

A

Major Project Report

on

**EARLY PREDICTION OF LIFESTYLE
DISEASES**

Submitted in Partial Fulfillment of
the Requirements for the Degree
of

Bachelor of Engineering

in

Computer Engineering

to

**Kavayitri Bahinabai Chaudhari
North Maharashtra University, Jalgaon**

Submitted by

**Rajput Pruthviraj Dineshsing
Chaudhari Harshada Manohar
Chouhan Sumit Amarnath
Patil Kirti Ravindra**

Under the Guidance of

Dr. Manoj. E. Patil



DEPARTMENT OF COMPUTER ENGINEERING
SSBT's COLLEGE OF ENGINEERING AND TECHNOLOGY,
BAMBHORI, JALGAON - 425 001 (MS)
2021 - 2022

**SSBT's COLLEGE OF ENGINEERING AND TECHNOLOGY,
BAMBHORI, JALGAON - 425 001 (MS)
DEPARTMENT OF COMPUTER ENGINEERING**

CERTIFICATE

This is to certify that the major project entitled *Early prediction of lifestyle diseases*, submitted by

**Rajput Pruthviraj Dineshsing
Chaudhari Harshada Manohar
Chouhan Sumit Amarnath
Patil Kirti Ravindra**

in partial fulfillment of the degree of *Bachelor of Engineering in Computer Engineering* has been satisfactorily carried out under my guidance as per the requirement of Kavayitri Bahinabai Chaudhari North Maharashtra University, Jalgaon.

Date: April 17, 2022

Place: Jalgaon

**Dr. Manoj. E. Patil
Guide**

Dr. Manoj. E. Patil
Head

Dr. G. K. Patnaik
Principal

Contents

Acknowledgements	2
Abstract	3
1 Introduction	4
1.1 Background	4
1.2 Motivation	5
1.3 Problem Definition	5
1.4 Scope	5
1.5 Objective	6
1.6 Selection Of Life Cycle Model	6
1.7 Organization of Report	7
1.8 Summary	8
2 Project Planning and Management	9
2.1 Literature Survey	9
2.2 Feasibility Study	10
2.2.1 Technical Feasibility	10
2.2.2 Economical Feasibility	11
2.2.3 Operational Feasibility	11
2.3 Risk Analysis	11
2.4 Project Scheduling	12
2.5 Effort Allocation	12
2.6 Cost Estimation	14
2.7 Summary	15
3 Analysis	16
3.1 Requirement Collection and Identification	16
3.2 Software Requirement Specification	17
3.2.1 Product Features	17
3.2.2 Operating Environment	17

3.2.3	Assumptions	17
3.2.4	Functional Requirements	17
3.2.5	Non-Functional requirements	18
3.2.6	External Interfaces	18
3.3	Modules	19
3.3.1	User Modules	19
3.3.2	Doctor Module	20
3.3.3	Admin Module	20
3.4	Summary	21
4	Design	22
4.1	System Architecture	22
4.2	E-R Diagram	23
4.3	Data Flow Diagram	24
4.3.1	level 0 DFD	24
4.3.2	level 1 DFD	25
4.4	UML Diagrams	26
4.4.1	Use Case Diagram	26
4.4.2	Sequence Diagram	27
4.4.3	Collaboration Diagram	31
4.4.4	Activity Diagram	32
4.4.5	Class Diagram	33
4.4.6	Component Diagram	35
4.4.7	Deployment Diagram	36
4.5	Summary	36
5	Implementation	37
5.1	Implementation Details	37
5.2	Implementation Environment	37
5.2.1	React Js	38
5.2.2	HTML	38
5.2.3	CSS	38
5.2.4	JavaScript	38
5.2.5	Node Js	38
5.2.6	Mongoose	38
5.2.7	Express Js	39
5.2.8	MongoDB	39
5.2.9	Netlify	39

5.2.10	Heroku	39
5.2.11	PWA (Progressive Web App)	39
5.3	Flow of system development	39
5.4	Summary	40
6	System Testing	41
6.1	How To Implement Testing	41
6.2	Black Box Testing	42
6.3	White Box Testing	42
6.4	Test Cases And Test Results	42
6.5	Summary	47
7	Results and Analysis	48
7.1	Database Structure	48
7.1.1	User Module Document Object	48
7.1.2	Doctor Module Document Object	49
7.1.3	Disease Module Document Object	49
7.1.4	Question Modules Document Object	50
7.1.5	Feedback Module Document Object	50
7.1.6	Contact Module Document Object	51
7.1.7	Chat Module Document Object	52
7.2	Overall System Structure	52
7.2.1	Snap Shorts	53
7.3	Summary	62
8	Conclusion and Future Work	63
8.1	Conclusion	63
8.2	Future Work	63
Bibliography		64
Index		64

List of Tables

2.1	Effort Allocation Table	13
6.1	Text Case 1	43
6.2	Text Case 2	43
6.3	Text Case 3	43
6.4	Text Case 4	44
6.5	Text Case 5	44
6.6	Text Case 6	44
6.7	Text Case 7	45
6.8	Text Case 8	45
6.9	Text Case 9	46
6.10	Text Case 10	46
6.11	Text Case 11	47

List of Figures

1.1	Selection Of Life Cycle Model (Waterfall Model)	7
2.1	Project Scheduling	12
2.2	Effort Allocation	13
2.3	Cost Estimation	15
4.1	System Architecture	23
4.2	ER Diagram	24
4.3	Level 0 DFD	25
4.4	Level 1 DFD	26
4.5	Use Case Diagram	27
4.6	Sequence Diagram Login Logout	28
4.7	User (People) Sequence Diagram	28
4.8	Doctors Sequence Diagram	29
4.9	Admin Sequence Diagram	30
4.10	Collaboration Diagram	31
4.11	Activity Diagram (User View)	32
4.12	Activity Diagram (Admin View)	33
4.13	Class Diagram	34
4.14	Component Diagram	35
4.15	Deployment Diagram	36
7.1	User Document Object	48
7.2	Doctor Document Object	49
7.3	Disease Document Object	50
7.4	Question Document Object	50
7.5	Feedback Document Object	51
7.6	Contact Document Object	52
7.7	Chat Document Object	52
7.8	Home Page	53

7.9	Symptom Checker	53
7.10	Symptom Checker Welcome Window	54
7.11	Symptom Checker Patient Age Window	54
7.12	Symptom Checker Symptoms Select Window	55
7.13	Symptom Checker Possible Cases Result	55
7.14	About Us Page	56
7.15	Diseases Display Page	56
7.16	Disease Information	57
7.17	Disease Related Ask Question	57
7.18	Specific Disease Add Feedback	58
7.19	User Profile	58
7.20	Login Page	59
7.21	Sign up Page	59
7.22	Login Page Doctor	60
7.23	Admin Dashboard	60
7.24	All Doctor Display	61
7.25	All Diseases Display	61
7.26	All Diseases Display	62
7.27	All Question Display	62

Acknowledgements

This project required a lot of guidance and assistance from many people. All that I have done is only due to such supervision and assistance and would not forget to thank them. Like to express my deep gratitude and sincere thanks to all who help us in this major project. Would like to thank Principal Dr. G. K. Patnaik, SSBT COET for having provided us facilities to complete work. Deep gratitude goes to Dr. Manoj. E. Patil, head of the department and major project guide Dr. Manoj. E. Patil, for granting us opportunity to conduct this major project work as well as his valuable suggestions and guidance at the time of need. Sincerely thankful to and fortunate enough to get constant encouragement, support and guidance from all Teaching staff of the Computer Department of SSBTs COET which helped us in our project work. Also would like to extend our sincere esteems to all staff in the laboratory or their timely support. I am also thankful to my Parents. Last but not least, thank God.

Rajput Pruthviraj Dineshsing
Chaudhari Harshada Manohar
Chouhan Sumit Amarnath
Patil Kirti Ravindra

Abstract

Lifestyle diseases are common among the population today not only in India but also in almost every country. Lifestyle diseases are caused because of the habits that we have on a day to day basis. The way one lives his life is the major cause of it. It includes heart disease, hypertension, etc. which all may hear of. In our life also, one also comes across at least one person who is either suffering from such diseases or the diseases became the reason for his death. We also came across many such people who died because they were not aware of their disease and were left with no appropriate time for treatment.

That is why we decided to develop the portal which will analyse the data entered by the user and will give the predictions of the diseases which he or she may have chances to suffer from. This not only gives the predictions but also gives you the preventive measures that are required to stay safe from the very common lifestyle diseases as well as in case of mild symptoms it provides you with the management techniques also. This project makes the person aware of his health so that he will have the treatment well in time if required and will save the lives of many people. This project covers three main aspects which are prediction, prevention and management of lifestyle diseases.

Keywords:- Lifestyle diseases, death,treatment, common lifestyle, prediction, prevention, management .

