatbot is created for the customer service that functions as public health service. The application uses N- gram, TF-IDF and cosine similarity. The knowledge base is created for storing the question and answer. The application clearly shows extracted the keyword from the question ad by using unigram, bigram, and trigram which helps in fast answering. [8]

Disadvantages in Existing system

It takes more time to response to the user question

Pay some charges to perform live chat

.

PROPOSED SYSTEM

In our proposed system the user can chat with the bot regarding the query through voice or text. The system uses an expert system to answer the queries. User can also view the available doctors for that particular disease. This system can be used by the multiple users to get the counselling sessions online. The data of the chatbot stored in the database in the form of pattern-template. Bot will provide analgesics and food suggestions that means which food you have to take based on the disease.

Advantages in proposed

system Reducing health care cost

Save the user time•

METHODOLOGY

This section discusses existing modern NLP techniques that can be used when developing a chatbot. Recently, NLP techniques have been combined with ML, as ML improves the chatbots' performance of finding patterns from large amounts of data. The following sections discuss the current stages of NLP and ML techniques applied in the field of chatbot model development. A. Data Preprocessing an arbitrary user input in human language needs to be processed to be "clean data," which is the data a chatbot machine can understand. There are many preprocessing methods currently used in chatbots, such as stop words removal, removing capitalization, and labelling. There are three features to consider when preprocessing the data: Lexical. The lexical feature is also called the "word form" feature because it focuses on each word rather than the sentence structure or grammar [46]. It uses the bag of n-grams method to group words together [47]. Three preprocessing methods using this lexical approach are word level n-grams, stemming, and lemmazation [44]. The word level n-grams method groups together in consecutive words and looks for word n-grams which might indicate the category of the question (for example, if the unigram "city" is used, the question is likely asking for a location). Stemming reduces words to their grammatical roots by removing suffixes [47]. This, however, fails when words change endings in plural form (for example, leaf and leaves would have different stems). Lemmazation is a more accurate approach which can identify the correct roots by referring to a lexical database of English [48]. Syntactic. Methods using syntactic principles include part-of-speech (POS) tagging and chunking. POS tagging [46] labels each word in a sentence with its part of speech (noun, pronoun, verb, etc.). Chunking is then used to partition the sentence into non-overlapping, non-recursive segments. Each partition has a chunk tag, which is its class label. The question classifier model then uses the POS tags, the surrounding context, and the class label to identify the question type [49]. Semantic. Whereas syntax focuses on sentence structure, semantics focuses on the meaning of words. One method using semantic

principles is Named Entity Recognition (NER) [50]. Most NER implementations use a coarse-grained hierarchical classifier consisting of a layered semantic hierarchy of answer types [51]. One paper designed an open-domain question answering chatbot using a two-layered hierarchy containing 6 coarse classes and 50 fine classes to answer 500 questions in the TREC competition. They classified user questions into different question types (with 98.8% accuracy), generated expected answer types, extracted keywords, and reformulated questions into semantically equivalent questions [51]. Vector Representations. Vector representation maps high dimensional word features to low dimensional feature vectors. Based on certain rules and relationships, words are represented by vector coordinates. For example, related words are closer together. The common techniques for vector representation are:

- word2vec [29]: Each word is represented by a vector in a specified vector space containing continuous bag-of-word (CBOW) and skip-gram (SG) architectures.
- doc2vec [52]: The vector representation for paragraphs and documents are found by taking the weighted average of all the words in the document.
- Global Vectors (GloVe) [6]: The global corpus statistics for the unsupervised learning of word representations which outperform other models on word analogy, word similarity, and named entity recognition (NER). The source code from Stanford can be found at http://nlp.stanford.edu/projects/glove/

B. Retrieval-Based A group of researchers used the information retrieval technique to tackle one of the difficult problems for chatbots: the short text conversation. By collecting short conversations on social media and using them to train different models, such as the translation model, latent space model (linear model), deep learning model (nonlinear model), and topic-word model, they reported that the retrieval-based model can perform more “intelligently” than some of the older approaches. This model collects data from many social media sources, such as Q&A forums, and find the differences between user inputs and questions online with the cosine similarities method [53]. Another research group used unstructured documents and examined their features on different levels, such as word level, phrase level, sentence level, document level, relation level, type level, and topic level. This

allowed them to respond to utterances in addition to question response (Q-R) pairs in which the response R is a short text and only depends on the last user utterance Q. Their method selects a sentence from given documents directly, by ranking all possible sentences based on features designed at different levels of granularity. They compared their chatbot's performance with a chitchat engine from China called XiaoIce, and found that their chatbot generated more formal and informative responses, whereas XiaoIce generated more colloquial responses. They also found that their chatbot generated either an equally relevant or more relevant response than XiaoIce in 109 out of 156 conversations, which is promising [41].

In various industries, chatbots are becoming a ubiquitous component of customer service. The usages of chatbots in different fields are summarized in Table III below. They are used in Customer Relationship Management (CRM) which helps companies stay connected to both current and potential customers for increased customer retention [56]. Both commercial and non-profit companies can improve their profitability if they understand their users' needs better. Most conversations are held on text-based platforms like email and online chat. An important variant on these conversational machines is the ability to think. It is why industries are moving towards a modern chatbot which uses AI technology to interact with a human more intelligently. In past years, most chatbots in the industries could only perform simple tasks because they are programmed to respond to a predefined list of questions. In order to become self-learning chatbots, which is what they may do in the future, they need to be trained using data from their past conversations and update its knowledge base autonomously to deliver personalized responses [57], [58].

