A PROJECT SYNOPSIS

ON

“PERSONAL HEALTH ASSISTANT”

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF
BACHELOR OF COMPUTER ENGINEERING

BY

GULSHANKUMAR BAKLE	**A-08**
GAYATRI ASODEKAR	**A-05**
VIJAY CHAUDHARI	**A-15**

UNDER GUIDANCE OF

PROF. CHANDRASHEKHAR RAUT

UNIVERSITY OF MUMBAI

DEPARTMENT OF COMPUTER ENGINEERING

DATTA MEGHE COLLEGE OF ENGINEERING

PLOT NO.98 SECTOR-3, AIROLI, NAVI MUMBAI

ACADAMIC YEAR 2018-19

DATTA MEGHE COLLEGE OF ENGINEERING

AIROLI, NAVI MUMBAI

CERTIFICATE

This is to certify that the project entitled “**Personal Health Assistant**” is bona fide work of “**Gulshankumar Bakle(A08), Gayatri Asodekar(A05), Vijay Chaudhari(A15)**“ submitted to the University of Mumbai in partial fulfilment of the requirement for the award of the degree of **“Undergraduate”** in **“Computer Engineering”**.

Prof. Chandrashekar Raut	Prof. A.P.Pande	Dr.S.D.Sawarkar
Project Guide	Head of the Department	Principal

DATTA MEGHE COLLEGE OF ENGINEERING

AIROLI, NAVI MUMBAI

PROJECT APPROVAL

This project report entitled “**Personal Health Assistant**” of the students “**Gulshankumar Bakle, Gayatri Asodekar, Vijay Chaudhari”** approved for the degree of **Computer Engineering.**

Internal Examiner External Examiner

Date: Date:

Place: Place:

DECLARATION

We declare that, this written submission represents our ideas in our own words and where others' ideas or words have been included; we have adequately cited and referenced the original sources. We also declare that, we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Name of the Students		Signature
Student -1	Gulshankumar Bakle	
Student -2	Gayatri Asodekar	
Student -3	Vijay Chaudhari	

ACKNOWLEDGEMENT

Motivation and guidance are the keys towards success. I would like to extend my thanks to all the sources of motivation.

We would like to grab this opportunity to thank **Dr.S.D.Sawarkar, Principal** for encouragement and support he has given for our project.

We express our deep gratitude to **Prof. A.P.Pande, Head of the Department** who has been the constant driving force behind the completion of this project.

We wish to express our heartfelt appreciation and deep sense of gratitude to my project guide **Prof. Chandrashekhar Raut** for his encouragement, invaluable support, timely help, lucid suggestions and excellent guidance which helped us to understand and achieve the project goal. His concrete directions and critical views have greatly helped us in successful completion of this work.

We extend our sincere appreciation to all professors for their valuable inside and tip during the designing of the project. Their contributions have been valuable in so many ways that we find it difficult to acknowledge of them individually.

We are also thankful to all those who helped us directly or indirectly in completion of this work.

Place: Name of the students

Date: Gulshankumar Bakle

Gayatri Asodekar

Vijay Chaudhari

INDEX

1.	**INTRODUCTION**		**1**
	1.1	Motivation	1
	1.2	Previous Work	2
	1.3	Application	3
	1.4	Problem Definition	3
	1.5	Organization of Report	4
2.	**LITERATURE REVIEW**		**6**
3.	**REQUIREMENT ANALYSIS**		**8**
	3.1	Functional Requirements	8
	3.2	Non-Functional Requirements	9
4.	**PROJECT DESIGN**		**12**
	4.1	Methodology	12
	4.2	Design Consideration	13
5.	**DESIGN IMPLEMENTATION**		**19**
	5.1	Programming environment	19
	5.2	Flow of Implementation	20
			24
6.	**TESTING**		**24**
	6.1	Software Testing	24
	6.2	Test Cases	25
7.	**RESULTS AND DISCUSSION**		**26**
	7.1	Results	26
	7.2	Discussion	28
8.	**CONCLUSIONS AND FUTURE SCOPE**		**30**
	7.1	Conclusion	30
	7.2	Future Work	30
10	**REFERENCES**		**31**

LIST OF FIGURES

SR. NO.	NAME OF THE FIGURE	PAGE NO.
Fig 4.1	DFD Level 0	13
Fig 4.2	DFD Level 1	14
Fig 4.3	Use case	15
Fig 4.4	Sequence Diagram	16
Fig 4.5	Activity Diagram	17
Fig 5.1	Proposed system of personal health assistant	20
Fig 7.1	Screenshot of home page of personal health assistant	26
Fig 7.2	Screenshot of web page, user selects body parts of ailment	26
Fig 7.3	Screenshot of web page, user selects symptoms as per body part	27
Fig 7.4	Screenshot of web page, table of probable diseases is displayed	27
Fig 7.5	Screenshot of web page, of location of doctors	28

LIST OF TABLE

SR. NO.	NAME OF THE TABLE	PAGE NO.
6.1	Test Case	25

ABSTRACT

Over the years the healthcare sector has seen significant advances in terms of diagnosis and prediction of various chronic diseases with machine learning. Since healthcare industry produces huge amount of varied data, which proves favorable for a machine learning model for predicting results with great accuracy and efficiency. With this knowledge, we propose a “Personal Health Assistant”.

When a person is infected with a disease, he is bound to suffer from various symptoms. Some of the symptoms are unique to a particular disease while many show mixed symptoms of two or more disease. During the early phases of disease, it becomes ambiguous to predict which disease a person is afflicted with. So, our web application will use various machine learning algorithms that would take symptoms of the person as input and predict the probabilities of various diseases he or she is suffering with. The output of model would give the chances of all the diseases that he or she may suffer with. Thus, enabling a person to have a better insight of his condition.

Depending on the severity of his/her disease, the model would suggest the necessity for a doctor checkup. This would be made easier by showing the nearby physicians and hospitals around him using Geo Location Google map. Once the person has visited the doctor, with our web app, he can know the comparative prices of various pharmacies providing the same prescription. Thus, allowing people to make wise and feasible choice.

Thus, this system would make a person more aware about his/her health condition and choose feasible drugs over costlier drugs.

Chapter 1

Introduction

The past decades have brought remarkable advances in our understanding of human disease. Artificial intelligence (AI) aims to mimic human cognitive functions. It is bringing a paradigm shift to healthcare, powered by increasing availability of healthcare data and rapid progress of analytics techniques. We survey the current status of AI applications in healthcare and discuss its future. AI can be applied to various types of healthcare data (structured and unstructured). Popular AI techniques include machine learning methods for structured data, such as the classical support vector machine and neural network, and the modern deep learning, as well as natural language processing for unstructured data. Major disease areas that use AI tools include cancer, neurology and cardiology. When it comes to effectiveness of machine learning, more data almost always yields better results—and the healthcare sector is sitting on a data goldmine. By using this knowledge of machine learning we propose a "Personal Health Assistant". In our system we will provide the user the probabilities of various disease on basis of some questions asked before. The output gives chances of the various disease he or she is suffering. And on the severity of his or her disease our system will provide the information of nearby physicians and hospital using the Geo location Google map. As we know that there is different price for same medicine according to their brand. The generic name medicine has low price than brand name medicine even though they have same contained in it. Thus, our system will also provide their generic name and their price by which the user can compare their price of the medicine which are been prescribe by the doctor.

1.1 Motivation

Many times, while we know the symptoms but don't know what exactly we are suffering from. So just by knowing the diseases we get an idea what could all we be suffering from and take necessary action as soon as possible. Sometimes many people find it difficult to locate a physician or specialist nearby after discovering his or her disease by the help of this type of system. Thus, our system not only would take symptoms of the person as input and predict the probabilities of various diseases he or she is suffering with but also help the person to locate the physician or hospital or specialist nearby. As we will be using GPS system with help of that the person can get information about the doctors nearby. Once the person has visited the doctor, with our web app, he can know the comparative prices of various pharmacies providing the same prescription. Thus, allowing people to make wise and feasible choice.

1.2 Previous Work

1.2.1 ADA: Your personal health assistance

ADA Health has an iOS and Android app called Ada that combines artificial intelligence (AI) with expertise from actual doctors to help people understand and manage their health. Ada Health, a fast-growing healthcare start up with 100 staff across Berlin, Munich, and London, has raised $47 million (€40 million; £36 million) from investors. Ada Health has an iOS and Android app called Ada that combines artificial intelligence (AI) with expertise from actual doctors to help people understand and manage their health. Daniel Nathrath, cofounder and CEO of Ada Health, told Business Insider that the money will be used to help Ada Health open a new US office and expand into new markets.

1.2.2 Medline Plus

The A.D.A.M. Medical Encyclopedia includes over 4,000 articles about diseases, tests, symptoms, injuries, and surgeries. It also contains an extensive library of medical photographs and illustrations. MedlinePlus [4] is the National Institutes of Health's Web site for patients and their families and friends. Produced by the National Library of Medicine, the world’s largest medical library, it brings you information about diseases, conditions, and wellness issues in language you can understand. MedlinePlus offers reliable, up-to-date health information, anytime, anywhere, for free.

1.3 Application

Our website is quite useful when any person wants to search for any diseases based on symptoms. The person needs to do research on the particular disease as one disease has many symptoms, so for searching for any disease it will be hectic task for a person. Thus, using this website a user can put its symptoms and it will predict the diseases based on those symptoms.

Not only that using this website the person can also get the location of doctors or hospitals nearby him/her. This reduces the stress of the person if he/she is new to that place or person is unaware of the specialists nearby him/her. Thus, search becomes easier and adds up saving time.

The next step is to buy medicines but we know that Generic name medicines have low price compared to brand name medicines. But most of the people do not know the generic name of the medicines so they tend to buy the medicines with high prices which can be available at low prices. So, our website provides the prices and generic name for the respective brand name, the user just has to give the brand name as input. Thus, this helps people to save their money.

1.4 Problem Definition

The basic idea is to predict the diseases by the symptoms as described by the person so he or she can get the knowledge of the severity of the disease. Thus, by depending on the severity the system will suggest the necessity for a doctor check-up and also show the nearby physicians and hospitals around him or her. He or she can also know the comparative prices of various pharmacies provided in the prescription by the doctors.

1.5 Organization of Report

The report is divided into 7 chapters. Each chapter has different sections along with its explanations.

Chapter 1: **Introduction**

This part introduces to the concept of Project PERSONAL HEALTH ASSITANT. The motivation to develop this project and previous work done in this domain are listed. Different applications of our system are also shown.

Chapter 2**: Literature Survey**

A literature survey is a text of a scholarly survey or research, which includes the current knowledge including substantive findings, as well as theoretical and methodological contributions to a particular topic. The findings of different research surveys based on/related to online learning are summarized in this chapter.

