# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belagavi – 590018

A Project Work on

# "SmartPill Companion MedApp using ML & IOT"

*A dissertation work submitted in partial fulfillment of the requirementfor the award of the degree*

**Bachelor of Engineering**

In

**Information Science and Engineering**

Submitted by


| | |
|---|---|
| **Abhinav Anand** | **1AY18IS004** |
| **Abhishek Prasad** | **1AY18IS006** |
| **Bhargavi C** | **1AY18IS026** |
| **Keerthi Srinivas** | **1AY18IS053** |


Under the Guidance of

**Dr. Chayapathi A R**

Assistant Professor

ACHARYA


**DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING**

ACHARYA INSTITUTE OF TECHNOLOGY

**(AFFILIATED TO VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELAGAVI.ACCREDITED BY NACC, RECOGNISED BY AICTE, NEW DELHI)**

Acharya Dr. Sarvepalli Radhakrishnan Road, Soldevanahalli, Bengaluru - 560107


**2021-2022**

**DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING**

# ACHARYA INSTITUTE OF TECHNOLOGY

**(AFFILIATED TO VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELAGAVI. ACCREDITED BY NACC, RECOGNISED BY AICTE, NEW DELHI)**

Acharya Dr. Sarvepalli Radhakrishnan Road, Soldevanahalli, Bangaluru-560107

**2021-2022**


# Certificate

Certified that the Project work entitled **"SmartPill Companion MedApp using ML & IOT"** is carried out by **Abhinav Anand (1AY18IS004)**, **Abhishek Prasad (1AY18IS006)**, **Bhargavi C (1AY18IS026)**, and **Keerthi Srinivas (1AY18IS053)**, are bonafide student of **Acharya Institute of Technology, Bengaluru** in partial fulfillment for the award of the degree of **Bachelor of Engineering** in **Information Science and Engineering** of the **Visvesvaraya Technological University**, Belagavi during the year 2021-2022. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporatedin the report deposited in the departmental library. The project has been approved as it satisfies the academic requirements in respect of project work prescribed for the Bachelor of Engineering Degree.

| Signature of the Guide | Signature of the HOD | Principal |
|---|---|---|
| **Dr. Chayapathi A R** | **Dr. Chayapathi A R** | **Dr. Rajath Hegde** |
| Assistant Professor | Assistant Professor | |

**External Vivo-Voce:**

| **Name of the Examiners** | **Signature with Date** |
|---|---|
| 1. _____ | _____ |
| 2. _____ | _____ |

# DECLARATION

We, Abhinav Anand (1AY18IS004), Abhishek Prasad (1AY18IS006), Bhargavi C (1AY18IS026), and Keerthi Srinivas (1AY18IS053), students of B.E, Information Science and Engineering, Acharya Institute of Technology, Bengaluru - 560107, hereby declare that the project entitled **"SmartPill Companion MedApp using ML & IOT"** is an authentic record of our own work carried out under the supervision and guidance of **Dr. Chayapathi A R**, Assistant Professor and Head of the department, Acharya Institute of Technology, Bengaluru. We have not submitted the matter embodied to any other university or institution for the award of any other degree.

| Name | USN | Signature |
|---|---|---|
| Mr. Abhinav Anand | 1AY18IS004 | |
| Mr. Abhishek Prasad | 1AY18IS006 | |
| Miss. Bhargavi C | 1AY18IS026 | |
| Miss. Keerthi Srinivas | 1AY18IS053 | |

Place:

Date:

# ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of **Phase - 2 Project Work** Report would be incomplete without the mention of the people who made it possible through constant guidance and encouragement.

We would take this opportunity to express our gratitude to **Sri. B. Premnath Reddy**, Founder Chairman, Acharya Institutes, **Dr. Rajath Hegade**, Principal, and **Prof. C K Marigowda,** Vice Principal, Acharya Institute of Technology for providing the necessary infrastructure to complete this **Phase - 2 Project Work** Report.

We wish to express our deepest gratitude to **Dr. Chayapathi A R**, Head of the Department, Information Science and Engineering, and also would like to thank **Phase – 2 Project Work** coordinator **Prof. Subhash Kamble** Assistant Professor for his constant support.

We wish to express our sincere thanks to our guide **Dr. Chayapathi A R,** Assistant Professor and Head of the Department of Information Science and Engineering for helping us throughout, and guiding us from time to time.

A warm thanks to the faculty of Department of Information Science and Engineering, who have helped us with their views and encouraging ideas.

**Abhinav Anand (1AY18IS004)**
**Abhishek Prasad (1AY18IS006)**
**Bhargavi C (1AY18IS026)**
**Keerthi Srinivas (1AY18IS053)**

# ABSTRACT

Automatic Vending Machines are very useful devices for dispensing items such as food and snacks and can be easily available for the use of the public. The Medical Vending machine can be used in healthcare facilities for dispensing medicines which will be easily accessible to the public 24/7. It gives availability of medicines all the time especially in rural areas where the healthcare sector is not that developed.

The proposed model relates to an automatic medicine vending machine that has the capability to dynamically receive input from the user via the application and then dispense the required type of medicine. The MedApp is a bi-lingual application that can be used either in English or French. The application will also be integrated with a medically trained chatbot which can suggest the diagnosis based on the symptoms provided by the user. The chatbot will also provide detailed description about the predicted disease and how the disease can be prevented.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# CHAPTER 1

# INTRODUCTION

# CHAPTER 1

# INTRODUCTION

## 1.1 Overview

Today's era is of informatization. With the advancement of technology and scientific theory, traditional medicine with biotechnology as its core, has gradually began to digitize and information. To reduce the human efforts many electronic devices and gadgets are developed. Same thing can be said about technology advancing in medical field. Smart Vending machines and chatbots are one of the advancements in the technology.

Vending machines are convenient, and they allow your employees and even clients and customers to get what they need in a time and cost-saving manner. Vending machines have good prices on them, and they are always convenient when they are in the right location. Medical chatbots are AI-powered conversational solutions that help patients, insurance companies, and healthcare providers easily connect with each other. These bots can also play a critical role in making relevant healthcare information accessible to the right stakeholders, at the right time.

## 1.2 SmartPill Companion MedApp using ML & IOT

SmartPilll Companion MedApp is an medicine dispensing vending machine which dispenses medicine based on the medicine selected by the user. This SmartPill Companion is also integrated with the medically trained chatbot. This chatbot is trained using Natural Language Processing and it uses K- Nearest Neighbor algorithm to classify and predict the disease. This chatbot provides the detailed description about the disease and also give the precautions to be taken .

The supporting MedApp is a bilingual app which consist of two languages that is English and French. This MedApp is connected to the IOT model through the Nodemcu WIFI module to send and receive signals from the IOT model.

## 1.3 Machine Learning

Machine learning is a branch of artificial intelligence (AI) and computer science which focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy. Machine learning is actively being used today, perhaps in many more places than one would expect.

Machine learning (ML) is a type of artificial intelligence (AI) that allows software applications to become more accurate at predicting outcomes without being explicitly programmed to do so. Machine learning algorithms use historical data as input to predict new output values.

### 1.3.1 Applications of Machine Learning

- **Medical Diagnosis**

  In medical science, machine learning is used for disease diagnoses. With this, medical technology is growing very fast and able to build 3D models that can predict the exact position of lesions in the brain. It helps in finding brain tumors and other brain-related diseases easily.

- **Chatbots**

  A chatbot (Conversational AI) is an automated program that simulates human conversation through text messages, voice chats, or both. It learns to do that based on a lot of inputs, and Natural Language Processing (NLP).

- **Virtual Personal Assistant**

  We have various virtual personal assistants such as Google assistant, Alexa, Cortana, Siri. As the name suggests, they help us in finding the information using our voice instruction. These assistants can help us in various ways just by our voice instructions such as Play music, call someone, open an email, Scheduling an appointment, etc.

### 1.3.2 Importance of Machine Learning

Machine learning is important because it gives enterprises a view of trends in customer behaviour and business operational patterns, as well as supports the development of new products. Many of today's leading companies, such as Facebook, Google and Uber, make machine learning a central part of their operations. Machine learning has become a significant competitive differentiator for many companies.

### 1.3.3 Types of Machine Learning

Classical machine learning is often categorized by how an algorithm learns to become more accurate in its predictions. There are four basic approaches: supervised learning, unsupervised learning, semi-supervised learning and reinforcement learning. The type of algorithm data scientists chooses to use depends on what type of data they want to predict.

- **Supervised learning:** In this type of machine learning, data scientists supply algorithms with labeled training data and define the variables they want the algorithm to assess for correlations. Both the input and the output of the algorithm is specified.

- **Unsupervised learning:** This type of machine learning involves algorithms that train on unlabeled data. The algorithm scans through data sets looking for any meaningful connection. The data that algorithms train on as well as the predictions or recommendations they output are predetermined.

- **Semi-supervised learning:** This approach to machine learning involves a mix of the two preceding types. Data scientists may feed an algorithm mostly labelled training data, but the model is free to explore the data on its own and develop its own understanding of the data set.

- **Reinforcement learning:** Data scientists typically use reinforcement learning, to teach a machine to complete a multi-step process for which there are clearly defined rules. Data scientists program an algorithm to complete a task and give it positive or negative cues as it works out how to complete a task. But for the most part, the algorithm decides on its own what steps to take along the way.

### 1.3.4 Advantages and Disadvantages of Machine Learning

Machine learning has seen use cases ranging from predicting customer behaviour to forming the operating system for self-driving cars. When it comes to advantages, machine learning can help enterprises understand their customers at a deeper level. By collecting customer data and correlating it with behaviours over time, machine learning algorithms can learn associations and help teams tailor product development and marketing initiatives to customer demand. Some companies use machine learning as a primary driver in their business models. Uber, for example, uses algorithms to match drivers with riders. Google uses machine learning to surface the ride advertisements in searches. But machine learning comes with disadvantages. First and foremost, it can be expensive. Machine learning projects are typically driven by data scientists, who command high salaries. These projects also require software infrastructure that can be expensive. There is also the problem of machine learning bias. Algorithms trained on data sets that exclude certain populations or contain errors can lead to inaccurate models of the world that, at best, fail and, at worst, are discriminatory. When an enterprise bases core business processes on biased models it can run into regulatory and reputational harm.

## 1.4 KNN Algorithm

K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique. *k*-NN is a type of classification where the function is only approximated locally, and all computation is deferred until function evaluation. Since this algorithm relies on distance for classification, if the features represent different physical units or come in vastly different scales then normalizing the training data can improve its accuracy dramatically. Both for classification and regression, a useful technique can be to assign weights to the contributions of the neighbors, so that the nearer neighbors contribute more to the average than the more distant ones. For example, a common weighting scheme consists in giving each neighbor a weight of 1/*d*, where *d* is the distance to the neighbor.

The neighbors are taken from a set of objects for which the class (for *k*-NN classification) or the object property value (for *k*-NN regression) is known. This can be thought of as the training set for the algorithm, though no explicit training step is required. A peculiarity of the *k*-NN algorithm is that it is sensitive to the local structure of the data.

## 1.5 Natural Language Processing

Natural language processing (NLP) is a subfield of linguistics, computer science, and artificial intelligence concerned with the interactions between computers and human language, in particular how to program computers to process and analyze large amounts of natural language data. The goal is a computer capable of "understanding" the contents of documents, including the contextual nuances of the language within them. The technology can then accurately extract information and insights contained in the documents as well as categorize and organize the documents themselves.

Challenges in natural language processing frequently involve speech recognition, natural-language understanding, and natural-language generation.

## 1.6 Internet of Things

The Internet of things (IoT) describes physical objects (or groups of such objects) with sensors, processing ability, software, and other technologies that connect and exchange data with other devices and systems over the Internet or other communications networks. Internet of things has been considered a misnomer because devices do not need to be connected to the public internet, they only need to be connected to a network and be individually addressable.

The field has evolved due to the convergence of multiple technologies, including ubiquitous computing, commodity sensors, increasingly powerful embedded systems, and machine learning. Traditional fields of embedded systems, wireless sensor networks, control systems, automation, independently and collectively enable the Internet of things. In the consumer market, IoT technology is most synonymous with products pertaining to the concept of the "smart home", including devices and appliances that support one or more common ecosystems, and can be controlled via devices associated with that ecosystem, such as smartphones and smart speakers. IoT is also used in healthcare systems.

### 1.6.1 Applications of IOT

- **Healthcare -** Healthcare industry has been utilizing the possibilities of Internet of Things for life saving applications. Starting from collecting vital data from bed side devices, real-time diagnosing process, accessing medical records and patient information across multiple departments, the entire system of patient care can be improved with IoT implementation.
- **Smart Home and Office -** Smart home applications with the use of smart sensors are becoming popular now. Any smart device can be configured and connected to the internet and control using simple mobile application.
- **Agriculture and Smart Farming -** There are lot of challenges in the agriculture and farming industry to produce more crops and vegetable to feed increasing human population. Internet of Things can assist farmers and researchers in this area to find more optimized and cost-effective ways to increase production.

## 1.7 IOT components used in the project

### ➢ Arduino UNO


**Fig – 1.1** Arduino UNO

The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button.

## ➢ RFID Reader

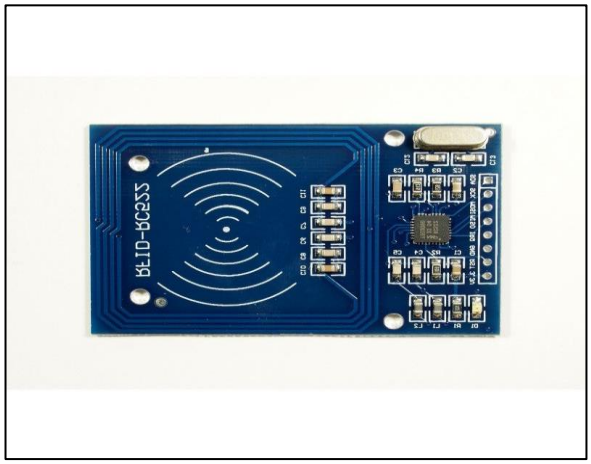

**Fig – 1.2** RFID Reader

Radio-frequency identification (RFID) uses electromagnetic fields to automatically identify, and track tags attached to objects. An RFID system consists of a tiny radio transponder, a radio receiver and transmitter. When triggered by an electromagnetic interrogation pulse from a nearby RFID reader device, the tag transmits digital data, usually an identifying inventory number, back to the reader.