FEASIBILITY STUDY

Feasibility study is conducted once the problem is clearly understood. The feasibility study which is a high-level capsule version of the entire system analysis and design process. The objective is to determine whether the proposed system is feasible or not and it helps us to the minimum expense of how to solve the problem and to determine, if the Problem is worth solving. The following are the three important tests that have been carried out for feasibility study.

This study talks about how this package is useful to the users and its advantages and disadvantages, and also it tells whether this package is cost effective are not. There are three types of feasibility study, they are

- Economic Feasibility.
- Technical Feasibility.
- Operational Feasibility.

3.1 TECHNICAL FEASIBILITY

Evaluating the technical feasibility is the trickiest part of a feasibility study. This is because, at this point in time, not too many detailed designs of the system, making it difficult to access issues like performance, costs on (on account of the kind of technology to be deployed) etc. A number of issues have to be considered while doing a technical analysis. Understand the different technologies involved in the proposed system before commencing the project we have to be very clear about what are the technologies that are to be required for the development of the new system. Find out whether the organization currently possesses the required technologies. Is the required technology available with the organization?

3.2 OPERATIONAL FEASIBILITY

Proposed project is beneficial only if it can be turned into information systems that will meet the organizations operating requirements. Simply stated, this test of feasibility asks if the system will work when it is developed and installed. Are there major barriers to Implementation? Here are questions that will help test the operational feasibility of a project:

- Is there sufficient support for the project from management from users? If the current system is well liked and used to the extent that persons will not be able to see reasons for change, there may be resistance.

- Are the current business methods acceptable to the user? If they are not, Users may welcome a change that will bring about a more operational and useful systems.
- Are there major barriers to Implementation? Here are questions that will help test the operational feasibility of a project
- Has the user been involved in the planning and development of the project?
- Since the proposed system was to help reduce the hardships encountered. In the existing manual system, the new system was considered to be operational feasible.

3.3 ECONOMIC FEASIBILITY

Economic feasibility attempts 2 weigh the costs of developing and implementing a new system, against the benefits that would accrue from having the new system in place. This feasibility study gives the top management the economic justification for the new system. A simple economic analysis which gives the actual comparison of costs and benefits are much more meaningful in this case. In addition, this proves to be a useful point of reference to compare actual costs as the project progresses. There could be various types of intangible benefits on account of automation. These could include increased customer satisfaction, improvement in product quality better decision-making timeliness of information, expediting activities, improved accuracy of operations, better documentation and record keeping, faster retrieval of information, better employee morale.

4.SYSTEM DESIGN

UML Diagrams

UML (Unified Modeling Language) is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. UML was created by the Object Management Group (OMG) and UML 1.0 specification draft was proposed to the OMG in January 1997. It was initially started to capture the behavior of complex software and non-software system and now it has become an OMG standard. This tutorial gives a complete understanding on UML.

UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems.
UML was created by the Object Management Group (OMG) and UML 1.0 specification draft was proposed to the OMG in January 1997.
OMG is continuously making efforts to create a truly industry standard.

- UML stands for Unified Modeling Language.
- UML is different from the other common programming languages such as C++, Java, COBOL, etc.
- UML is a pictorial language used to make software blueprints.
- UML can be described as a general-purpose visual modeling language to visualize, specify, construct, and document software system.
- Although UML is generally used to model software systems, it is not limited within this boundary. It is also used to model non-software systems as well. For example, the process flow in a manufacturing unit, etc.

UML is not a programming language but tools can be used to generate code in various languages using UML diagrams. UML has a direct relation with object- oriented analysis and design. After some standardization, UML has become an OMG standard.

Components of the UML

UML diagrams are the ultimate output of the entire discussion. All the elements, relationships are used to make a complete UML diagram and the diagram represents a system. The visual effect of the UML diagram is the most important part of the entire process. All the other elements are used to make it complete.

UML includes the following nine diagrams, the details of which are described in the subsequent chapters.

- Class diagram
- Object diagram
- Use case diagram
- Sequence diagram
- Collaboration diagram
- Activity diagram
- State chart diagram
- Deployment diagram
- Component diagram

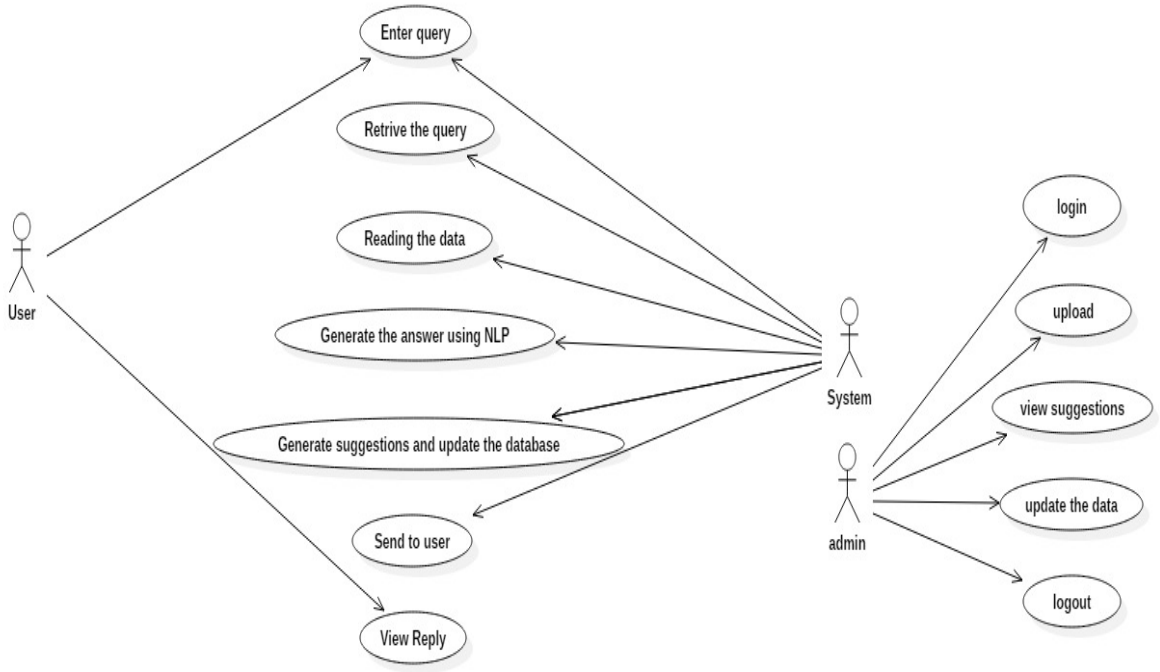
The following are the main components of UML: -

