**CHAPTER 1**

**INTRODUCTION**

## 1.1 ABOUT THE PROJECT

In the modern Era of technology, Chatbots are the next big thing in the era of conversational services. Chatbot is a virtual person who can effectively talk to any human being using interactive textual skills.

## The proposed system is designed to automate the process of fixing appointments with doctors conveniently. It takes in information such as the specialist they wish to consult and the date and time of the appointment. The chatbot would process this information and would reply positively if there is a possibility for the appointment to get fixed, otherwise, it provides other options. It also gives each patient a unique token number which the patient must verify during the consultation. This saves valuable customer time and avoids the necessity of a receptionist at the hospital end.

## 1.2 CHATBOTS

A chatbot is a piece of [software](https://en.wikipedia.org/wiki/Software_agent) that conducts a [conversation](https://en.wikipedia.org/wiki/Conversation) via auditory or textual methods. Such programs are often designed to convincingly simulate how a human would behave as a conversational partner, although as of 2019, they are far short of being able to pass the [Turing test](https://en.wikipedia.org/wiki/Turing_test).Chatbots are typically used in [dialog systems](https://en.wikipedia.org/wiki/Dialog_system) for various practical purposes including customer service or information acquisition. Some chatbots use sophisticated [naturallanguage processing](https://en.wikipedia.org/wiki/Natural_language_processing) systems, but many simpler ones scan for keywords within the input, then pull a reply with the most matching keywords, or the most similar wording pattern, from a [database](https://en.wikipedia.org/wiki/Database).

Chatbots are increasingly present in businesses and often are used to automate tasks that do not require skill-based talents. With customer service taking place via messaging apps as well as phone calls, there are growing numbers of use-cases where chatbot deployment gives organisations a clear return on investment. Call centre workers may be particularly at risk from AI-driven chatbots.

**1.3 TELEGRAM BOT**

Telegram is a [cloud](https://en.wikipedia.org/wiki/Cloud_computing)-based [instant messaging](https://en.wikipedia.org/wiki/Instant_messaging) and [voice over IP](https://en.wikipedia.org/wiki/Voice_over_IP) service. Telegram [client](https://en.wikipedia.org/wiki/Client_(computing)) apps are available for [Android](https://en.wikipedia.org/wiki/Android_(operating_system)), [iOS](https://en.wikipedia.org/wiki/IOS), [Windows Phone](https://en.wikipedia.org/wiki/Windows_Phone), [Windows NT](https://en.wikipedia.org/wiki/Windows_NT), [macOS](https://en.wikipedia.org/wiki/MacOS) and [Linux](https://en.wikipedia.org/wiki/Linux). Users can send messages and exchange photos, videos, [stickers](https://en.wikipedia.org/wiki/Sticker_(messaging)), audio and files of any type. Default messages and media in Telegram are encrypted when stored on its servers, but can be accessed by the Telegram service provider, who holds the [encryption keys](https://en.wikipedia.org/wiki/Encryption_key). In addition Telegram provides optional [end-to-end encrypted](https://en.wikipedia.org/wiki/End-to-end_encryption) "secret" chats between two online users, yet not for groups or channels.

A bot is created in telegram using the BOTFATHER. The botfather is one bot to handle all the bots. Botfather takes in the details and provides a token for the new bot created. The token and the URL assigned are unique to the bot created and will be used to program the bot.

**1.4 EXISTING SYSTEM**

The existing system that hospitals use to book appointments is by employing a receptionist who manages the appointment requests by verifying all the booked appointments and provides a token for the available slots. In case of cancelling appointments, the task becomes tedious as the person must identify the free slots and accommodate new requests in them.

**1.5 PROBLEM STATEMENT**

The appointment booking process in any firm involves the presence of a personal assistant or a receptionist. This process is often carried out either by meeting the concerned staff in person or contacting them through a telephone call. This situation can be improvised by automating this process through the creation of a chatbot.

**1.6 CHAPTER OVERVIEW**

The project report is organized with various chapters that denote the various functionalities and aspects of the system being developed.

**Chapter 1** gives a general description about the project. It represents the basic idea of the project and introduces the topics of the existing system and proposed system.

**Chapter 2** deals with the related works of the project. A literature review for each related work is explained in detail.

**Chapter 3**presents the system architecture and requirements. It specifies the hardware and software components that are required. It also lists the technologies used in the implementation of the project.

**Chapter 4**explains the system design with the use of UML diagrams and the data flow diagrams.

**Chapter 5** contributes a detailed description of different modules that are there in the design and how they are implemented.

**Chapter 6** gives a detailed description of the different test cases that were performed on the system.

**Chapter 7**provides the conclusion. It also elucidates how the project can be further enhanced.

**CHAPTER 2**

**LITERATURE SURVEY**

**2.1 CHATBOT AND BULLYFREE CHAT**

The paper was referred from the IEEE Xplore and was published in IEEE International Conference on System, Computation and Automation in the year 2019 (29-30 March’2019). In this paper the concept behind cyberbullying is used to detect the bully words and the project is mainly based on text. The chatbot is created using python and trained using Machine Learning. The system showed better performance than much of the existing systems. In this technology world, a recent technology called chatbot which have been in demand and usage for every business purpose and have hit the market.

Chatbots is an interaction between person and bot which gives us a efficient service and it also gives the way to develop customer engagement and efficiency by reduction of cost by using these service. It can be accessible at anytime,which can handle capacity that is chatbot can chat with thousands of people at a time,It has a flexible attribute as well as customer satisfaction. A chatbot is constructed using natural language processing with the help of machine learning algorithm for training the bot and to make up the bot to perform in a right way and so training and testing is done using ML.This paper gives an overview of chatbot and challenges we faced behind the chatbot with extra features of images.

# **2**.**2 DESIGN OF E-COMMERCE CHAT ROBOT FOR AUTOMATICALLY ANSWERING CUSTOMER QUESTION**

The paper was referred from the IEEE Xplore and was published in IEEE International Conference on Computing, Engineering and Design (ICCED)in the year 2019 (15th April’2019). This research was carried out by building a remote control device using ESP8266 to access local control with Artificial Intelligence Chatbot by using Telegram Messenger. This was done to make it easy for the employees to control devices on their workspace through smartphones or Personal Computers. In order to provide an excellent service, the seller in e-commerce world is required to actively involves in communication with its customer. Nevertheless, in several condition e.g. in vacation or during a rest, the seller might be unable to communicate with his/her customer.

While employing a customer service can be a partial solution, it may involves additional costs for paying customer service persons. In this paper, we propose the design and implementation of e-commerce chatbot system which provides an automatic response to the incoming customer-to-seller question. In general, the proposed system consists of two main agents : communication and intelligent part. In order to get the question message sent by the customer, the communication agent periodically performs a request to Telegram server using a standard HTTP protocol. Upon reception, it forwards that question to intelligent agent which then find the closest instance in predefined question-answer corpora. Notice that, we utilize the Levenstein distance to measure the difference between a submitted question with that of in predefined question-answer corpora. Once an closest instance is selected, the intelligent agent forward the answer to communication agent which then send the answer back to the sender through Telegram chat service. From usability and performance testing result, the proposed system can deliver the automatic answer in less than 5 seconds with relatively good matching accuracy.

**2.3 A CONVERSATIONAL SMART HOME ASSISSTANT BUILT ON TELEGRAM AND GOOGLE DIALOG FLOW**

The paper was referred from the IEEE Xplore and was published in IEEE TENCON 2019, Technology, Knowlegde and Society in the year 2019 (12th December’2019). In this project, a chatbot application is created using a very user-friendly interface “Telegram” along with other technologies like Natural Language Processing, Machine Learning. The system can aid the user in smart decision making, predictive and preventive analysis. In this work, the domain of home automation is considered from the area of the Internet of Things, and a Chatbot application built using technologies like Natural Language Processing, Machine Learning, and Service-Oriented Computing is designed as an intuitive user-interface for Smart home products. The aim of this paper is to build easy to implement and integrate DIY Smart Home Assistant using available technologies. The proposed Conversational Artificial Intelligence system can aid the user in smart decision making, predictive and preventive analytics, and showed promising results during experimental evaluation.