Chapter 1

Introduction

Today, people do not have time for the regular checkup. They are so busy with their work that they rarely have time for their own health. But the thing is that, they can do the analysis if the appropriate application can provide them the overall health status of the person. This is because they need not to give the time separately for this, rather they can just utilise the time for example the time of travelling, etc. The only thing required is the smartphones which almost everyone has with them in this century. So, this can be considered as the portable health checker which everyone can use easily just through a web application. In this, we have decided to give the sign up page where the user can sign up using his name, id and password. Then further modules will have the diseases portion. Though the wholehealth check is a somewhat more difficult task, so initially we are adding some of the very common diseases data like heart disease, breast cancer, etc. Later on, we will keep adding more and more diseases.

The organization of this Chapter is as follows. Section 1.1 describes Background of the project. The motivation behind this project selection is represented in Section 1.2. Section 1.3 represents the Problem Definition of the project. Scope of the project is described in Section 1.4. Section 1.5 describes the Objective of the project. Selection of Life Cycle Model for Development of the project is represented in Section 1.6. Section 1.7 represents the Organization of the project. Finally, The Summary is described in the last Section 1.8.

1.1 Background

Prediction Of the disease during the pandemic is critical for public health planning of efficient health care allocation and monitoring the effects of policy interventions.

In another study, an algorithm is proposed to achieve local updates and global updates which is critical for the learning process. ML/AI is also used to solve wireless network problems. Cheat al. represented how artificial neural networks can be used to solve various problems in wireless networks.

Diabetes has been in society for a very long time. Diabetes is further dependent on an individual's body, diet, and way of living. In another study, pre-diabetes is predicted using different applications on the Korean population.

Death in intensive care units (ICU s) worldwide and its recognition, particularly in the early stages of the disease, remains a medical challenge.

1.2 Motivation

Lifestyle diseases are common among the population today not only in India but also in almost every country. Lifestyle diseases are caused because of the habits that we have on a day to day basis. The way one lives his life is the major cause of it. It includes heart disease, hypertension, etc. which all may hear of. In our life also, one also comes across at least one person who is either suffering from such diseases or the diseases became the reason for his death. We also came across many such people who died because they were not aware of their disease and were left with no appropriate time for treatment. That is why we decided to develop a portal for showing early disease information and solutions or precautions.

1.3 Problem Definition

In this platform, the people and doctors are joined and they are registered login logout and also they have early knowledge of what is happening in their area and the true information is given to this platform. About the disease which is a large amount of spearing in this area first of all the people see and their area which diseases is spread and also check their symptoms, also in bottom, and also give what the precaution do for the disease not come, also if you have any other disease then also have a chat box to say your problem there was the best doctor, and they convey you to what to do for these diseases. And also doctors joined this platform as a sub-admin. They solve the people's posted questions, and also add current disease information and safety tips and what precautions we do for these diseases and also know about these doctors who give this information. Admin doctor verifies this doctor's added information and verifies the doctor's status and manages the site settings.

1.4 Scope

This project will help the user to overcome the lifestyle diseases which are in themselves a big threat to humans, will reduce the unawareness about the diseases and will help people to remain healthy which is of utmost importance in today's fast-growing world. It will also

change the lifestyle of people for the better. It will also give clarity about the health of a person or his current status.

In future mostly AI and ML is going to be implemented everywhere. Everyone will get so busy with their work that they will not get enough time to visit the doctor. Exceptionally when they get serious. But will ignore the minor and common disease which will eventually become more serious in future. Like in tuberculosis, a person starts to cough and only with preventive measures they can be submerged at that time. so this app will become more and more common in future. Without visiting doctors' people will get to know about their symptoms.

1.5 Objective

Objective of this project is as follows;

- Easly predicate diseases possibilities .
- Show diseases information.
- Clear people doutetes about any disease.
- Installable in Mobile or iOS as well as Desktop.

1.6 Selection Of Life Cycle Model

There are many different software development life cycle models. They represent the logical constructed sequence of actions, starting with the need identification and finishing with the software production. Each model represents the process, which consists of the stages aimed at ensuring the integrity of the appropriate actions. Each fully exhaustive stage reduces the degree of project risk due to the application of the criteria of entry and exists to determine the future actions.

The software development life cycles are the techniques, which cover all standards and procedures which impact on the planning, requirements process, design, construction (coding and testing), implementation of the software system. The popular generalized models provide only possibility of its use for certain project types. The life cycle must be carefully selected according to tasks and goals of the certain project to ensure the efficiency of the life cycle.

We are the select waterfall life cycle model because all requirements are easy to understand and also development team members have less domain knowledge.

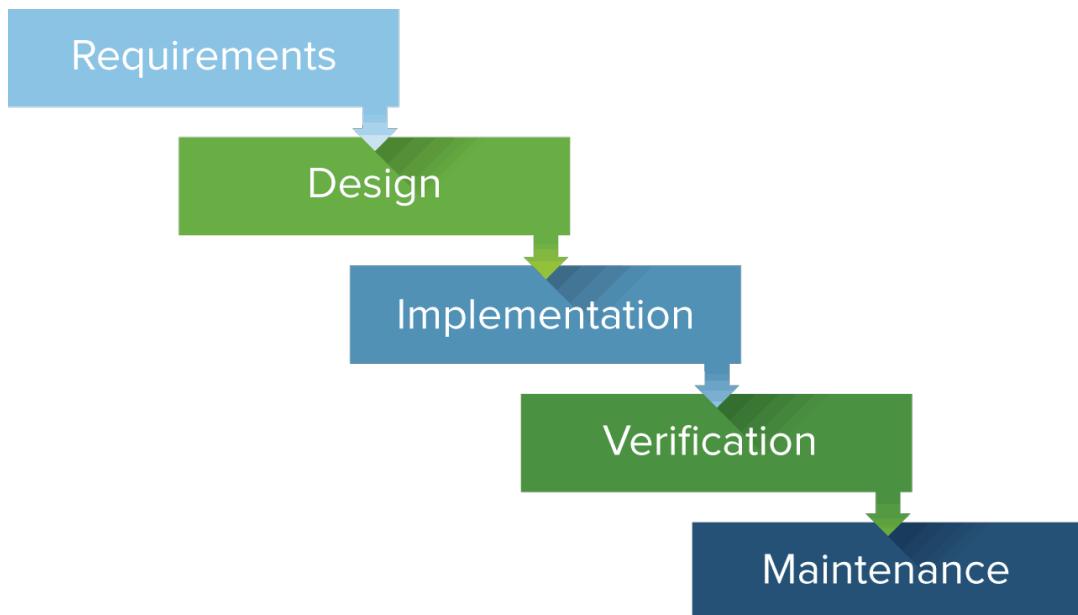


Figure 1.1: Selection Of Life Cycle Model (Waterfall Model)

1.7 Organization of Report

The Organization of the Report is structured in the following way:

- **CHAPTER 1**, titled Introduction, presents introduction to the proposed system, problem statement, problem definition, objective and future scope.
- **CHAPTER 2**, titled Project Planning Management, presents a literature survey, existing system for data perturbation approach, proposed system, feasibility study and risk analysis.
- **CHAPTER 3**, titled System Requirements specifications, presents hardware and software requirements needed for the proposed system.
- **CHAPTER 4**, titled System Design, presents proposed system Flow which provide detail information about project flow, data flow diagrams, E-R Diagram, UML Diagrams.
- **CHAPTER 5**, titled Implementation, presents flow of system development, implementation details, implementation Environment used for project development.
- **CHAPTER 6**, titled System Testing, presents how to perform testing, strategy for testing conventional software, test cases which performed on developed system to determine system works properly.

- **CHAPTER 7**, titled Result and Analysis, presents perturbation methods Result analysis with existing methods, which proves that proposed mechanism gives very high performances and low error rate compared with existing methods.
- **CHAPTER 8**, titled Conclusion and Future Scope, presents conclusion for proposed approach and future scope provide the direction for future development in data perturbation method.

1.8 Summary

In this chapter, Introduction is presented. In the next chapter, System Analysis is presented.

Chapter 2

Project Planning and Management

Project planning is a procedural step in project management. It is the practice of initiating, planning, executing, controlling and closing the work team to achieve specific goals. Project planning and management is important because it ensures that the right people do the right things, at the right time. It also ensures the proper project life cycle.

The organization of this Chapter is as follows. Section 2.1 Literature Survey is presented. Section 2.2 presents Feasibility Study. Risk Analysis is described in Section 2.3. Project Scheduling is presented in Section 2.4. Effort Allocation is described in Section 2.5. Section 2.6 describes Cost Estimation Finally summary is presented in the last section 2.7.

2.1 Literature Survey

After a survey of this topic. We are developing this project using a machine learning concept that is Data Mining. Also another concept like speech recognition, many others.

Data Mining is a technique of analyzing the huge amount of data in different aspects to discover useful information or knowledge discovery. It combines the concepts of artificial intelligence, statistics, probability, machine learning, deep learning and database system technology. The processes of data collection, selection, cleaning, handling the missing values, transformation, mining, evaluation of pattern, and knowledge visualization involved in the data mining process.