## ➢ WIFI Module

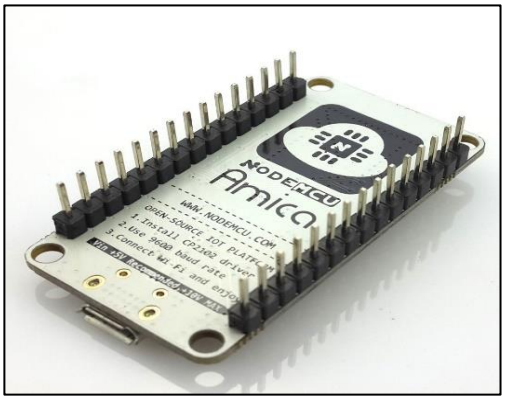


**Fig – 1.3** NodeMCU Wifi Module

NodeMCU is a low-cost open source IoT platform. It initially included firmware which runs on the ESP8266 Wi-Fi SoC from Espress if Systems, and hardware which was based on the ESP-12 module. The prototyping hardware typically used is a circuit board functioning as a dual in-line package (DIP) which integrates a USB controller with a smaller surface-mounted board containing the MCU and antenna.

## ➢ Servo Motors


**Fig – 1.4** Servo Motor

A Servo Motor is a small device that has an output shaft. This shaft can be positioned to specific angular positions by sending the servo a coded signal. As long as the coded signal exists on the input line, the servo will maintain the angular position of the shaft. If the coded signal changes, the angular position of the shaft changes. In practice, servos are used in radio-controlled airplanes to position control surfaces like the elevators and rudders. They are also used in radio-controlled cars, puppets, and of course, robots.

## ➢ LCD Display


**Fig – 1.5** LCD Display

A liquid crystal display is a thin, flat electronic visual display that uses the light modulating properties of liquid crystals (LCs). LCs does not emit light directly. The primary factor was size, an LCD consisting of primarily with some liquid crystal material between them of two glass plates. In general, LCDs uses very low power than the cathode-ray tube (CRT) counterparts. Many LCDs are ruminative, means that they use only atmosphere light to illuminate the display.

# CHAPTER 2

# DETAILED LITERATURE SURVEY

# CHAPTER 2

# DETAILED LITERATURE SURVEY

## 2.1 Literature Survey on Vending Machine

1. **[2020] IoT Based Medicine Dispenser,** Aditya Patil , Darshan B.G., Dr. D.V. Ashoka, NethravathiB

   The Medicine Dispenser is a machine which dispenses the medicine based on the authentication of the user by using the user name and the password given to the user by the administrator of the machine. This machine will dispense medicine according to the user prescription provided by the user's doctor and also the number of medicine is also determined by the user's choice. In this medicine dispenser the money is handled through the digital points stored in the user's card which is given by the administrator and also the user can restore the points by talking to the administrator of the card and use his identification to add the digital points to the card. The main goal of the dispensing machine is to provide the medicine to the users 27/7 so it can help the village people who are for away from the hospital or the clinic to buy the medicine and also it can be installed like a ATM machine so it is easy to use by the people even though if they cannot read and write. The medicine can be replaced by the administrator of the machine by time to time based on the expiration date or is it is empty the machine will alarm the administrator to refill the machine through SMS or telecom messenger.

2. **[2020] Automatic Chocolate Vending Machine,** Kanagasabapathi V, Naveenraj K., Neelavarna

   This paper proposes the design of an Automatic chocolate vending machine. The primary point of the thought is to dispatch new innovation applications in the public eye.Vending machines that sell different types of products. Here we use Radio frequency Identification (RFID) along with Arduino, to overcome the coin based vending machine which does not return the balance amount if no change is available. The customer can select the product before the RFID is read, after the card is scanned the product can be collected at the chocolate collector. There are three units present in the machine,the cashless payment provided by the RFID in the first unit. In

second unit ,Arduino UNO executed the programming section and Finally,the machine delivers the product and displays the information on the display.This paper made solution to the machine that do not returns the balance amount in a coin based vending machine.

3. **[2019] RFID Based Vending Machine,** Aye Aye Nyein, Ohnmar Win

   The vending machine which provides beverages like snacks, cold drinks, is also used for ticketing. These systems are operated on either coin or note or manually switch operated. This paper presents the system which operates not on coin or note, it operates on RFID system. This system gives the access through only RFID which avoids the misuse of the machine. A small RFID reader is fitted on the machine. The identity card which contains the RFID tag is given to each person. If the ID card owner wants to get some snacks, by showing the ID tag to the reader is the only way to get the desired snacks and the amount to money to pay will be deducted from the account.

4. **[2019] Coin Acceptor Based Vending Machine using Microcontroller**, Hay Man Oo, Khin Thandar Tun, Su Mon Aung

   This paper investigates a coin acceptor vending machine using a microcontroller system. Technology has become a part of the different aspects of peoples' lives as it makes most of their work faster and easier. One of the fast-paced technologies is the vending machine. It is a machine that dispenses automatically, products such as beverages, tickets, snacks, etc., by inserting currency or credit to the machine. Vending machines appear in different forms and functions. These are generally used in public and private areas such as malls, markets, business and government offices, schools and along the streets.This paper proposes the design of Arduino based automatic vending machines.The main objective of this paper is to launch new technology applications in society. In this proposed system, Arduino Mega board, Liquid Crystal Display (LCD), coil acceptor, servo motor, stepper motor and push button are used.The rectifier circuit is used for power supply and step down DC to DC mod is used to reduce the rectifier output voltage 12V to 5V. Mega is mainly used to run the program for the vending machine. LCD is used for showing the information, for inserting coils and making a selection. The user can choose the product by touching the related button.

5. **[2019] Design and Implementation of RFID based Smart Shopping Booth,** Hay Man Oo, Khin Thandar Tun, Su Mon Aung

   Some of the most dangerous web attacks, such as Cross-Site Scripting and SQL injection, exploit vulnerabilities in web applications that may accept and process data of uncertain origin without proper validation or filtering, allowing the injection and execution of dynamic or domain-specific language code. These attacks have been constantly topping the lists of various security bulletin providers despite the numerous counter measures that have been proposed over the past 15 years. The paper provided an analysis on various defense mechanisms against web code injection attacks and proposed a model that highlights the key weaknesses enabling these attacks, and that provides a common perspective for studying the available defenses. It then categorizes and analyzes a set of 41 previously proposed defenses based on their accuracy, performance, deployment, security, and availability characteristics. Detection accuracy is of particular importance, as findings of the paper shows that many defense mechanisms have been tested in a poor manner. In addition, it was also observed that some mechanisms can be bypassed by attackers with knowledge of how the mechanisms work.

## 2.2 Literature Survey on Chatbot

1. **[2021] Intelligent Chatbot,** Bhaumik Kohli , Tanupriya Choudhury, Shilpi Sharma, Praveen

   A Chat-bot is a software application used to conduct an online chat conversation via text or text-to-speech, instead of providing direct contact with a live human agent. Designed to convincingly simulate the way a human would behave as a conversational partner. In the proposed system, we presented a chatbot that generates a dynamic response for online client's queries. The Proposed System is based on Artificial Intelligence-powered Chatbot. The web-based platform provides a vast intelligent base that can help simulate problem- solving for humans. This proposed chatbot identifies the user context which triggers the particular intent for a response. Since it is a dynamic response the desired answer will be generated for the user. The proposed system used machine learning algorithms to learn the Chatbot by experiencing various user's responses and requests. After referring to 17 IEEE papers and 13 Standard papers, our research found a state that the strong point of Chat-bot is that it comes to use in numerous fields of our daily life.

2. **[2021] Design and Development of CHATBOT: A Review**, Rohit Tamrakar, Niraj Wani

   This paper focuses on a newly emerging tool for learning from Chabot, which is a learning-cum-assisted tool. A Chatbot is an artificially created virtual entity that interacts with users using interactive textual or speech skills. This Chatbot directly chats with the people using artificial intelligence and Machine Learning concepts. This paper reviews the technique, terminology, and different platforms used to design and develop the Chatbot. It also presents some actual practical life typical applications and examples of Chatbot. The utility of the Chatbot tool for Computer-Aided Design (CAD) applications is proposed from this review. The different methods used are ,Parsing, Pattern matching, Artificial Intelligence Markup Language, Chat Script, SQL, Markov Chain, Language Trick, Ontologies.

3. **[2020] Development of An e-commerce Sales Chatbot**, Mohammad Monirujjaman Khan

   This paper presents the development of an e-commerce sales chatbot in order to provide customer support and increase sales. The system uses machine learning for natural language understanding. It is developed on an modular chatbot framework.The intent of the proposed solution is to push the boundaries of natural language processing and natural language understanding and take them a step closer to understanding context. The biggest challenge in the NLP/NLU as of now, is to develop a chatbot that can render an experience that mimics real human sales agents, bringing together two industries, e-commerce sales and AI to create an immersive model to perform synergy.This modular system is available for more platforms.This present NLU engine trains its classifier from the classified training data provided by the admins. Also, it is based on SVM. Artificial Neural Network can be used to improve the accuracy of the NLU Engine. This project may help improve the relationship with customers which can lead to more sales. Also, it has the potential to make customer service cheaper and more satisfying.

4. **[2019] Chatbot: An automated conversation system for educational domain,** Anupam Mondal, Monisha Dey, Dipankar Das, Sachit Nagpal, Kevin Garda

   Speech and textual information play a crucial role in communicating between humans. An article in "The New York Times" published that now-a-days the adults are spending more than 8 hours a day on screens of computers or mobiles. So the major communication between humans is conducted through web applications such as WhatsApp, Facebook, and Twitter etc as a form

of speech and textual conversation. In the present paper, we have focused on designing a textual communication application namely chatbot in the educational domain. The proposed chatbot assists in answering questions provided by the users. To develop the system, we have employed an ensemble learning method as a random forest in the presence of extracted features from our prepared dataset. Besides, the validation system offers an average F-measure 0.870 score on various K-values under random forest for the proposed chatbot. Finally, we have deployed the proposed system in the form of a telegram bot.

5. **[2019] Chatbot System for Cancer Patient**, Pooja Nehul, Bhakti Lohar, Uttkarsha Jagtap, Shreya Rajurkar, Prof. Gauri Virkar

   Normally Users are not aware about all the treatment or symptoms regarding the particular disease for small problem users have to go personally to the hospital for check-up which is more time consuming. Also handling the telephonic calls for the complaints is quite hectic. Such a problem can be solved by using a medical Chat Bot by giving proper guidance regarding healthy living. The medical chat-bots functioning depends on Natural language processing that helps users to submit their problems about their health. The User can ask any personal query related to health care through the chat-Bot without physically being available to the hospital by Using Google API for voice-text and text voice conversion. Query is sent to the Chat Bot and gets related answer and display answer on android app.

## 2.3 Literature Survey on Multilingual Indexing

1. **[2021] Graph based Multilingual Product Retrieval in E-Commerce Search** , Hanqing Lu, Youna

   Nowadays, with many e-commerce platforms conducting global business, e-commerce search systems are required to handle product retrieval under multilingual scenarios. Moreover, compared with maintaining per-country specific e-commerce search systems, having a universal system across countries can further reduce the operational and computational costs, and facilitate business expansion to new countries. In this paper, a universal end-to-end multilingual retrieval system is introduced and learnings and technical details when training and deploying the system to serve billion-scale product retrieval for e-commerce search is discussed.

2. **[2016] User Model-Based Intent-Aware Metrics for Multilingual Search Evaluation,** Alexey Drutsa, Andrey Shutovich, Philipp Pushnyakov, Evgeniy Krokhalyov, Gleb Gusev, Pavel Serdyukov

   Despite the growing importance of multilingual aspects of web search, no appropriate offline metrics to evaluate its quality have been proposed so far. At the same time, personal language preferences can be regarded as intents of a query. This approach translates the multilingual search problem into a particular task of search diversification. Furthermore, the standard intent-aware approach could be adopted to build a diversified metric for multilingual search on the basis of a classical IR metric such as ERR. The intent-aware approach estimates user satisfaction under a user behavior model. We show however that the underlying user behavior models are not realistic in the multilingual case, and the produced intent-aware metric does not appropriately estimate the user satisfaction. We develop a novel approach to build intent-aware user behavior models, which overcome these limitations and convert to quality metrics that better correlate with standard online metrics of user satisfaction.

3. **[2014] Indexing and Weighting of Multilingual and Mixed Documents**, Mohammed Mustafa, Izzedin Osman, Hussein Suleman

   Non-English-speaking users, such as Arabic speakers, are not always able to express terminology in their native languages, especially in scientific domains. Such difficulty forces many Arabic authors and scholars to use English terms in order to explain precise concepts, particularly when they address technical topics, resulting in mixed/multilingual queries with both English and Arabic terms. Cross Language Information Retrieval (CLIR) allows users to search documents that are written in a language different from the query. However, current algorithms are optimized for monolingual queries, even if they

   are translated. This paper attempts to address the problem of multilingual querying in CLIR. New techniques that are better suited to the unique characteristics of this problem, in terms of indexing and weighting, are proposed. A new multilingual and mixed test collection containing mixed-language (Arabic and English) computer science documents and mixed-language queries has been created. Experimentally, results show that current CLIR techniques were not designed for these types of multilingual queries and documents and are found to perform poorly whereas the proposed techniques are found to be promising.

4. **[2014] Query Transformation for Multilingual Product Search**, Qie Hu, Vishnu Narayanan, Ivan Davchev, Hsiang-Fu Yu, Rahul Bhagat, Inderjit S. Dhillon

   In this paper, we study the problem of enabling multilingual product search for a global shopping store. Given an existing search system and product catalog in a primary language, and a search query in a secondary language, transform the query into a semantically equivalent one in the primary language in order to retrieve the most relevant products. A Query Transformation system that consists of 1) a language identifier to detect the language of the input query, 2) a deep neural machine translation model fine-tuned on human-curated parallel query corpus and learned, during training, to copy entities such as model numbers, and 3) a traffic re-ranker which selects the transformation that may help the search system retrieve the most relevant products. Compelling offline and online results: 11% and 3% in improvements in offline nDCG@8 for Spanish (ES) → English (EN) and French (FR) → EN, and 10% and 22% in reduction in online product type search defects for ES→EN and FR→EN, respectively, over a state-of-the-art statistical machine translation system for product search, are presented.

5. **[2010] Multi-Language Ontology-based Search Engine,** Lily Popova Zhuhadar, Olfa Nasraoui, Robert Wyatt, Elizabeth Romero

   This paper addresses the problem of a user who is capable to read or use documents written in a specific language, but he/she is not fluent in this specific language to query for the right terms to find the document.The majority of information retrieval systems are monolingual and more precisely English-based.This proposed model uses an open-source search engine and a scalable Information Retrieval (IR) library that allows indexing and searching capabilities. It applies the Boolean Model first to select the most relevant documents for the query,then it uses the Vector Space Model for content-based ranking. The hierarchical structure of the ontology defines the relationship between concepts. Term Vector Translation approach maps statistical information about term usage between languages using techniques which map sets of weights from English to Spanish and vice-versa.