Chapter 3**: Requirement Analysis**

Requirements analysis, also called requirements engineering, is the process of determining user expectations for a new or modified product. The functional and nonfunctional requirements for the development of the project are listed.

Chapter 4: **Project Design**

This part includes overview of different elements which will be used in the design and Different type diagram which shows the flow of project.

Chapter 5: **Design Implementation**

The tools and programming language useful for implementation of the project are specified in this chapter.

Chapter 6: **Testing**

In this chapter the system is tested number of times for software errors and hardware faults. Considering various test cases, the errors and faults are eliminated.

Chapter 7: **Conclusion and Future Scope**

The summary, concluding remarks and how the project can be continued in the future is described here.

References

Different sources and papers used for the development of the project are cited here. Each element in the project synopsis is in an organized manner and is in order as mentioned above.

Chapter 2

Literature Survey

Literature survey aims at gaining an understanding of fundamentals and state of the art of the area of the research by learning the definitions of the concepts and theories. Here it includes various analyses and research made in the field of Medical field and the results already published, considering the various parameters of the project and the extent of the project. It gives the clarity &amp; better understanding of the project.

2.1 Prediction of Heart Disease Using Machine Learning

The above IEEE paper was given by Aditi Gavhane and was conference on 29-31 March 2018[6]. We have summarized that in this paper they propose to develop an application which can predict the vulnerability of a heart disease given basic symptoms like age, sex, pulse rate etc. In their proposed system they used the neural network algorithm multi-layer perceptron (MLP), the system is developed using python code using PyCharm IDE.

2.2 Prediction of Urinary System Disease Diagnosis

According to the IEEE paper IEEE to the Prediction of Urinary System Disease Diagnosis dated on 2014. We have summarized the in this paper, the main objective is to demonstrates the ability of DM to develop a prediction model for a presumptive diagnosis of two familiar urinary diseases: the acute inflammation of the urinary bladderand nephritis of renal pelvis. This research evaluates the supervised machine learning algorithms Ridor, OneR, and J48.

2.3 Prediction of Probability of Disease Based On Symptoms

According to the IRJET paper IRJET to the Prediction of Probability of Disease based on Symptoms dated on May-2018[5]. We have summarized paper using machine learning disease is predicted on the base of general symptoms. In this paper, Latent factor model is used. A new convolutional neural network based multimodal disease risk prediction (CNN-MDRP) algorithm is proposed. Use three algorithms for predicting the diseases. One is KNN, second is Naïve Bayesian and third is Decision tree.

2.4 Diagnosis of Liver Diseases Using Machine Learning

According to the IEEE paper IEEE to the Diagnosis of Liver Diseases using Machine Learning dated on 3-5 Feb. 2017. In this paper with the help of machine learning they are diagnosing liver disease. This paper aims to compare 2 methods of computer aided medical diagnoses. This method involves the training of an Artificial Neural Network to respond to several patient parameters such as age, Bilirubin, Alkaline Phosphotase, Alamine Aminotransferase, and Aspartate Amino transferase among others. The second method studied in this paper involves a genetic approach to the diagnosis. The proposed approach is the application of Artificial Neural Networks and Multi-Layer Perceptron’s to Micro-Array Analysis.

2.5 Disease Prediction Using Machine Learning Over Big Data

The above CSEIJ paper was given by Vinitha S, Sweetlin S, Vinusha H and Sajini Sand was date on February 2018. In this paper they classifying the various sub-disease using big data. In this proposed system, it provides machine learning algorithms for effective prediction of various disease occurrences in disease-frequent societies. Using structured and unstructured data from hospital it uses Machine Learning Decision Tree algorithm and Map Reduce algorithm.

Chapter 3

Requirements Analysis

Requirement analysis encompasses those tasks that go into determining the needs or conditions to meet for a new or altered product or project, taking account of the possibly conflicting requirements of the various stakeholders, analyzing, documenting, validating and managing software or system requirements.

3.1 Functional Requirements

3.1.1 Availability

The system should be available to the end user for use all the time. Specific technology/domain requirements listed in the system should be available and accessible to the end user. Availability of a system is typically measured as a factor of its reliability. As reliability increases, so does availability. Availability of a system may also be increased by the strategy of focusing on increasing testability, diagnostics and maintainability. Improving Testability & diagnostics during the early design phase is generally easier than reliability.

3.1.2 Interoperability

All the resources provided must be rendered and plugged in seamlessly. Interoperability is a characteristic of a system, whose interfaces are completely understood, to work with other products or systems, at present or future, in either implementation or access, without any restrictions.

3.1.3 Symptom Classifier

The symptom checker uses symptoms diseases database, which is pre-processed and trained using machine learning algorithm- random forest. The information within this database is transformed into dataset. A dataset is the collection of related sets of information that is composed of separate elements but can be manipulated as single unit.

3.1.4 Find Doctors

Under this functionality, the person based on the severity of his diseases would be able to track the doctors or physicians around him. This would be done with the help of Google map API and Geo locationof that person.

3.1.5 Price comparator

Once prescription is given, the person can use the system's price comparison module. Here, he or she will be able to enter the prescribed drugs name, and system would be giving out the generic medicines under the same chemical formula.

3.2 Non-Functional Requirements

3.2.1 Portability

Portability is a characteristic attributed to a computer program if it can be used in an operating system other than the one in which it was created without requiring major rework. Porting is the task of doing any work necessary to make the computer program run in the new environment. As the system requires only internet connection it would be able to work on all browsers and systems.

3.2.2 Usability

Usability is the degree to which software can be used by specified consumers to achieve quantified objectives with effectiveness, efficiency, and satisfaction in a quantified context of use. The amount of human intervention that the system would be requiring should be less or minimal. The user should not feel tired out of choosing many questions in order to classify the disease.

3.2.3 Performance

System should be well optimized to exhibit high performance. Performance is a degree which explains the responsiveness and stability of a system under a particular workload. It can also serve to investigate measure, validate or verify other quality attributes of the system, such as scalability, reliability and resource usage. It strives to build performance standards into the implementation, design and architecture of a system.

3.2.4 Dependency

The entire system should be put together as a whole package and should work without requiring any additional dependencies. It is a state in which one object uses a function of another object. It describes a dependence relation between statements in a program. It is a relationship between more than one element in the Unified Modeling Language.

3.2.5 Functionality

System should be capable of running on machines with both low and high configuration. Functionality is the sum or any aspect of what a product, such as a software application or computing device, can do for a user. A product's functionality is used by marketers to identify product features and enables a user to have a set of capabilities. Functionality may or may not be easy to use.

3.2.6 Summary Efficiency

System should be capable of summarizing an efficient summary of the diseases that the person may be facing with. He should be able render proper inference from the summary provided by system.

System Requirements:

Hardware Requirements:	1. Power supply 2. Keyboard 3. Mouse tracker
Software requirements:	1. Anaconda IDE 2. Python-Django 3. HTML, CSS, JavaScript 4. Windows XP or higher 5. 512 MB RAM 6. 500 MB free space

Thus, we have studied and analyzed all the requirements of our system. All the stated requirements are clear, complete, consistent and unambiguous. This will help us in the successful development of our project.

Chapter 4

Project Design

4.1Methodology

In our project we have used Random Forest Algorithm which is one of the supervised algorithms in Machine learning. It is used for classification as well as for regression. It operates by preparing multiple numbers of decision trees on the same dataset, once multiple decision trees are constructed, all the decisions of individual trees are merged together to obtain more accurate and stable result. Random forest solves the issues of over-fitting to training dataset, as it operates by calculating the mean of obtained classes of decision tree.

Random forests can be termed as upgraded variant of decision trees. The major difference between them being, decision trees are built over entire dataset, while random forests are used to create multi-decision trees on subset of dataset.Every decision tree consists of following components:

1) Root node: This is the top most node present in decision tree. It splits the entire dataset into two halves. Root node is the node which has lowest value of impurity.
2) Internal node: Root node splits into internal nodes. All the results made out on branches are internal nodes. These internal nodes further branch out into leaf nodes.
3) Branch: Every branch indicates an outcome or possible action to be taken.
4) Leaf nodes: These are the end nodes of our decision tree.

Random forest construct trees using certain metrics. These metrics help in measuring best way to make splits in tree. The metrics are Gini impurity, information gain and variance reduction. Random forest algorithm works on the principle of bootstrap aggregation. The bootstrap is a powerful statistical method which is used for estimation of a quantity from a data sample. The bootstrapping algorithm leads to better performance since it reduces variances of model without increasing the bias. Simply training many trees on a single training set would give strongly correlated trees (or even the exactly the same tree many times, if the training algorithm used in problem is deterministic); bootstrap sampling is a way which helps in de-correlating the trees using different training sets for separate trees.

Random forests differ in only one way from bagging algorithm methodology: they use a modified tree learning algorithm that selects, at each candidate split in the learning process, a randomised subset of the features from the training dataset. This process of section randomly is called "feature bagging"

4.2 Design Consideration

4.2.1 Data Flow Diagram

A data flow diagram (DFD) is a graphical representation of the “flow” of data through an information system, modelling its process aspects. A DFD is often used as a preliminary step to create an overview of the system without going into great detail, which can later be elaborated.

4.2.1.1 DFD level 0

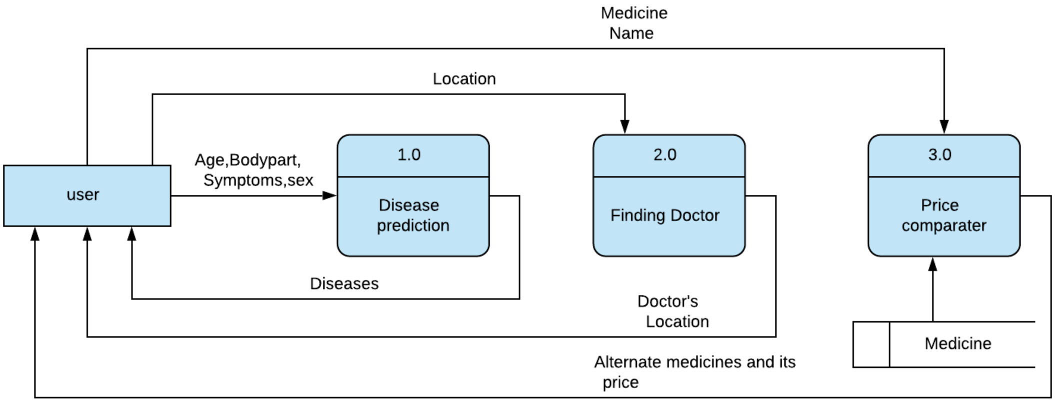


Fig 4.1: DFD Level 0

In above DFD level 0 fig 4.1, User give required details to the Disease prediction, Finding Doctor and Price comparator. On Which User gets the probability of the diseases, Doctors nearby him/her and Generic names with its prices.