# **2.4 A PLATFORM FOR HUMAN-CHATBOT INTERACTION USING PYTHON**

As numerous chatbot platforms already exist, there are still some problems in building data-driven system because a huge amount of data is required for its development. Thus, this paper describes various such agents which depend upon natural expressions implemented in Python. Moreover, to provide a better platform, web connectivity is also provided to evaluate the chatbot on a web-based platform which will help in analysing Human-Chatbot interactions.

**CHAPTER 3**

**SYSTEM ARCHITECTURE**

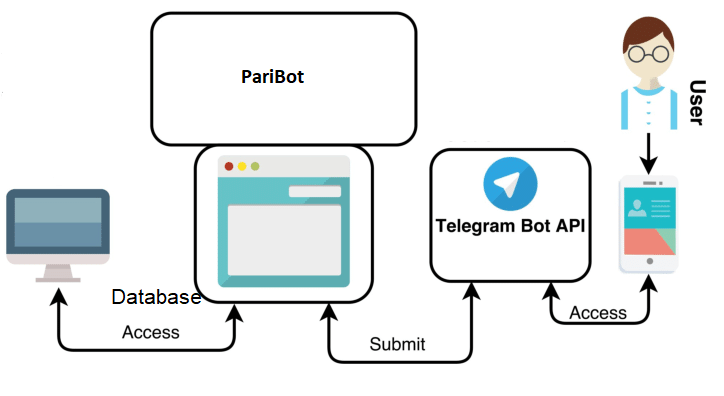
**3.1 PROJECT ARCHITECTURE**

System Architecture defines a comprehensive solution based on principles, concepts, and properties logically related and consistent with each other. The architecture has features, properties, and characteristics satisfying, as far as possible, the problem or opportunity expressed by a set of system requirements and life cycle concepts (e.g., operational, support) and is implementable through technologies (e.g., software, services, procedures, human activity).

The Architecture explains how the raw data set is preprocessed and is used to decide the credibility of the bidder in an online auction. This is hence a basic conceptual picture of what is going on behind the screen and the components that are involved in predicting the accuracy.

**3.2 SYSTEM ARCHITECTURE**

**Figure 3.1 System Architecture**



The user uses the bot through telegram. The user searches for PariBotin Telegram and the bot’s API takes the messages from the user. The telegram server recognizes the change in the “updates” of the bot’s API and invokes the bot’s server to handle the messages. The program running on a server comes up with the response by performing the required steps. The steps include the access of database to verify whether the slot is free and to reply the result.

**3.3 HARDWARE REQUIREMENTS**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **REQUIREMENTS** | **RECOMENDED** | **MINIMUM REQUIREMENTS** |
| 1 | Operating System | Win 10 | Win 10 |
| 2 | RAM | 4 GB | 2 GB |
| 3 | HDD | 1 TB | 1TB |
| 4 | Processor | Intel Quad Core | Intel Dual Core |

The most common set of requirements defined by any [operating system](http://en.wikipedia.org/wiki/Operating_system) or [software application](http://en.wikipedia.org/wiki/Software_application) is the physical computer resources, also known as [hardware](http://en.wikipedia.org/wiki/Computer_hardware). A hardware requirements list is often accompanied by a [hardware compatibility list](http://en.wikipedia.org/wiki/Hardware_compatibility_list) (HCL), especially in case of operating systems. A HCL lists tested, compatible, and sometimes incompatible hardware devices for a particular operating system or application. The following sub-sections discuss the various aspects of hardware requirements.

**Table 3.1 Hardware Requirements**

**3.4 SOFTWARE REQUIREMENTS**

|  |  |
| --- | --- |
| **Requirements** | **Specification** |
| FRONT END | Telegram’s UI |
| DATABASE | SQLite 3 |
| CODING LANGUAGE | Python |

**Table 3.2 Software Requirements**

**3.4.1 TELEGRAM USER INTERFACE**

Telegram android application or web application is used to invoke the chatbot. Telegram is a [cloud](https://en.wikipedia.org/wiki/Cloud_computing)-based [instant messaging](https://en.wikipedia.org/wiki/Instant_messaging) and [voice over IP](https://en.wikipedia.org/wiki/Voice_over_IP) service. Telegram [client](https://en.wikipedia.org/wiki/Client_(computing)) apps are available for [Android](https://en.wikipedia.org/wiki/Android_(operating_system)), [iOS](https://en.wikipedia.org/wiki/IOS), [Windows Phone](https://en.wikipedia.org/wiki/Windows_Phone), [Windows NT](https://en.wikipedia.org/wiki/Windows_NT), [macOS](https://en.wikipedia.org/wiki/MacOS) and [Linux](https://en.wikipedia.org/wiki/Linux).[[15]](https://en.wikipedia.org/wiki/Telegram_(software)#cite_note-teleapps-15) Users can send messages and exchange photos, videos, [stickers](https://en.wikipedia.org/wiki/Sticker_(messaging)), audio and files of any type.

**3.4.2 SQLITE 3**

SQLite is a C-language library that implements a [small](https://www.sqlite.org/footprint.html), [fast](https://www.sqlite.org/fasterthanfs.html), [self-contained](https://www.sqlite.org/selfcontained.html), [high-reliability](https://www.sqlite.org/hirely.html), [full-featured](https://www.sqlite.org/fullsql.html), SQL database engine. SQLite is the [most used](https://www.sqlite.org/mostdeployed.html) database engine in the world. SQLite is built into all mobile phones and most computers and comes bundled inside countless other applications that people use every day. The SQLite [file format](https://www.sqlite.org/fileformat2.html) is stable, cross-platform, and backwards compatible and the developers pledge to keep it that way through at least the year 2050. SQLite database files are commonly used as containers to transfer rich content between systems and as a long-term archival format for data . There are over 1 trillion SQLite databases in active use .SQLite [source code](https://sqlite.org/src) is in the [public-domain](https://www.sqlite.org/copyright.html) and is free to everyone to use for any purpose.Normally, an RDBMS such as MySQL, PostgreSQL, etc., requires a separate server process to operate. The applications that want to access the database server use TCP/IP protocol to send and receive requests. This is called client/server architecture. Because of the serverless architecture, you don’t need to “install” SQLite before using it. There is no server process that needs to be configured, started, and stopped.In addition, SQLite does not use any configuration files.SQLite uses dynamic types for tables. It means you can store any value in any column, regardless of the data type.SQLite allows a single database connection to access multiple database files simultaneously. This brings many nice features like joining tables in different databases or copying data between databases in a single command.SQLite is capable of creating in-memory databases which are very fast to work with.All transactions in SQLite are fully ACID-compliant. It means all queries and changes are Atomic, Consistent, Isolated, and Durable.In other words, all changes within a transaction take place completely or not at all even when an unexpected situation like application crash, power failure, or operating system crash occurs.

**3.4.3 PYTHON**

Python is interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, and a syntax that allows programmers to express concepts in fewer lines of code, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales.Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural.Python Database API supports a wide range of database servers such as −

1. GadFly
2. mSQL
3. MySQL
4. Oracle
5. Sybase

Some of the features of python are listed below:

1. A variety of basic data types are available: numbers (floating point, complex, and unlimited-length long integers), strings (both ASCII and Unicode), lists, and dictionaries.
2. Python supports object-oriented programming with classes and multiple inheritance.
3. Code can be grouped into modules and packages.
4. The language supports raising and catching exceptions, resulting in cleaner error handling.