6. **[2005] A Multilingual (Chinese, English) Indexing, Retrieval, Searching Search Engine**, Wen-hui ZHANG, Hua-lin QIAN, Wei MAO, Guo-nian SUN

   This paper presents the concepts, technologies, algorithms and detailed measures to achieve the goal of providing highly relevant, almost complete and up-to-date multilingual information search, index, and retrieve. As more and more people on the Internet are from non-English-speaking countries and Web documents can be found in languages other than English, Multilingual search engine is strongly required for users to be able to retrieve information from multilingual databases. In this paper, this problem is addressed for Chinese, a non-romanised language. The proposed system consists of two subsystems: organization based subsystem and Web-page based subsystem. The process of indexing is to generate index objects and then create the inverted index on resource profiles. In the proposed multilingual architecture, the search engine has as many index tables as the number of languages. The search engine employs the so called TFxIDF search and ranking algorithm which assigns relevance scores to Web documents using the vector space model. Retrieved documents are ranked in decreasing order of similarity to the query.

## 2.4 Literature Survey Summary Table

The table 2.4 represents the summary of the Literature Survey review which consist of the paper referred to the problem that have been encountered in the paper, the authors approach in solving the problem and finally the results that were obtained.

| SL.No | Title of the Paper | Problem Addressed | Authors Approach / Method | Results |
|---|---|---|---|---|
| **Automatic Vending Machine** | | | | |
| 1 | IoT Based Medicine Dispenser<br><br>[2020] | To design and develop a dispensing machine which uses RFID tag as an authorization card and displays the user data with prescription. The system that will send notification to the administrator if the medicine needs to be refilled. | RFID tag system with an Arduino mega board that uses LCD, nodemcu, RFID and Wi-Fi Module. Embedded system which will provide all the functions needed to fulfill the goal of creating the medicine dispenser. | The medical dispenser offers a flexible and simple solution for extending basic healthcare to all places including remote places, at a very less cost. The machine will dispense prescribed medicineThe medical dispenser is technically feasible to all the people. |
| 2 | Automatic Chocolate Vending Machine<br><br>[2020] | Technical problems in methods of payment, level of security, heavy structure, and product capacity. Cause problem in giving balance amount. Fraud cases are common in means of hacking the system. | The Arduino Uno acts as a controller along with an RFID tag and a reader. Many external devices like keypad, DC motor, power supply unit, ATMega328 AVR, and a switch are used. | The RFID scans the tag, and the scanned signal is fed to Arduino and after that the user can select the product for their own needs. Finally, the machine delivers the product and displays the information on the display. |

| | | | | |
|---|---|---|---|---|
| 3 | RFID Based Vending Machine<br><br>[2019] | The option of vending machines is critical because many people are determined by them to access products conveniently. Vending machines are always convenient when they are in the right location. | The main components used in the system are microcontroller, motor driver, DC motor, LCD display, RFID RC522 and power supply. | The user needs to insert a RFID card and press a button of user choice and the vending machine will dispense the corresponding item for the user. |
| 4 | Coin Acceptor Based Vending Machine using Microcontroller<br>[2019] | The problem addressed in this paper refers to the advancement and automation of every product in society and hence use of different functions and forms of vending machines in public and private areas such as hospitals, malls, schools and along the streets. | In this proposed system, Arduino MEGA board, coin acceptor, LCD, four servo motors and four push buttons are used.The system rectifier circuit is used and step to reduce the rectifier output voltage 12V to 5V. | All the wiring for the system is shown . LCD shows to make selection and the servo working. Hence, the ejection of the chosen item from the vending machine is shown. |
| 5 | Design and Implementation of RFID based Smart Shopping Booth<br><br>[2019] | VM that holds traditional methods has no passcode protection. Storage of coins and various sizes of coins are also an alarming situation. Maintenance and auditing of cash is also a severe issue | In this proposed model the transaction is made by RFID based rechargeable prepaid card. RFID reader unit is interfaced with an Arduino Mega microcontroller. The coding in the controller is done by using proton IDE Language. LCD display, Keypad are other components | The user can access the VM with an RFID card. The microcontroller reads the info from the reader. The student should enter the password with a keypad and after successful security check the item is dispensed. |

| | | | | |
|---|---|---|---|---|
| **Multilingual Indexing** | | | | |
| 6 | Graph-based Multilingual Product Retrieval in E-Commerce Search<br><br>[2021] | The traditional models rely on keyword matching which leads to poor result when exact term match is unavailable.<br>Maintaining country specific models increases both operational burden and model iteration risks among countries. | 1. The query encoder employs word-piece tokenization that is robust to spelling errors and allows to share vocabulary between different languages.<br>2. The transformer encoder layer extracts the features of a product and its neighbor queries. The graph convolution layer that aggregates the extracted features to compute the final embeddings for a given product. | The proposed paper can handle multilingual text data and the vocabulary gap issues between queries and product descriptions. Fine tuning of the proposed model based on the retrieval tasks. |
| 7 | User Model-Based Intent-Aware Metrics for Multilingual Search Evaluation<br><br>[2016] | Despite the growing importance of multilingual aspects of web search, no appropriate offline metrics to evaluate its quality are proposed so far. | Metrics investigated in this paper include the state-of-the-art ERR as a baseline and its different modifications, which are based on the special cases of the Dynamic Bayesian Network (DBN) click model. | As demonstrated, the straightforward intent-aware modifications of user models do not take such aspects of user behavior such as users who can be satisfied by documents written in one language, while searching documents in another language into account. |

| | | | | |
|---|---|---|---|---|
| 8 | Query Transformation for Multilingual Product Search<br><br>[2014] | The problem of enabling multilingual product search for a global shopping store has been addressed. | New behavior metrics to directly evaluate machine translation systems based on their performance on the task of multilingual product search. The proposed behavior metrics measure how easily customers can find products that are relevant to their shopping intent using a transformed query. | The proposed QT system's performance is compared with two models:<br>1. A state-of-the-art SMT model for Product Search (PS-SMT).<br>2. AWS Translate, a state-of-the-art NMT model for general translation.<br>The proposed QT model achieves the highest behavior metric scores for both ES→EN and FR→EN: increasing the accuracy by 11.31% and 3.45%, respectively. |
| 9 | Indexing and Weighting of Multilingual and Mixed Documents<br><br>[2014] | Non-English-speaking users are not always able to express terminology in their native languages, especially in scientific domains. Such difficulty forces many authors and scholars to use English terms in order to explain precise concepts, resulting in multilingual queries with both English and the other language. | The proposed technique combines centralized and distributed architectures, taking advantage of their benefits, and trying to minimize their drawbacks. It also uses a variant of the structured query model to re-weight documents into the centralized multilingual sub-collection. | The source multilingual queries are bi-directionally translated into Arabic and English, as explained in the experimental setup. |

| | | | | |
|---|---|---|---|---|
| 10 | Multi-Language Ontology-based Search Engine<br><br>[2010] | Most information retrieval systems are monolingual and more precisely English-based. | This proposed model uses an open-source search engine and a scalable Information Retrieval (IR) library that allows indexing and searching capabilities. It applies the Boolean Model first to select the most relevant documents for the query, then it uses the Vector Space Model for content-based ranking. | The hierarchical structure of the ontology defines the relationship between concepts. Term Vector Translation approach maps statistical information about term usage between languages using techniques which map sets of weights from English to Spanish and vice-versa. |
| 11 | A Multilingual (Chinese, English) Indexing, Retrieval, Searching Search Engine<br><br>[2005] | As more and more people on the Internet are from non-English-speaking countries and Web documents can be found in languages other than English, Multilingual search engine is strongly required for users to be able to retrieve information from multilingual databases. | The proposed system consists of two subsystems: organization-based subsystem and Web-page based subsystem. The process of indexing is to generate index objects and then create the inverted index on resource profiles. In the proposed multilingual architecture, the search engine has as many index tables as the number of languages. | The search engine employs the so called TFxIDF search and ranking algorithm which assigns relevance scores to Web documents using the vector space model. Retrieved documents are ranked in decreasing order of similarity to the query. |

| Chatbot | | | | |
|---|---|---|---|---|
| 12 | Intelligent Chatbot<br><br>[2021] | As numerous chatbot platforms already exist, there are still some problems in building data-driven systems because a huge amount of data is required for their development. | The methodology includes API of Chatbot that will be developed with Cascading style sheet which covers all the styling part and the JavaScript is used for functioning . | A sample chatbot using the methods enlisted with twitch as an online platform that provides a chatbot platform to the online clients. |
| 13 | Design and Development of CHATBOT: A Review<br><br>[2021] | Nowadays, speech and textual conversation are primary communication forms between humans and computers that occur through web applications. The purpose of a CHATBOT is to help answer user queries. | The different methods used are, Parsing, Pattern matching, Artificial Intelligence Markup Language, Chat Script, SQL, Markov Chain, Language Trick, Ontologies. | The CHATBOT utilizes the concepts of and Machine Learning to interact with people virtually. Various design techniques and varying platforms of build Bot are reviewed, followed by the advancement in CHATBOT is presented. |
| 14 | Development of an e-commerce Sales Chatbot<br><br>[2020] | Development of an e-commerce sales chatbot in order to provide customer support and increase sales. | The intent of the proposed solution is to push the boundaries of natural language processing and natural language understanding and take them a step closer to understand context. The biggest challenge in the NLP/NLU as of now, is to develop a chatbot that can render an experience | This modular system is available for more platforms. This present NLU engine trains its classifier from the classified training data provided by the admins. Also, it is based on SVM. Artificial Neural Network can be used to improve the accuracy of the NLU Engine. |

| | | | | |
|---|---|---|---|---|
| 15 | Chatbot System for Cancer Patient<br><br>[2019] | There is no instant response given to the patients; they must wait for the expert's acknowledgment for a longer time. They might also charge an amount for live chat with doctors. | Using NLP processing and sentiment analysis negation level of a complaint is detected. Google API is used for conversion of text-voice and vice-versa. Lexical analysis parsing is involved in analyzing the structure of words.<br>Random - Forest algorithm is used for classification of disease. | An Android application provides answers to be analyzed queries of the user. The medical chat-bots functioning depends on NLP that helps the user to share their problem. |
| 16 | Chatbot: An automated conversation system for educational domain<br><br>[2019] | A person may not react quickly and positively at the time of multiple conversations in each time. Lack of manpower and time difference it is not possible to answer or reply to each student's queries. | Data cleaning removes noise and inconsistencies in the data. Bag-of-words model is used for this retrieval based chatbot. Contextual and semantic based features are extracted which assists in identifying the response keyword and help to carry the contextual knowledge from question to answer. | The proposed chatbot assists in communicating between a visitor and a machine in the domain of education. |

**Table - 2.1** Summary of literature of SmartPill Companion MedApp

# CHAPTER 3

# PROBLEM STATEMENT AND OBJECTIVES

# CHAPTER 3

# PROBLEM STATEMENT AND OBJECTIVES

## 3.1 Problem Statement

Diagnosis is always a concern for the people living in rural areas and for those traveling long distances. The problem arises when the need for some medicine is urgent, and the stores aren't open. In remote and rural areas, where public turnover is a smaller amount, the supply of medicines within the patient's reach may be a critical issue. Automating the flow of pharmacies is an urgent matter such that the need of medicines is no more a problem for anyone, anywhere.

## 3.2 Objectives

- The Vending machine in proposed model is to dispense the medicines that are prescribed by the doctor.
- The proposed MedApp is a bi-lingual application used for connecting the IoT model with the mobile which can send commands through API's over the Wi-Fi.
- The application consists of a medical chatbot which can predict diseases.
- The Chatbot application is to predict the diagnosis based on the symptoms of the user and even give the description of the disease predicted and how to prevent them..
- The objective is to intimate human dialogue to provide a more natural user interface to program.

# CHAPTER 4

# SYSTEM ANALYSIS

# CHAPTER 4

# SYSTEM ANALYSIS

## 4.1 System Specifications

### 4.1.1 Hardware Requirements

- **Processor:** Intel Core3 Quad @ 2.4Ghz on Windows® Vista 64-Bit / Windows® 7 64- Bit / Windows® 8 64-Bit / Windows® 8.1 64-Bit.
- **RAM:** 4GB of RAM
- **Memory:** 256GB Hard drive
- **Keyboard:** MS compatible keyboard
- **Mouse:** MS compatible mouse

### 4.1.2 Software Requirements

- **Operating system:** Windows® Vista 64-Bit / Windows® 7 64-Bit / Windows® 8 64- Bit / Windows® 8.1 64-Bit.
- **Libraries:** OpenCV, NumPy, scikit-learn, scikit-image, TensorFlow 2.0, Keras
- **IDE:** IDLE Python 3.8, Arduino IDE

## 4.2 Tools Used

### ➢ Python 3.8

- **Python** is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. It was created by Guido van Rossum during 1985- 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL).
- This **tutorial** gives enough understanding on **Python programming** language.
- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.

- It supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.
- **Easy-to-learn** − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read** − Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** − Python's source code is fairly easy-to-maintain.
- **A broad standard library** − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable** − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases** − Python provides interfaces to all major commercial databases.
- **GUI Programming** − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable** − Python provides a better structure and support for large programs than shell scripting.

❖ **Libraries used in Python**:

- **Pandas**
  - Pandas is a fast, powerful, flexible and easy to use open-source data analysis and manipulation tool, built on top of the Python programming language.
  - Pandas Series is a one-dimensional labeled array capable of holding data of any type (integer, string, float, python objects, etc.). The axis labels are collectively called indexes. Pandas Series is nothing but a column in an excel sheet. Labels need not be unique but must be a hashable type.

- **Scikit-learn**
  - scikit-learn is an open-source Python library that implements a range of machine learning, pre-processing, cross-validation, and visualization algorithms using a unified interface.
  - Simple and efficient tools for data mining and data analysis. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means, etc.
  - Accessible to everybody and reusable in various contexts.
  - Built on the top of NumPy, SciPy, and matplotlib.
  - Open source, commercially usable – BSD license.

- **Numpy**
  - NumPy is a package that defines a multi-dimensional array object and associated fast math functions that operate on it. It also provides simple routines for linear algebra and fft and sophisticated random-number generation.
  - NumPy replaces both Numeric and Numarray.
  - It is the core library for scientific computing in Python. It provides a high-performance multidimensional array object, and tools for working with these arrays.