The data is increasing exponentially as in the case of the health sector. It is also a major data producing sector which is not only heterogeneous but also valuable as it stores the sensitive health information of the person which can even cost the life of a person. The majority of the methods are used to predict, prevent and manage the diseases appropriately and efficiently. The medical diagnosis is subjective and important in other aspects and depends upon the data available and in this case the data entered by the user.

Healthcare related data mining is a difficult field as some minor changes may lead to the huge difference in the predictions and will further affect the output. It explores the

hidden patterns which further helps in discovery and extracting knowledge in a database to predict diseases that a person may suffer from. We will use both the core models of data mining i.e., descriptive as well as predictive in big data. In case of descriptive data analysis, it uses user data to identify the patterns in the data and analyze the relationship between various variables and samples. Descriptive models are prior association rule, data clustering, summarizing and visualization. These models are generally developed by using complete data set but we will try to reduce the number of variables or samples required to predict the output which increases its performance as well as the efficiency.

While in case of predictive data analysis, it uses historical data and current data for predicting the probabilities of the future lifestyle diseases or used for diagnosing and curing the diseases as well. But in case of severe symptoms, it will always suggest the user to consult the doctor as soon as possible. Further enhancement of the model can include the nearby hospitals or the clinics available using the google maps. This can be done by several techniques like Dijkstra's algorithm. CART Decision trees, artificial neural network (ANN), random forecasting and the regression (linear, logistic and ridge) are the commonly used predictive data models

2.2 Feasibility Study

A Feasibility Study is an analysis of the viability of an idea. It ensures that a project is technically, economically feasible, and operationally justifiable. The early Prediction of Life cycle Diseases Software / Healthcare is among the most popular tools that support the viewing and understanding of spread of Life cycle and Pandemic disease. It also provides a detailed documented status to the user. Feasibility is also useful to identify the risk, cost, and benefits related to economics, technology, and user operation. Feasibility is the first stage in the process of project development. There are several types of feasibility depending on the aspects they cover. Some important feasibility's are as follows:

2.2.1 Technical Feasibility

The early Prediction of Life cycle Diseases Software / Healthcare is among the most popular tools that support the viewing and understanding of spread of pandemic disease. The technical feasibility study assesses the details of how developers will deliver a product or service in the available market place. Technical feasibility is very important and significant because it is helpful to decide realistic business models for a particular product. This Software / Healthcare is to provide the platform to the user on this platform, the people and doctors are joined and they are registered login logout and also they have early knowledge of what

is happening in their area and the true information is given to this platform.

2.2.2 Economical Feasibility

This includes an evaluation of incrementally costs and benefits expected if the beginning opposed system has economized. The users will be economically satisfied with this application because this application is free of cost to the users and users can only require log-in in this application to add their reviews. So developing this application is economically feasible.

2.2.3 Operational Feasibility

Operational feasibility helps in taking advantage of the opportunities and fulfills the requirements as identified during the development of the project. It takes care that business clients and users support the project. This software is to provide they have early knowledge of what is happening in their area and the true information is given to this platform.

In this application, doctors also joined this platform as a sub-admin. They solve the people's posted questions, and also add current disease information and safety tips and what precautions we do for these diseases and also know about these doctors who give this information. Admin doctor verifies this doctor's added information and verifies the doctor's status and manages the site settings. So this application is operationally feasible.

2.3 Risk Analysis

Risk analysis in software development is a method of software testing where software risk is analyzed and measured. It is the process of prioritizing risks for further analysis or action by combining and assessing their probability of occurrence and impact. Plan risk management should take place early in the project, it can impact various aspects for example cost, time, scope, quality, and procurement.

As the healthcare industry begins to use new technologies such as predictive analytics, government health agencies, doctors, and primary health providers must be aware of the risk and agree on standards. Technology is playing an integral role in health care worldwide as predictive analytics has become increasingly useful. Health care has a long track record of evidence-based clinical practice and ethical standards in research.

Most of the projects have their own risks but developers need to understand all risks and find the better and best solution for the risk.

In our technology-based health care application, there is a risk of getting the wrong information about the disease or risk of any type of fraud. But this risk has not affected

the project or application, because they are only verified and the best doctors are allowed to give advice or suggestions to the general public or people.

2.4 Project Scheduling

Project scheduling activity is used to schedule the software development lifecycle. Project scheduling is important to note, however, that the schedule evolves overtime. During the early stages of project planning and a microscopic schedule is developed. The schedule identifies all major software engineering activities. Also identifies the product functions to which they are applied.

We are planning how much time is required to develop actual software. Figure 2.1 shows the project scheduling.

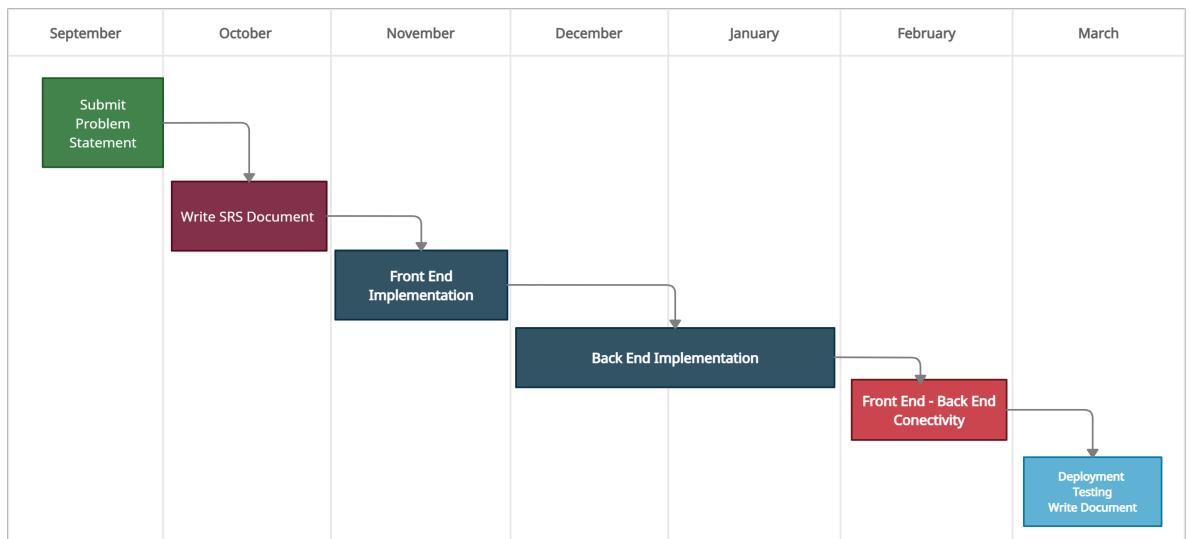


Figure 2.1: Project Scheduling

2.5 Effort Allocation

Identification of project, requirements gathering and study of existing system accounts is 10 percent of effort. 15 percent of effort is normally applied to data modeling and coding. Identification of functional and nonfunctional requirements, testing results by using some test cases take 5 percent of project effort. Designing requires 30 percent of effort.

Table 2.1 shows the Effort Allocation Chart. Figure 2.2 Show the Efforts Allocation in percentage format.

Sr. No	Task	Pruthviraj	Kirti	Harshada	Sumit
1.	Submiting Problem Statement	YES	YES	YES	YES
2.	Writing Problem Definition	YES	YES	YES	YES
3.	Writing Abstract	YES	NO	YES	NO
4.	Background and Scope of Project	NO	YES	YES	NO
5.	Objectives	NO	NO	NO	YES
6.	Selection of Life-Cycle Model	YES	YES	YES	YES
7.	Literature Survey	YES	YES	YES	YES
8.	Feasibility Study	NO	YES	NO	NO
9.	Risk Analysis	NO	NO	YES	NO
10.	Project Scheduling	YES	YES	YES	YES
11.	Analysis	YES	YES	YES	YES
12.	Design	YES	NO	NO	YES
13.	Implementation	YES	YES	YES	YES
13.	System Testing	NO	YES	YES	NO
13.	Results and Analysis	YES	YES	YES	YES
13.	Conclusion and Future Work	YES	YES	YES	YES
14.	Project Report Writing in LaTeX	YES	NO	NO	NO

Table 2.1: Effort Allocation Table

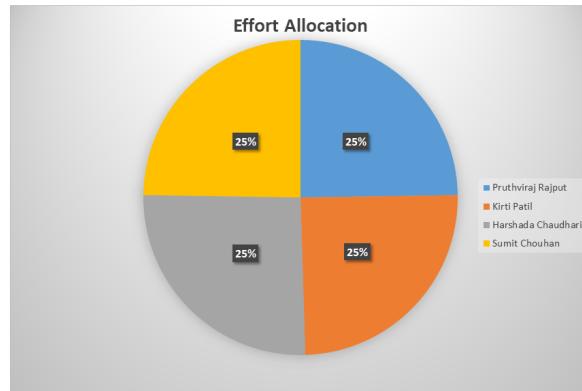


Figure 2.2: Effort Allocation

2.6 Cost Estimation

For any new software project, It is necessary to know how much it will cost to develop and how much development time it will take. These estimates cost estimation in software engineering is typically concerned with the financial spend on the effort to develop and test the software. These evaluations are needed before development is started and conveyed to the team. The software industry has inconsistently defined and explained metrics or atomic units of measure, data from real and actual projects are largely and highly suspect in terms of consistency and comparability.