**Used Python Packages :**

1. **SQLITE3**

The sqlite3 module was written by Gerhard Häring. It provides a SQL interface compliant with the DB-API 2.0 specification described by [**PEP 249**](https://www.python.org/dev/peps/pep-0249).To use the module, you must first create a [**Connection**](https://docs.python.org/2/library/multiprocessing.html#Connection) object that represents the database. You can also supply the special name :memory: to create a database in RAM.Once you have a [**Connection**](https://docs.python.org/2/library/multiprocessing.html#Connection), you can create a [**Cursor**](https://docs.python.org/2/library/sqlite3.html#sqlite3.Cursor) object and call its [**execute()**](https://docs.python.org/2/library/sqlite3.html#sqlite3.Cursor.execute) method to perform SQL commands.

## (ii)URLLIB

[urllib](https://docs.python.org/3.1/library/urllib.request.html#module-urllib.request) is a module built into the Python standard library and uses [http.client](https://docs.python.org/3.1/library/http.client.html) which implements the client side of HTTP and HTTPS protocols. There’s no pip install required because urllib is distributed and installed with Python. If you value stability, this is for you. The [twilio-python helper library](https://www.twilio.com/docs/libraries/python) uses urllib. urllib can require more work than using the libraries built on top of it. For example, you have to create a URL object before making the HTTP request.

**(iii)REQUESTS**

Requests allows you to send HTTP/1.1 requests extremely easily. There’s no need to manually add query strings to your URLs, or to form-encode your PUT and POST method – instead use the JSON method. Requests is ready for the demands of building robust and reliable HTTP–speak applications according to the needs.

**CHAPTER 4**

**SYSTEM MODELLING**

**4.1 UNIFIED MODELING LANGUAGE (UML)**

Unified Modeling Language is a standardized modeling language consisting of an integrated set of diagrams, developed to help system and software developers for specifying, visualizing, constructing, and documenting the artifacts of software systems, as well as for business modeling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems. The UML is a very important part of developing object-oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects. Using the UML helps project teams communicate, explore potential designs, and validate the architectural design of the software.

The primary goals in the design of the UML as follows:

* Provide users with a ready-to-use, expressive visual modeling language so they can develop and exchange meaningful models.
* Provide extensibility and specialization mechanisms to extend the core concepts.
* Be independent of particular programming languages and development processes.
* Provide a formal basis for understanding the modeling language.
* Encourage the growth of the OO tools market.
* Support higher-level development concepts such as collaborations, frameworks, patterns and components.
* Integrate best practices.

**4.2 USE CASE DIAGRAM**

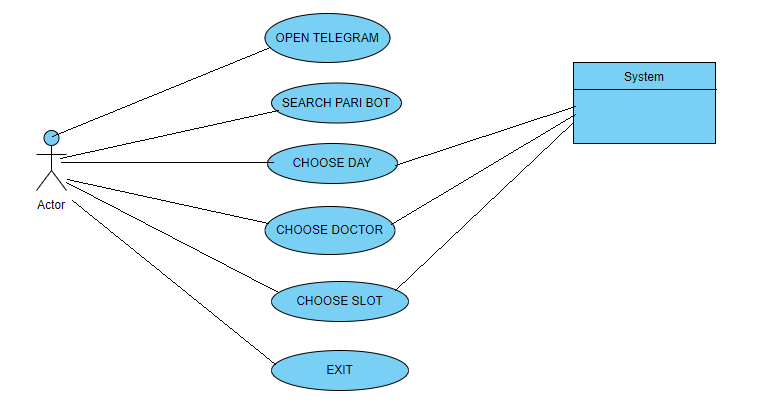
The use case diagram is used to define the core elements and processes that make up a system. The key elements are termed as “actors” and the processes are called “use cases”. The use case diagram shows which actors interacts with each use case. This definition defines what a use case diagram is primarily made up of – actors and use cases.

In software and system engineering, a use case is a list of steps, typically defining interactions between a role (known in UML as an “actor”) and a system, to achieve a goal. The actor can be a human or an external system. In system engineering, use cases are used at a higher level than within software engineering, often representing missions or stakeholder goals.

The purposes of use case diagrams can be as follows:

1. Used to gather requirements of a system.
2. Used to get an outside view of a system.
3. Identify external and internal factors influencing the system.
4. Show the interacting among the requirements are actors.

Use cases help in identifying the operations that can be performed by an actor. actor can be real time human or a system. It helps in identifying the various modules present in the system. A single use case diagram captures a particular functionality of a system. Hence to model the entire system, a number of use case diagrams are used. The actor can be real time human or a system. It helps in identifying the various modules present in the system. A single use case diagram captures a particular functionality of a system. Hence to model the entire system, a number of use case diagrams are used.



**Figure 4.1 Use case Diagram**

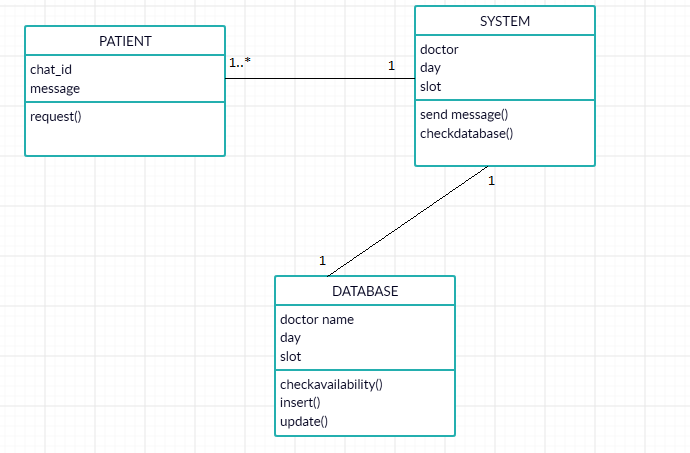
In this system, the patient who needs a doctor appointment is the primary actor, the subordinate actor is the system. The patient is engaged in various use cases such as opening the telegram to search for the Pari Bot. The user interacts with the system to book the appointment by choosing day, the doctor and slot. The system gets these user inputs and replies the user with the appropriate answers.

**4.3 CLASS DIAGRAM**

The class diagram describes the attributes and operations of a class as well as the constraints imposed on the system. The class diagrams are widely used in the modeling of object oriented systems because they are the only UML diagrams which can be mapped directly with the object oriented languages. The purpose of the class diagram is to model the static view of an application. The class diagram shows a collection of classes, interfaces, associations, collaborations and constraints. It is also known as structural diagram.

The purpose of the class diagram can be summarized as:

1. Analysis and design of the static view of an application.
2. Description of the responsibilities of a system.
3. Development the base for the component and deployment diagrams.
4. Forward and reverse engineering.



**Figure 4.2 Class Diagram**

In this system, each class possesses various attributes or privileges and functions. The objects in the project are patient, system and database. Every patient has a chat id and a message as their attributes and the functions performed by them is sending request messages. The attributes of the object system are doctor, day and slot whose values are obtained from the user. The other object database has the attributes, doctor name, day and slot which are the coloumns in the database and the manipulation functions such as insert and update is performed.

**4.4 SEQUENCE DIAGRAM**

A sequence diagram is a kind of interaction diagram that shows how processes operate with one another and in which order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence.

It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. A sequence diagram shows different processes or objects that live simultaneously as parallel vertical lines (lifelines) and,the messages exchanged between them and the order in which they occur as horizontal arrows.

The main purpose of the Sequence diagram is

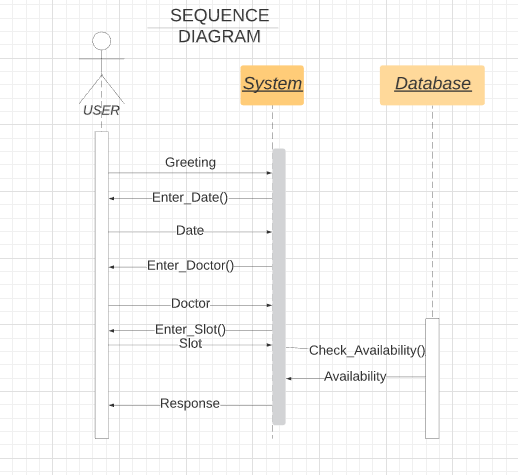
* To capture the dynamic behavior of a system.
* To describe the message flow in the system.
* To describe the structural organization of the objects.
* To describe the interaction among objects.