4.2.1.2 DFD level 1

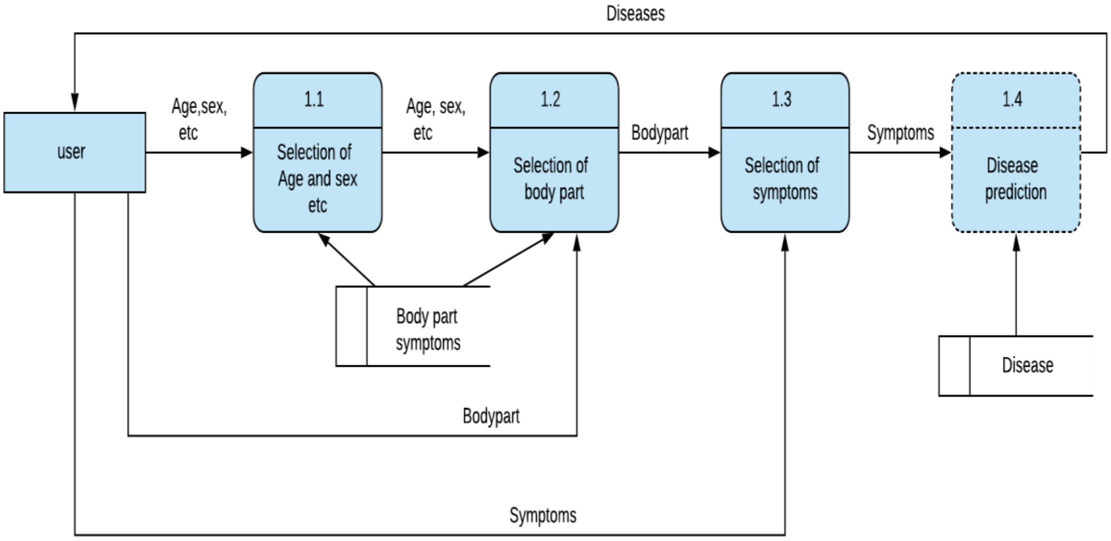


Fig 4.2:DFD Level 1

In DFD level 1, Fig 4.2 The user provides the system with its details like age, gender, etc. then select the body parts after that user select the symptoms h/she is suffering from. The selected system is being processed and calculate the probability of diseases using the Random forest method. The outcome is then displayed to the user.

4.2.2 Use Case Diagram

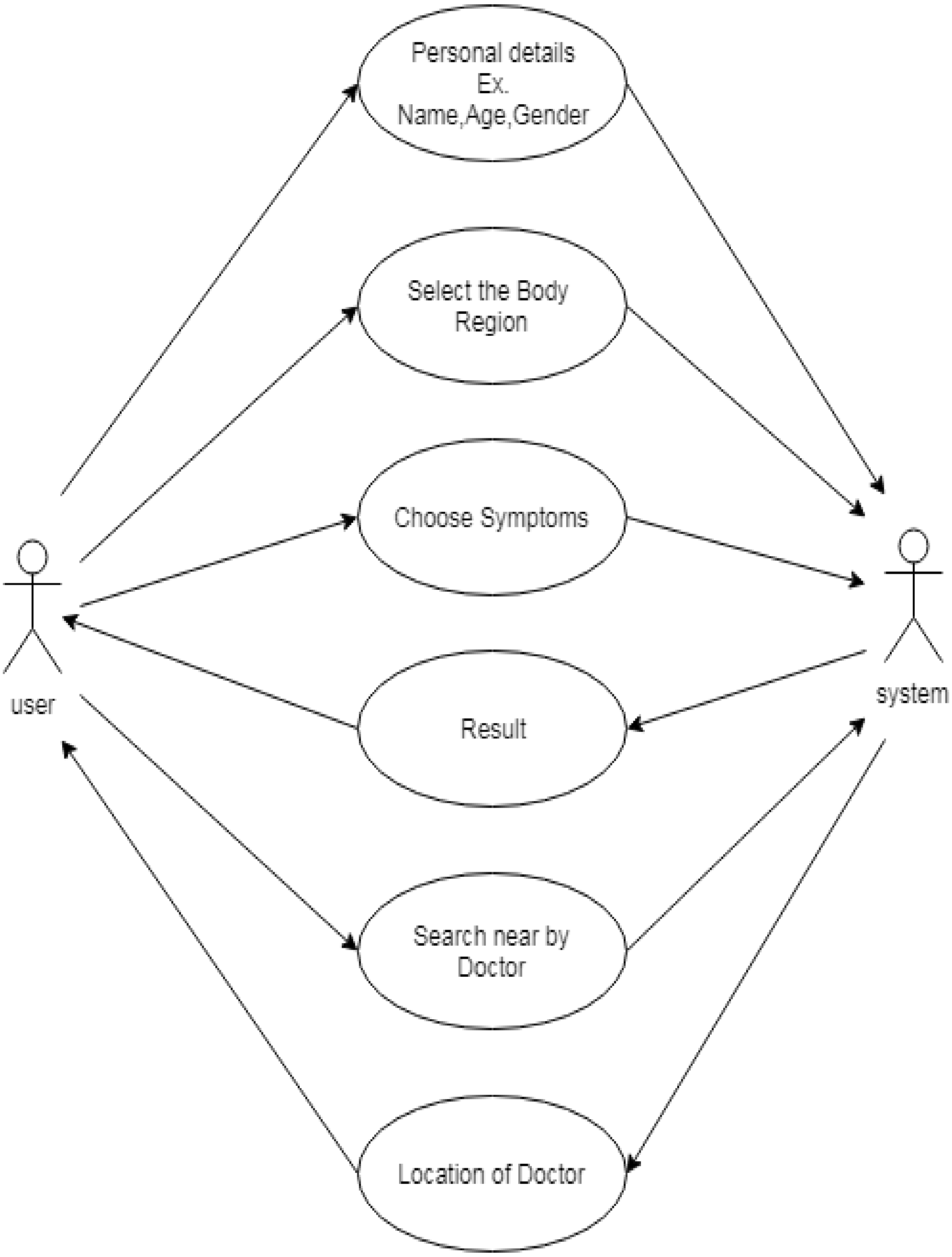


Fig 4.3: Use case

The user will Enter His/ Her Personal Details like Name, Age, Sex. After that he/she will select the body part and also select the symptoms he/she is suffering from. The system will calculate the probabilities of the diseases the person may have based on the selected symptoms. After that person may search for doctors nearby him/her for consultation.

4.2.3 Sequence Diagram

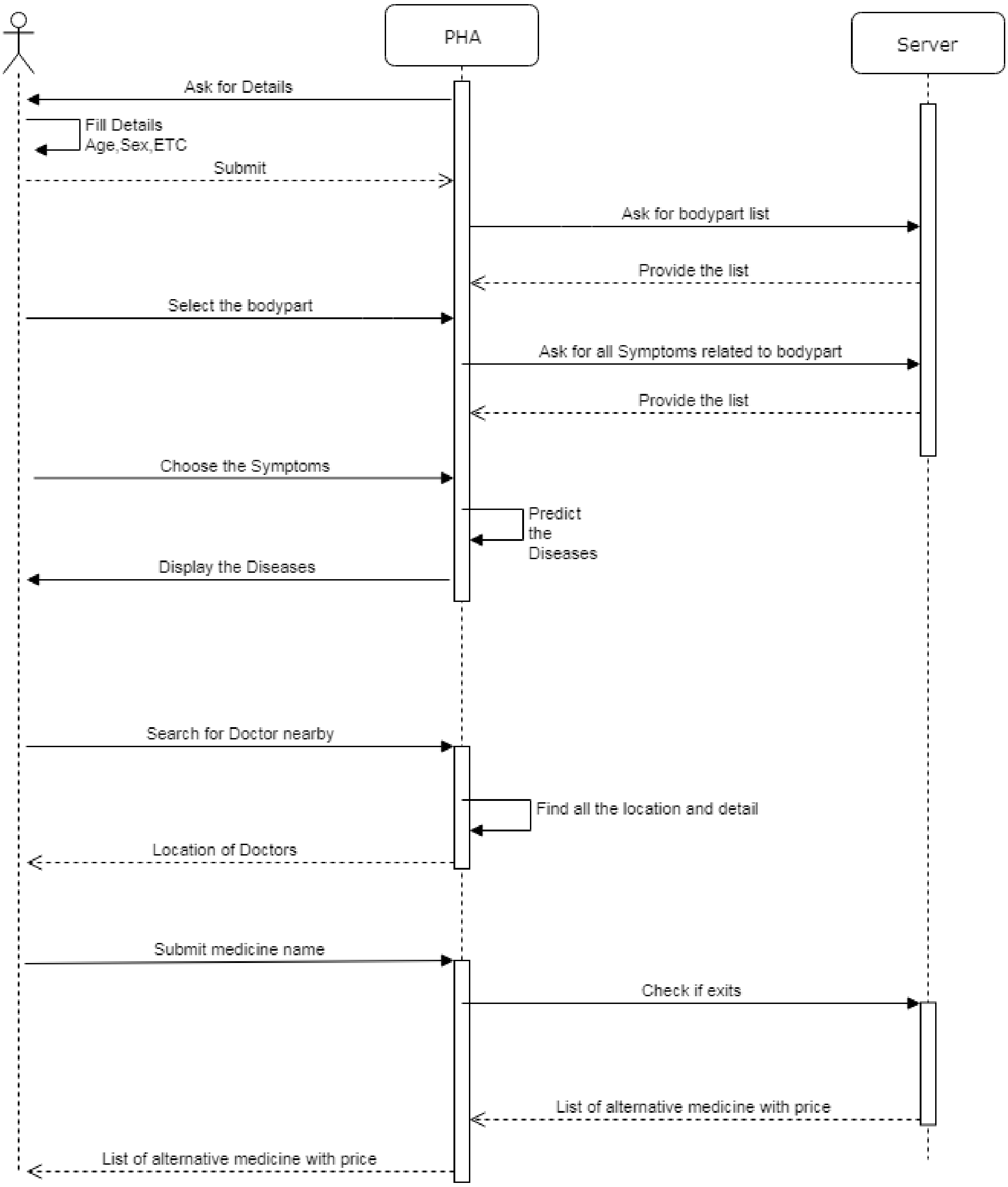


Fig 4.4: Sequence Diagram

The sequence diagram given above shows the interaction of various modules in the system. The 'User' gives reply to 'PHA' object by giving the details like age, gender, etc. The 'Server' object provides the list of body parts and symptoms on which the

‘User’ selects the body part and symptoms. After that ‘PHA’ object predicts the probability of diseases based on the selected symptoms and displays it to ‘User’. Later, if ‘User’ wants to search for doctors nearby him/her then ‘PHA’ object will return the doctors nearby user by taking user location. ‘User’ object can also get Generic Name with Its price ‘User’ just have to give the ‘PHA’ object the name of medicine the ‘PHA’ object will return if exists the Generic name and its price.