## ➢ C Programming

- It is a very powerful and general-purpose language used in programming. We can use C to develop software such as databases, operating systems, compilers, and many more.
- This programming language is excellent to learn for beginners in programming.
- The C language is imperative, procedural, and general-purpose in nature, developed by Dennis M. Ritchie in 1972 at the Bell Telephone for developing the UNIX OS.
- As of now, the C language is one of the most widely used computer languages along with Java, which is mostly used among modern programmers.
- This language helps users comprehend a computer's internal architecture. It assists you in knowing how a computer would store information within and retrieve it.
- This language was initially utilized for the development of systems- particularly those programs that would make up an OS (operating system).

- The C programming language was adopted in the form of a language for system development since it generates codes that run as fast as those codes that exist in the assembly language.
- **Procedural Language:** The execution of the instructions present in a C program happens step by step.
- **Speed:** The C language is much faster as compared to a majority of the programming languages, such as Python, Java, and many more.
- **Portable:** A C program can be moved from any given platform to another one, and we can also run it on that platform without any of the charges.
- **General Purposes:** We can use the C programming language for developing operating systems, databases, embedded systems, etc.

## ➢ Java

- Java Programming is a powerful general-purpose programming language.
- It is used to develop desktop and mobile applications, big data processing, embedded systems, and so on.
- According to Oracle, the company that owns Java, Java runs on 3 billion devices worldwide, which makes Java one of the most popular programming languages.
- Java is a class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible.
- It is a general-purpose programming language intended to let application developers write once, run anywhere meaning that compiled Java code can run on all platforms that support Java without the need for recompilation.
- Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of the underlying computer architecture.
- **Object Oriented:** In Java, everything is an Object. Java can be easily extended since it is based on the Object model.
- **Platform Independent:** Unlike many other programming languages including Cand C++, when Java is compiled, it is not compiled into platform specific machine, rather into platform independent byte code. This byte code is distributed over the web and interpreted by the Virtual Machine (JVM) on whichever platform it is being run on.

- **Simple:** Java is designed to be easy to learn. If you understand the basic concept of OOP Java, it would be easy to master.
- Secure: With Java's secure feature it enables to develop virus-free, tamper-free systems. Authentication techniques are based on public-key encryption.
- **Architecture-neutral:** Java compiler generates an architecture-neutral object file format, which makes the compiled code executable on many processors, with the presence of Java runtime system.
- **Portable:** Being architecture-neutral and having no implementation dependent aspects of the specification makes Java portable. Compiler in Java is written in ANSI C with a clean portability boundary, which is a POSIX subset.
- **Robust:** Java makes an effort to eliminate error prone situations by emphasizing mainly on compile time error checking and runtime checking.
- **Multithreaded:** With Java's multithreaded feature it is possible to write programs that can perform many tasks simultaneously. This design feature allows the developers to construct interactive applications that can run smoothly.
- **Interpreted:** Java byte code is translated on the fly to native machine instructions and is not stored anywhere. The development process is more rapid and analytical since the linking is an incremental and light-weight process.
- **High Performance:** With the use of Just-In-Time compilers, Java enables high performance.
- **Distributed:** Java is designed for the distributed environment of the internet.
- **Dynamic:** Java is considered to be more dynamic than C or C++ since it is designed to adapt to an evolving environment. Java programs can carry extensive amount of run-time information that can be used to verify and resolve accesses to objects on run-time.

## ➢ Visual Studio Code

- Visual Studio Code, also commonly referred to as VS Code, is a source-code editor made by Microsoft for Windows, Linux and macOS.
- Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality.

- Visual Studio Code was first announced on April 29, 2015, by Microsoft at the 2015 Build conference. A preview build was released shortly thereafter.
- On November 18, 2015, the source of Visual Studio Code was released under the MIT License, and made available on GitHub. Extension support was also announced. On April 14, 2016, Visual Studio Code graduated from the public preview stage and was released to the Web.
- Visual Studio Code is a source-code editor that can be used with a variety of programming languages, including Java, JavaScript, Go, Node.js, Python, C++ and Fortran.
- It is based on the Electron framework, which is used to develop Node.js Web applications that run on the Blink layout engine.
- Out of the box, Visual Studio Code includes basic support for most common programming languages. This basic support includes syntax highlighting, bracket matching, code folding, and configurable snippets.
- Visual Studio Code also ships with IntelliSense for JavaScript, TypeScript, JSON, CSS, and HTML, as well as debugging support for Node.js.
- Support for additional languages can be provided by freely available extensions on the VS Code Marketplace.

## ➢ Android Studio

- Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on IntelliJ IDEA.
- On top of IntelliJ's powerful code editor and developer tools, Android Studio offers even more features that enhance your productivity when building Android apps
- **Android Emulator:** The Android Emulator simulates Android devices on your computer so that you can test your application on a variety of devices and Android API levels without needing to have each physical device.
- The emulator provides almost all of the capabilities of a real Android device. You can simulate incoming phone calls and text messages, specify the location of the device, simulate different network speeds, simulate rotation and other hardware sensors, access the Google Play Store, and much more.

- Testing your app on the emulator is in some ways faster and easier than doing so on a physical device. For example, you can transfer data faster to the emulator than to a device connected over USB.
- The emulator comes with predefined configurations for various Android phone, tablet, Wear OS, and Android TV devices.
- **Android Build System:** The Android build system compiles app resources and source code, and packages them into APKs or Android App Bundles that you can test, deploy, sign, and distribute.
- **Gradle:** Android Studio uses Gradle, an advanced build toolkit, to automate and manage the build process, while allowing you to define flexible custom build configurations.
- Each build configuration can define its own set of code and resources, while reusing the parts common to all versions of your app.
- The Android plugin for Gradle works with the build toolkit to provide processes and configurable settings that are specific to building and testing Android applications.
- Gradle and the Android plugin run independent of Android Studio.
- **Layout Editor:** It enables you to quickly build layouts by dragging UI elements into a visual design editor instead of writing layout XML by hand.
- The design editor can preview your layout on different Android devices and versions, and you can dynamically resize the layout to be sure it works well on different screen sizes.

# CHAPTER 5

# DESIGN

# CHAPTER 5

# DESIGN

## 5.1 System Design

**System Design** is the process of designing the architecture, components, and interfaces for a system so that it meets the end-user requirements. System Design for tech interviews is something that can't be ignored! Almost every IT giant whether it be Facebook, Amazon, Google, or any other ask various questions based on System Design concepts such as scalability, load-balancing, caching, etc. in the interview. This specifically designed System Design tutorial will help you to learn and master System Design concepts in the most efficient way from basics to advanced level.

If the broader topic of product development "blends the perspective of marketing, design, and manufacturing into a single approach to product development," then design is the act of taking the marketing information and creating the design of the product to be manufactured. Systems design is therefore the process of defining and developing systems to satisfy specified requirements of the user. The basic study of system design is the understanding of component parts and their subsequent interaction with one another.

## 5.2 Physical Design

The physical design relates to the actual input and output processes of the system. This is explained in terms of how data is input into a system, how it is verified/authenticated, how it is processed, and how it is displayed. In physical design, the following requirements about the system are decided.

1. Input requirement,
2. Output requirements,
3. Storage requirements,
4. Processing requirements,
5. System control and backup or recovery.

Put another way, the physical portion of system design can generally be broken down into three sub-tasks:

1. User Interface Design
2. Data Design
3. Process Design

## ➢ High Level Design (HLD) :

High Level Design in short HLD is the general system design means it refers to the overall system design. It describes the overall description/architecture of the application. It includes the description of system architecture, data base design, brief description on systems, services, platforms and relationship among modules. It is also known as macro level/system design. It is created by solution architect. It converts the Business/client requirement into High Level Solution. It is created first means before Low Level Design.

## ➢ Low Level Design (LLD) :

Low Level Design in short LLD is like detailing HLD means it refers to component-level design process. It describes detailed description of each and every module means it includes actual logic for every system component and it goes deep into each modules specification. It is also known as micro level/detailed design. It is created by designers and developers. It converts the High Level Solution into Detailed solution. It is created second means after High Level Design.

## 5.3 Design of Proposed System

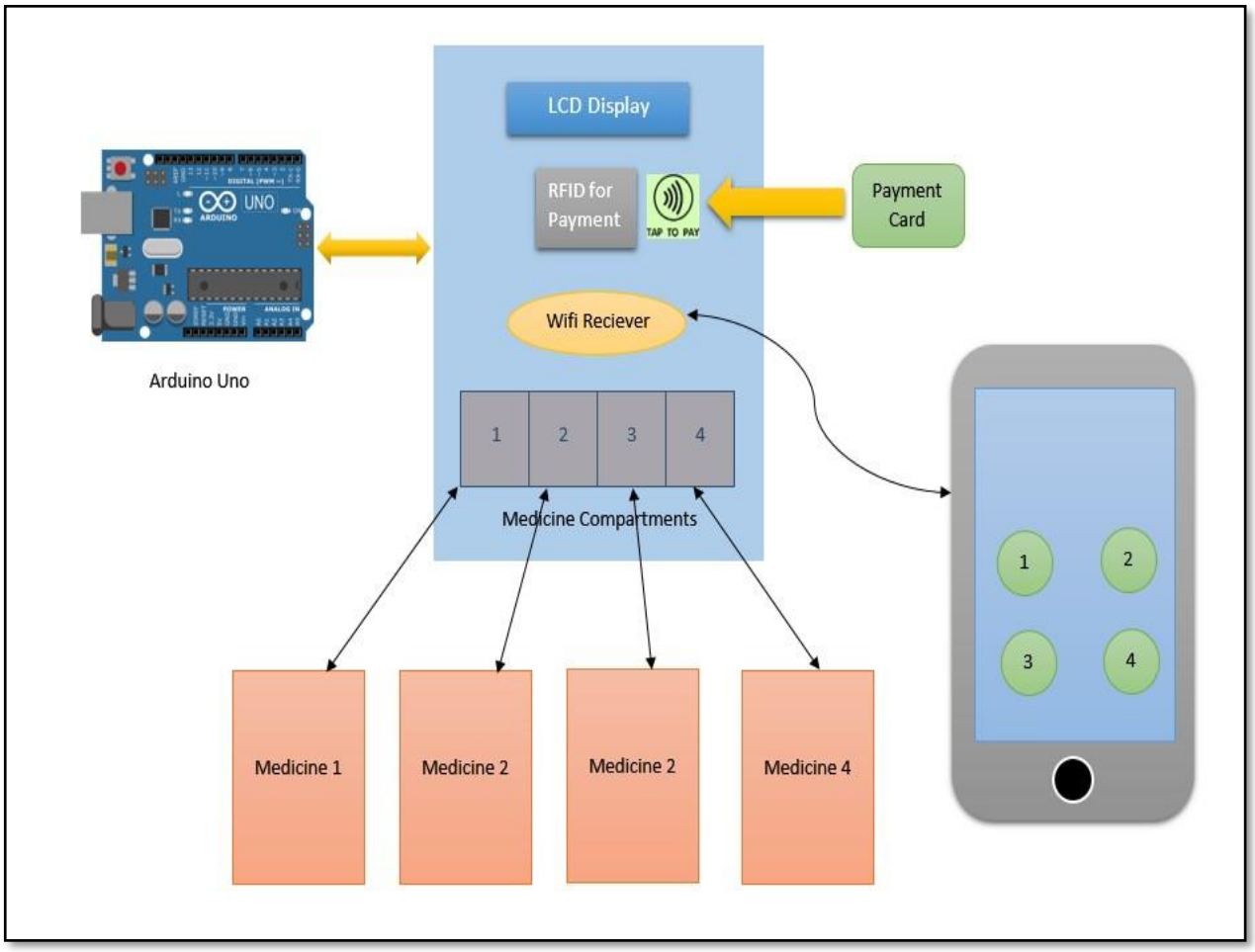


**Fig - 5.1** Diagram for the Proposed System

This the implementation design of the proposed model of the SmartPill Companion MedApp in the figure. It shows the idea how the IoT model of the medicine machine is created and how the IoT model is connected through the mobile application. In this proposed system the vending machine has an LCD display on which the machine displays the medicine name and also displays a message to swipe card for initiating the payment and making the payment successfully. The payment is authenticated by a RFID receiver For dispensing the medicines the machine is connected with the app through the Wifi module Node MCU and the compartments are made using the servo motors to open the medicine compartments.


**Fig - 5.2** Design for the Chatbot

A chatbot or chatterbot is a software application used to conduct an on-line chat conversation via text, in of providing direct contact with a live human agent. In this is the basic design of the chatbot that is created as an web app through which the patient can interact with the med Bot by giving the personal details and symptoms.

## 5.4 Data Flow Diagram(HLD)

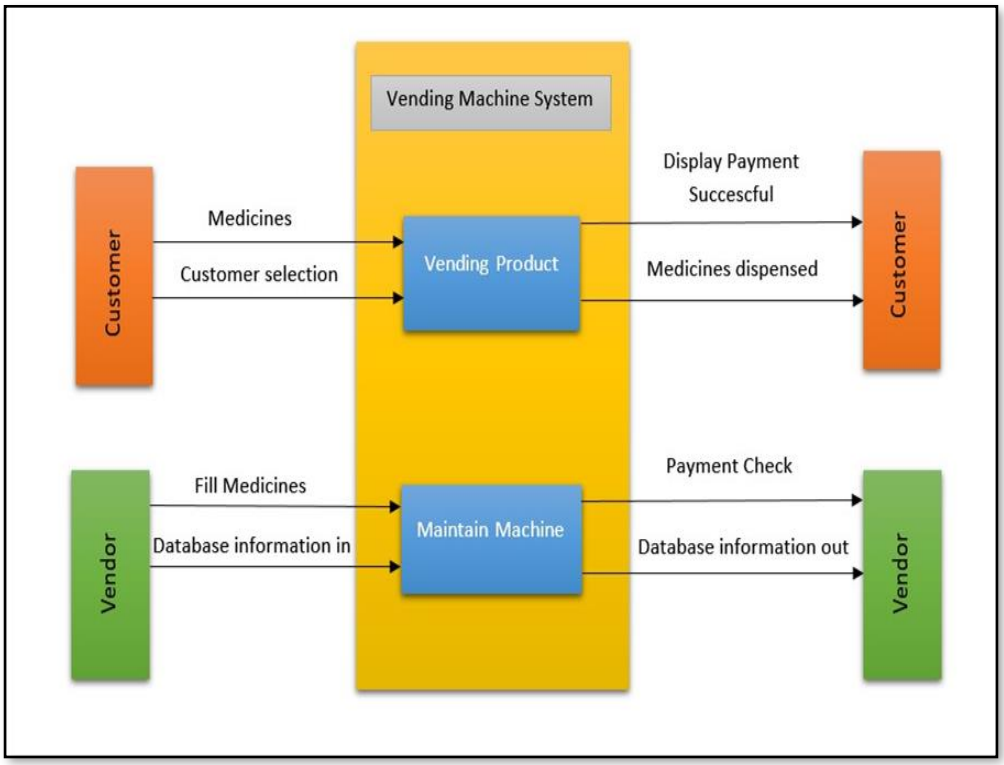


**Fig – 5.3** Data Flow Diagram of the Proposed System

A data-flow diagram is a way of representing a flow of data through a process or a system (usually an information system). The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow, there are no decision rules and no loops. The above figure shows the dataflow diagram for the vending machine interactions with the user. This shows how the customer and the vendor has access to the machine. The customer can select medicines and after successful payment the medicines will be dispensed to the customer. The vendor can fill the machine with medicines after checking the database. The vendor can check the payment status as well in the background and keep a track on the transactions.