1. Use-case Diagram
2. Class Diagram
3. Sequence Diagram
4. Activity Diagram
5. Collaboration Diagram

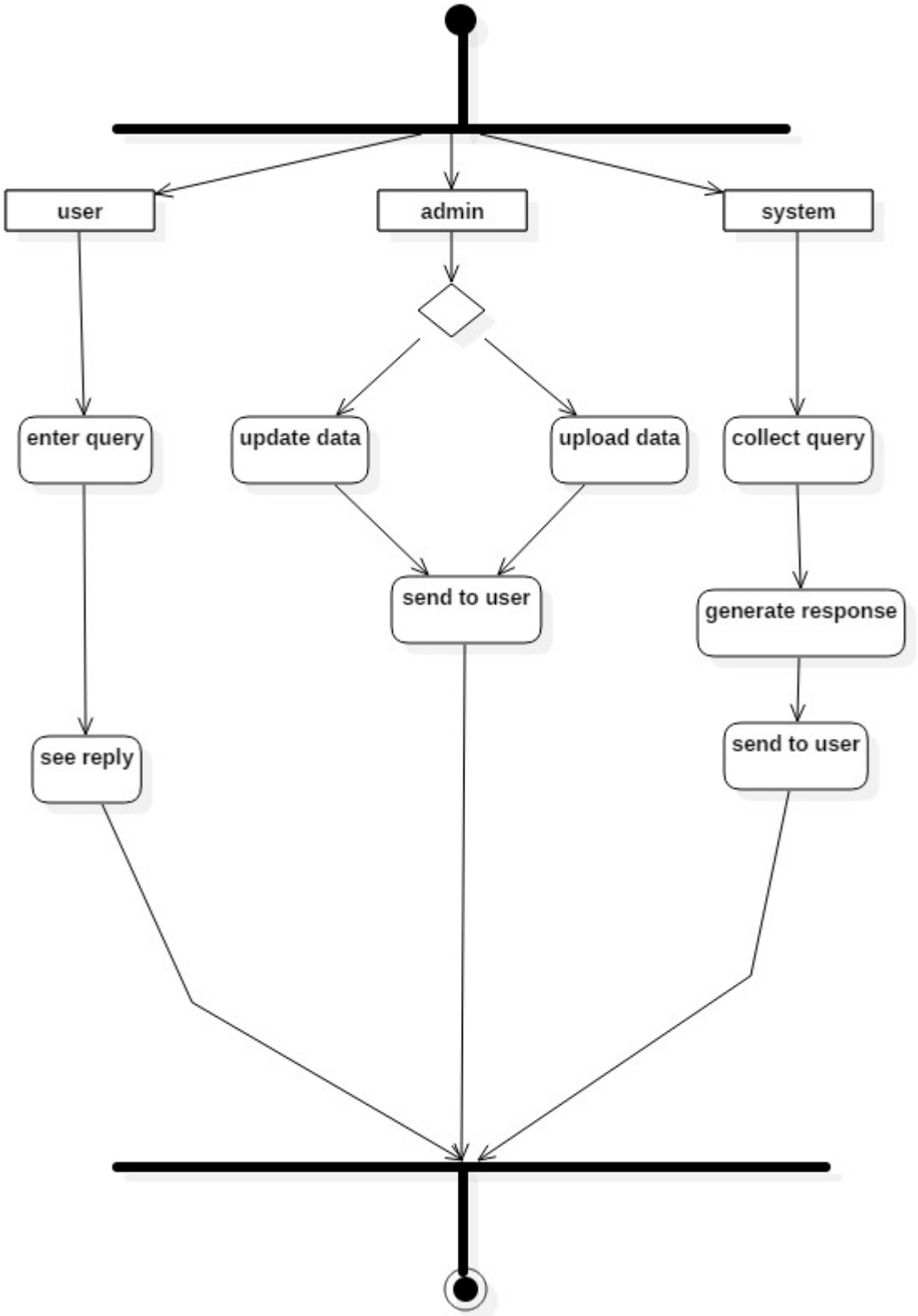
UML DIAGRAMS

USE CASE DIAGRAM

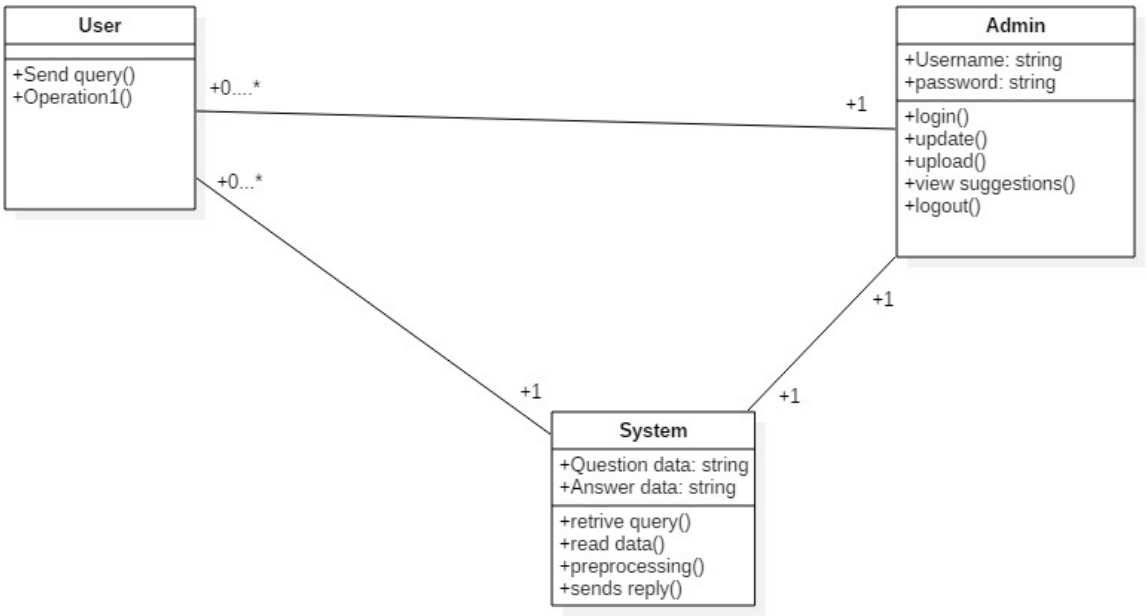
A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