4.2.4Activity Diagram

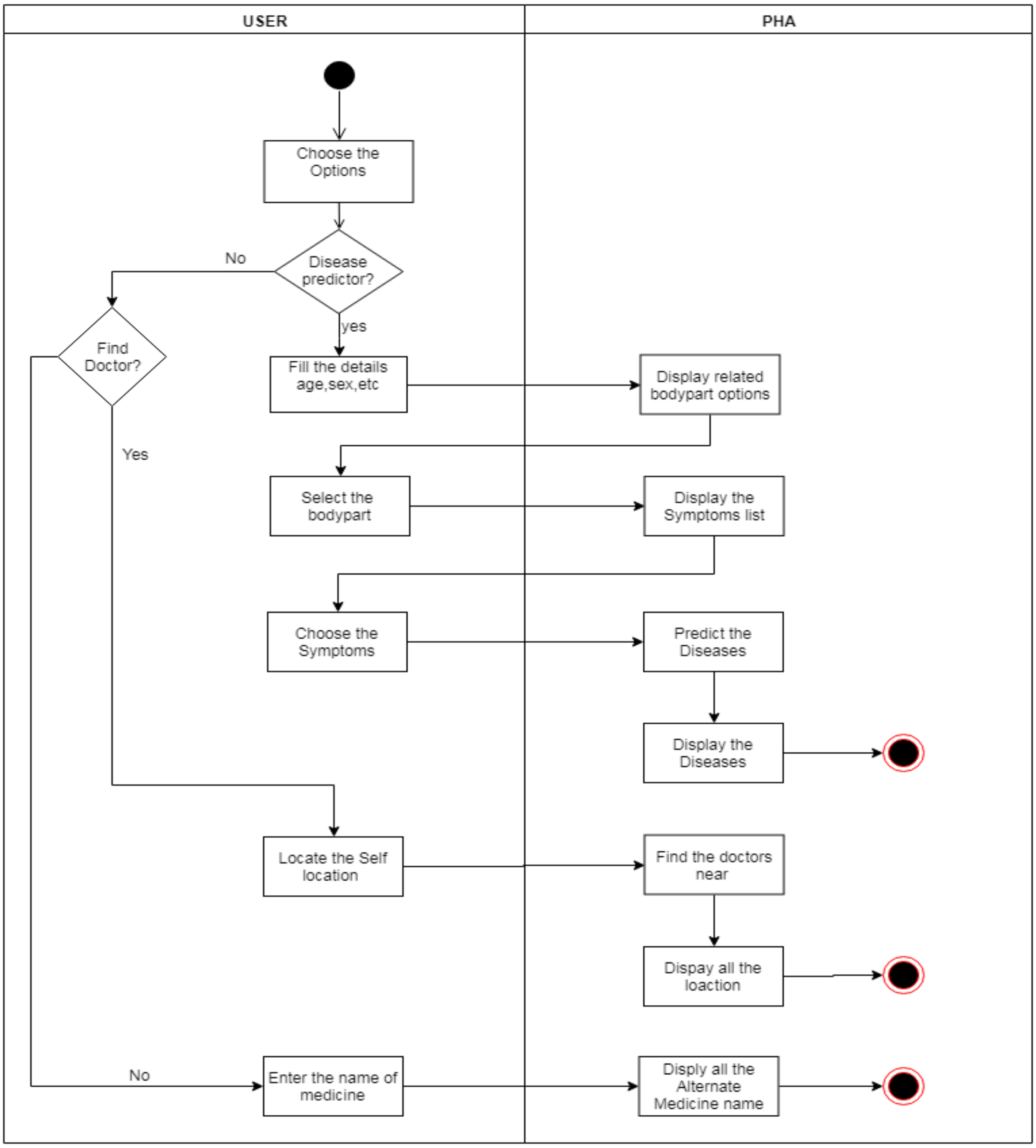


Fig 4.5: Activity Diagram

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.

Chapter 5

DESIGN IMPLEMENTATION

5.1 Programming environment

5.1.1 Software Used

PYCHARM:

PyCharm is an integrated development environment (IDE) used in computer programming, specifically for the Python language. It is developed by the Czech company JetBrains. It provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems (VCSes), and supports web development with Django.

5.1.2 LANGUAGES: -

HTML:

The system will use HTML for rendering the analysed and forecasted data in the web page. HTML is the standard markup language for creating Web pages. HTML describes the structure of Web pages using markup. HTML elements are the building blocks of HTML pages.

CSS:

CSS stands for Cascading Style Sheets. Basically, CSS is a language that manages the design and presentation of web pages -- the way things look. It works together with HTML, or Hyper Text Markup Language, which handles the content of web pages.

JavaScript:

The system will make the use of JavaScript to incorporate different plugins in our web page. JavaScript is the programming language of HTML and the Web. JavaScript resides inside HTML documents, and can provide levels of interactivity to web pages that are not achievable with simple HTML.

PHP:

The system will use PHP to interact with the database since our data are stored in the database. PHP is a server-side scripting language designed primarily for web development but also used as a general-purpose programming language.

PYTHON:

Python is used as our Machine Learning language, Model was made from Libraries like Numpy Pandas and Sklean.

SQL:

Structured Query Language is a special-purpose programming language designed for managing data held in a relational database management system (RDBMS), or for stream processing in a relational data stream management system (RDBMS)

5.2 Flow of Implementation

At beginning, our system will mimic like a real doctor, by asking which body part is being affected. The patient will need to select the body part from the displayed list. The second step as a doctor asks about is symptoms, similarly our personal health assistant will display a set of symptoms pertaining to those selected body parts. The patient would be able to select multiple symptoms that he or she is suffering with. On the basis of symptoms given by patient, the system will make predictions of the probable diseases in the form percentages. The predictions are made by our random forest classifier object which is trained beforehand on dataset of symptoms and disease. The patient would be able to see a list of predicted diseases given by our random forest classifier.

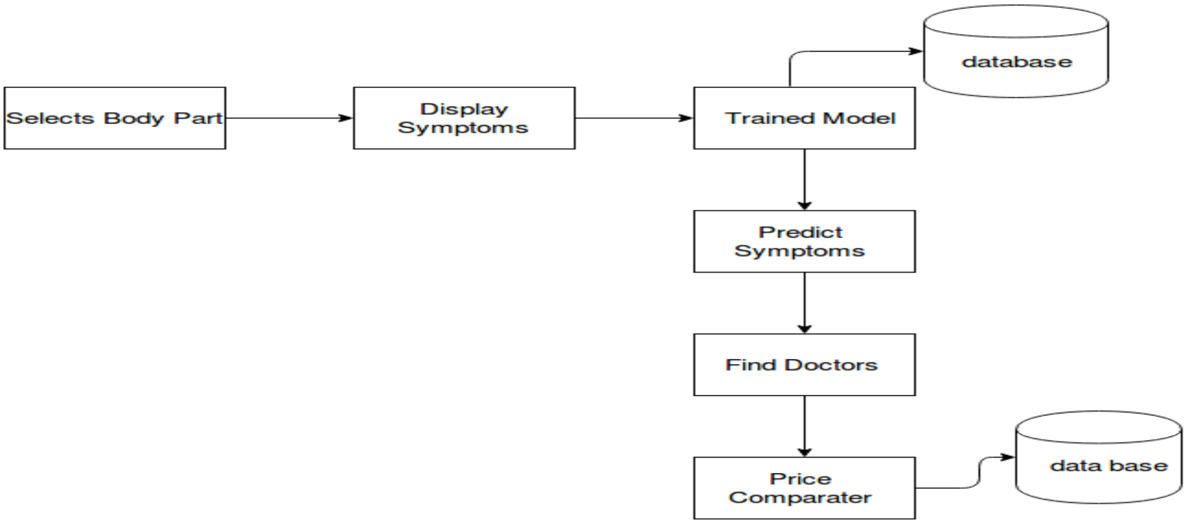


Fig 5.1 Proposed system of personal health assistant

Since the dataset present online is raw data, it must be pre-processed using various data pre-processing technique. Our dataset consisted of three columns- symptoms, diseases and weight. The dataset is processed such that all the symptoms are turned into columns and last column would be of disease column. The symptoms pertaining to particular diseases are marked as 1, while rest 0. This processed data is now trained using random forest classification algorithm.

CODE SNIPPET

```
import pandas as pd
import numpy as np
data = pd.read_csv("Clean_dataset.csv",encoding="ISO-8859-1")
df = pd.DataFrame(data)
Unique_df = pd.get_dummies(df.Target)
df_source = df['Source']
df_final = pd.concat([df_source,Unique_df],axis=1)
df_final.drop_duplicates(keep='first',inplace=True)
df_final = df_final.groupby('Source').sum()
df_final = df_final.reset_index()
df_final[:5]
print(len(df_final))
cols=df_final.columns
cols=cols[1:]
x=df_final[cols]
y=df_final['Source']
df_final.head()
x.to_pickle('symptom_list')
y.to_pickle('disease_list')
diseases=pd.read_pickle('disease_list')
symptomList=pd.read_pickle('symptom_list')
import matplotlib.pyplot as plt
%matplotlib inline
from sklearn.cross_validation import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
x_train, x_test, y_train, y_test =train_test_split(x,y,test_size=0.3)
```

```
clf = RandomForestClassifier()
trained_model = clf.fit(x,y)
print(trained_model.score(x,y))
print(accuracy_score(y, trained_model.predict(x)))
from sklearn.externals import joblib
def check_disease(test_text):
    test_symptoms = test_text.split(';')
    sym_list = list(x)
    test_data = np.zeros(len(sym_list))
    flag = 0
    for i in test_symptoms:
        print(i)
        try:
            ind = sym_list.index(" ".join(i.strip(' ').split()))
            print(ind)
            test_data[ind] = 1
            flag += 1
        except ValueError:
            pass
    tdf = pd.DataFrame([test_data], columns = sym_list).astype(int)
    print(clf.predict(tdf))
    out = clf.predict_proba(tdf)
    out1 = out.tolist()
    out_list = out1[0]
    print("XXX", len(out_list))
    dis_list = list(y)
    #print(dis_list)
    final_dis_list=[]
    final_dis_prob_list=[]
    print("Probable Diseases:\t% Probability\n\n")
    for i in range(len(out_list)):
        if not out_list[i] == 0:
            final_dis_list.append(dis_list[i])
            final_dis_prob_list.append(((out_list[i]) * 100))
```

```
print(final_dis_list)
print(final_dis_prob_list)
test_text = 'shortness of breath;orthopnea;jugular venous
 distention;rale;dyspnea;cough;wheezing'
 check_disease(test_text)
```

The second functionality added in our personal health assistant being, locating the nearby doctors with the help of current Geo Positioning System (GPS) location of the patient using our web application. The patient would be able to see a list of doctors, physicians and hospitals around him or her on map.