## 5.5 Class Diagram(LLD)

Class diagram describes the attributes and operations of a class and also the constraints imposed on the system as shown in the Fig.

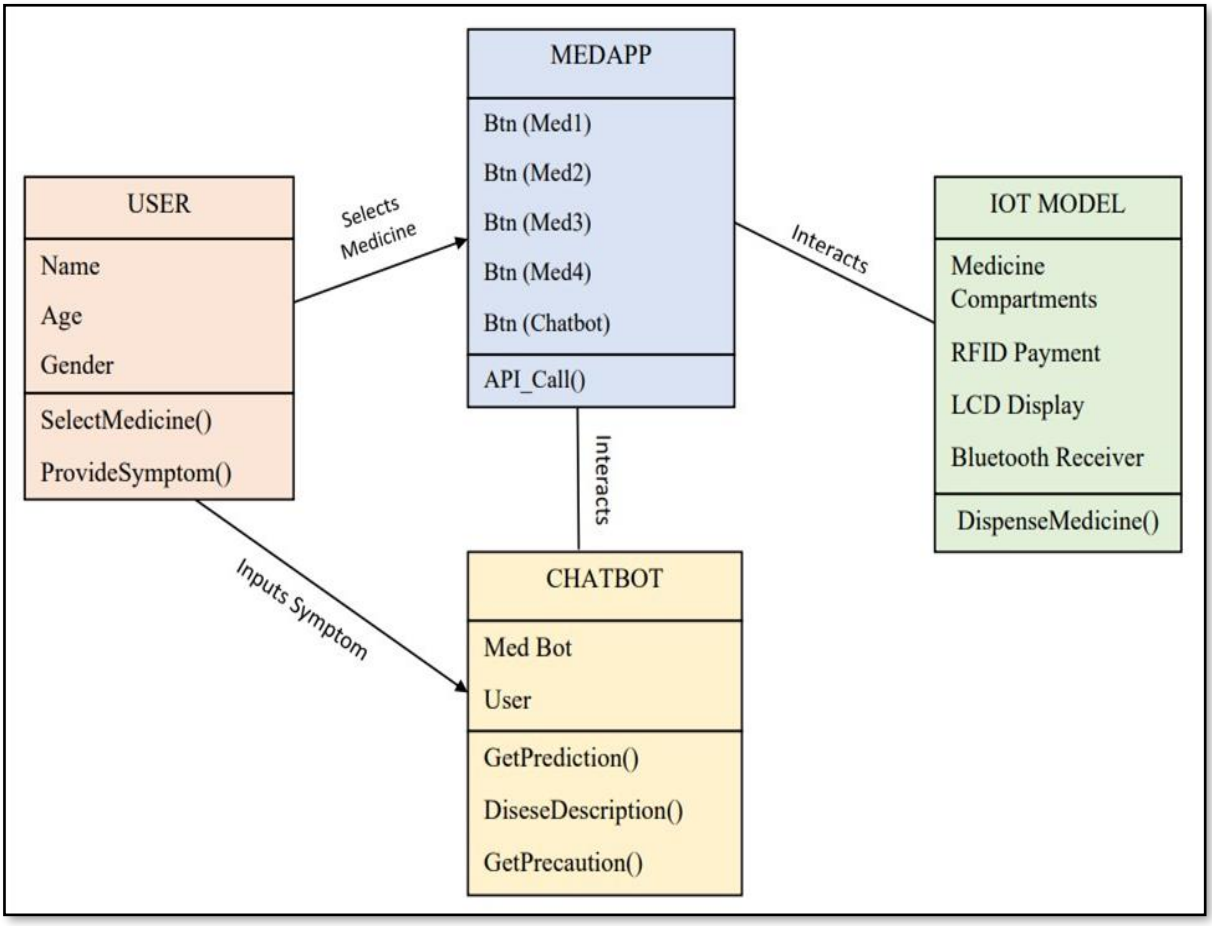


**Fig - 5.4** Class Diagram for the Proposed System

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application. Class diagram describes the attributes and operations of a class and also 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 object-oriented languages. The class diagram shown above in the figure shows how mapping is done with all the objects of the class of the proposed system and how the operations are done.

## 5.6 ER Diagram(LLD)

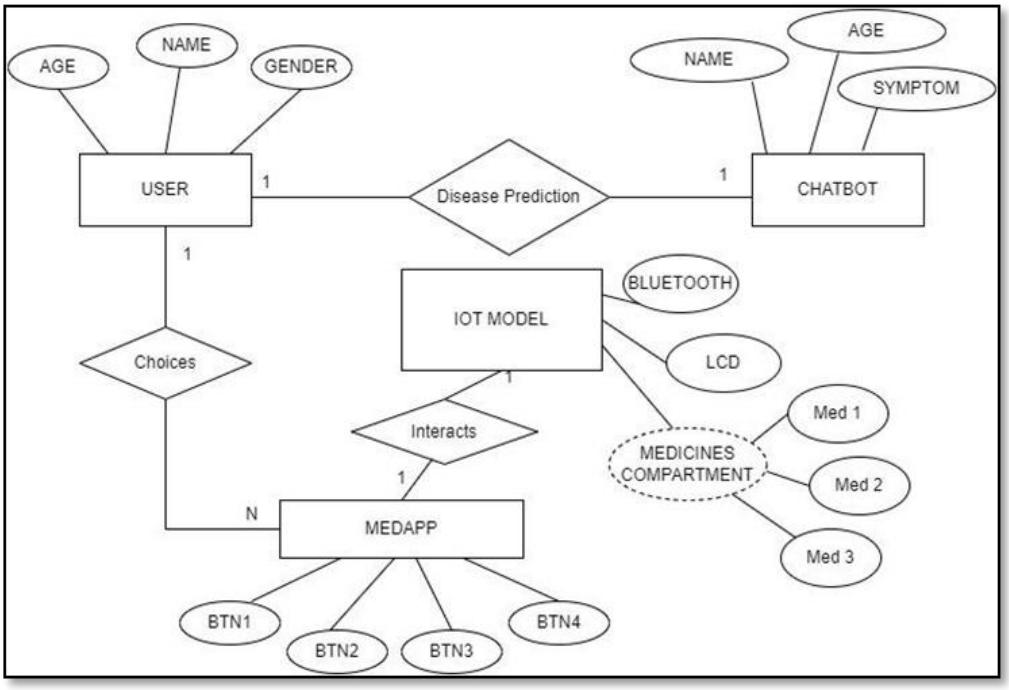


**Fig- 5.5** ER Diagram for the Proposed System

**ER Diagram** stands for Entity Relationship Diagram, also known as ERD is a diagram that displays the relationship of entity sets stored in a database. In other words, ER diagrams help to explain the logical structure of databases. ER diagrams are created based on three basic concepts: entities, attributes and relationships. The purpose of ER Diagram is to represent the entity framework infrastructure. The above Entity-Relationship (ER) Diagram of the proposed system shows the relationship of the entities with each other how many attributes are there in each entity. The main attributes here are USER, CHATBOT, IOT MODEL, MEDAPP.

## 5.7 Activity Diagram(LLD)

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system as shown in Fig.

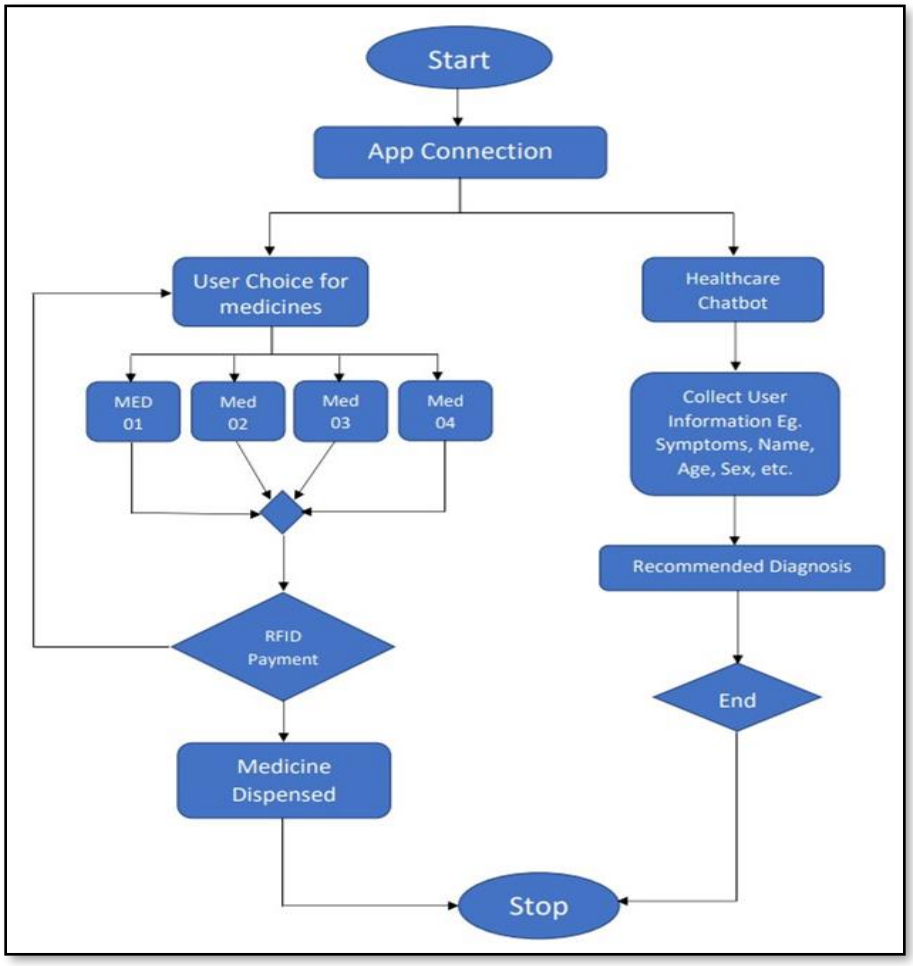


**Fig - 5.6** Activity Diagram for the Proposed System

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. 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. The above the activity diagram for the proposed system shows the different activities going on in the system and the flow of events how the activities are taking place.

# CHAPTER 6

# IMPLEMENTATION

# CHAPTER 6

# IMPLEMENTATION

## 6.1 Chatbot Implementation

### 6.1.1 Training and Test Datasets

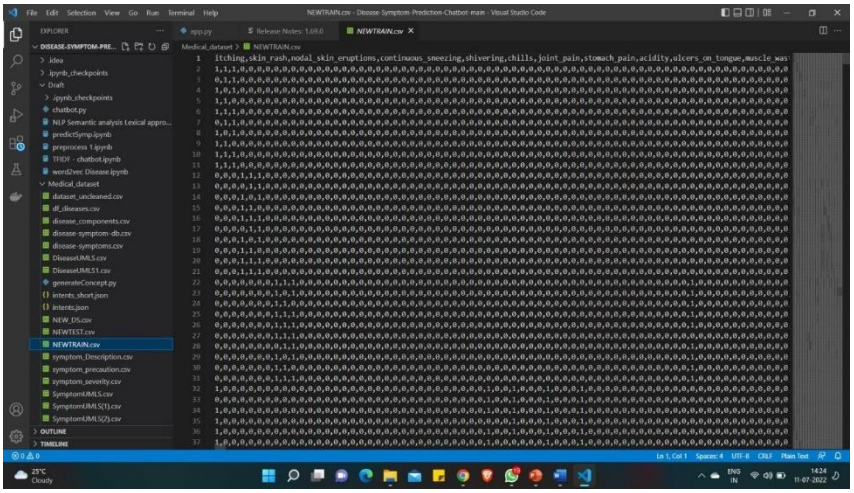

**Fig- 6.1** Training Dataset

This is the trained dataset which we have collected from publicly accessible databases Kaggle. The first component of building the chatbot to train and gather our datasets.

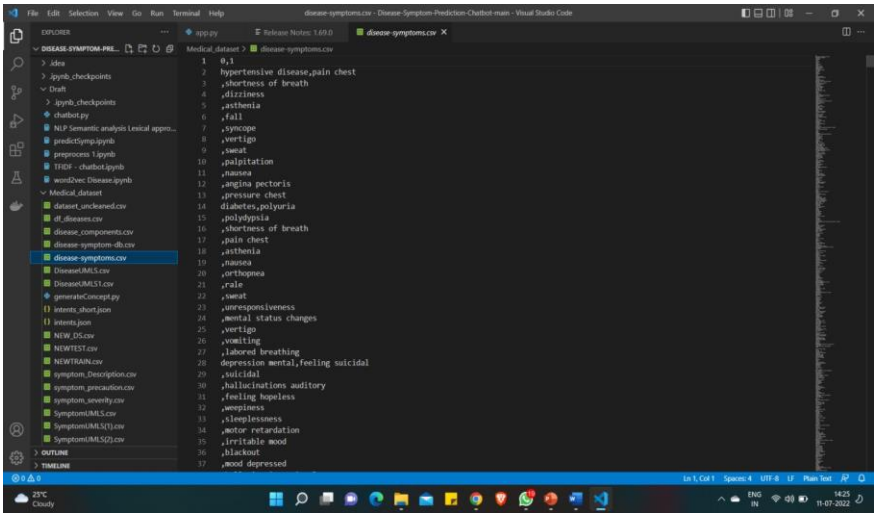

**Fig – 6.2** Disease Symptom Dataset

The above dataset contains the different variety of symptom where patients can provide them as an input to the chatbot.