Sequence diagrams can be used

* To model the flow of control by time sequence.
* To model the flow of control by structural organizations.
* For forward engineering.
* For reverse engineering.

**Figure 4.3 Sequence Diagram**

In this system, the sequence of operations between the various objects are:

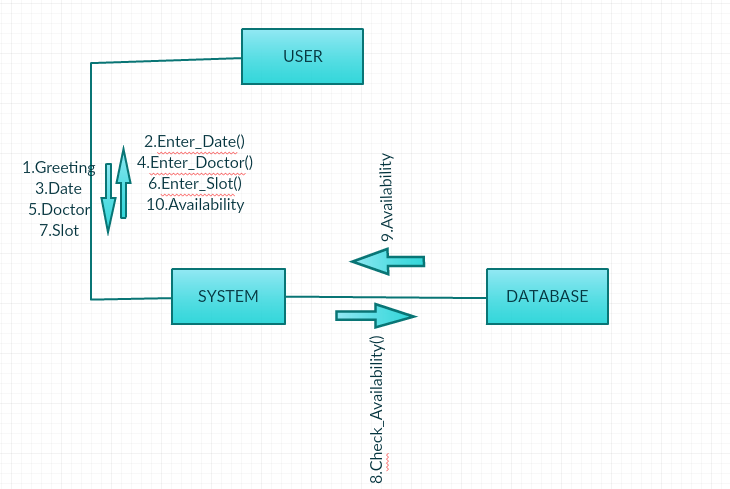


* Firstly, the user sends a greeting message to the system.
* The system prompts the user to enter a date and user requests for a date.
* Similarly, the system prompts the user to enter the doctor and the slot in which they wish to fix the appointment.
* Based on these information, the system requests the database to check the availability of the slot
* The database returns the availability status to the system.
* Lastly, the system sends a positive or negative response to the user based on the availability

**4.5 COLLABORATION DIAGRAM**

A collaboration diagram, also called a communication diagram or interaction diagram, is an illustration of the relationships and interactions among objects in the Unified Modeling Language (UML). Collaboration diagrams convey the same information as sequence diagrams, but focus on object roles instead of the timings of messages. It illustrates messages being sent between classes and objects (instances).

Collaboration diagrams represent a combination of information taken from class, sequence and use case diagrams describing both the static structure and dynamic behavior of a system.



**Figure4.4 Collaboration Diagram**

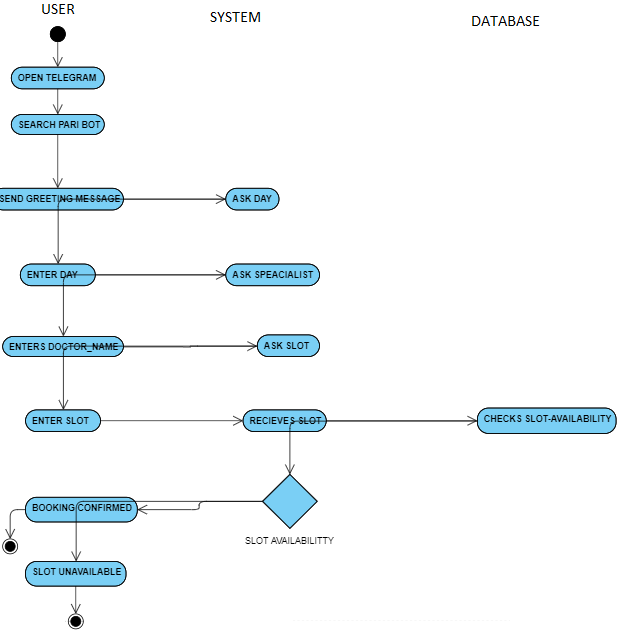
**4.6 ACTIVITY DIAGRAM**

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams are intended to model both computational and organizational processes (i.e., workflows), as well as the data flows intersecting with the related activities. Although activity diagrams primarily show the overall flow of control, they can also include elements showing the flow of data between activities through one or more data stores.

Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc.Activity diagrams are constructed from a limited number of shapes, connected with arrows.

The most important shape types:

* rounded rectangles represent actions;
* diamonds represent decisions;
* bars represent the start (split) or end (join) of concurrent activities;
* an encircled black circle represents the end (final node).
* Black circle represents the initial code.



**Figure 4.5 Activity Diagram**

In this system, the sequence of actions are:

* Firstly, the user sends a greeting message to the system.
* The system prompts the user to enter a date and user requests for a date.
* Similarly, the system prompts the user to enter the doctor and the slot in which they wish to fix the appointment.
* Based on these information, the system requests the database to check the availability of the slot
* The database returns the availability status to the system.
* Lastly, the system sends a positive or negative response to the user based on the availability

**4.7 STATE CHART DIAGRAM**

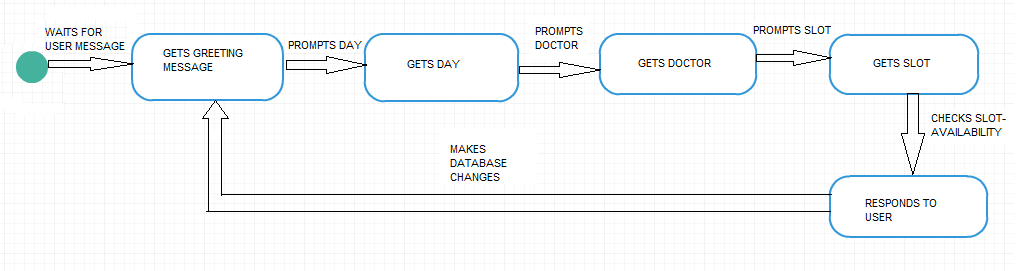
A State chart diagram describes a state machine. State machine can be defined as a machine which defines different states of an object and these states are controlled by external or internal events. It describes different states of a component in a system. The states are specific to a component/object of a system.State chart diagram is used to model the dynamic nature of a system. They define different states of an object during its lifetime and these states are changed by events. State chart diagrams are useful to model the reactive systems.

State chart diagram describes the flow of control from one state to another state. States are defined as a condition in which an object exists and it changes when some event is triggered. The most important purpose of State chart diagram is to model lifetime of an object from creation to termination.

The main purposes of using State chart diagrams

* To model the dynamic aspect of a system.
* To model the life time of a reactive system.
* To describe different states of an object during its life time

**Figure 4.6 State Chart Diagram**



In this system, the initial or the starting state is reached after a greeting message is obtained from the user. The system then moves into the subsequent states such as, get day, get doctor and get slot by getting prompt messages from the user. After these states are reached, the last state is reached which is replying to the user, eitherly positively or negatively based on the availability of the slot and the corresponding changes are made in the database and the system goes back to the initial state and the process goes on endlessly.

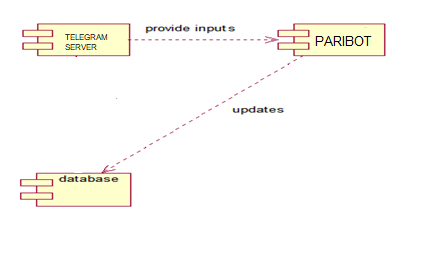
**4.8 COMPONENT DIAGRAM**

In the Unified Modeling Language, a component diagram depicts how components are wired together to form larger components or software systems. They are used to illustrate the structure of arbitrarily complex systems. Component diagrams are different in terms of nature and behavior. Component diagrams are used to model the physical aspects of a system. Physical aspects are the elements such as executables, libraries, files, documents, etc. which reside in a node.

Component diagrams are used to visualize the organization and relationships among components in a system. These diagrams are also used to make executable systems.Component diagram is a special kind of diagram in UML. The purpose is also different from all other diagrams. It does not describe the functionality of the system but it describes the components used to make those functionalities. Component diagrams can also be described as a static implementation view of a system. Static implementation represents the organization of the components at a particular moment.