COCOMO MODEL:

Boehm proposed COCOMO in 1981. COCOMO is one of the most generally used software estimation models in the world. COCOMO predicts the efforts and schedule of a software product based on the size of the software.

The necessary steps in this model are:

- Get an initial estimate of the development effort from the evaluation of thousands of delivered lines of source code (**KDLOC**).
- Determine a set of 15 multiplying factors from various attributes of the project.
- Calculate the effort estimate by multiplying the initial estimate with all the multiplying factors i.e., multiply the values in step1 and step2.

Three modes of software development are considered in this model: organic, semi-detached, and embedded. In the organic mode a small team of experienced developers In the organic model, a small team of experienced developers develops software in a very familiar environment. The size of software development in this mode ranges from small (few KLOC to medium (few tens of KLOC), while in the other two modes the size ranges from small to very large (a few hundred KLOC). According to Boehm, software cost estimation should be done through three stages:

- Basic Model
- Intermediate Model
- Detailed Model

Basic COCOMO Model: The basic COCOMO model provides an accurate size of the project parameters. The following expressions give the basic COCOMO estimation model:

$$Effort = a1 * (KLOC) \quad a2PMTdev = b1 * (efforts) \quad b2Month \quad (2.1)$$

Where,

-KLOC is the estimated size of the software product indicate in , a1,a2,b1,b2 are constants for each group of software products,

-Tdev is the estimated time to develop the software, expressed in months,

-Effort is the total effort required to develop the software product, expressed in person-months (PMs)

TASK	MONTHS	MEMBER	SALARY PER MONTH	AMOUNT
Research / Planning	2	4	20,000/-	20,000 X 4 X 2 TOTAL: 160,000/-
Front End Development Team	1	4	23,000/-	23,000 X 4 X 1 TOTAL: 92,000/-
Back End Development Team	3	2	25,000/-	25,000 X 3 X 2 TOTAL: 150,000/-
Testing/Bug fixes	15 Days	4	18,000/-	18,000 X ½ X 4 TOTAL: 36,000/-
Launch	15 Days	4	18,000/-	18,000 X ½ X 4 TOTAL: 36,000/-
Hosting	1 Years Plane		10,000/-	TOTAL: 10,000/-
Domain	1 Years Plane		8,000/-	TOTAL: 8,000/-
				TOTAL: 492,000/-

Figure 2.3: Cost Estimation

2.7 Summary

In this chapter, Planning of project management is discussed. In the next chapter Analysis of the project is explained.

Chapter 3

Analysis

System Analysis is the process of gathering and interpreting facts, diagnosing problems and using the facts to improve the system. System analysis chapter will show overall system analysis of the concept, description of the system, meaning of the system. System analysis is the study of sets of interacting entities, including computer system analysis. The development of computer-based information systems includes the system analysis phase which produces or enhances the data model which itself is to create or enhance a database. There are a number of different approaches to system analysis. The analysis is the process which is used to Analyze, Portal for Early Prediction Of Lifestyle Diseases. Analysis activity provides a graphical view of the entire System.

The organization of this Chapter is as follows. Section 3.1 represents Requirement Collection and Identification. Software Requirements Specification (SRS) of the project is described in Section 3.2. Section 3.3 represents Summary of this chapter.

3.1 Requirement Collection and Identification

We have identified some requirements during the requirements collection phase;

- User means people check the symptoms , View all diseases information, search disease information using text enter otherwise speak and they also post questions and give the feedback.
- Doctors can register themself also login. They can add disease details, update details, delete details. They solve people's questions.
- Admin can be verifying the register doctor details. Also they can verify diseases with added details. Also see all the user details and doctors details. Manage site settings.

3.2 Software Requirement Specification

3.2.1 Product Features

- The website is easy to navigate.
- The website content is fully authoritative.
- The website has a fast load time.
- The website has Browser consistency which is helpful in user interference.
- The website has a Contrasting color scheme which is very attractive from a presentation point of view.

3.2.2 Operating Environment

This Platform will operate with the following software components and applications:- It is a Web-based application run on a web application server Node js and access data on an enterprise information system, such as a MongoDB database and node js server. The components of web-based applications are spread across multiple tiers, or layers. Node js provides capabilities to create its own web server which will handle HTTP requests asynchronously. Node js provides a global variable process. An object that contains all environment variables available to the user running the application.

3.2.3 Assumptions

This platform will link doctors to people. The entrepreneur will get all the resources they need. The team members they are dealing with have all the required skills. The scope of the project will not change throughout the life cycle. All equipment will be in working conditions through the project cycle. Project cost will stay the same as initially budgeted cost. Project will follow an Incremental methodology throughout execution. The website is well developed for all devices.

3.2.4 Functional Requirements

A functional requirement defines a function of a software system on its component. A function is described as a set of inputs, the behaviour and output. Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that defines what a system is supposed to accomplish.

This platform doctors and people join and the people check their symptoms, post questions and perform this activity. Doctors are added to the disease information, answers to asked questions also admin verify doctors details these are our project function requirements.

3.2.5 Non-Functional requirements

Nonfunctional requirements deny the needs in terms of performance, logical database requirements, design constraints, standard compliance, reliability, availability, security, maintainability and portability.

A non-functional requirement is a requirement that specific criteria that can be used to judge the operation of a system, rather than specific behaviors. It should be contrasted with functional requirements that define specific behavior or functions. The plan for implementing functional requirements is detailed in the system design. The plan for implementing non functional requirements is detailed in the system architecture.

3.2.6 External Interfaces

■ *Hardware Requirements*

- **System :** Any.
- **RAM :** 4 Gb or above.
- **Hard Disk :** 1Tb or above.
- **Input Device :** Keyboard or Mouse.
- **Output Device :** Monitor, PC or Laptop.

■ *Software Requirements*

- **Operating System :** Any.
- **IDE :** Visual Studio Code.
- **Front End :**
 - HTML (Hypertext Markup Language).
 - CSS (Cascading Style Sheets).
 - Java Script.
- **Back End :** Node Js.

- **Library :**
 - React Js.
 - Chart Js.
- **FrameWork :** Express.
- **Ui FrameWork :** React-Bootstrap.
- **Database :** MongoDB.
- **Project Managed :** GitHub Git.

■ *Communication*

State the requirements for any communication functions the product will use, including email, Web browser, network communications protocols, and electronic forms. Define any pertinent message formatting. Specify communication security or encryption issues, data transfer rates, and synchronization mechanisms.

3.3 Modules

3.3.1 User Modules

- Registration
- Login
- Contact
- Profile
- Diseases Display
- Disease Details
- Comments Display
- Ask Question
- Gives Feedback (Comments)
- Feedback Display
- Symptoms Checker

- Question Ask Details Display
- Gives Feedback Details Display
- Contact Details Display
- Chat common module
- Home Page

3.3.2 Doctor Module

- Registration
- Login
- Add / Edit Disease Information
- Reply Question
- Update Comments Verify
- Display Added Diseases Record
- See Profile Edit Profile

3.3.3 Admin Module

- Login
- Verify Doctor
- View All Doctor and Users Details
- Add Disease / Edit Disease / update Disease
- Verify Disease
- Verify Feedback
- Reply to contact
- Reply Question
- Remove Doctor or User Account

3.4 Summary

In this chapter, Analysis of the project is described. In the Next chapter, detailed Design of Project is explained.

Chapter 4

Design

System design provides the understanding and procedural details necessary for implementing the system. Design is an activity concerned with making major decisions, often of a structural nature. Design builds coherent, well planned representations of programs that concentrate on the interrelationships of parts at the higher level and the logical operations involved at the lower levels. Software design is the rest of the three technical activities: designs, coding and tests which are required to build and verify the software.

In Section 4.1 System Architecture is presented. E-R Diagram is presented in Section 4.2. In Section 4.3 Data Flow Diagram is presented in Section 4.4. UML Diagrams present in Section 4.4 Finally summary is presented in the last Section 4.5.

4.1 System Architecture

The system architecture is the conceptual model that denotes the structure, behavior and more views of a system. An architecture description is a formal description and representation of a system. It provides a broad understanding of the portal.

The figure 4.1 shows the system architecture of our project.