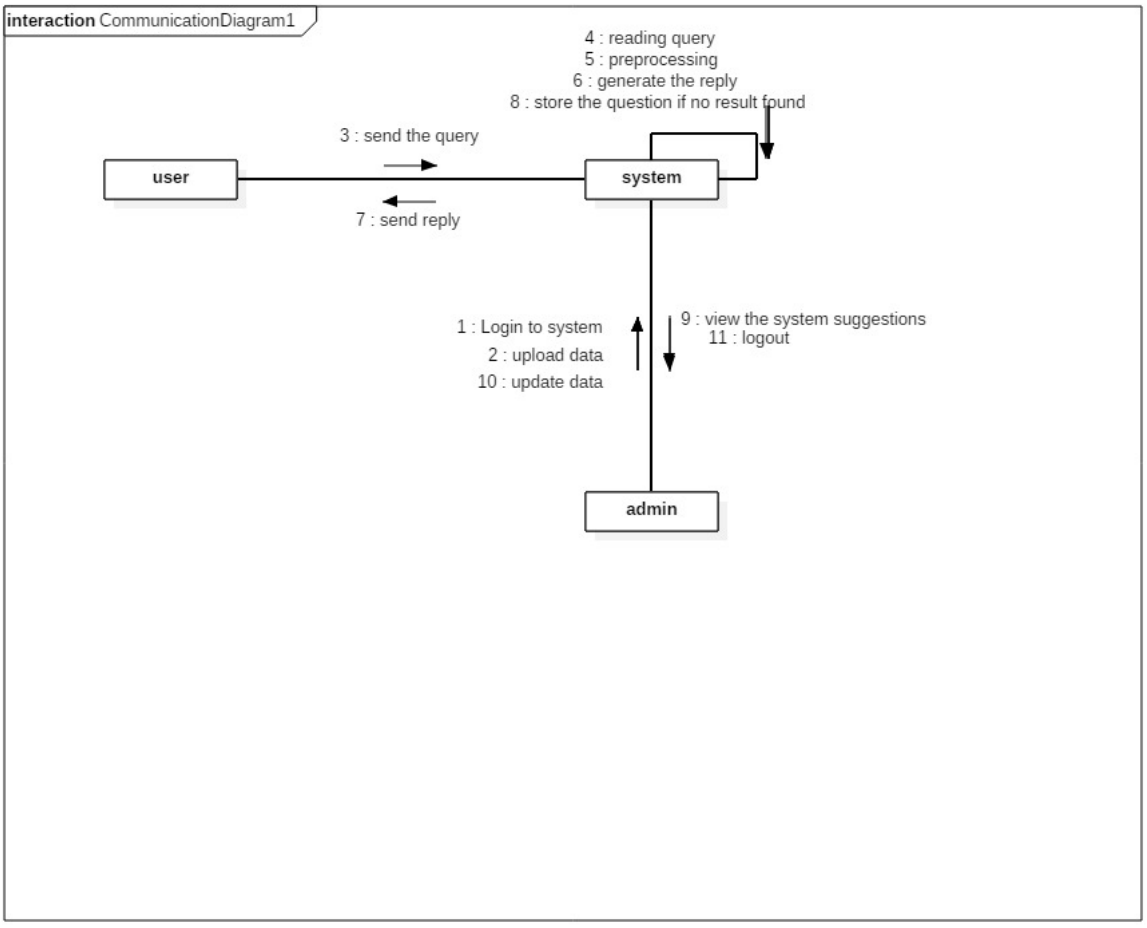
Activity Diagram



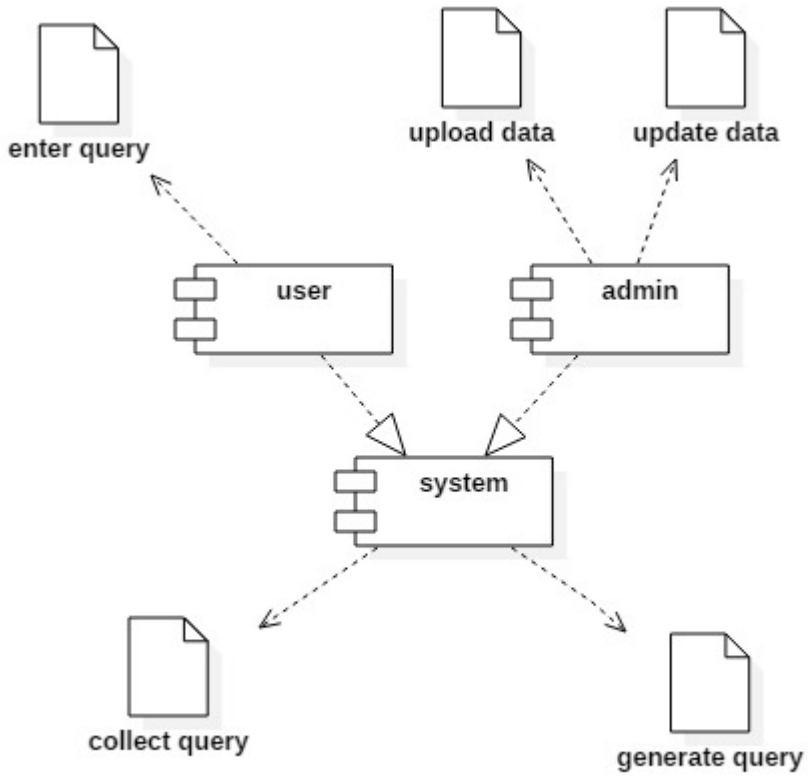
Class diagram



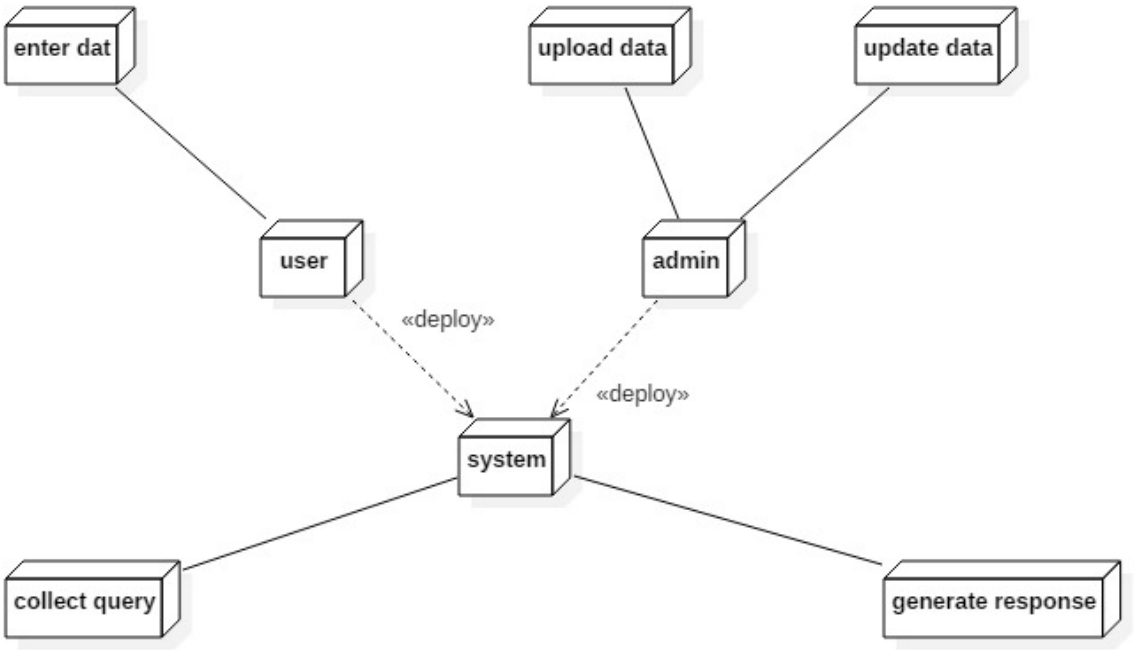
Communication Diagram



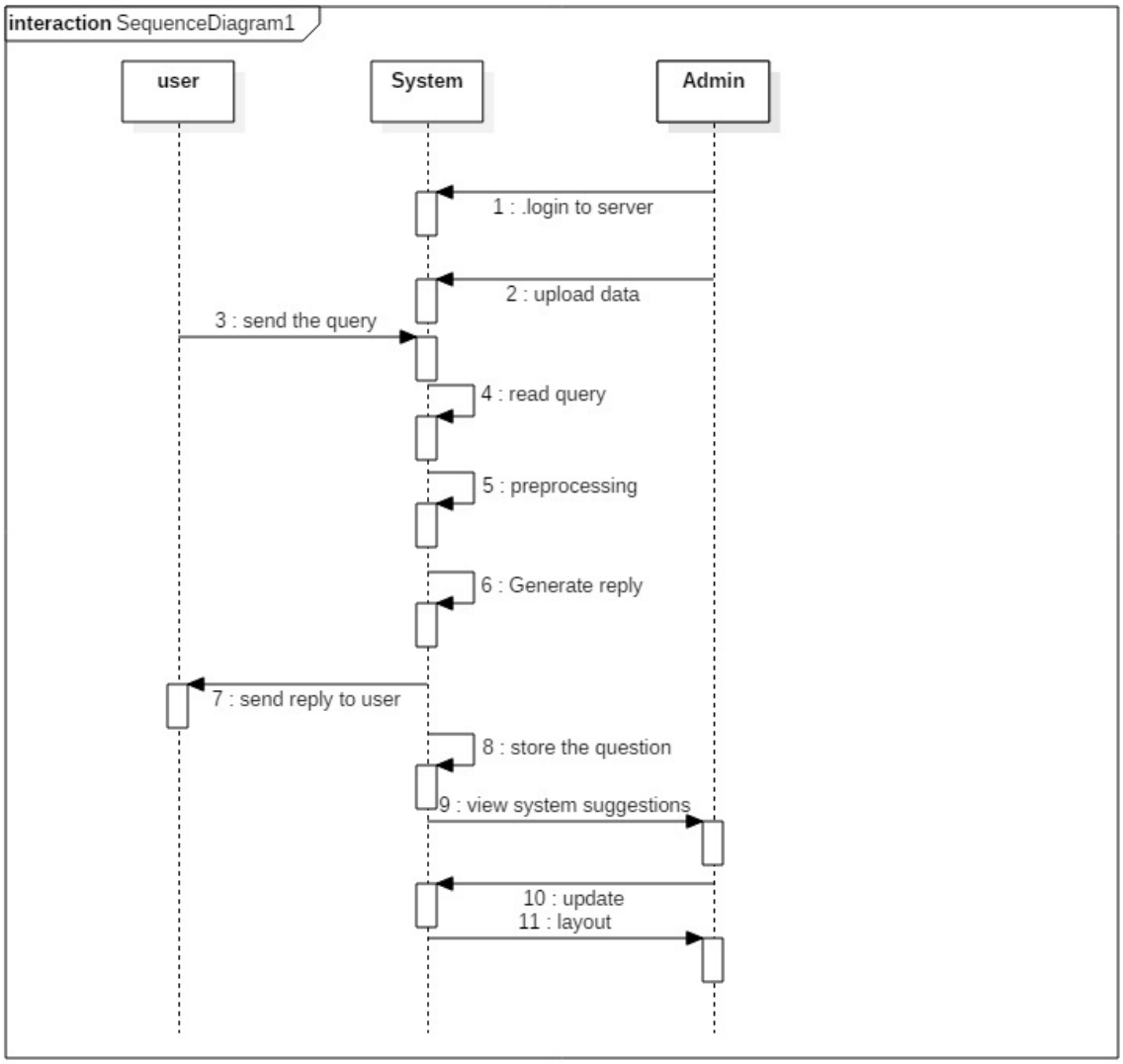
Component Diagram



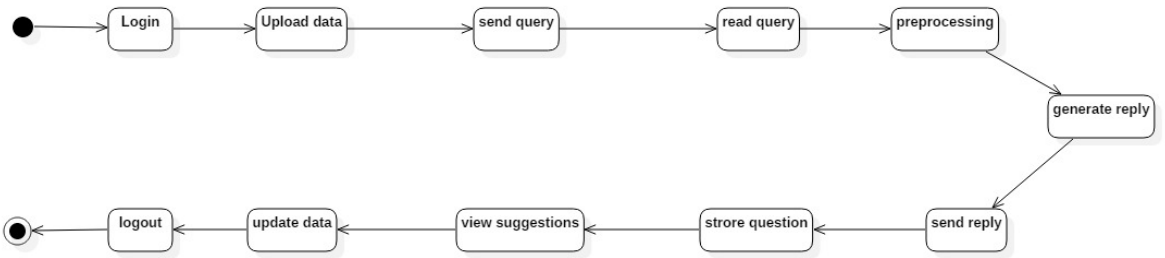
Deployment Diagram



Sequence Diagram State



State Chart Diagram



5.OVERVIEW OF TECHNOLOGIES

ALGORITHM USED

Natural Language Processing (NLP) which can help computers analyze text easily i.e., detect spam emails, autocorrect. We'll see how NLP tasks are carried out for understanding human language.

Natural Language Processing

NLP is a field in machine learning with the ability of a computer to understand, analyze, manipulate, and potentially generate human language.

6.TESTING

SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

TYPES OF TESTS

Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration

testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.