The purpose of the component diagram can be summarized as

* Visualize the components of a system.
* Construct executables by using forward and reverse engineering.
* Describe the organization and relationships of the components.



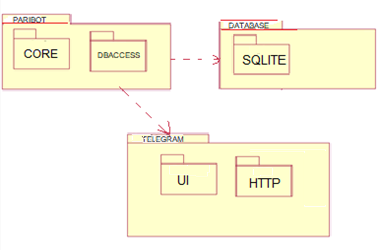
**Figure 4.7 Component Diagram**

In this system, there are three components- Telegram Server, Paribot and the database. The telegram server acts as the web server which gets the input from the user through the Telegram platform. The paribot is implemented using python and the backend, tables for storing the information is done through SQlite3.

**4.9 PACKAGE DIAGRAM**

Package diagram is a UML structure diagram which shows packages and dependencies between the packages. The package diagram shows the arrangement and organization of the model elements in middle to large scale project. The package diagram can show both the structure and dependencies between sub-systems or modules.

A package is rendered as a tabbed folder – a rectangle with a small tab attached to the left side of the top of the rectangle. If the members of the package are not shown inside the package rectangle, then the name of the package should be placed inside. The members of the package may be shown within the boundaries of the package. In this case, the name of the package should be placed on the tab. A diagram showing a package with content can show only a subset of the contained elements according to some criterion. Members of the package may be shown outside of the package by branching lines from the package to the members. A plus sign (+) within a circle is drawn at the end attached to the namespace (package).



**Figure 4.8 Package Diagram**

The package diagram of the system consists of three main packages which represent the system, the telegram server and the database package. The system uses the core module and the Database access package. The core package implements the functionality of the system and also uses the database package to effectively access the SQLite database. The database package implements the SQLite database. The telegram package offers the user interface and the communication platform for the application server.

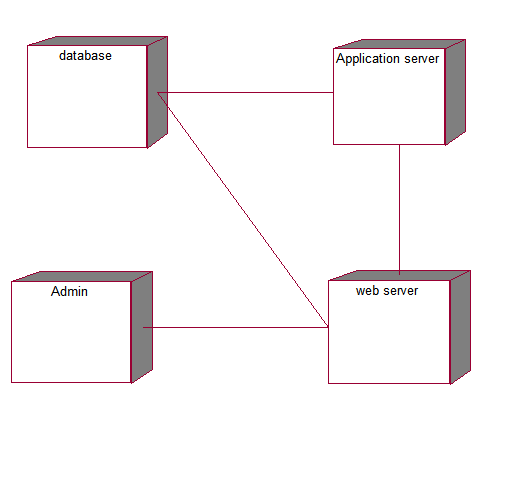
**4.10 DEPLOYMENT DIAGRAM**

A deployment diagram in the Unified Modeling Language models the physical deployment of artifacts on nodes. To describe a website, for example, a deployment diagram would show what hardware components (“nodes”) exist (e.g., a web server, an application server, and a database server), what software components (“artifacts”) run on each node (e.g., web application, database), and how the different pieces are connected (e.g., JDBC, REST, RMI).

The nodes appear as boxes, and the artifacts allocated to each node appear as rectangles within the boxes. Nodes may have sub nodes, which appear as nested boxes. A single node in a deployment diagram may conceptually represent multiple physical nodes, such as a cluster of database servers. Deployment diagrams are used by system engineers.

The purposes of deployment diagrams can be as follows:

1. Visualize the hardware topology of a system.
2. Describe the hardware components used to deploy software components.
3. Describe the runtime processing nodes.



**Figure 4.9 Deployment Diagram**

In this system, there are users who upload the data to the system using the Telegram API. The application serveris used to handle the operations between the users and databases.The web server is used to store and process the collected datasets and to deliver the web pages to the users.

**CHAPTER 5**

**SYSTEM IMPLEMENTATION**

* 1. **PROPOSED SYSTEM**

The proposed system takes information such as the doctor name, day of visit and the slot and books an appointment was developed.The Bot was built on Telegram platform using the chat service provided by Telegram. The Telegram server forwards the updates to the application server and delivers the replies to the user.The application server processes the updates and uses Sqlite database to check the availability and responds accordingly. This method is very user friendly and and an appointment can be easily fixed through direct messages with the system-“the doctor chatbot” without much complications.

**5.2 MODULE DESCRIPTION**

The system here is divided into three modules as listed below:

1. Core module
2. Data module
3. Integration module

**5.2.1 CORE MODULE**

The core module represents the proposed system. This module takes in key information- date, time and the doctor. It then checks the availability of the doctor and blocks the slot. So, this module uses both the data module and the integration module and processes it and come up with the responses. This module uses the *requests, json* and other imported modules for communicating by performing HTTP requests and responses.

**5.2.2DATA MODULE**

This module extracts the key information from the conversation and passes it to the core module. This module also uses the *requests* package to form requests and get responses. The key attributes are the doctor name, the day and the requested slot.

**5.2.3 INTEGRATION MODULE**

This module establishes a connection with an SQLite database and creates a handle to manage all the database access operations. The module creates a SQLite database. The database contains two tables representing today and tomorrow. These tables have the available doctors and the slots for each doctor. The database handle searches this database to check whether the requested slot is free or not.

**CHAPTER 6**

**SYSTEM TESTING**

**6.1 INTRODUCTION**

Testing is the major quality control measure employed during software development. Its basic function is to detect the errors in the software. For this, different levels of testing are used, which performs different tasks on the aim to test different aspects of the system.

System testing is performed on the entire system in the context of a Functional Requirement Specification and/or a System Requirement Specification. System testing tests not only the design, but also the behavior and even the believed expectations of the customer.

**6.2WHITE BOX TESTING**

White-box testing is a method of testing software that tests internal structures or workings of an application, as opposed to its functionality. While white-box testing can be applied at the unit, integration and system levels of the software testing process, it is usually done at the unit level. It can test paths within a unit, paths between units during integration, and between subsystems during a system–level test. Though this method of test design can uncover many errors or problems, it might not detect unimplemented parts of the specification or missing requirements.

**Advantages of White Box Testing**

* Powers test designer to reason precisely about usage.
* Uncovers blunders in "concealed" code.

**6.3BLACK BOX TESTING**

Black-box testing is a method of software testing that examines the functionality of an application without peering into its internal structures or workings. This method of test can be applied to virtually every level of software testing, unit, integration, system and acceptance. It typically comprises all higher-level testing, but can also dominate unit testing as well.

**Advantages of Black Box Testing**

* Tests are done from a client's perspective and will help in uncovering disparities in the details.
* Analyzer require not know programming dialects or how the product has been executed.

**6.4VALIDATION TESTING**

After black box testing is done the errors have been uncovered and corrected and final series of software tests i.e. validation test begins. Validation can be defined in many ways, but simple definition is that validation succeeds if the software is in a manner that is reasonably expected by the client.

Software validation is achieved through a series of black box test that demonstrate conformity with requirements. Both the plan and procedures are designed to ensure that all

* Functional requirements are satisfied
* Performance and requirements are achieved
* Documentation is correct and human engineered

**Validation is determining** if the system complies with the requirements and performs functions for which it is intended and meets the organization’s goals and user needs.While validating the product if some deviation is found in the actual result from the expected result then a bug is reported or an incident is raised.Validation was not done manually within the website. It was provided by client-side web browser. As per browser records, there were no invalidated request that was able to access the data and 99% of the validated users were redirected to the corresponding page within seconds and the remaining 1% where delayed due to their internet connection.

**6.5 MANUAL TESTING**

**Manual Testing** is a process of finding out the defects or bugs in a software program. In this method the tester plays an important role of end user and verifies that all the features of the application are working correctly**. The tester manually executes test cases without using any automation tools.**The tester prepares a test plan document which describes the detailed and systematic approach to testing of software applications. Test cases are planned to cover almost 100% of the software application. As manual testing involves complete test cases it is time consuming.

The differences between actual and desired results are treated as defects. The tester retests the defects to ensure that defects are fixed. The goal of Manual testing is to ensure that application is defect & error free and is working fine to provide good quality work to customers.