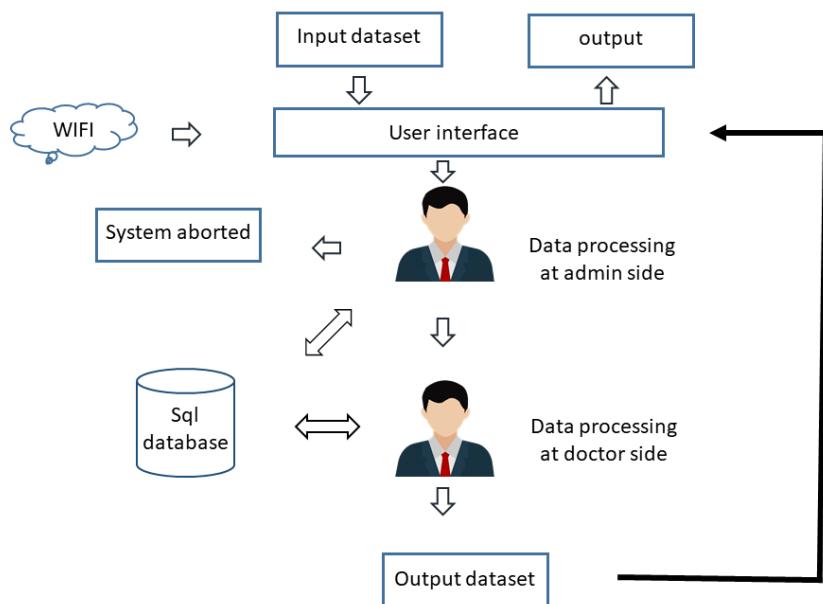


Figure 4.1: System Architecture

4.2 E-R Diagram

An entity–relationship model (or ER model) describes interrelated things of interest in a specific domain of knowledge.[2] A basic ER model is composed of entity types (which classify the things of interest) and specifies relationships that can exist between entities (instances of those entity types). In software engineering, an ER model is commonly formed to represent things a business needs to remember in order to perform business processes. Consequently, the ER model becomes an abstract data model that defines a data or information structure which can be implemented in a database, typically a relational database.

In This project use database structure mentioned in figure 4.2. The database structure defined Like Entity people this entity attributes like name, id is primary key, email, address, password like this all database structure defined.

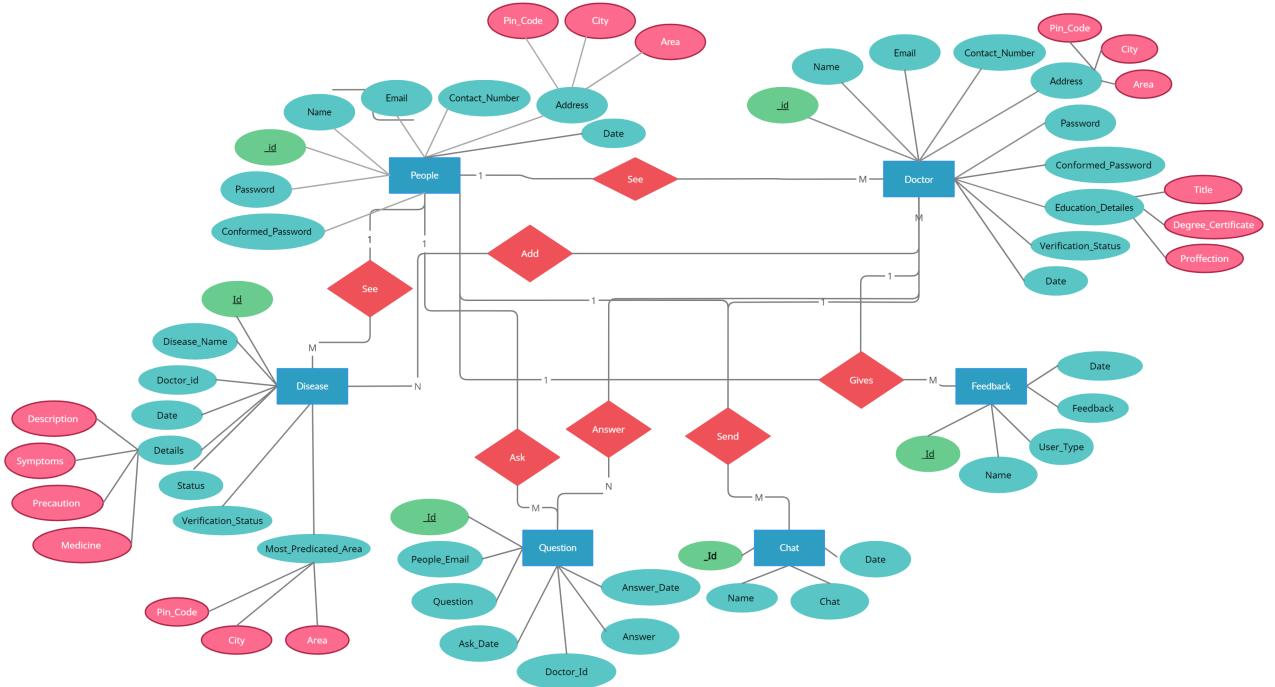


Figure 4.2: ER Diagram

4.3 Data Flow Diagram

DFD is concerned with designing a sequence of functions that converts the system input into the required output.

4.3.1 level 0 DFD

Level 0 contains one input and one output. The system provides information to the user means the system is input and the user is output.

Figure 4.3. shows the level 0 DFD.

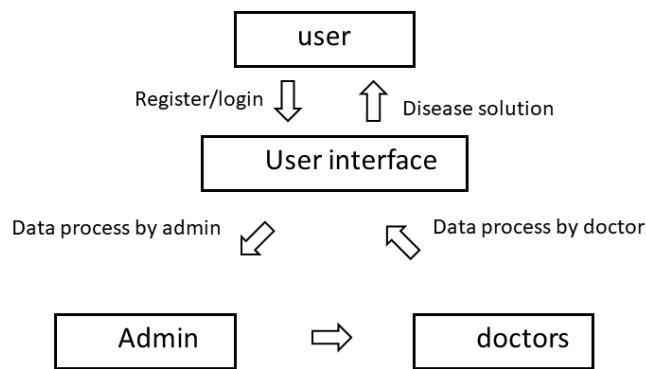


Figure 4.3: Level 0 DFD

4.3.2 level 1 DFD

Scheme management helps to generate the scheme and criteria management helps to provide the information according to the user's criteria.

Figure 4.4. shows the level 0 DFD.

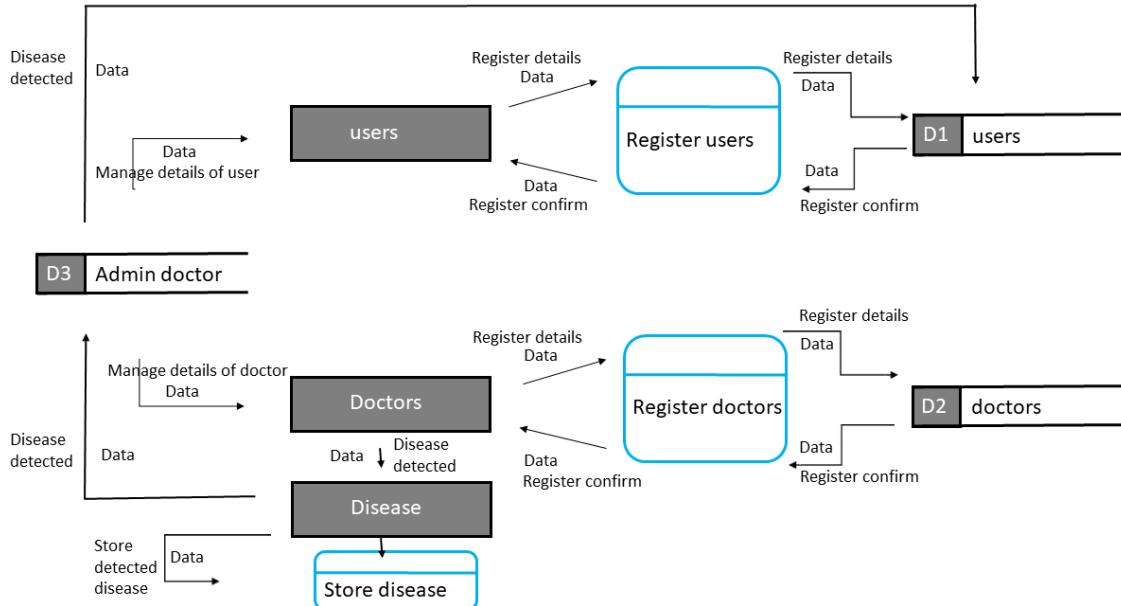


Figure 4.4: Level 1 DFD

4.4 UML Diagrams

The Unified Modeling Language is a language that defines the industry's best engineering practices for the modeling systems. The goal of UML is to be a ready-to-use expressive visual modeling language that is simple and extensible. Use case diagram shows a set of use cases, actors and their relationships. Use case diagrams address the static use case view of a system. These diagrams are especially important in organizing and modeling the behaviour of the system.