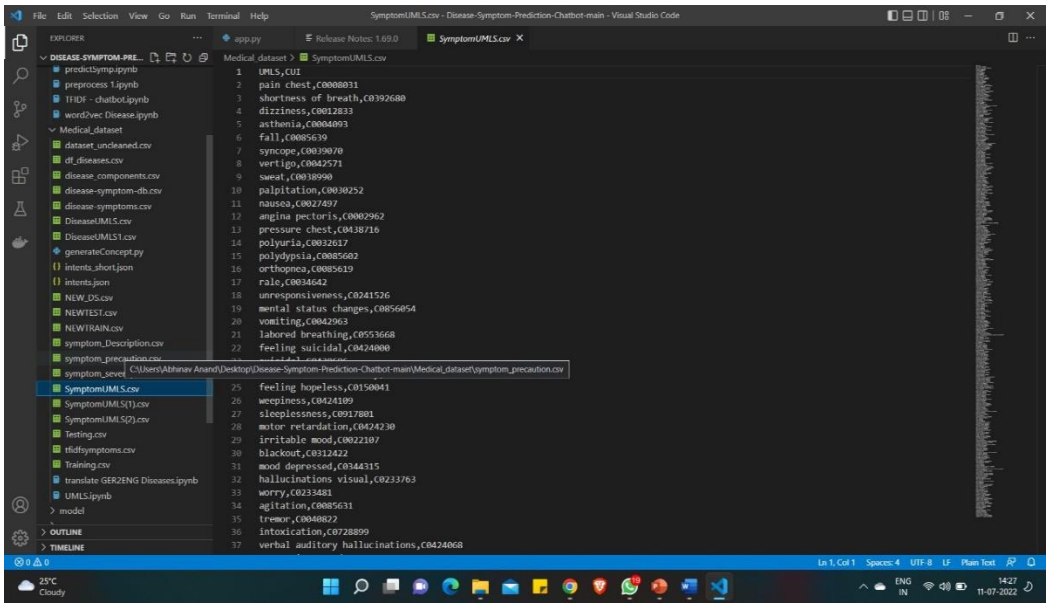


**Fig-6.3** Symptoms UML

The above dataset contains the symptoms UML that is intended to provide a standard way to visualize the design of the system.

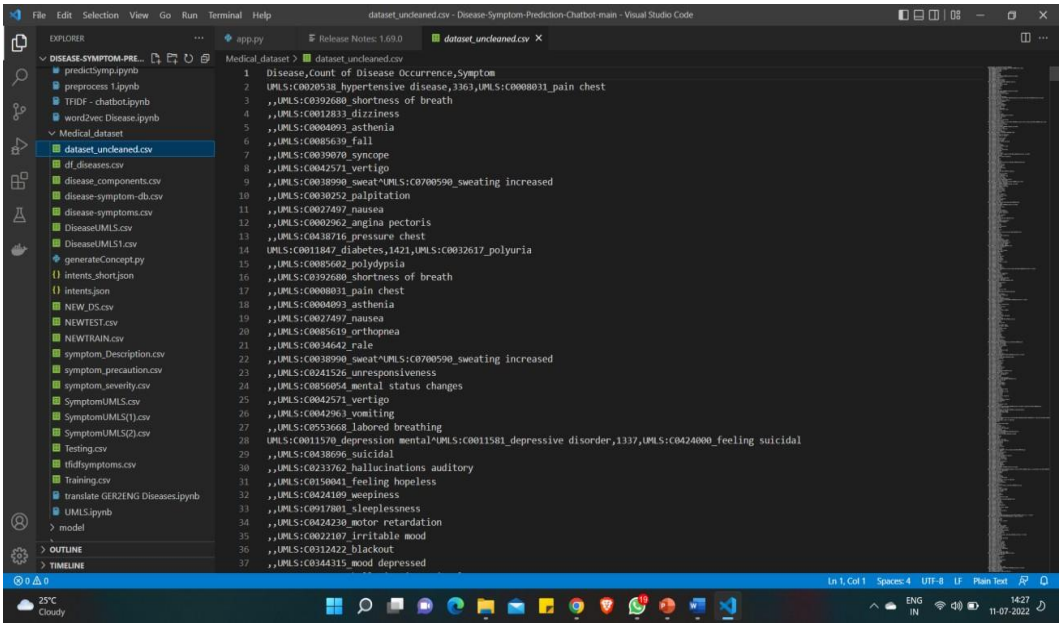


**Fig.-6.4** Dataset Uncleaned

The above dataset is downloaded from the Kaggle website where the data is pre-processed needs to be processed to remove the extra or unnecessary data.

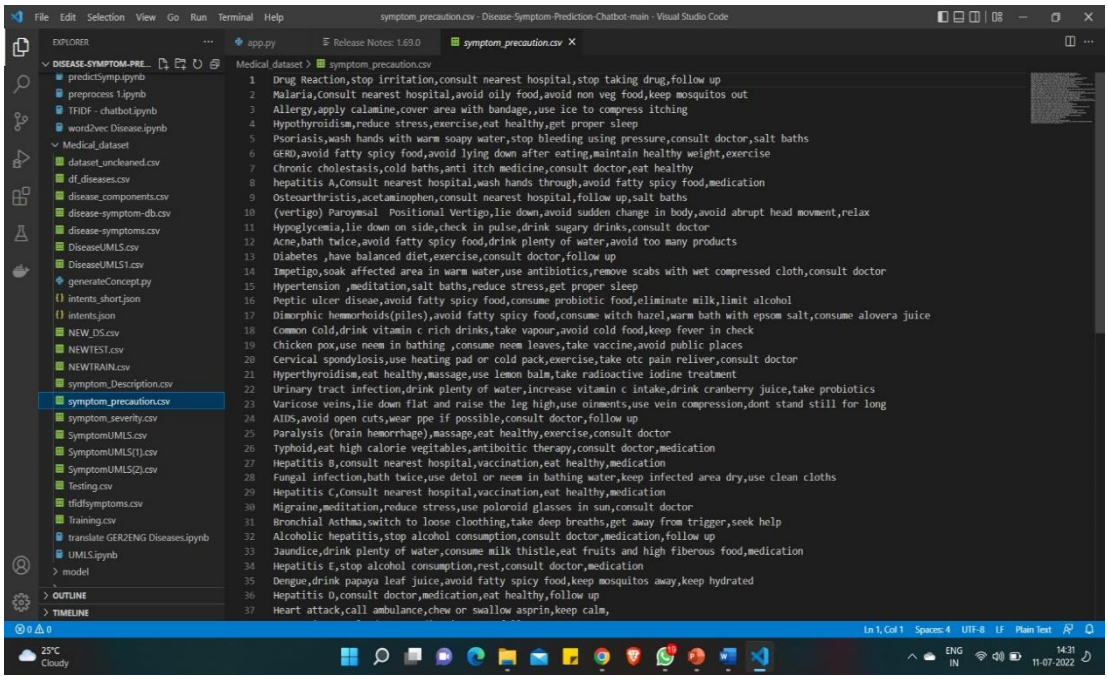


**Fig-6.5** Symptom Precaution

The dataset contains the different variety of precautions to be taken for the required diagnosis and provided to the user.

## 6.1.2 Chatbot Implementation Code:

- **Chatbot UI**

```
<body>
  <!-- partial:index.partial.html -->
  <section class="msger">
    <header class="msger-header">
      <div class="msger-header-title">
        MedApp
      </div>
    </header>
    <main class="msger-chat">
      <div class="msg left-msg">
        <div class="msg-img" style="background-image: url(https://image.flaticon.com/icons/svg/327/327779.svg)"></div>
        <div class="msg-bubble">
          <div class="msg-info">
            <div class="msg-info-name">MedApp</div>
          </div>
          <div class="msg-text">
            Hello,I am Chatbot and I will be happy to help diagnose your disease.
          </div>
        </div>
      </div>
      <div class="msg left-msg">
        <div class="msg-img" style="background-image: url(https://image.flaticon.com/icons/svg/327/327779.svg)"></div>

        <div class="msg-bubble">
          <div class="msg-info">
            <div class="msg-info-name">MedApp</div>
          </div>
          <div class="msg-text">
            To start, we need to ask some basic questions, tap OK to continue !
          </div>
        </div>
      </div>
    </main>
    <form class="msger-inputarea">
      <input type="text" class="msger-input" id="textInput" placeholder="Enter your message...">
      <button type="submit" class="msger-send-btn">Send</button>
    </form>
```

**Fig- 6.6** UI Design of the Chatbot

This is the UI of the chatbot screen in which the user can interact with the chatbot for the medical assistance.

- **Input Symptoms**

```
symp = []
disease = []
for i in range(len(df_tr)):
    symp.append(df_tr.columns[df_tr.iloc[i] == 1].to_list())
    disease.append(df_tr.iloc[i, -1])

# # I- GET ALL SYMPTOMS
all_symp_col = list(df_tr.columns[:-1])
def clean_symp(sym):
    return sym.replace('_', ' ').replace('.1', '').replace('(typhos)', '').replace('yellowish', 'yellow').replace(
        'yellowing', 'yellow')
all_symp = [clean_symp(sym) for sym in (all_symp_col)]
def preprocess(doc):
    nlp_doc = nlp(doc)
    d = []
    for token in nlp_doc:
        if (not token.text.lower() in STOP_WORDS and token.text.isalpha()):
            d.append(token.lemma_.lower())
    return ' '.join(d)
all_symp_pr = [preprocess(sym) for sym in all_symp]
# associate each processed symp with column name
col_dict = dict(zip(all_symp_pr, all_symp_col))
```

**Fig-6.7** Fetched Symptoms from the UI

In this section the user can provide his symptoms to the chatbot.

- **Disease Prediction:**

```
# IV- Prediction Model (KNN)
# load model
knn_clf = joblib.load('model/knn.pkl')
severityDictionary = dict()
description_list = dict()
precautionDictionary = dict()
def getDescription():
    global description_list
    with open('Medical_dataset/symptom_Description.csv') as csv_file:
        csv_reader = csv.reader(csv_file, delimiter=',')
        line_count = 0
        for row in csv_reader:
            _description = {row[0]: row[1]}
            description_list.update(_description)
def getSeverityDict():
    global severityDictionary
    with open('Medical_dataset/symptom_severity.csv') as csv_file:

        csv_reader = csv.reader(csv_file, delimiter=',')
        line_count = 0
        try:
            for row in csv_reader:
                _diction = {row[0]: int(row[1])}
                severityDictionary.update(_diction)
        except:
            pass
def getprecautionDict():
    global precautionDictionary
    with open('Medical_dataset/symptom_precaution.csv') as csv_file:
        csv_reader = csv.reader(csv_file, delimiter=',')
        line_count = 0
        for row in csv_reader:
            _prec = {row[0]: [row[1], row[2], row[3], row[4]]}
            precautionDictionary.update(_prec)
```

**Fig-6.8** Disease Prediction Code

Here the chatbot predicts the disease based on the symptoms provided by the user.

## ➢ Predicted Disease Description and Precautions:

```
            session['step'] = "Description"
            session["disease"] = result[0]
            return "Well Mr/Ms " + session["name"] + ", you may have " + result[
                0] + ". Tap D to get a description of the disease ."
        else:
            session['step'] = "Q_C"  # test if user want to continue the conversation or not
            return "can you specify more what you feel or Tap q to stop the conversation"
    if session['step'] == "Description":
        y = {"Name": session["name"], "Age": session["age"], "Gender": session["gender"], "Disease": session["disease"],
             "Sympts": session["all"]}
        write_json(y)
        session['step'] = "Severity"
        if session["disease"] in description_list.keys():
            return description_list[session["disease"]] + " \n <br>  How many days have you had symptoms?"
        else:
            if " " in session["disease"]:
                session["disease"] = session["disease"].replace(" ", "_")
            return "please visit <a href='" + "https://en.wikipedia.org/wiki/" + session["disease"] + "'>  here  </a>"
    if session['step'] == "Severity":
        session['step'] = 'FINAL'
        if calc_condition(session["all"], int(s)) == 1:
            return "you should take the consultation from doctor <br> Tap q to exit"
        else:
            msg = 'Nothing to worry about, but you should take the following precautions :<br> '
            i = 1
            for e in precautionDictionary[session["disease"]]:
                msg += '\n ' + str(i) + ' - ' + e + '<br>'
                i += 1
            msg += ' Tap q to end'
            return msg
```

**Fig- 6.9** Predicted Disease Description and Prediction Output

In this section the chatbot provides detailed description about the predicted disease and provides with the precautions to be taken care.

## 6.2 Arduino Implementation

### 6.2.1 Arduino IDE Code

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(13, 12, 8, 7, 5, 4);
#include <SoftwareSerial.h>
SoftwareSerial Rfidserial(2, 3);
#include<Servo.h>
Servo Myservo1;
Servo Myservo2;
Servo Myservo3;
Servo Myservo4;
String Rx_data_arr;
unsigned char
RFID_TAG[6][9]={"xxxxxxxx","14853278","09645355","09645377","09645398"};//23250046
char card[]={'0','0','0','0'};
int Tag_Detect_Flag = 0,Card_Detect=0;
void RFID_Tag_Compare_Select();
int Rfid_flag;
//=================================================================
==================
//                    SETUP
//=================================================================
==================
void setup()
{
  Serial.begin(9600);
  Rfidserial.begin(9600);
  lcd.begin(16,2);
  Myservo1.attach(9);
  Myservo2.attach(10);
```

```
  Myservo3.attach(11);
  Myservo4.attach(6);
   Myservo1.write(45);
   Myservo2.write(45);
   Myservo3.write(45);
   Myservo4.write(45);
   delay(2000);
   Myservo1.write(0);
   Myservo2.write(0);
   Myservo3.write(0);
   Myservo4.write(0);
   lcd.setCursor(0,0);
   lcd.print("    WELCOME     ");
   delay(2000);
}
void loop()
{
    lcd.setCursor(0,0);
    lcd.print("    WELCOME     ");
    if(Serial.available()>0)
    {
        char Rx_data=Serial.read();
        //Serial.println(Rx_data);
        if(Rx_data=='2')
        {
            lcd.setCursor(0,0);
            lcd.print("   SWIPE CARD    ");
            while(Rfid_flag==0)
            {
                if(Rfidserial.available()>0)
                {
```

```
            Rfid_flag=1;
            Rx_data_arr = Rfidserial.readString();
     RFID_Tag_Compare_Select();
      }

    }
    Rfid_flag=0;
    if(Card_Detect==1)
    {
       lcd.setCursor(0,0);
       lcd.print("  VALID CARD  ");
       lcd.setCursor(0,1);
       lcd.print("    DOLO 650    ");
       Myservo1.write(45);
       delay(3000);
       Myservo1.write(0);
    }
    else
    {
       lcd.setCursor(0,0);
       lcd.print("  INVALID CARD  ");
       delay(2000);
    }

}
else if(Rx_data=='3')