Manual Testing is a process in which you compare the behavior of a developed piece of code (software, module, API, feature, etc.) against expected behavior (Requirement). By reading or listening to the requirements and understanding it fully.

**6.6 UNIT TESTING IN PYTHON**

**PYUNIT:**

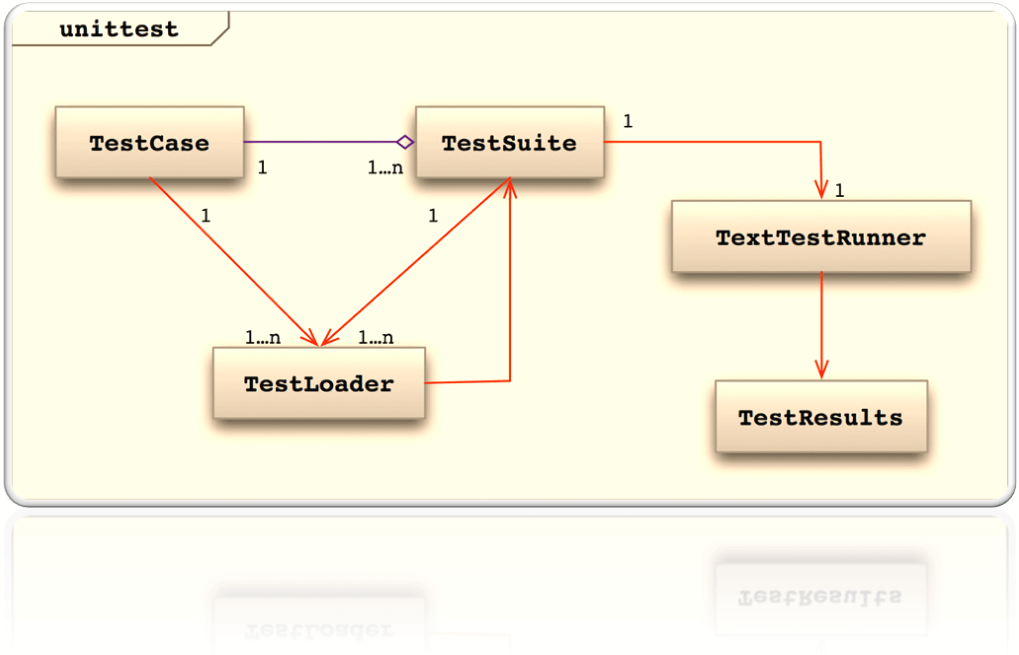
This software system is tested using a testing tool available in python calles the “PYUNIT”. Python Unit Testing mainly involves testing a particular module without accessing any dependent code. Developers can use techniques like stubs and mocks to separate code into "units" and run unit level testing on the individual pieces. PyUnit makes it easier to perform unit testing as it comes with its own framework:

* **PyUnit:**PyUnit supports fixtures, test cases, test suites and a test runner for the automated testing of the code. In PyUnit, you can organize test cases into suites with the same fixtures
* **Nose:**Nose's built in plug-ins helps you with output capture, code coverage, doctests, etc. Nose syntax is pretty simpler and reduces the barriers to writing tests. It extends Python unittest to make testing easier.
* **Doctest :**Doctest testing script goes in docstring with small function at the bottom of file. Doctest allows you to test your code by running examples included in the documentation and verifying that they returned the expected results. The use-case of doctest is less detailed and don't catch special cases. They are useful as an expressive documentation of the main use case of a module and its components.

**PYUNIT CLASSES:**

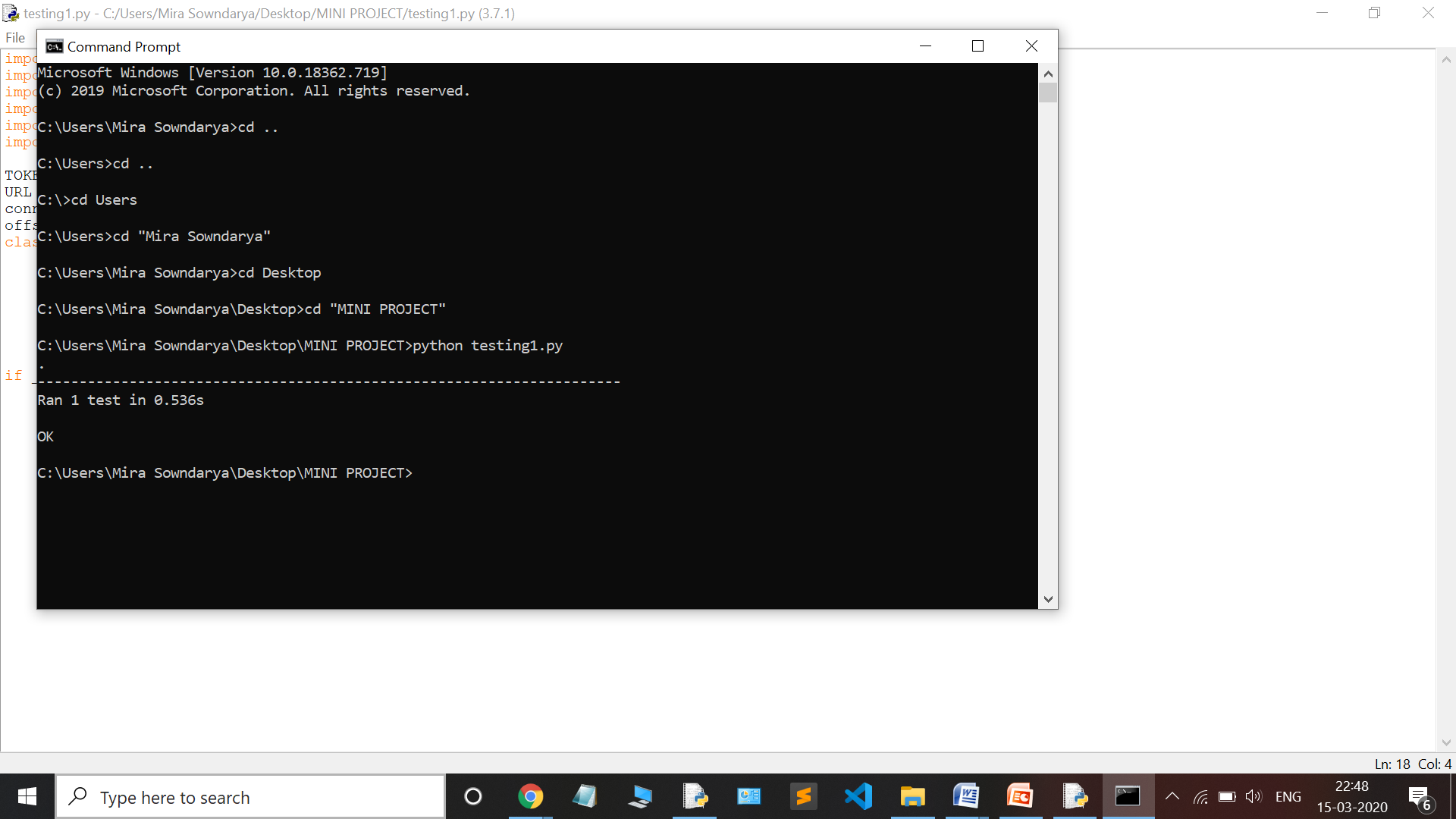
PyUnit performs testing of the modules using the following classes:

* **TestCase class**: The TestCase class bears the test routines and delivers hooks for making each routine and cleaning up thereafter
* **TestSuite class**: It caters as a collection container, and it can possess multiple testcase objects and multiple testsuites objects
* **TestLoader class**: This class loads test cases and suites defined locally or from an external file. It emits a testsuite objects that posseses those suites and cases
* **TextTestRunner class**: To run the tests it caters a standard platform to execute the tests
* **The TestResults class**: It offers a standard container for the test results



**FIGURE 6.1**

The snapshot of the implementation of unit test for the module is given in figure 6.2. It returns OK to show that the system has passed the unit test.