Functional testing is centered on the following items:
Valid Input: identified classes of valid input must be accepted.
Invalid Input: identified classes of invalid input must be rejected.
Functions: identified functions must be exercised.
Output: identified classes of application outputs must be exercised.
Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box. you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

6.1 Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

6.2 Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g., components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

6.3 Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

7.CODING & TESTING

Run.py

```
from flask import Flask, render_template, request
from chatterbot import ChatBot
from chatterbot.trainers import ChatterBotCorpusTrainer
import os

from chatterbot import ChatBot
from chatterbot.trainers import ListTrainer

filenumber=int(os.listdir('saved_conversations')[-1])
filenumber=filenumber+1
file= open('saved_conversations/'+str(filenumber),"w+")
file.write('bot : Hi There! I am a medical chatbot. You can begin
conversation by typing in a message and pressing enter.\n')
file.close()

app = Flask(__name__)


english_bot = ChatBot('Bot',

storage_adapter='chatterbot.storage.SQLStorageAdapter',
             logic_adapters=[
   {
       'import_path': 'chatterbot.logic.BestMatch'
   },

],
trainer='chatterbot.trainers.ListTrainer')
english_bot.set_trainer(ListTrainer)

@app.route("/")
def home():
    return render_template("index.html")

@app.route("/get")
def get_bot_response():
    userText = request.args.get('msg')
    response = str(english_bot.get_response(userText))

    appendfile=os.listdir('saved_conversations')[-1]
```

```
    appendfile= open('saved_conversations/'+str(filenumber),"a")
    appendfile.write('user : '+userText+'\n')
    appendfile.write('bot : '+response+'\n')
    appendfile.close()

    return response


if __name__ == "__main__":
    app.run()
```