{
     lcd.setCursor(0,0);
    lcd.print("   SWIPE CARD    ");
    while(Rfid_flag==0)
    {
       if(Rfidserial.available()>0)
```

```
        {
          Rfid_flag=1;
          Rx_data_arr = Rfidserial.readString();
      Serial.println(Rx_data_arr);

     RFID_Tag_Compare_Select();
      }
    }
    Rfid_flag=0;
    if(Card_Detect==1)
    {
      lcd.setCursor(0,0);
      lcd.print("  VALID CARD  ");
      lcd.setCursor(0,1);
      lcd.print("     CROCIN     ");
      Myservo2.write(45);
      delay(3000);
      Myservo2.write(0);

    }
    else
    {
      lcd.setCursor(0,0);
      lcd.print("  INVALID CARD  ");


      delay(2000);
    }
  }
  else if(Rx_data=='4')
  {
    lcd.setCursor(0,0);
    lcd.print("   SWIPE CARD    ");
```

```
    while(Rfid_flag==0)
    {
       if(Rfidserial.available()>0)
       {
          Rfid_flag=1;
          Rx_data_arr = Rfidserial.readString();
     Serial.println(Rx_data_arr);
     RFID_Tag_Compare_Select();
      }
    }
    Rfid_flag=0;
    if(Card_Detect==1)
    {
       lcd.setCursor(0,0);
       lcd.print("  VALID CARD  ");
       lcd.setCursor(0,1);
       lcd.print("  REMDESIVIR  ");
       Myservo3.write(45);
       delay(3000);
       Myservo3.write(0);
    }
    else
    {
       lcd.setCursor(0,0);
       lcd.print("  INVALID CARD  ");
       delay(2000);
    }
}
else if(Rx_data=='5')
{
    lcd.setCursor(0,0);
```

```
        lcd.print("   SWIPE CARD    ");
        while(Rfid_flag==0)
        {
           if(Rfidserial.available()>0)
           {
              Rfid_flag=1;
              Rx_data_arr = Rfidserial.readString();
          Serial.println(Rx_data_arr);
RFID_Tag_Compare_Select();
            }
        }
        Rfid_flag=0;
        if(Card_Detect==1)
        {
           lcd.setCursor(0,0);
           lcd.print("  VALID CARD  ");
           lcd.setCursor(0,1);
           lcd.print("  PARACETAMOL  ");
           Myservo4.write(45);
           delay(3000);
           Myservo4.write(0);
        }
        else
        {
           lcd.setCursor(0,0);
           lcd.print("  INVALID CARD  ");
           delay(2000);
        }
      }
      lcd.clear();
```

```
    }
 }
 void RFID_Tag_Compare_Select( void )
 {
  unsigned char i,j;
  RFID_TAG[0][0] = Rx_data_arr[0];
  RFID_TAG[0][1] = Rx_data_arr[1];
  RFID_TAG[0][2] = Rx_data_arr[2];
  RFID_TAG[0][3] = Rx_data_arr[3];
  RFID_TAG[0][4] = Rx_data_arr[4];
  RFID_TAG[0][5] = Rx_data_arr[5];
  RFID_TAG[0][6] = Rx_data_arr[6];
  RFID_TAG[0][7] = Rx_data_arr[7];
  Tag_Detect_Flag = 0;
  Card_Detect = 0;
  for( i=1; i<=4; i++ )
  {
   j = 0;
   while(1)
   {
    if( RFID_TAG[0][j] == RFID_TAG[i][j] )
    {
      j++;
      if( j == 8 )
      {
        Tag_Detect_Flag = 1;
        break;
      }
    }
    else
      break;
    }
```

```
    if( Tag_Detect_Flag == 1 )
    {
      Card_Detect = i;
      break;
    }
  }
}
```

## 6.3 Application Implementation

### 6.3.1 Block Leve Figures of the Application

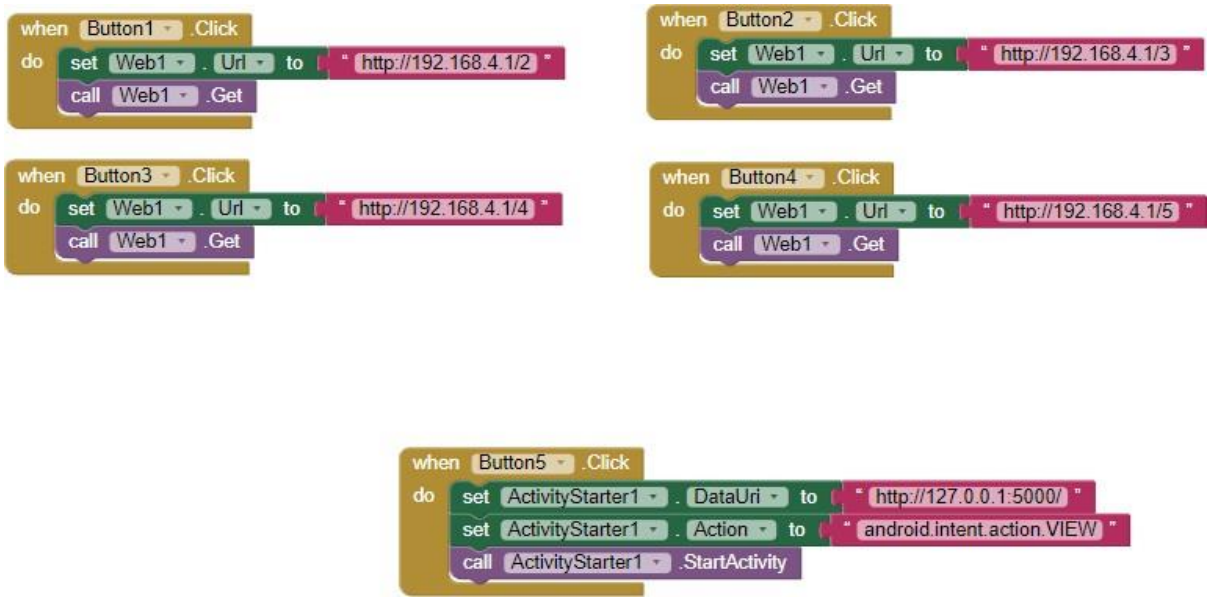


**Fig - 6.10** Button Blocks

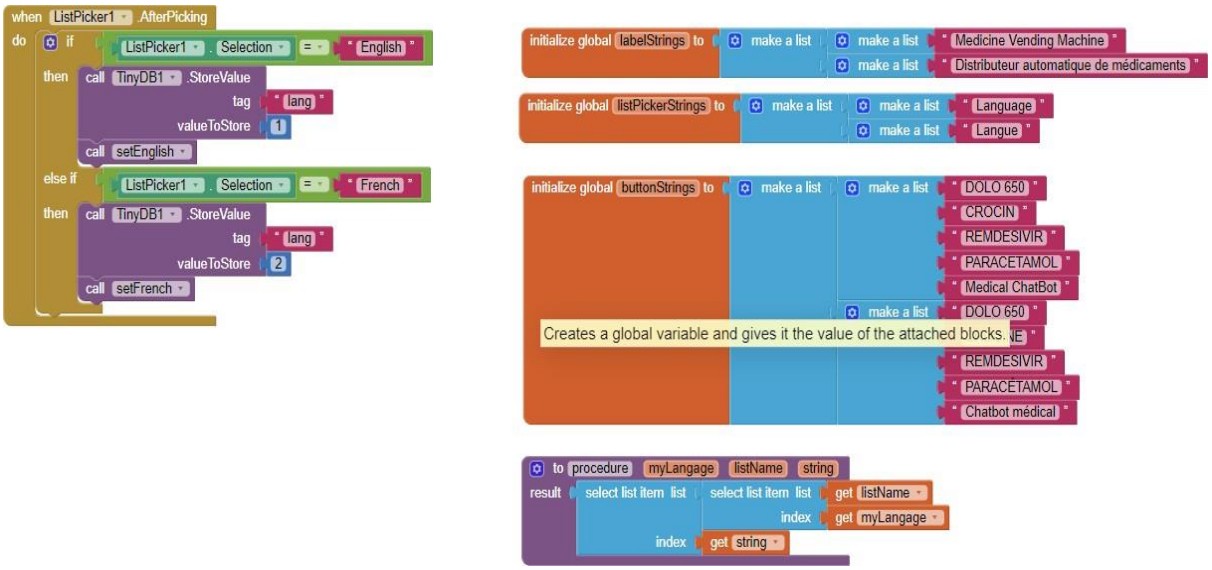


**Fig - 6.11** Medicine Dispensing Button Blocks

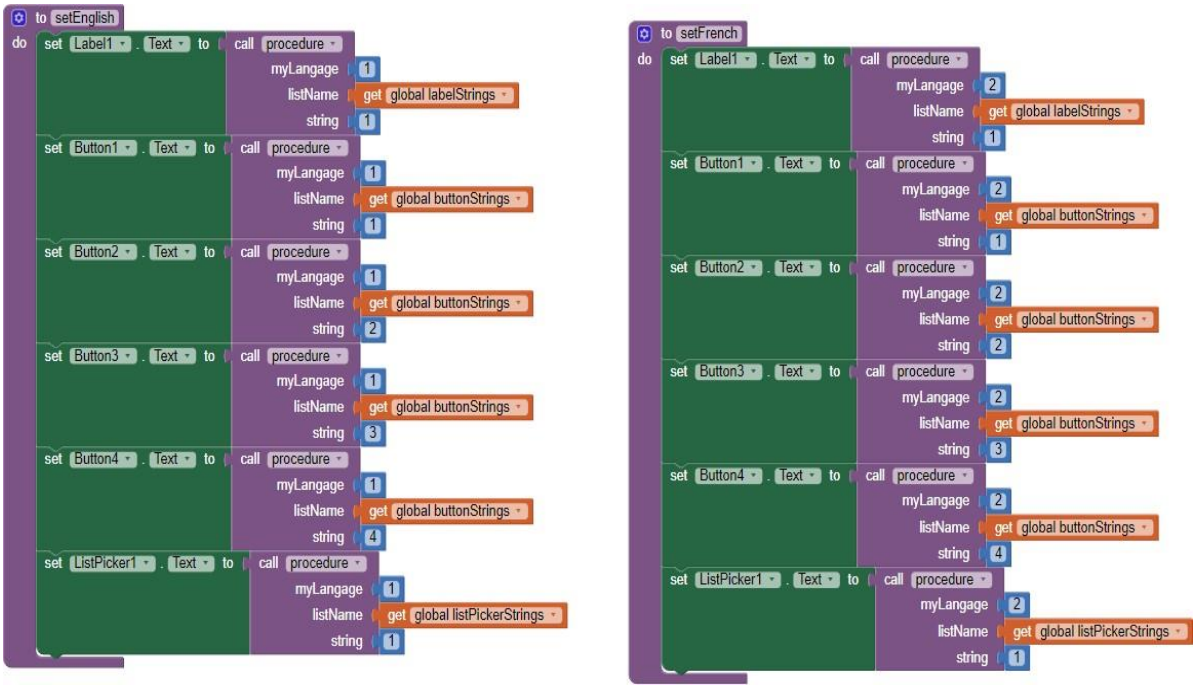


**Fig - 6.12** Language Conversion Blocks

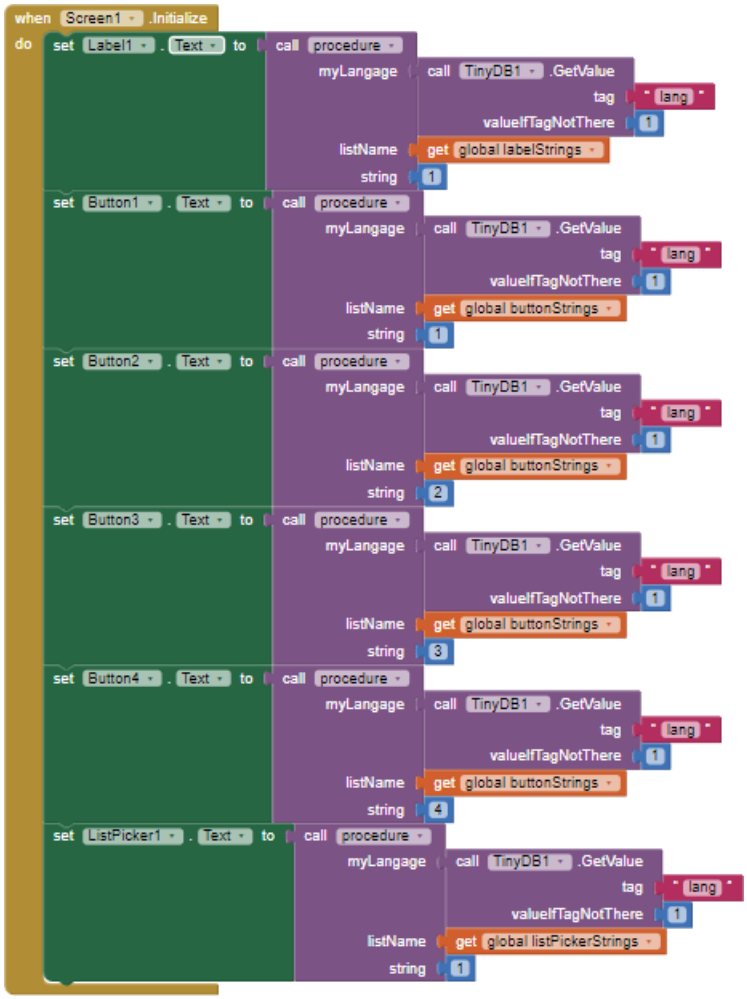


**Fig - 6.13** Last Saved Version in Tiny DB Block

### 6.3.2 Java Code of the Application

```
package org.appinventor;
import com.google.appinventor.components.runtime.HandlesEventDispatching;
import com.google.appinventor.components.runtime.EventDispatcher;
import com.google.appinventor.components.runtime.Form;
import com.google.appinventor.components.runtime.Component;
...
...
import com.google.appinventor.components.runtime.Web;
import com.google.appinventor.components.runtime.ActivityStarter;
import com.google.appinventor.components.runtime.TinyDB;
class Screen1 extends Form implements HandlesEventDispatching {
  private HorizontalArrangement HorizontalArrangement1;
  private Label Label1;
  private HorizontalArrangement HorizontalArrangement2;
  private HorizontalArrangement HorizontalArrangement11;
  ...
  ...
  ...
  private ListPicker ListPicker1;
  private Web Web1;
  private ActivityStarter ActivityStarter1;
  private TinyDB TinyDB1;
  protected void $define() {
    this.AppName("SmartPill Companion");
    this.BackgroundImage("136920.jpg");
    this.Scrollable(true);
    this.Title("SmartPill Companion");
    HorizontalArrangement1 = new HorizontalArrangement(this);
    ...
    ...
```