**FIGURE 6.2**

**CHAPTER 7**

**CONCLUSION AND FUTURE ENHANCEMENT**

**7.1 CONCLUSION**

The conventional way of making doctor appointments is a tedious process as it involves either making a telephone call to the clinic directly or visiting the hospital in person for the same. Thus the “PARI BOT” aims at eliminating this tedious process of appointment fixing. It only involves simple messages by which the patient is guaranteed of the doctor visit. a receptionist at the hospital end can be eliminated if this method is implemented. This method is very user friendly and and an appointment can be easily fixed through direct messages with the system-“the doctor chatbot” without much complications. Also the need of a receptionist at the hospital end can be eliminated if this method is implemented.

A Chatbot that takes information such as the doctor name, day of visit and the slot and books an appointment was developed.The Bot was built on Telegram platform using the chat service provided by Telegram. The Telegram server forwards the updates to the application server and delivers the replies to the user.The application server processes the updates and uses Sqlite database to check the availability and responds accordingly. This method is very user friendly and and an appointment can be easily fixed through direct messages with the system-“the doctor chatbot” without much complications.

The project can be implemented for various clinics, both small scale and large scale by which the human resource costs can be reduced or in certain cases even eliminated.

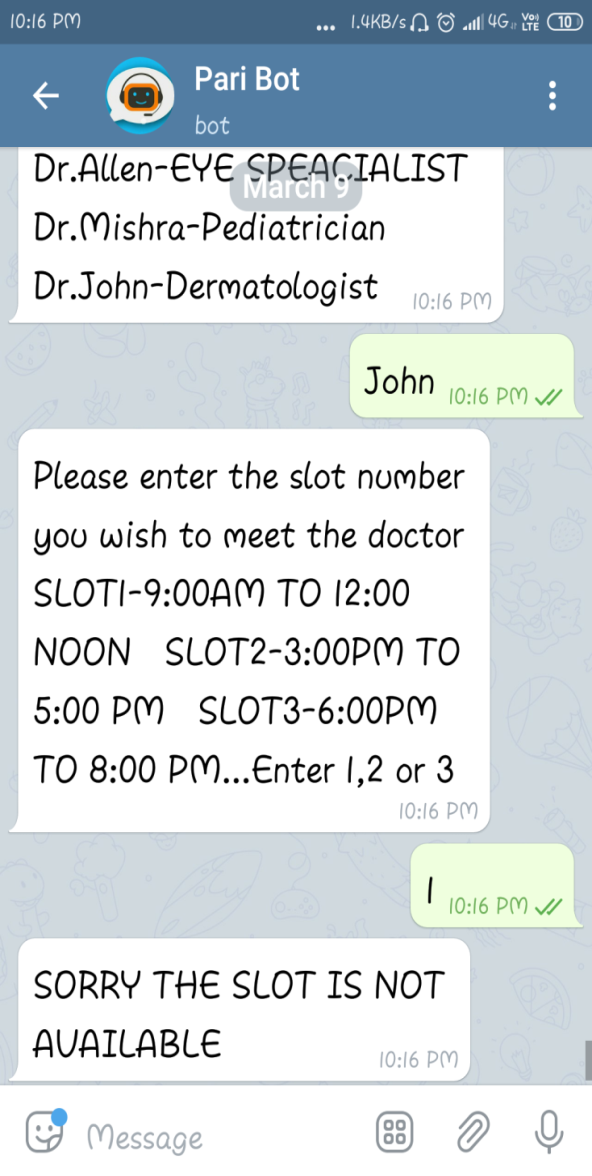
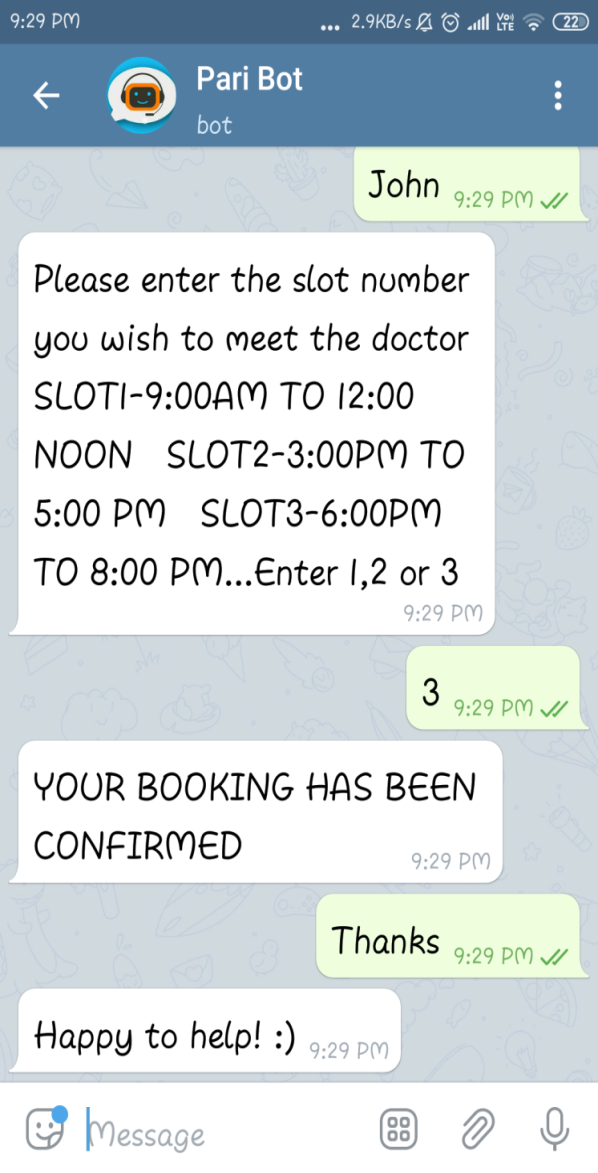
**7.2 FUTURE ENHANCEMENT**

In the future, the next step would ideally be to make a complete and universal model that the organizations can use to help them reduce their human resource costs. Furthermore, can also be combined with other machine learning algorithms to create a highly robust and powerful system. Various other concepts of machine learning can be implemented to improve the efficiency of the system as a whole. The more the training data the more the accuracy. Also interfaces independent to the hospital can be created for more privacy of the users.