The third functionality that our personal health assistant would provide is a price comparator for medicines. As the prices of branded medicines are high, not everyone is able to afford them. A substitute for them is generic medicines, which contains the same chemical component as that of other medicines but at lower prices. The patient would need to enter the prescribed medicine name in the system, which in return will fetch out a list of all generic medicines with same chemical components as that of prescribed medicines. This will allow the patient to make a proper choice for purchasing the medicines. A figurative representation of entire system is shown in figure 5.1.

Chapter 6

TESTING

In hardware and software development, testing is used at key checkpoints in the overall process to determine whether objectives are being met. After the project is developed it is tested for different test cases to make sure that there are no errors.

6.1 Software Testing

Software testing is a process, to evaluate the functionality of a software application, with an intent to find whether the developed software met the specified requirements or not and to identify the defects to ensure that the product is defect free in order to produce the quality product.

6.1.1 Unit testing:

Unit Testing is done to check whether the individual modules of the source code are working properly that is testing each and every unit of the application separately. In this project all the software modules are checked thoroughly which includes the radio buttons command buttons like check Now or Get location, text boxes where user will type the text, and other selection buttons.

6.1.2 Integrated testing:

Integration Testing is the process of testing the connectivity or data transfer between a couple of units tested modules. All the software modules in this project are well tested and inter-connectivity between modules of the software are also working properly. If the Geo location is not correctly taking the location due to internet issues or due other reason then it will show a message “The Geolocation service failed” or “Your browser doesn't support geolocation”.

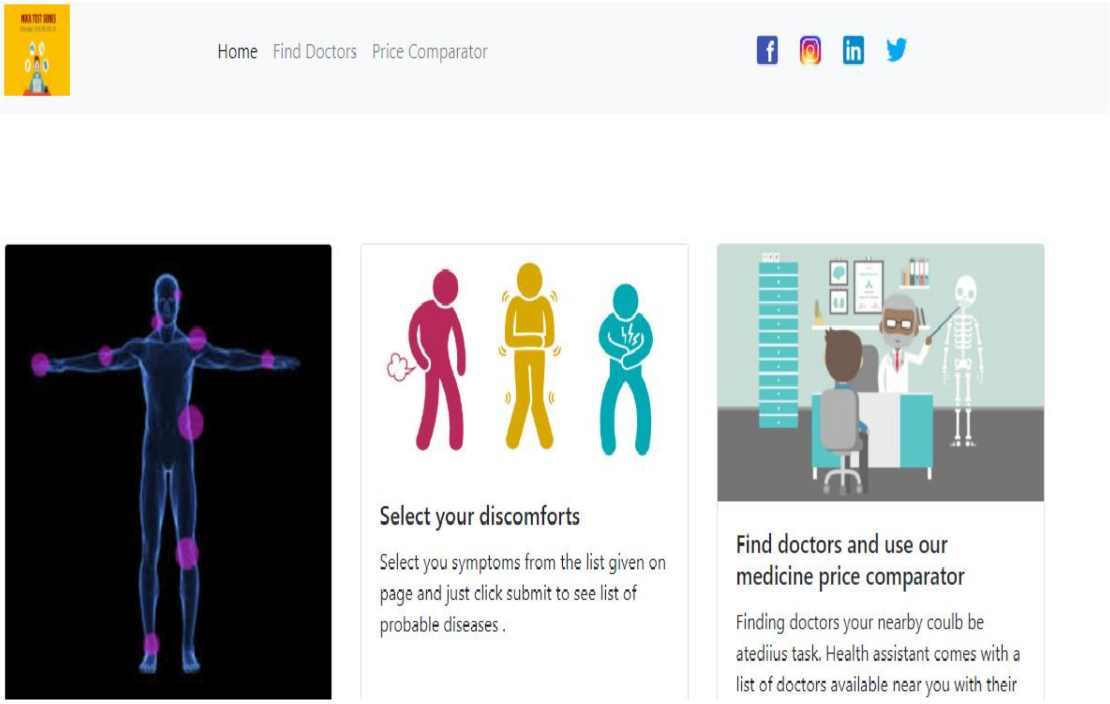
6.2 Test Case

Test cases	Description	Pre-requisite	Expected Result	Actual result	Status
1	Run the web app	Any browser with internet connection	Display the home page	Home page appears	Pass
2	Functionality that is to be used by the user	Any browser with internet connection	Display selected page between home, find doctors and price comparator	Selected page is displayed	Pass
3	Select body part with ailment	Web browser	User should be able to enter name, age, gender and select body part	User selects multiple body parts with ailment	Pass
4	Enter symptoms pertaining to selected body parts	Web browser	User should be able to enter his or her symptoms in the form of text	The page displays all symptoms of body parts, from which user can select symptoms he is suffering and input is text format	Pass
5	Predict diseases	Web browser	A table containing list of diseases and percentage probabilities	The page displays the all diseases the person could be suffering with, with their percentage probabilities	Pass
6	Find doctors	Web browser and Geo Positioning System (GPS)	A list of doctors nearby to location of user	The page asks permission to access location of user and displays a map locating nearby doctors and also a list of those doctors with their address	Pass
7	Price compare	Web browser	A list of generic medicines for branded ones	On entering name of branded medicine, the search query displays a table of generic medicines	Pass

Table 6.1Test Case

Chapter 7
RESULTS AND DISCUSSION

7.1 Results



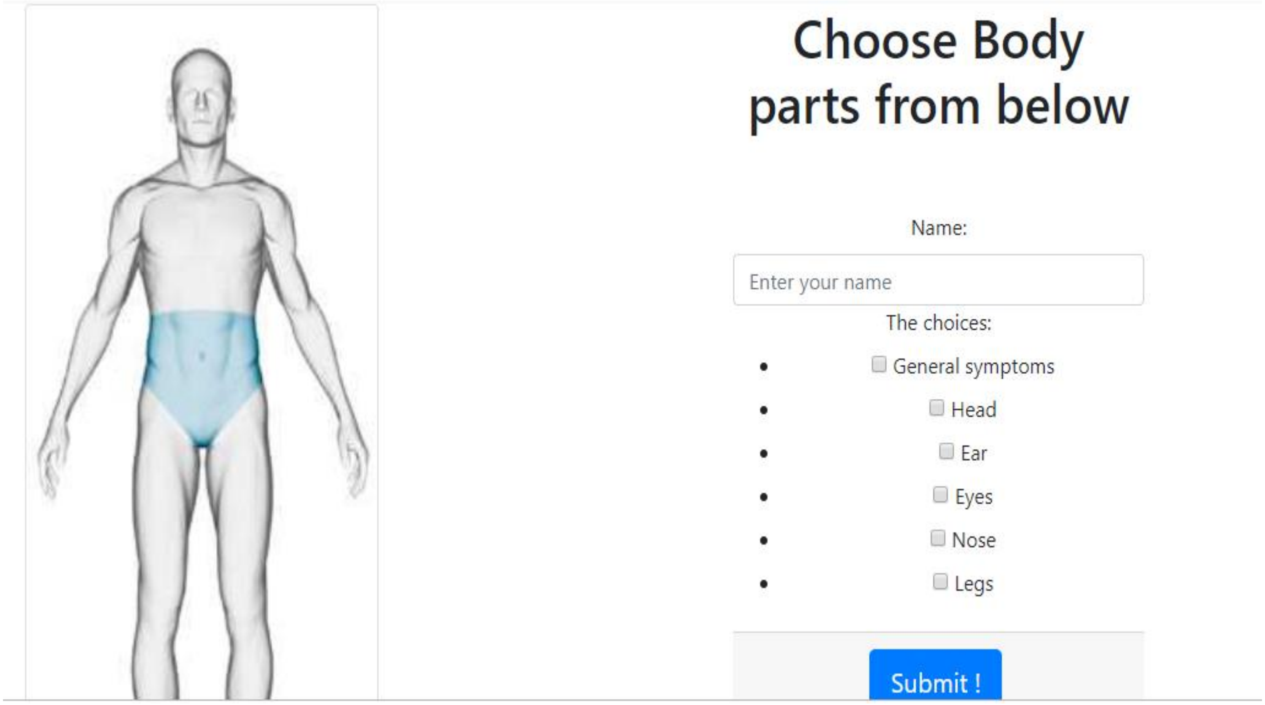
Select your discomforts

Select you symptoms from the list given on page and just click submit to see list of probable diseases .