Subsection 4.4.1 shows the use case diagram. Sequence diagrams show the subsection 4.4.2. Subsection 4.4.3 shows the collaboration diagram. Activity diagrams show the subsection 4.4.4. Subsection 4.4.5 shows the class diagram. Components diagram shows the subsection 4.4.6. Subsection 4.4.7 shows the state chart diagram. Finally subsection 4.4.8 shows the deployment diagram.

4.4.1 Use Case Diagram

The Use Case diagram of the project disease prediction system consists of all the various aspects a normal use case diagram requires .The use case diagram shows the various actors like users (Peoples), Doctors, admin doctor and the relation and dependency between them. It also shows the functionality of the entire system.

Figure 4.5. shows the Use Case Diagram.

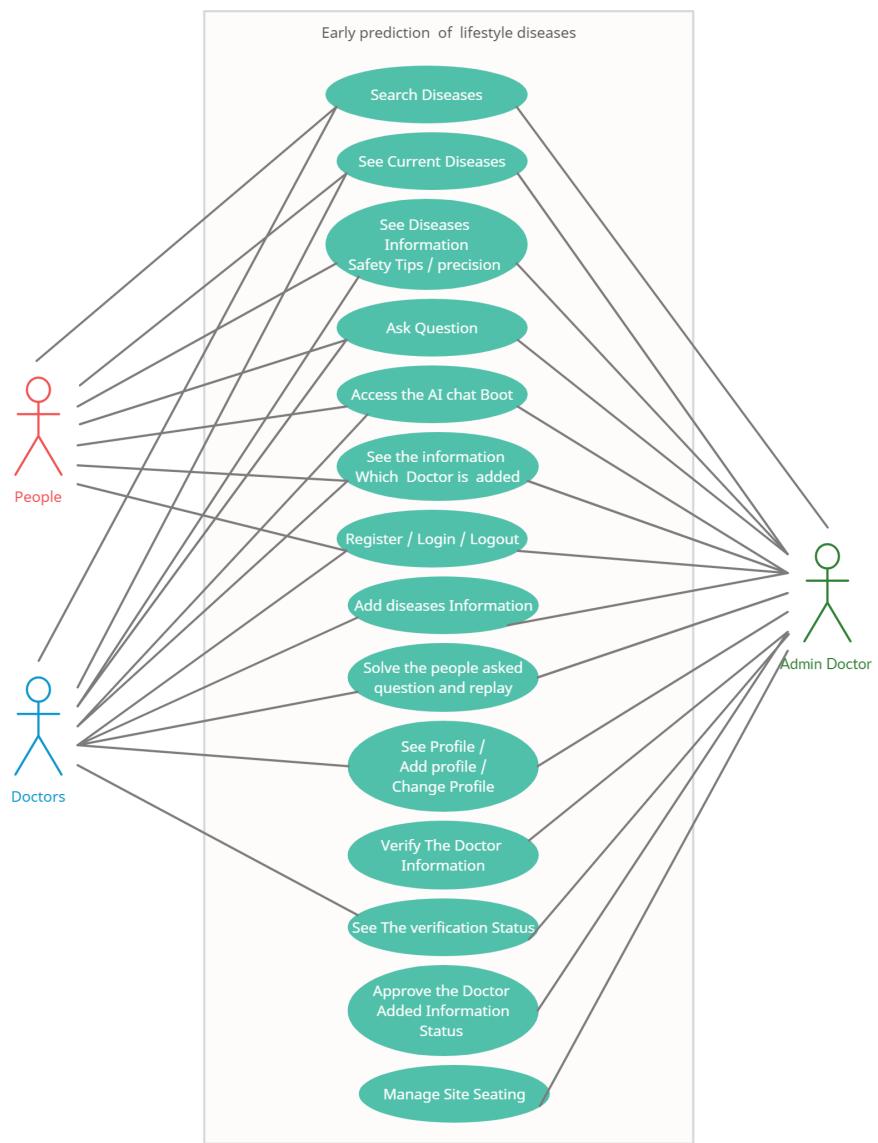


Figure 4.5: Use Case Diagram

4.4.2 Sequence Diagram

The Sequence diagram of the project disease prediction system consists of all various aspects a normal sequence diagram requires.

The Figure 4.6. shows the sequence of interaction between the various objects like login, register and logout functionalities. The Figure 4.7. shows the sequence of interaction between the user (people) Interaction to the portal. The Figure 4.8. shows the sequence of interaction between the Doctors Interaction to the portal. The Figure 4.9. shows the sequence of interaction between the Admin Doctors Interaction to the portal.