**APPENDIX-I**

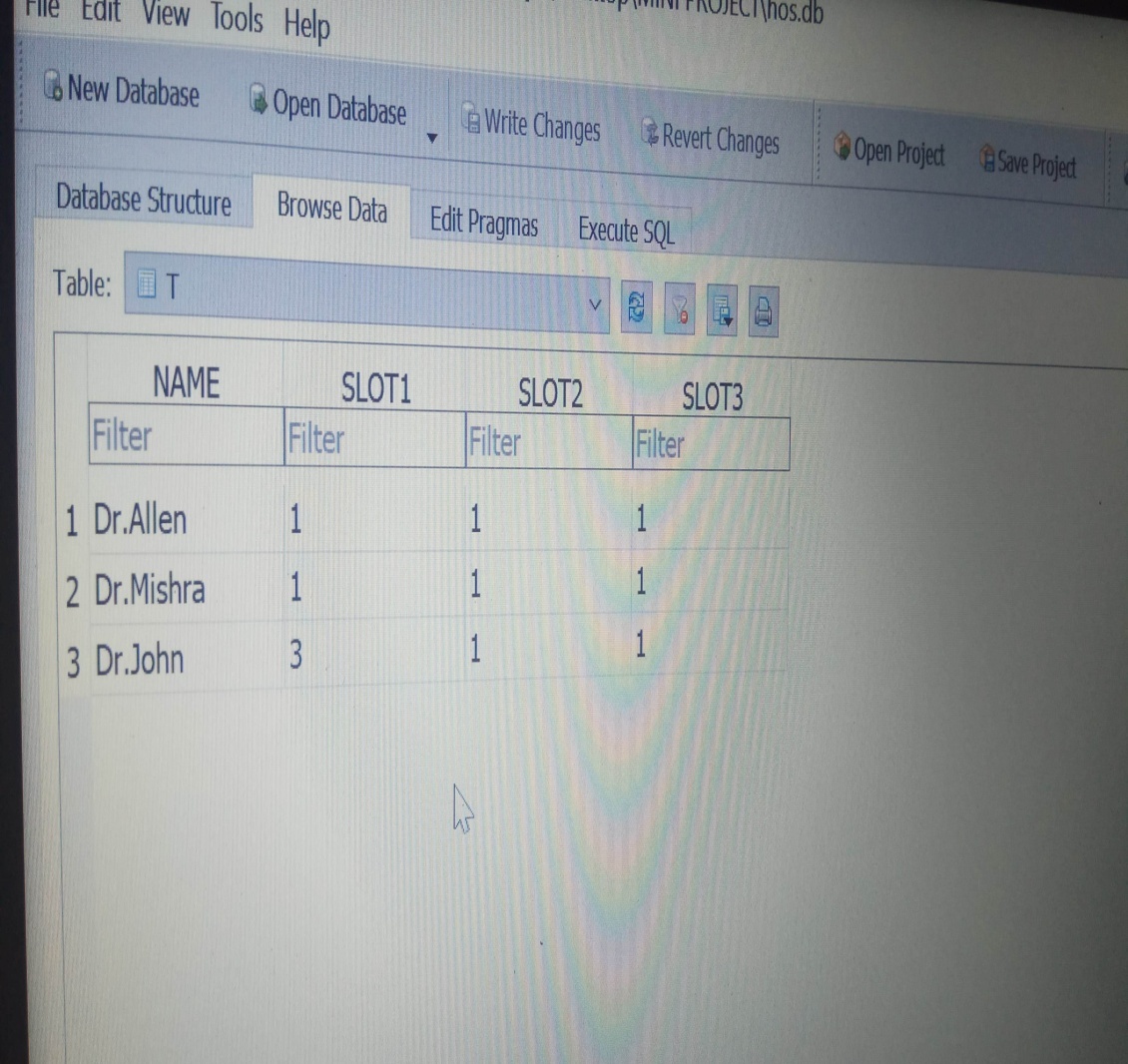
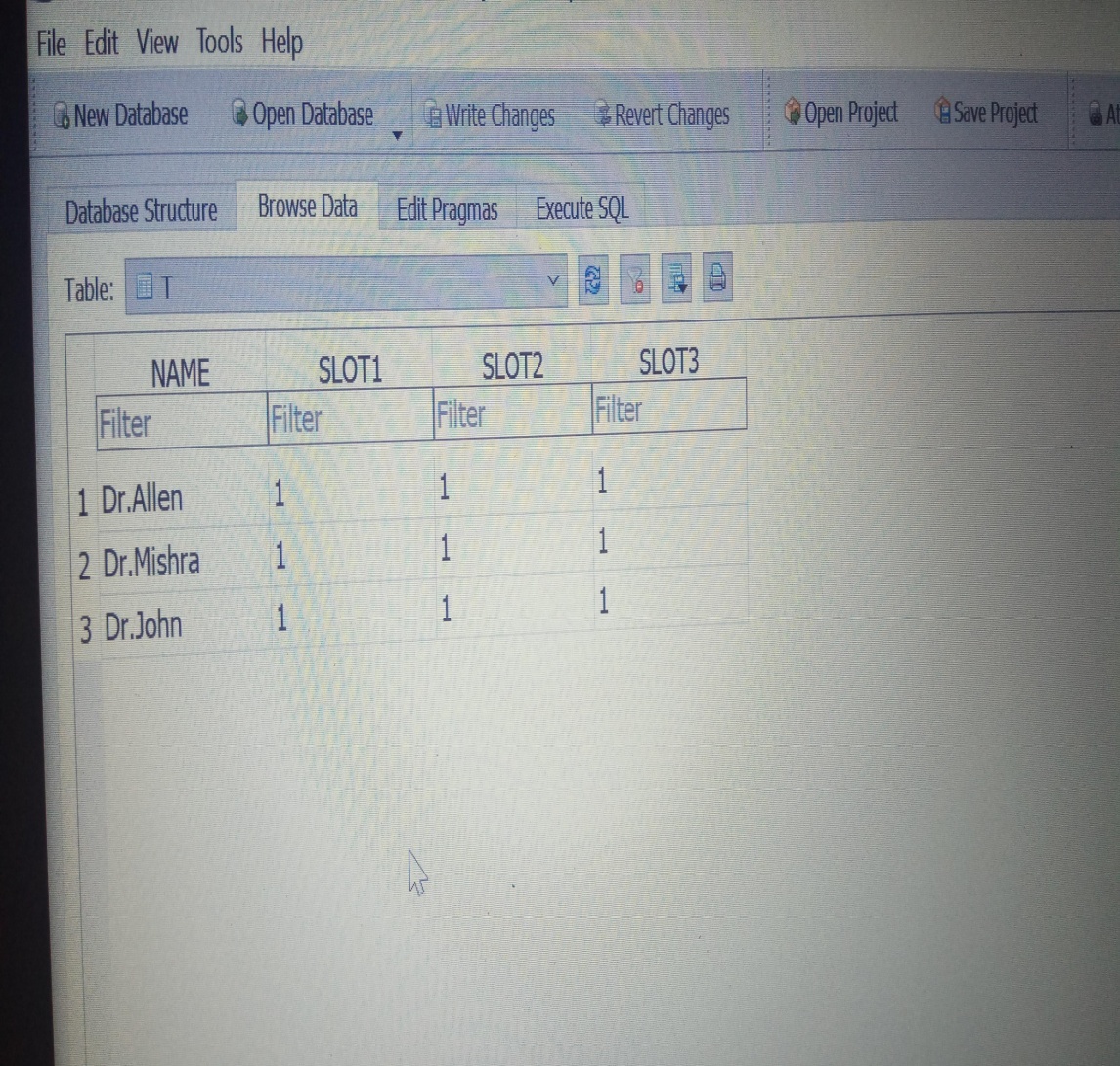
**IMPLEMENTATION SCREENSHOTS**

**TELEGRAM CHAT:**

****

**FIGURE 1.1 BOOKING CONFIRMED FIGURE 1.2 BOOKING CANCELLED**

**DATABASE –SQLITE3 BROWSER:**

****

**FIGURE 1.3 SLOT AVAILABLE FIGURE 1.4 SLOT FULL**

**REFERENCES**

**1.** Mrs. V. Selvi, M.E, Ms.SaranyaS,Ms.Chidida.k,Ms.Abarna R Assistant Professor CSE, UG Scholar Dept. of CSE, ManakulaVinayagar Institute of Technology, Puducherry, “Chatbot and bullyfree chat”, IEEE International Conference on System, Computation and Automation, 29-30 March’2019.

**2.**MuhamadMuslih, Somantri, DediSupardi, ElpidMutlipi, YusuphMavlanaNyaman, “Developing Smart Workspace Based IOT with AI using Telegram Chatbot”, IEEE International Conference(ICCED), 15 April’19.

**3.** SanketSahni, V Geetha, S SowmyaKamath, “A conversational smart home assistant built on Telegram and Google Dialogflow”, IEEE Conference- tencon 2019, 12 December ‘2019.

4.[BhaumikKohli](https://ieeexplore.ieee.org/author/37086887040); [TanupriyaChoudhury](https://ieeexplore.ieee.org/author/37085794967); [Shilpi Sharma](https://ieeexplore.ieee.org/author/37076088600); [Praveen Kumar](https://ieeexplore.ieee.org/author/37085348077),”A Platform for Human-Chatbot Interaction Using Python”, IEEE [2018 Second International Conference on Green Computing and Internet of Things (ICGCIoT)](https://ieeexplore.ieee.org/xpl/conhome/8743504/proceeding), 16-18 Aug. 2018.