Find doctors and use our medicine price comparator

Finding doctors your nearby coulb be atediius task. Health assistant comes with a list of doctors available near you with their

Fig 7.1 Screenshot of home page of personal health assistant



Choose Body parts from below

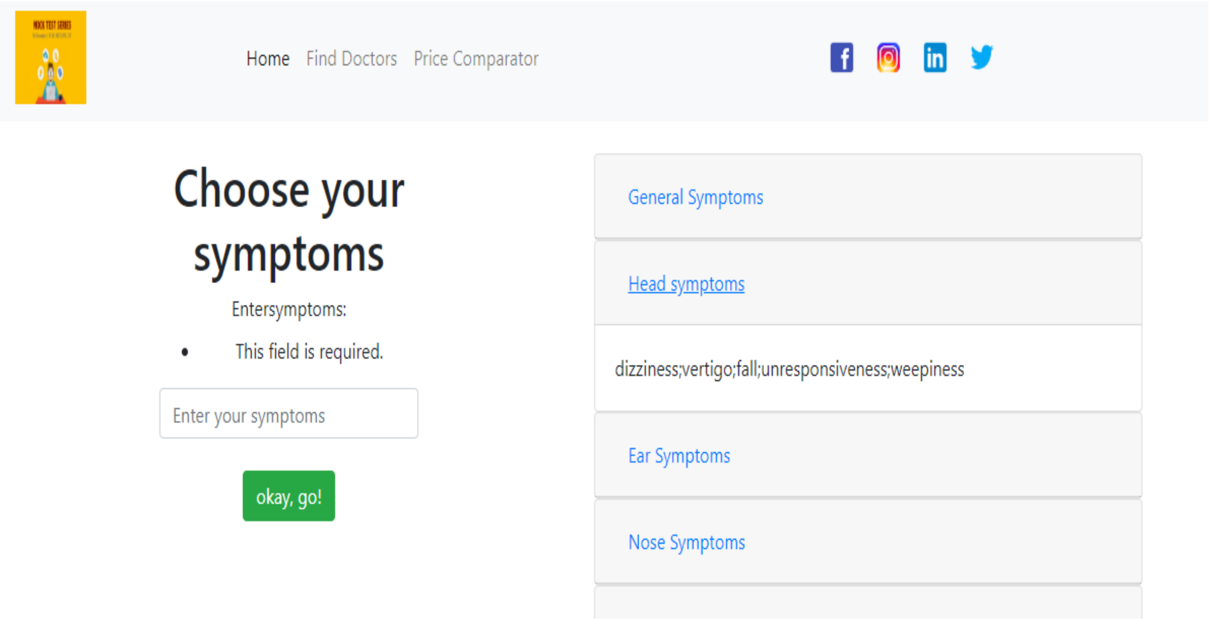
Name:

Enter your name

The choices:

- □ General symptoms
- □ Head
- □ Ear
- □ Eyes
- □ Nose
- □ Legs

Fig 7.2 Screenshot of web page, user selects body parts of ailment



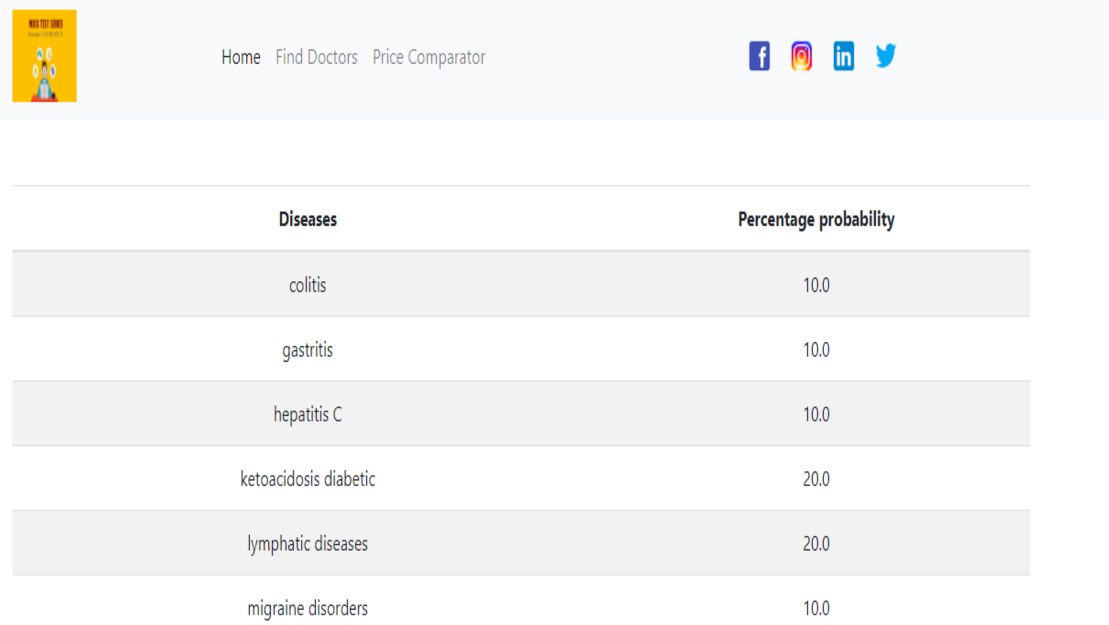
Choose your symptoms

Entersymptoms:

- This field is required.

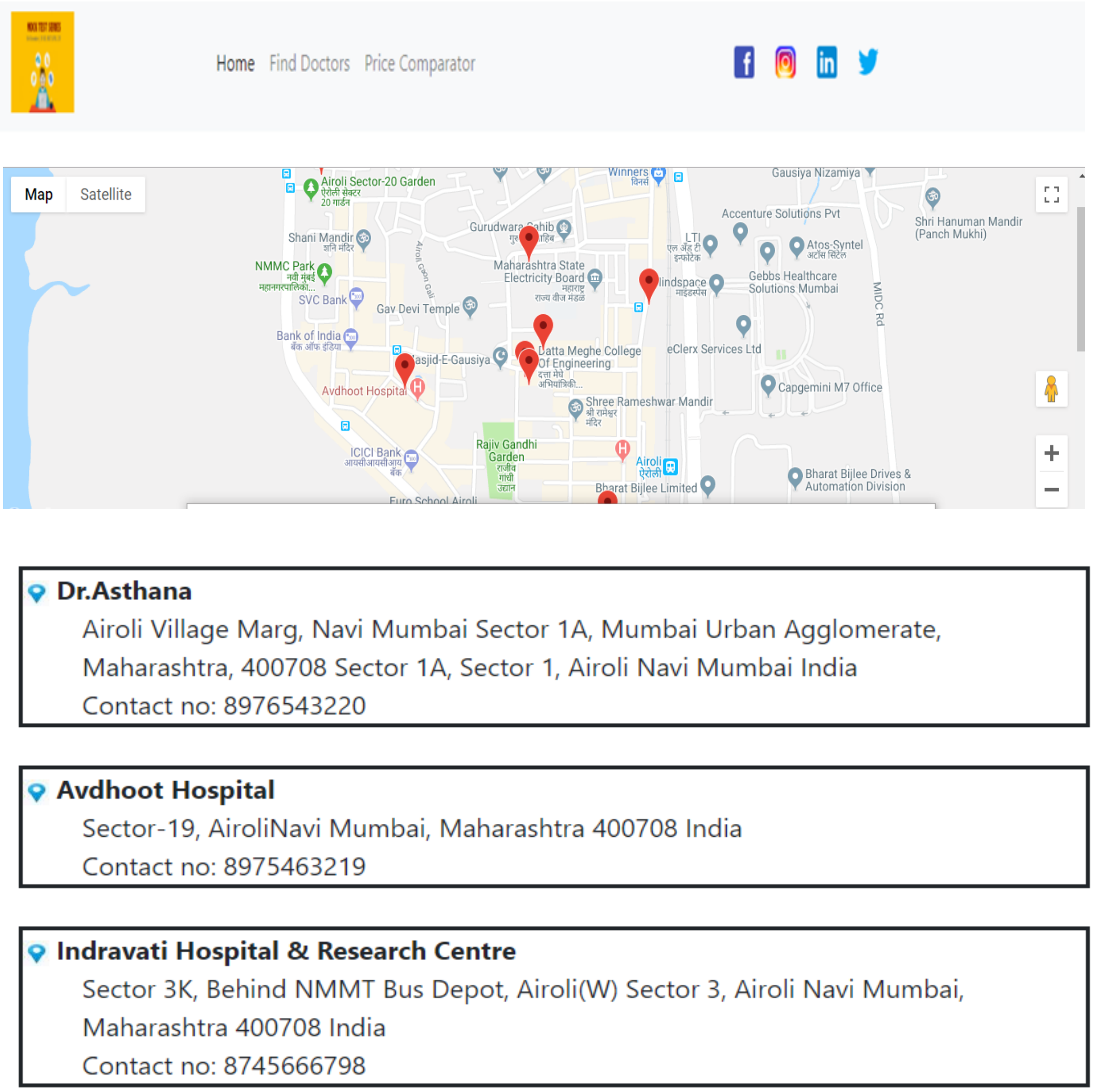
dizziness;vertigo;fall;unresponsiveness;weepiness

Fig 7.3 Screenshot of web page, user selects symptoms as per body part



Diseases	Percentage probability
colitis	10.0
gastritis	10.0
hepatitis C	10.0
ketoacidosis diabetic	20.0
lymphatic diseases	20.0
migraine disorders	10.0

Fig 7.4 Screenshot of web page, table of probable diseases is displayed



Dr.Asthana

Airoli Village Marg, Navi Mumbai Sector 1A, Mumbai Urban Agglomerate, Maharashtra, 400708 Sector 1A, Sector 1, Airoli Navi Mumbai India

Contact no: 8976543220

Avdhoot Hospital

Sector-19, AiroliNavi Mumbai, Maharashtra 400708 India

Contact no: 8975463219

Indravati Hospital & Research Centre

Sector 3K, Behind NMMT Bus Depot, Airoli(W) Sector 3, Airoli Navi Mumbai, Maharashtra 400708 India

Contact no: 8745666798

Fig 7.5 Screenshot of web page, of location of doctors

7.2 DISCUSSION

The system overall execution is described in above diagrams. The end user will interact to our system by means of a web-application. This web app is convenient and easily understood by the user as the entire functionality is present in the form of a flow. The users need to navigate from one page to other in step wise manner.

The figure 7.1 is the landing home page, which is followed by selection of body parts by the user. Fig 7.2 displays a screen shot of body part selection web page, which is followed by entering the symptoms. Fig 7.3 displays screenshot of web page, where user can enter symptoms pertaining to each body part. On submit, the list of diseases, with their individual percentage probabilities, based on symptoms is displayed in the form of table.

Once the user has used symptom checker, he or she can navigate towards finding doctors nearby to his location. The page asks for accessing location of user and then displays list of doctors near to him, in the form of map and tables. The tables also display address of those doctors and contact information. This is shown in figure 7.5.

The user can also use price comparator module, where he or she on entering names of branded prescribed medicines will fetch out a list of generic medicines.

Chapter 8

Conclusion and Future Work

8.1 Conclusion

We have discussed how this application can help person to detect any chances of disease, may be before turn into any serious disease so he or she can consultant doctor. We highlighted how machine learning algorithm can used to detect the chances of the disease. We have described different tools and technologies that can be used to build a system that will help us to identify the disease. Moreover, depending on the severity of disease our application will also provide the information of the nearby physicians and hospitals using the Geo location Google maps.

8.2 Future Work

The current system focuses on predicting the diseases and giving the information of the physicians and hospital using Geo location Google maps. In future, the system may become more interactive. For example, it will be able to schedule a doctor’s appointment nearby to the patient the system can be improvised to Chatbot instead of graphical based interface. In future, we would be taking helps of doctors and physicians to make our dataset even more powerful and robust, which would help in detecting any type of disease. Here we have used supervised learning algorithms; but this system can be made even more robust and accurate by choosing unsupervised learning algorithms like reinforcement learning methods. The system may detect complex diseases too, like skin diseases through image processing and neural network.