Train.py

```
from chatterbot import ChatBot
from chatterbot.trainers import ListTrainer
import os

try:
     os.remove("db.sqlite3")
     print("Old database removed. Training new database")
except:
     print('No database found. Creating new database.')

english_bot = ChatBot('Bot')
english_bot.set_trainer(ListTrainer)
for file in os.listdir('data'):
        print('Training using '+file)
        convData = open('data/' + file).readlines()
        english_bot.train(convData)
        print("Training completed for "+file)
```

8.INSTRUCTIONS

Requirements:

```
# Used by pip to install required python packages
# Usage: pip install -r requirements.txt

Flask==0.12.3
chatterbot==0.8.4
SQLAlchemy==1.1.11
```

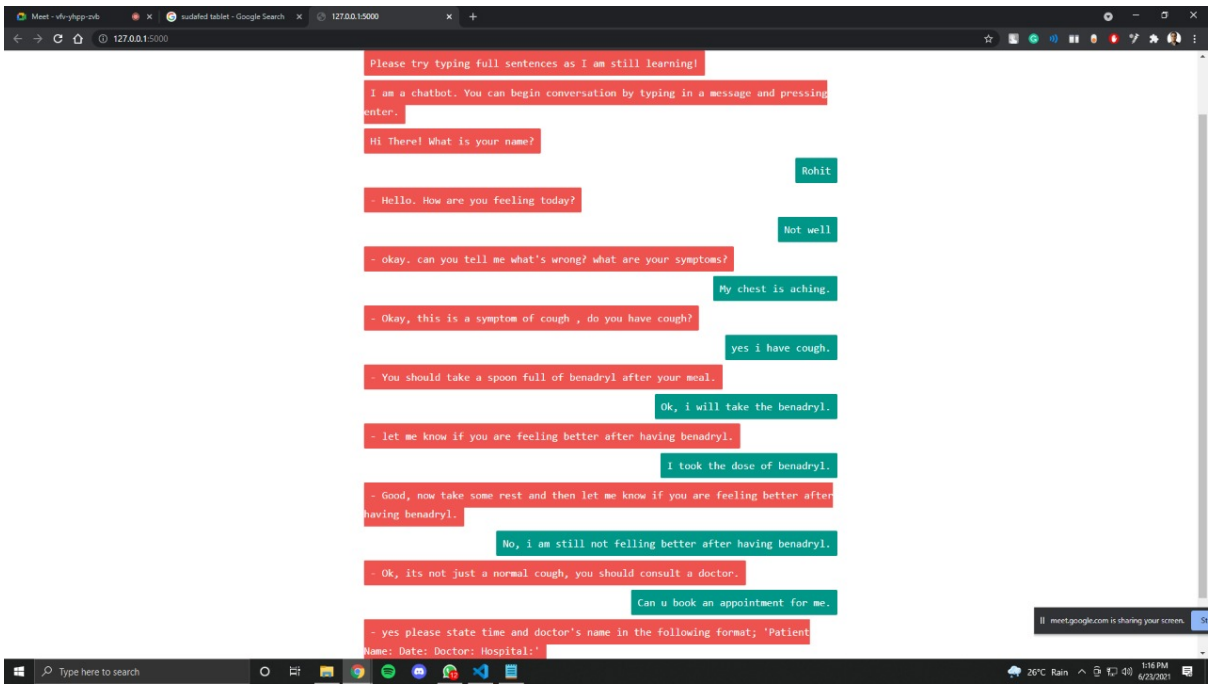
How To Run:

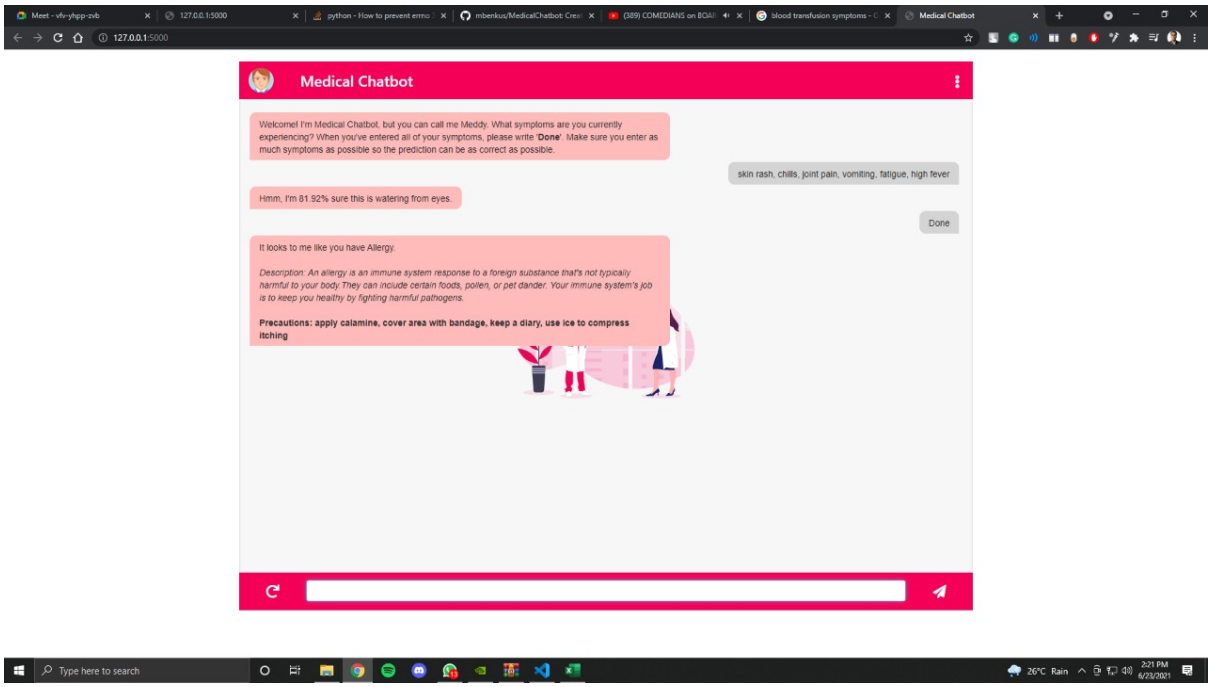
```
flask-chatbot
Built on python 3.6
Flask==0.11
chatterbot==0.8.4
SQLAlchemy==1.1.11

#### A web implementation of
[ChatterBot](https://github.com/gunthercox/ChatterBot) using Flask.

Local Setup:
 1. Open command prompt and locate folder. run 'pip install -r
requirements.txt'
 2. Run *train.py*
 3. Run *run.py*
 4. Demo will be live at http://localhost:5000/
```

9.RESULTS


Please try typing full sentences as I am still learning!
I am a chatbot. You can begin conversation by typing in a message and pressing enter.
Hi There! What is your name?
Rohit
- Hello. How are you feeling today?
Not well
- okay. can you tell me what's wrong? what are your symptoms?
My chest is aching.
- Okay, this is a symptom of cough , do you have cough?
yes i have cough.
- You should take a spoon full of benadryl after your meal.
Ok, i will take the benadryl.
- let me know if you are feeling better after having benadryl.
I took the dose of benadryl.
- Good, now take some rest and then let me know if you are feeling better after having benadryl.
No, i am still not felling better after having benadryl.
- Ok, its not just a normal cough, you should consult a doctor.
Can u book an appointment for me.
- yes please state time and doctor's name in the following format; 'Patient Name: Date: Doctor: Hospital:'


Medical Chatbot
Welcome! I'm Medical Chatbot, but you can call me Meddy. What symptoms are you currently experiencing? When you've entered all of your symptoms, please write 'Done'. Make sure you enter as much symptoms as possible so the prediction can be as correct as possible.
skin rash, chills, joint pain, vomiting, fatigue, high fever
Hmm, I'm 81.92% sure this is watering from eyes.
Done
It looks to me like you have Allergy.
Description: An allergy is an immune system response to a foreign substance that's not typically harmful to your body.They can include certain foods, pollen, or pet dander. Your immune system's job is to keep you healthy by fighting harmful pathogens.
Precautions: apply calamine, cover area with bandage, keep a diary, use ice to compress itching

10.CONCLUSION

Chatbot is great tool for conversation language between human and machine. The application is developed for obtaining a fast response from the bot which implies with none delay it provides the correct result to the user. It's ended that, the usage of chatbot is user friendly and might be utilized by someone who is aware of the way to sort in their own language. Chatbot provides personalised diagnosis supported symptoms. The future era is that the era of messaging app as a result of people spend longer time in messaging app than the other apps. The implementation of personalized drugs would with success save several lives and build a medical awareness among the people. No matter how far people are, they will have this medical voice communication. The sole demand they have easy desktop or smartphone with active web association. The economical of chatbot will be improved by adding a lot of combination of words and increasing the use of database information so of the medical chatbot may handle all type of diseases.

11.References

[1] K. Oh, D. Lee, B. Ko and H. Choi, "A Chatbot for Psychiatric Counseling in Mental Healthcare Service Based on Emotional Dialogue Analysis and Sentence Generation," 2017 18th IEEE International Conference on Mobile Data Management (MDM), Daejeon, 2017, pp. 371-375. doi: 10.1109/MDM.2017.64

[2] Du Preez, S.J. & Lall, Manoj & Sinha, S. (2009). An intelligent web-based voice chat bot. 386 - 391.10.1109/EURCON.2009.5167660

[3] Bayu Setiaji, Ferry Wahyu Wibowo, "Chatbot Using a Knowledge in Database: Human-to-Machine Conversation Modeling", Intelligent Systems Modelling

[4] and Simulation (ISMS) 2016 7th International Conference on, pp. 72-77, 2016.

[5] Dahiya, Menal. (2017). A Tool of Conversation: Chatbot. INTERNATIONAL JOURNAL OF COMPUTER SCIENCES AND ENGINEERING. 5. 158-161.2017.

[6] C.P. Shabariram, V. Srinath, C.S. Indhuja, Vidhya (2017). Ratatta: Chatbot Application Using Expert System, International Journal of Advanced Research in Computer Science and Software Engineering,2017

[7] Mrs Rashmi Dharwadkar1, Dr.Mrs. Neeta A. Deshpande, A Medical ChatBot, International Journal of Computer Trends and Technology (IJCTT) – Volume 60 Issue 1-June 2018

[8] N-gram Accuracy Analysis in the Method of Chatbot Response, International Journal of Engineering & Technology. (2018)