■ Register Login Logout

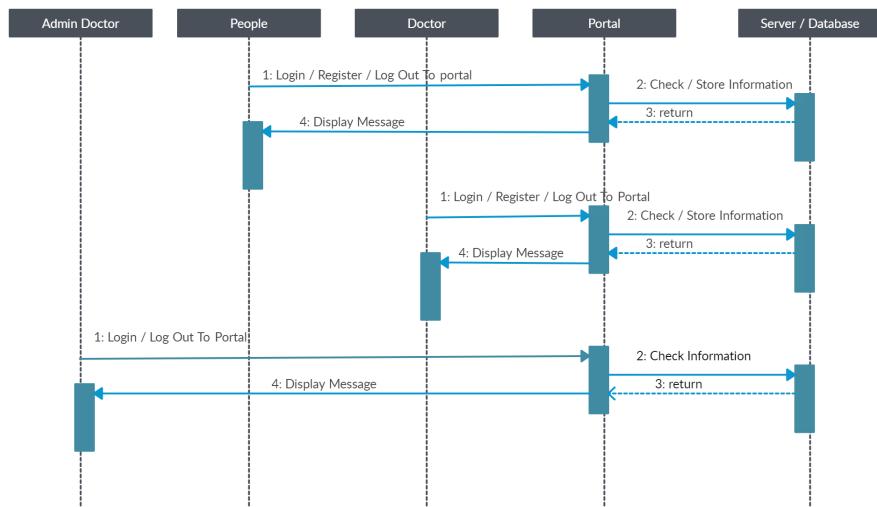


Figure 4.6: Sequence Diagram Login Logout

User (People) Sequence Diagram

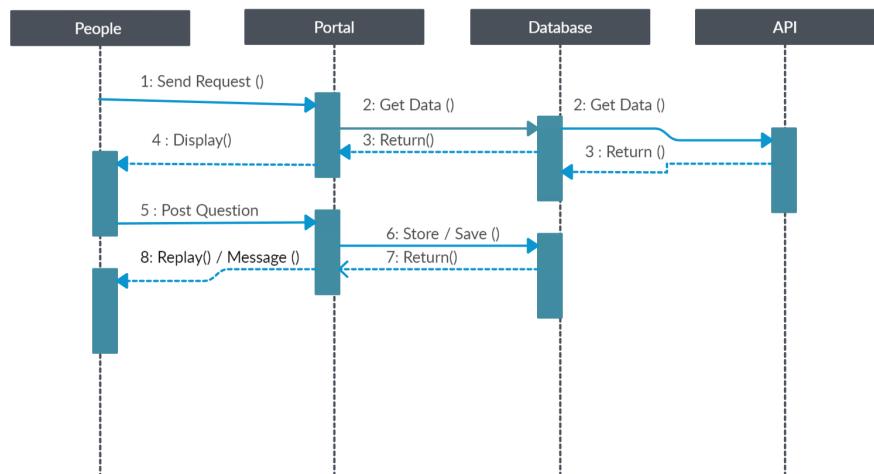


Figure 4.7: User (People) Sequence Diagram

■ Doctors Sequence Diagram

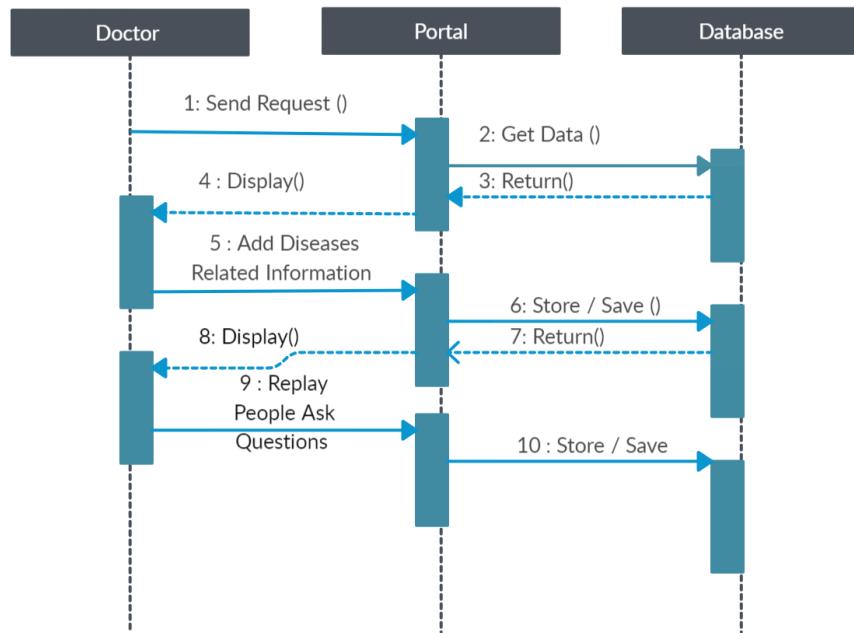


Figure 4.8: Doctors Sequence Diagram

■ Admin Sequence Diagram

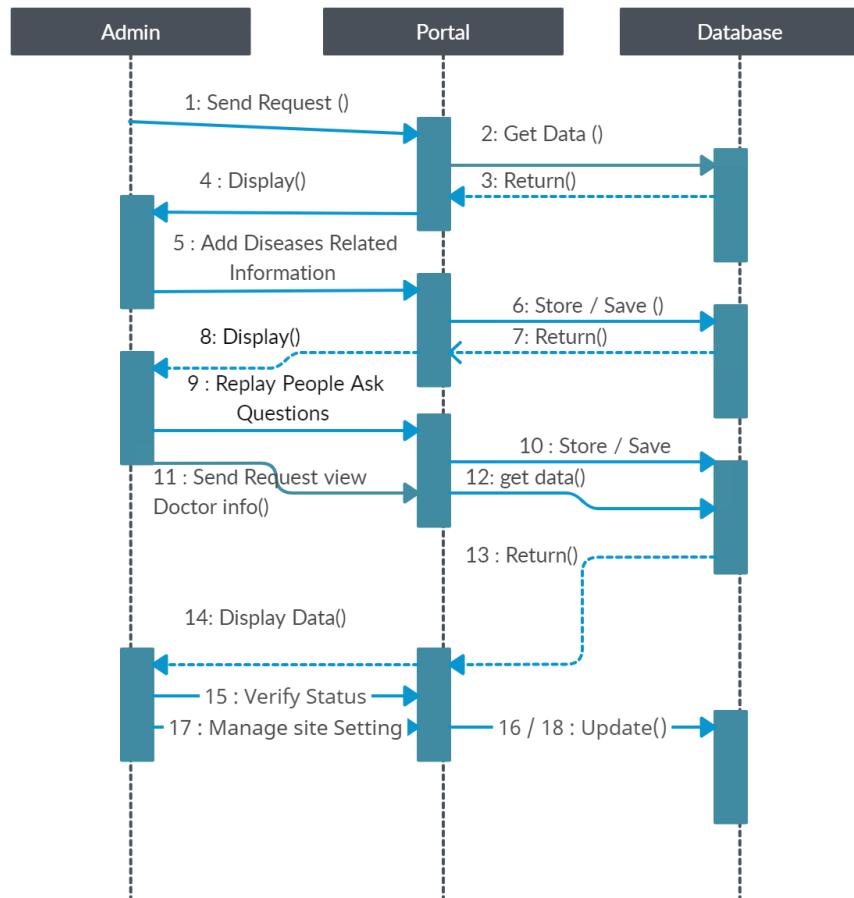


Figure 4.9: Admin Sequence Diagram

4.4.3 Collaboration Diagram

A collaboration diagram, also known as a communication diagram, is an illustration of the relationships and interaction among software objects in the UML. These diagrams can be used to portray the dynamic behaviour of a particular use case and define the role of each object.

Here this figure 4.10 shows how all the modules are connected to show the correct result starting from people, where he opens the portal then using the portal he does registration and the registration data is saved into database and using those data people / admin logs to the portal and then he provides all the necessary information in order to get the accurate result, then portal evaluates the people / admin entered information and finally gives the correct result.

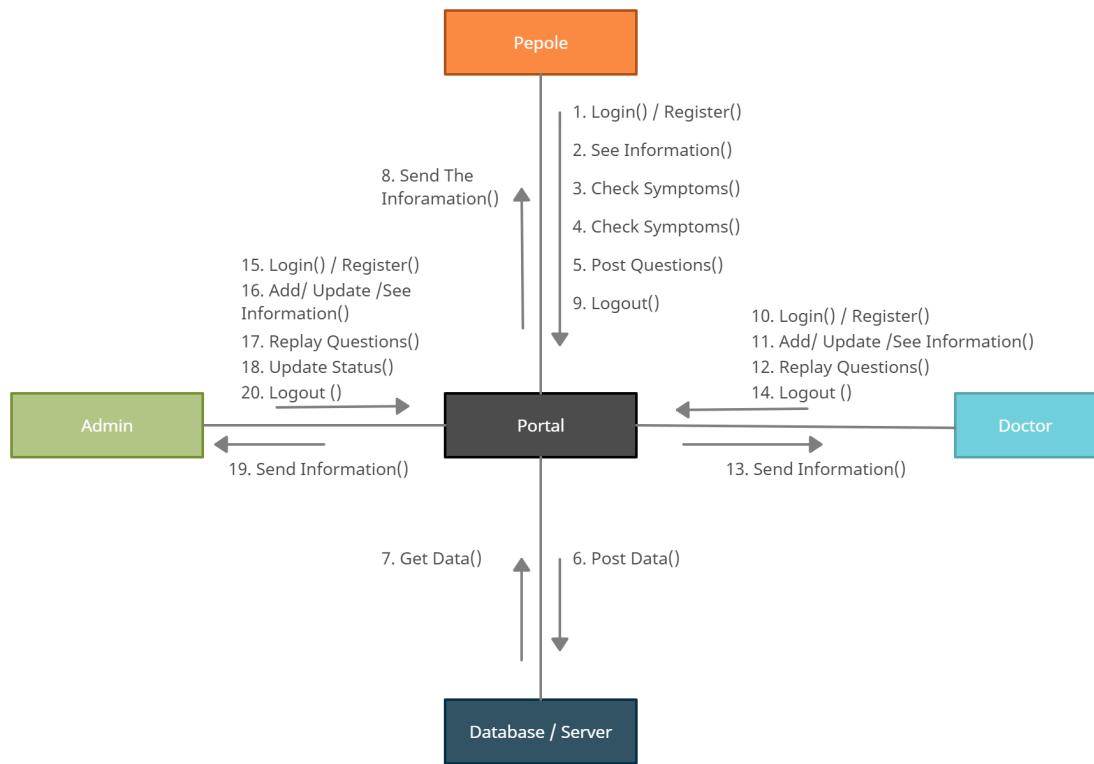


Figure 4.10: Collaboration Diagram

4.4.4 Activity Diagram

Activity diagram is another important diagram UML describes the dynamic aspect of the system. Activity diagram is basically 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

Here in the figure 4.11 describe user view activities starts from checking symptoms and also search disease like this. And figure 4.12 describes admin view activities starting from login like this.

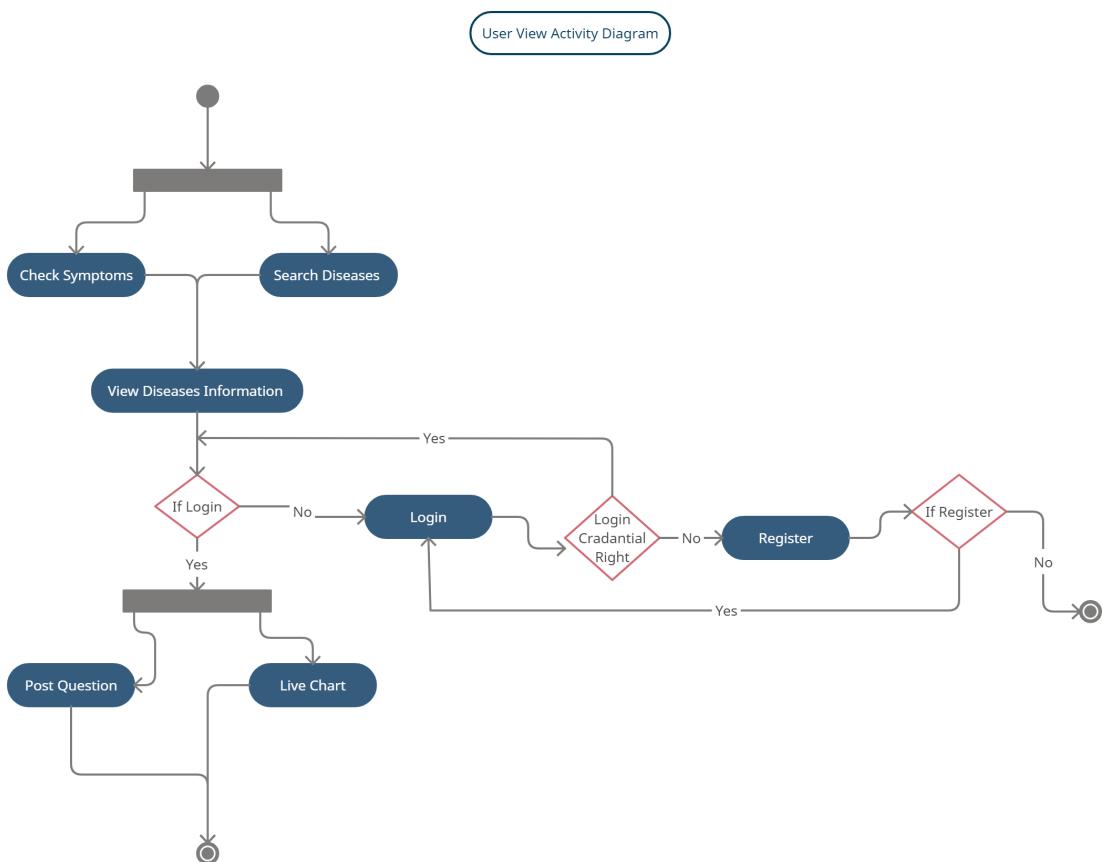


Figure 4.11: Activity Diagram (User View)