References

[1] http://people.dbmi.columbia.edu/~friedma/Projects/DiseaseSymptomKB/index.html

[2] https://www.nature.com/articles/ncomms5212

[3] https://www.nlm.nih.gov/research/umls/sourcereleasedocs/current/DDB/

[4] https://medlineplus.gov/encyclopedia.html

[5] "Prediction of probability of disease based on symptoms" by Harini D K and Natesh

[6] "Prediction in Heart Diseases using Data Mining" by Monica Gandhi and Shailendra Narayan Singh.

[7] https://en.wikipedia.org/wiki/Decision_tree.

[8] http://people.dbmi.columbia.edu/~friedma/Projects/DiseaseSymptomKB/index.html

Personal Health Assistant

GulshanKumar Bakle#1, Gayatri Asodekar#2, Vijay Chaudhari#3 , Prof. Chandrashekhar Raut #4

1,2,3Student, 4Assistant Professor

Department of Computer Engineering, Mumbai University

1gulshanbakle@gmail.com

2asodekarg@gmail.com

3vijay.c4dec1997@gmail.com

4cmr.cm.dmce@gmail.com

Abstract— **While initiating treatment for a particular disease, the main problem that arises, is mapping of symptoms with many diseases. Many symptoms may show presence of multiple diseases in the body of a patient. There has been a significant rise in data generated from healthcare sector, which could be efficiently exploited to give meaningful insights using data analysis and machine learning techniques. This paper aims to propose a system which would use disease-symptoms data and predict chances of various diseases a patient could be suffering with on taking inputs as symptoms. The system will use data analysis tools and machine learning algorithm- Random Forest in order to classify diseases. The system further enables patient to locate nearby doctors around him on maps and use a price comparator functionality that would fetch prices of generic medicines under given prescription.**

***Keywords*— data analysis, machine learning, random forest, classification, price comparator**

I. Introduction

Data has been scaling at an exponential rate in recent years. The scenario in past decade was as such; producer would produce huge amounts of data and on other leg, consumer would consume it. But the recent years has witnessed a swift change in scenario; both producer and consumer are producing and consuming data, which has led to immense rise in data being generated, especially in healthcare and financial sectors. Generally, in healthcare sector, diagnosis of a disease is done on the basis of symptoms that are visible in patient. With an increase in usage of internet and presence of information in the forms of blogs, forums, questions and answer portals, people have started spending more time over search engines to conduct self-diagnosis for the symptoms visible to them. Sometimes search based self-diagnosis may outcome with limited and incomplete information pertaining to prediction of diseases. Often self-diagnosis can lead to low quality results and unsubstantiated information.

Over the years, the symptoms with respect to different diseases have also seen tremendous variations. Many symptoms can be related to occurrence of more than one disease. Such instances, while diagnosis makes the treatment process to be extremely tedious and ambiguous. In order to facilitate with proper self-diagnosis and reasonable accuracy, concept of symptom checker has been proposed. A symptom checker comprises of three parts as, selection of affected body part, selection of symptoms being visible and prediction of diseases. Out of these three parts, prediction of diseases is an important task, since the predicted diseases should hold efficient accuracies. The other crucial task in building a symptom checker would be to provide a smooth user experience in sharing their symptoms in symptom checker.

The system will make use of database consisting of all symptoms pertaining to various diseases. Since every industrial database is present in a raw format, it needs to be converted into usable and clean format using data pre-processing techniques. Now this clean and processed data would be used for training our machine learning model using random forest algorithm. The result of our system will give out probability percentages of various diseases the patient might be suffering with. Along with this, we have added two more extra functionalities in our system that will able it to be used as a personal health assistant. First, using Geo-Positioning System (GPS), the patient would be able to locate the nearby physicians, doctors and hospitals present around him or her, for treatment of those predicted diseases. Second being a generic medicine comparator that would fetch out the list of various low priced medicines of the same chemical components as that of branded medicines; prescribed by doctors.

The section II of this article will explore about the related work being done in this field. The section III will throw light on random forest algorithm which would be used in classifying the chances of various diseases based on symptoms. The section IV will focus on proposed system for personal health assistant. Section V will refer to the experiment and result about each module which is used in this system. Section VI will conclude the paper.

II. LITERATURE REVIEW

There are many proposed systems that predict diseases based on the symptoms. Md. Tahmid Rahman Laskar, Md. Tahmid Hossain, Abu Raihan Mostofa Kamal, Nafiul Rashid proposed a system in a paper[1] that works on Relevant Attribute (RA) Data Structure which takes five relevant parameters S = Symptom name T = Time I = Intensity O = Organ name D = Duration from the user as input. The user will give symptoms in the form of text input. The input is then scanned and tagging of each word is done according to the RA data structure. Synonym Parent Tree, Symptom Reference Tag and Decision Tree Relevant Attribute Array techniques are used for tagging. Then a Data matrix will be formed using RA, from which the symptoms will be retrieved and mapped with the symptoms that are already in the databases. After that asymmetric binary similarity factor is calculated.

Prediction System for Diseases and Suggestion of Appropriate Medicines in this paper[2] author Disha Mahajan, Mrudula Phalak, Shubhangi Pankore, Saniya Pathan, Deepa Abin proposed a system that not only predicts the disease but also suggests medicines for it. This system uses data mining for predicting diseases. They used the dataset of MedlinePlus. This system takes symptoms as input from the user then checks the symptoms in the database after that it maps the symptoms with the disease and checks the condition for supplement medicine and display the predicted disease with suggested medicines.

In the paper [3] Disease Prediction and Doctor Recommendation System published by Dhanashri Gujar, Rashmi Biyani, Tejaswini Bramhane, Snehal Bhosale, Tejaswita P. Vaidya proposed a system that predicts the disease and also recommends a doctor near the user location. The prediction of disease is done by data mining. The system uses Naive Bayes that is implemented using WEKA libraries which contain a collection of tools for visualization and also contain algorithms for data analysis and predictive modelling. Doctors are recommended based on the location as well as the reviews of the doctors that are done by fetching the various doctors based on reviews. Core NLP is used for processing the data that are fetched.

Miss Swati Y. Dugane, Prof. Karuna G. Bagde proposed a system [4] that predicts the disease as well as gives detail information of disease in question like form. The disease prediction is done using deep learning method. Deep learning is part of machine learning. The user can search the symptoms or can ask the questions to the system will respond in the form of the associated data set.

Harini D K, Natesh M in a paper [5] proposed a system that predicts the probability of disease-based on symptoms using machine learning algorithm. The system combines the structure and unstructured data and then to reconstruct missing data system uses the latent factor model. After that to it uses statistical knowledge to determine the major chronic disease. Then, it uses data that are been consulted with hospital expert to extract useful features so it can be used as structured data. Using CNN algorithm Features of unstructured data can be selected. The system uses three algorithms: KNN algorithm, Naïve Bayesian and Decision tree for prediction of disease.

III. RANDOM FOREST ALGORITHM

Random Forest algorithm is one of supervised algorithm in machine learning, which can be effectively used for binary as well as multi label classification. Random forest or random decision forests are an ensemble learning method, which is used for classification as well as regression problems. It operates by preparing multiple numbers of decision trees on the same dataset, involving different sequence and range of parameters or features. Once multiple decision trees are constructed, all the decisions of individual trees are merged together to obtain more accurate and stable result. Often decision trees are bound to suffer from over-fitting to their training dataset, random forest tends to solve this issue since it operates by calculating the mean of obtained classes of decision tree. Random forest algorithm finds its application in many fields, since it is easy to use and can be effectively used for both classification and regression purposes.

Random forests can be termed as upgraded variant of decision trees. The major difference between them being, decision trees are built over entire dataset, while random forests are used to create multi-decision trees on subset of dataset.

Any decision tree works according to general principle, for predicting any sample or class label, it creates a set of general rules on the features present in dataset. Among all the features, one of the features is selected as root node, and other features as internal nodes. We continue to split out our internal nodes until we reach at particular classified or predicted class. Every decision tree consists of following components:

5) *Root node*: This is the top most node present in decision tree. It splits the entire dataset into two halves. Root node is the node which has lowest value of impurity.
6) *Internal node*: Root node splits into internal nodes. All the results made out on branches are internal nodes. These internal nodes further branch out into leaf nodes.
7) *Branch*: Every branch indicates an outcome or possible action to be taken.
8) *Leaf nodes*: These are the end nodes of our decision tree.

Every decision tree works in a flow chart like structure, where each internal node denotes a test on attribute, branch indicating outcome of a test and each leaf node representing class labels. Both decision tree and random forest construct trees using certain metrics. These metrics help in measuring best way to make splits in tree. The metrics are Gini impurity, information gain and variance reduction.

Random forest algorithm works on the principle of bootstrap aggregation. The bootstrap is a powerful statistical method which is used for estimation of a quantity from a data sample. Given a training set as A = a_1, a_2, a_3 ..., a_n with responses, let's say as Y = y_1, ..., y_n, bagging repeatedly (*B* times) selects a random sample with replacement of the training set and fits trees to these samples:

For *b* = 1,2, 3,..., *B*:

1) Sample, with replacement, *n* training examples from *A*, *Y*; call these A_b, Y_b.
2) Train a classification or regression tree f_b on A_b, Y_b.

After training our model, it can be tested for unseen samples *a'*, final decision can be made by averaging the predictions from all regression trees drawn on sample *a'*.

$$f' = \frac{1}{B}\sum_{b=1}^{B} f(x')$$

Or by taking the majority vote in the case of classification trees.

This bootstrapping procedure leads to better model performance since it reduces the variance of the model, without increasing the bias. Since the training and prediction of single tree is much prone to noise, the average of large number of trees is not, as long as the trees are not correlated. Simply training many trees on a single training set would give strongly correlated trees (or even the exactly the same tree many times, if the training algorithm used in problem is deterministic); bootstrap sampling is a way which helps in de-correlating the trees using different training sets for separate trees. The number of samples per trees, *B*, is a free parameter. Typically, a few hundred to several thousand trees are used, depending on the size and nature of the training set of our problem.

The procedure described above is original bagging algorithm for trees. Random forests differ in only one way from this methodology: they use a modified tree learning algorithm that selects, at each candidate split in the learning process, a randomised subset of the features from the training dataset. This process of section randomly is called "feature bagging". The reason for doing this is the correlation of the trees in an ordinary bootstrap sample: if one or a few features are very strong predictors for the response variable (target output), these features will be selected in many of the *B* trees, causing them to become correlated. Thus, bootstrapping in random forest helps in making the trees de-correlated and improving the accuracy by taking mean of all tree results.