```
Label1.BackgroundColor(0xFF888888);
Label1.FontBold(true);
Label1.FontSize(20);
Label1.Text("MEDICINE VENDING MACHINE");
HorizontalArrangement2 = new HorizontalArrangement(this);
...
...
...
HorizontalArrangement3 = new HorizontalArrangement(this);
HorizontalArrangement3.Width(LENGTH_FILL_PARENT);
VerticalArrangement2 = new VerticalArrangement(HorizontalArrangement3);
HorizontalScrollArrangement1 = new HorizontalScrollArrangement();
Button1 = new Button(HorizontalArrangement3);
Button1.BackgroundColor(0xFF444444);
...
...
...
Button1.TextColor(0xFFFFFFFF);
VerticalArrangement3 = new VerticalArrangement(HorizontalArrangement3);
Button2 = new Button(HorizontalArrangement3);
Button2.BackgroundColor(0xFF444444);
Button2.FontBold(true);
Button2.Height(100);
Button2.Width(120);
Button2.Shape(1);
Button2.Text("CROCIN");
Button2.TextColor(0xFFFFFFFF);
HorizontalArrangement8 = new HorizontalArrangement(this);
…
HorizontalArrangement5.Width(LENGTH_FILL_PARENT);
VerticalArrangement1 = new VerticalArrangement(HorizontalArrangement5);
```

```
HorizontalScrollArrangement2 = new HorizontalScrollArrangement();
Button3 = new Button(HorizontalArrangement5);
Button3.BackgroundColor(0xFF444444);
Button3.FontBold(true);
Button3.Height(100);
Button3.Width(120);
Button3.Shape(1);
Button3.Text("REMDESIVIR");
Button3.TextColor(0xFFFFFFFF);
VerticalArrangement4 = new VerticalArrangement(HorizontalArrangement5);
Button4 = new Button(HorizontalArrangement5);
Button4.BackgroundColor(0xFF444444);
Button4.FontBold(true);
Button4.Height(100);
Button4.Width(120);
Button4.Shape(1);
Button4.Text("PARAACETEMOL");
Button4.TextColor(0xFFFFFFFF);
HorizontalArrangement12 = new HorizontalArrangement(this);
HorizontalArrangement13 = new HorizontalArrangement(this);
Image2 = new Image(this);
Image2.Height(40);
HorizontalArrangement14 = new HorizontalArrangement(this);
...
Button5 = new Button(HorizontalArrangement14);
Button5.BackgroundColor(0xFF444444);
Button5.Shape(1);
Button5.Text("Medical ChatBot");
Button5.TextColor(0xFFFFFFFF);
Image1 = new Image(this);
Image1.Height(40);
```

```
    HorizontalArrangement15 = new HorizontalArrangement(this);
    HorizontalArrangement18 = new HorizontalArrangement(HorizontalArrangement15);
    ...
    ...
    HorizontalArrangement21.Width(60);
    ListPicker1 = new ListPicker(HorizontalArrangement15);
    ...
    ...
    ListPicker1.TextColor(0xFFFFFFFF);
    Web1 = new Web(this);
    ActivityStarter1 = new ActivityStarter(this);
    TinyDB1 = new TinyDB(this);
  }
  public boolean dispatchEvent(Component component, String componentName,
  String eventName, Object[] params){
    return false;
  }
}
```

# CHAPTER 7

# TESTING AND RESULTS

# CHAPTER 7

# TESTING AND RESULTS

## 7.1 Test Cases

| Test Case ID | Test Case description | Expected Output | Actual Output | Remarks |
|---|---|---|---|---|
| 1 | Payment using valid RFID card. | Payment Successful. Displays the name of the medicine. | Payment Successful. Displays the name of the medicine. | Medicine compartment opens. |
| 2 | Payment using invalid RFID card. | Payment Unsuccessful. Displays Invalid Card. | Payment Unsuccessful. Displays Invalid Card. | Medicine compartment does not open. |
| 3 | Chatbot disease prediction with valid symptoms input. | Disease correctly predicted. | Disease correctly predicted. | Predicts the disease and suggests doctor visit and tells description of disease. |
| 4 | Chatbot disease prediction with invalid symptoms input. | Disease is not predicted. | Disease is not predicted. | Index error generated while predicting the disease. |

**Table - 7.1** Test Case Data

## 7.2 Results

### 7.2.1 Snapshot of ChatBot:


**Fig - 7.1** Chatbot asking basic details of patient

The above snapshot is interaction between the chatbot and the patients, where the chatbot is asking the basic details of the patient like name, gender, age and so on.

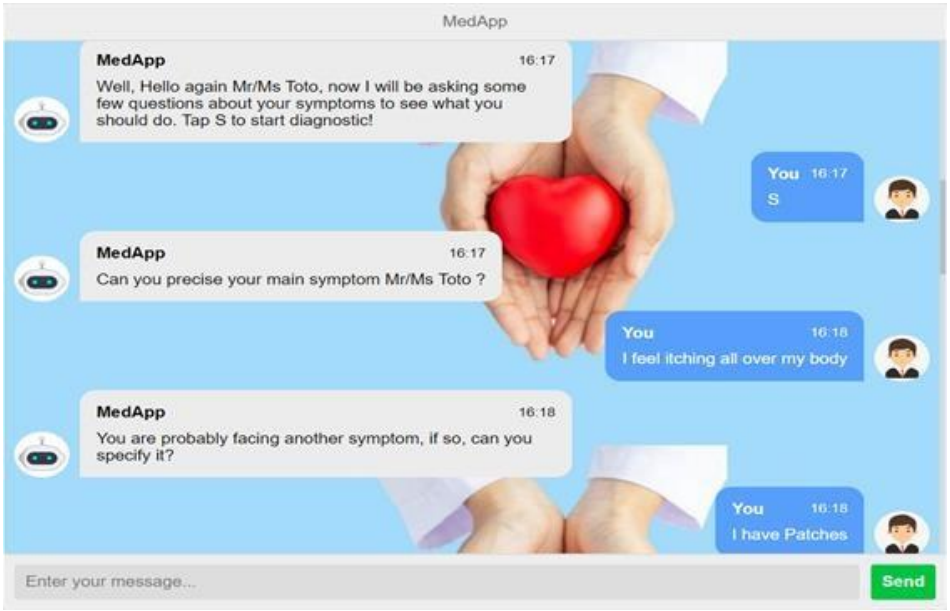


**Fig - 7.2** Chatbot asking for symptoms from patient

This snapshot shows that the chatbot is asking for symptoms faced by the patients and the patients can provide to input the symptoms to the chatbot.


**Fig - 7.3** Chatbot asking specific details of the symptoms

This snapshot shows that the chatbot is asking specific details of the symptoms from the patients to provide better diagnosis.


**Fig - 7.4** Chatbot predicting disease for the patient

The above snapshot is interaction of the chatbot predicting the disease based on the symptoms provided by the patients.


**Fig - 7.5** Chatbot giving description disease to the patient

The above snapshot is based on the diagnosis provided by the patients gives the detailed description of the disease and checks with the patient that how many days are they facing the symptoms and suggest the precautions to be taken.

### 7.2.2 Snapshot of Mobile app for IoT:

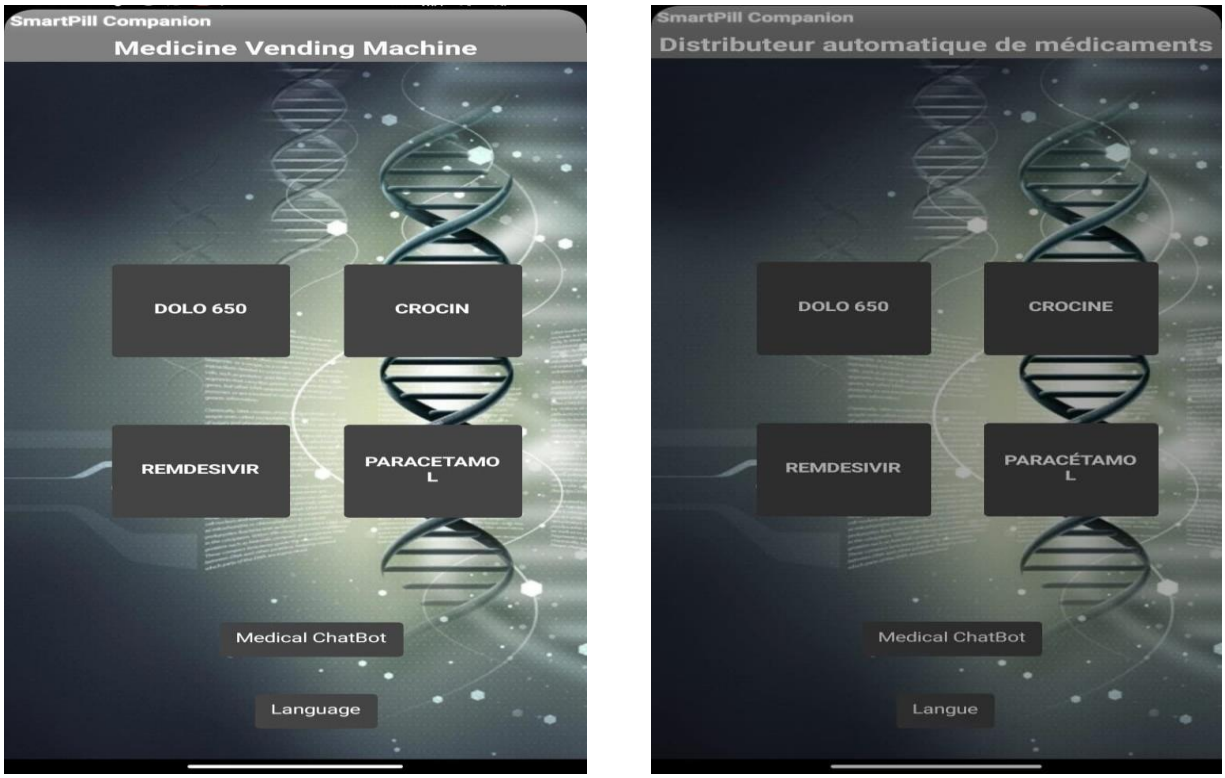


**Fig - 7.6** Mobile app for controlling the vending machine

The above snapshot describes the UI Interface of the mobile application that connects with the IOT model. This application is provided with an option to select any of the two languages that is either English or French. The user can choose the medicine of this choice. A button is provided to connect with the chatbot in this mobile application.

### 7.2.3 Snapshot of IoT Model for medicine vending machine:


**Fig - 7.7** The prototype model built for Medicine vending machine

The above snapshots depict the IOT model. This model consists of an Arduino board, LCD Display, Wi-Fi module, and Servo motors. This servo motors are controlled by the Arduino code written into the board. These servo motors are connected to each compartment to control the doors of each compartment.

# CONCLUSION AND FUTURE ENHANCEMENT

## Conclusions

Automatic Vending Machines are very useful devices for dispensing items such as food and snacks and can be easily available for the use of the public. These machines are more reliable, easily accessible and much more practical than the convention method of purchase and consumes less power so that the system is used everywhere. The Medical Vending machine can be used in healthcare facilities for dispensing medicines which will be easily accessible to the public 24/7. It gives availability of medicines all the time especially in rural areas where the healthcare sector isnot that developed. The model presented is an automatic medicine vending machine that has the capability to dynamically receive input via the application from the user and then dispense the required type of medicine. This can help most of the non-english speaking users to get the proper results in their desired languages. The proposed chatbot provides instant response to the queries of the client in the field of medical field. These Chatbots uses different methodologies to improve the quality of the conversation between the clients and the bots. The application is also be integrated with a medically trained chatbot which can predict a diagnosis based on the symptoms provided by the user and can suggest to refer to the concerned doctor.

## Limitations

Some Limitations of our study should be acknowledged. Needs Human Surveillance although ML has come a long way in the medical world, human surveillance is still essential. Secondly the payment sectorwhere patients need to be protected both medically and personally, confidentiality is a top priority.

Therefore, significant thought and consideration need to be put into the healthcare payment processing system your practice chooses. By doing so, you can ensure the protection of payments as well as the personal data shared through these transactions.

# Future enhancements

- The Chatbot can be enhanced to be Multilingual or Cross lingual.
- Enhancement of Payment Security
- Addition of more meds
- Search options in the app can be added
- Accuracy of the symptom prediction can be improved.

# REFERENCES

[1] Aye Aye Nyein, Ohnmar Win, "RFID Based Vending Machine", IJTSRD Volume 3 Issue 5, 2019.

[2] Munira Ansari, Mohammed Saad Parbulkar, Saalim Shaikh, Talha Khan, Anupam Singh, "Intelligent Chatbot", IJERT, 2021.

[3] Mohammed Mustafa, Izzedin Osman, Hussein Suleman, "Indexing and Weighting of Multilingual and Mixed Documents", Research Gate, 2014.

[4] Leyla Zhuhadar, Olfa Nasraoui, Robert Wyatt, Elizabeth Romero, "Multi-Language Ontology-based Search Engine", Research Gate, 2016.

[5] Ebtsam Mohamed, Samir Elmougy, Mostafa Aref, "Toward multilingual information retrieval system based on internet linguistic diversity measurement", Ain Shams Engineering Journal, 2019.

[6] Alexey Drutsa, Andrey Shutovich, Philipp Pushnyakov, Evgeniy Krokhalyov, Gleb Gusev, Pavel Serdyukov, "User Model-Based Intent-Aware Metrics for Multilingual Search Evaluation", 2016.

[7] Hanqing Lu, Youna Hu, Tong Zhao, Tony Wu, Yiwei Song, Bing Yin, "Graph-based Multilingual Product Retrieval in E-Commerce Search, 2021.

[8] Kanagasabapathi V, Naveenraj K., Neelavarnan V., Naveen raj S, "Automatic chocolate vending machine", 5th International Conference on Advanced Computing & Communication Systems (ICACCS), 2019.

[9] Ehtisham Lodhi, Fenghua Zhu, Zeeshan Lodhi, Qasim Saleem, Gang Xiong, FeiYue Wang,"Design and Implementation of RFID based Smart Shopping Booth", 6th International Conference on Information Science and Control Engineering (ICISCE), 2019.

[10] Hay Man Oo, Khin Thandar Tun, Su Mon Aung, “Coin Acceptor Based Vending Machine using Microcontroller”, IJTSRD, Volume 3 Issue 5, August 2019.

[11] Aditya Patil , Darshan B.G., Dr. D.V. Ashoka, Nethravathi B, “IoT Based Medicine Dispenser”, IJERT , 2020.