IV.PROPOSED SYSTEM

At beginning, our system will mimic like a real doctor, by asking which body part is being affected. The patient will need to select the body part from the displayed list. The second step as a doctor asks about is symptoms, similarly our personal health assistant will display a set of symptoms pertaining to those selected body parts. The patient would be able to select multiple symptoms that he or she is suffering with. On the basis of symptoms given by patient, the system will make predictions of the probable diseases in the form percentages. The predictions are made by our random forest classifier object which is trained beforehand on dataset of symptoms and disease [6]. The patient would be able to see a list of predicted diseases given by our random forest classifier.

Since the dataset present online is raw data, it must be pre-processed using various data pre-processing technique. Our dataset consisted of three columns- symptoms, diseases and weight. Here weight parameter signifies the number of instances of such diseases. The dataset is processed such that all the symptoms are turned into columns and last column would be of disease column. The symptoms pertaining to particular diseases are marked as 1, while rest 0. This processed data is now trained using random forest classification algorithm. The reason behind choosing random forest classifier, as machine learning algorithm, is its ability construct multiple decision trees on features and glue them together to get a more accurate and stable prediction. The forest that it builds is an ensemble of many decision trees which are trained using bagging method.

The second functionality added in our personal health assistant being, locating the nearby doctors with the help of current Geo Positioning System (GPS) location of the patient using our web application. The patient would be able to see a list of doctors, physicians and hospitals around him or her on map.

The third functionality that our personal health assistant would provide is a price comparator for medicines. As the prices of branded medicines are high, not everyone is able to afford them. A substitute for them is generic medicines, which contains the same chemical component as that of other medicines but at lower prices. The patient would need to enter the prescribed medicine name in the system, which in return will fetch out a list of all generic medicines with same chemical components as that of prescribed medicines. This will allow the patient to make a proper choice for purchasing the medicines. A figurative representation of entire system is shown in figure 1.

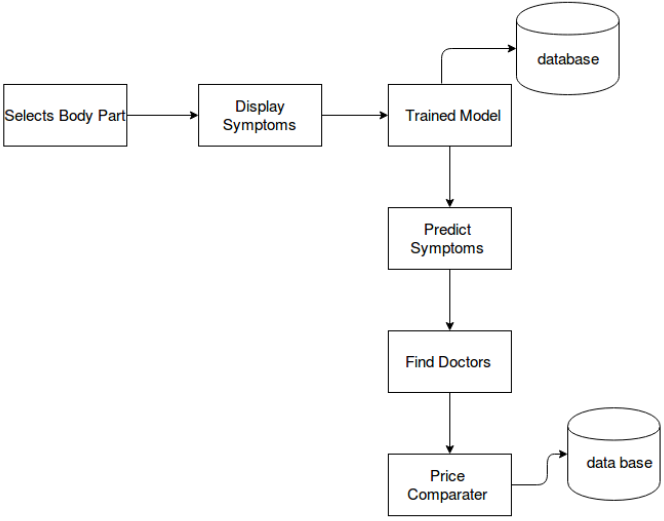


Fig. 1 Proposed system of personal health assistant

These all functionalities are represented in a form of a web application. For front end purposes, we have used HTML, CSS and JavaScript in order to bring out responsiveness in real time. The back-end part of this web app is made with Django framework. Django provides an easy way to integrate database with website, and also indulges in easy implementation of machine learning in a form of web app. For machine learning part we have used python language, since it contains wide range of inbuilt data analysis libraries like numpy, pandas, matplotlib, seaborn, etc.

IV. EXPERIMENTS AND RESULTS

The personal health assistant consists of three important functionalities- symptom checker, doctor locater and medicine price comparator. Working of each module is explained below in detail:

A. Symptom Checker:

The symptom checker uses symptoms diseases database, which is pre-processed and trained using machine learning algorithm- random forest. The information within this database is transformed into dataset. A dataset is the collection of related sets of information that is composed of separate elements but can be manipulated as single unit. As our database consisted of three columns- symptoms, disease and weight of instances of symptoms noticed for that disease. We used python libraries like numpy and pandas for processing these three columns into encoded form. The raw dataset is converted into clean dataset, thus removing any missing, repetitive and faulty data. This clean dataset is now turned into a form that can be used for training our model. All the unique symptoms are turned into columns and a separate label column for unique disease names. The cells of symptoms pertaining to each disease are filled with 1 and rest cells as 0. There are in total 148 unique diseases in our dataset and 404 unique symptoms.

Following libraries were used while building our symptom checker:

1) Scikit learn library: Scikit-learn is a free machine learning library especially built for python language. It features many algorithms like decision tree, random forests, and Multinomial Naive Bayes, and it also supports python numerical and scientific libraries like NumPy and SciPy.
2) Numpy: NumPy arrays facilitate advanced mathematical and other types of operations on large numbers of data. Such operations are executed more efficiently and in less lines of code as compared to other python operations.
3) Pandas: Pandas is popular package used widely in machine learning, commonly much used for data manipulation and analysis. Much of data pre-processing task is done using Numpy and Pandas.

We have used scikit learn library, from which we will import our random forest classifier.

Before applying random forest classifier on our dataset, we split the dataset into two parts- training set and testing set. The split is done in the ratio of 80:20. The total length of our dataset is 148, that is, 148 unique diseases which are split in the mentioned ratio. Training set consists of 118 diseases while testing set with 30 diseases. The training set was made up of two components, x train and y train; x train consisting of all the symptoms and y train list of diseases of those symptoms. When random forest classifier was applied on our entire dataset, we obtained an accuracy score of 90.54 %. When this classifier was run on our training dataset, it gave us an accuracy of 88.98%, while on testing dataset, accuracy rose to 96.66%. This result has been formulated in table 1. There were 13 such instances, when our classifier gave false predicted diseases.

Table 1

Dataset	Accuracy achieved in percentage
Complete dataset with 148 diseases	90.54
Training dataset with 118 diseases	88.98
Testing dataset with 30diseases	96.66

There are in total 404 unique symptoms which act like features in our classifier. Some of the features are important in constructing multiple decision trees, based on how efficiently the nodes reach out to predict class labels. Feature importance is generally calculated as the decrease in impurity of node, which is weighted by the probability of reaching till that node in tree. The probability of node can be calculated by dividing the number of samples that reach the node, by the total number of samples. Higher the value of probability, more important the feature would be.

Table 2

shows list of top features, which showed higher node probability, thus come under important features:

Table 2

Feature index	Feature name
255	Pain abdominal
70	Cough
122	Fever
80	Diarrhoea
45	Breathiness sound decreased
87	Dizziness

We have used matplot library and seaborn library in order to plot most important features among 404 symptoms. As we can observe from table 2, pain abdominal forms most important feature in the entire set of features which is followed by cough, fever, diarrhoea and breathiness sound decreased. The feature importance values for each one of above features (symptoms) is shown in figure 2.

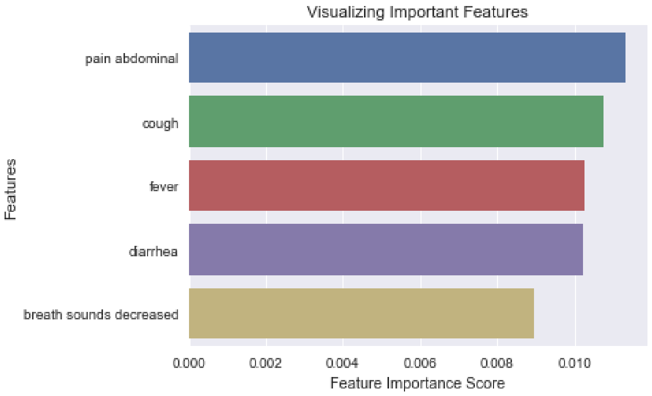


Fig. 2 Feature importance values of symptoms

From figure 2 we can interpret that pain abdominal has highest feature importance value as 0.011 and cough as 0.0107. These features would be present in most of the decision trees as internal or leaf nodes while traversing towards classified disease. Higher the values of feature importance, indicates that feature to reduce the impurity across all trees in the forest.

B. Doctors locator:

Once our symptom checker has shown the list of probable diseases that the patient might be suffering with, our system will also enable them to locate some of the nearby doctors and physicians treating those diseases. This module of the system would require current location of the patient, in order to locate doctors around him or her. The location can be known with the help of Geo Positioning System (GPS), which will track the current coordinates of patient and use it for locating nearby hospitals. The patient at the end of it would be able to see a list of hospitals near to him on map and in a list format as well.

C. Price Comparator:

Since the prices of all branded medicines are a bit higher, but there is a substitute for them in the form generic medicines. These generic medicines have same chemical content as present in branded ones, but are sold at cheaper price as compared to former. Thus, our system will provide a functionality where patient can enter names of prescribed medicines, and will fetch out details of all generic medicines under that prescription.

IV. CONCLUSION

We have discussed how this application can help person to detect any chances of disease, may be before turn into any serious disease so he or she can consultant doctor. We highlighted how machine learning algorithm can be used to detect the chances of the disease. We have described different tools and technologies that can be used to build a system that will help us to identify the disease. Moreover, depending on the severity of disease our application will also provide the information of the nearby physicians and hospitals using the “GPS” (Geo Positioning System) and Google maps.

ACKNOWLEDGEMENT

The authors wish to thank the Management and Principal of Datta Meghe College of Engineering, for their support in carrying out this research work. Also, would like to thank our guide Prof. Chandrashekhar Raut for guiding us throughout the paper.

REFERENCES

[1] Md.Tahmid Rahman Laskar, Md. Tahmid Hossain, Abu Raihan MostofaKamal ,Nafiul Rashid ”Automated Disease Prediction System (ADPS): A User Input-based Reliable Architecture for Disease Prediction”International Journal of Computer Applications (0975 – 8887) Volume 133 – No.15, January 2016

[2] Disha Mahajan, MrudulaPhalak, ShubhangiPankore, Saniya Pathan, Deepa Abin “Prediction System for Diseases and Suggestion of Appropriate Medicines” International Journal of Innovations & Advancement in Computer Science IJIACS ISSN 2347 – 8616 Volume 6, Issue 11 November 2017.

[3] Dhanashri Gujar, Rashmi Biyani, TejaswiniBramhane, Snehal Bhosale, Tejaswita P. Vaidya“Disease Prediction and Doctor Recommendation System” International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056 Volume: 05 Issue: 03 Mar-2018.

[4] Miss Swati Y. Dugane (ME, CSIT) Prof. Karuna G. Bagde “Framework for Disease Prediction from Symptoms and Health related data”International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)Volume 6, Issue 9, September 2017.

[5] Harini D K, NateshM” Prediction of Probability of disease-based o symptoms using machine learning algorithm”International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056 Volume: 05 Issue: 05 May-2018.

[6] http://people.dbmi.columbia.edu/~friedma/Projects/DiseaseSymtpomKB/index.html