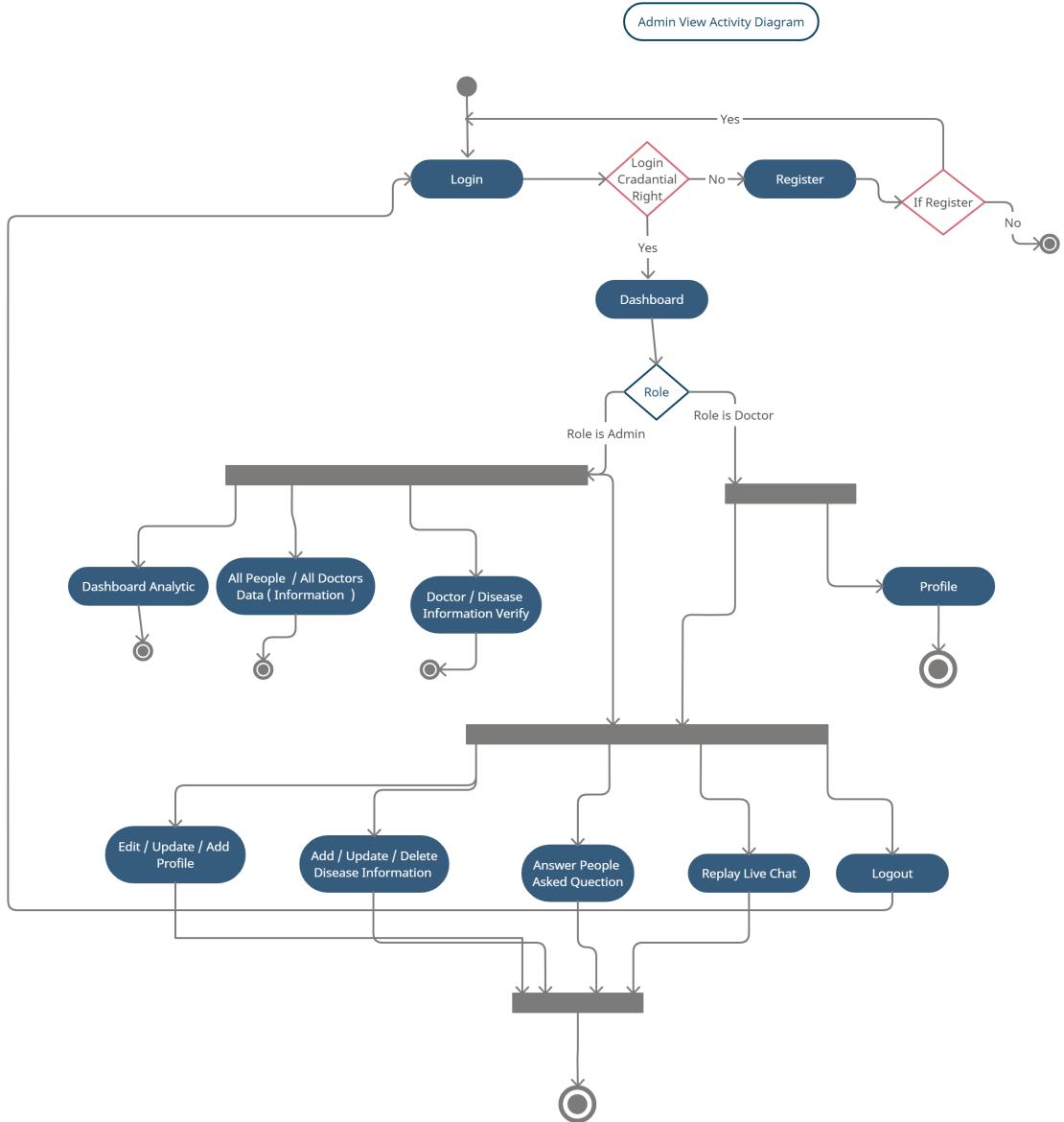


Figure 4.12: Activity Diagram (Admin View)

4.4.5 Class Diagram

Disease prediction system consists of a class diagram that all the other applications that consist of the basic class diagram, here the class diagram is the basic entity that is required in order to carry on with the project. Class diagram consist information about all the classes that is used and all the related datasets, and all the other necessary attributes and their relationships with other entities, all these information is necessary in order to use the concept of the prediction, where the people will enter all necessary information such as name,email,contact number, and many more attribute that is required in order to login into the system and using the database we will store the information of the people, admin,

doctors who are register into the system and retrieves those information later while logging into the system. The figure 4.13. shows a class diagram of the overall system.

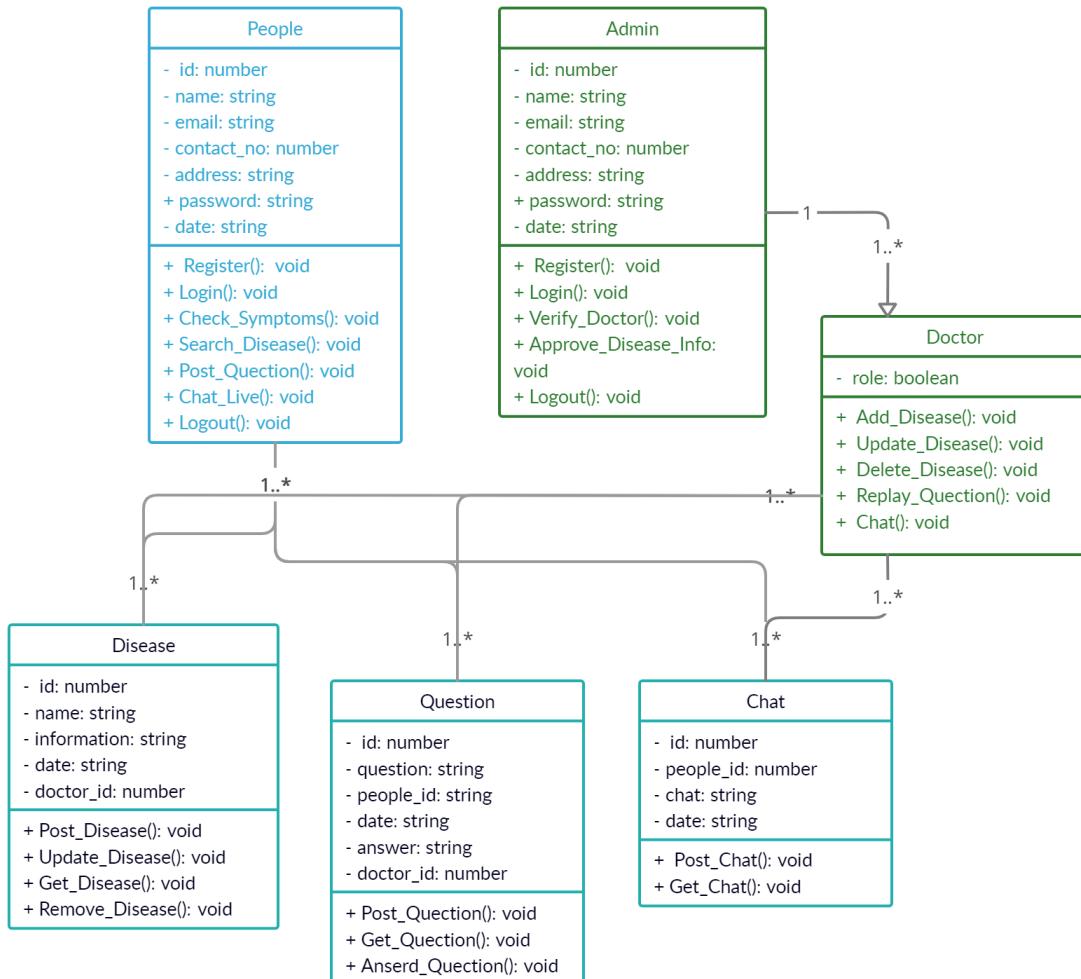


Figure 4.13: Class Diagram

4.4.6 Component Diagram

A components diagram, also known as a UML component diagram, describes the organization and writing of the physical components in a system. Component diagrams are often drawn to help model implementation details and double-check every aspect of the system's required function is covered by planned development. Here the component diagram consists of all major components that are used to build a system. Figure 4.14. describes the components diagram.

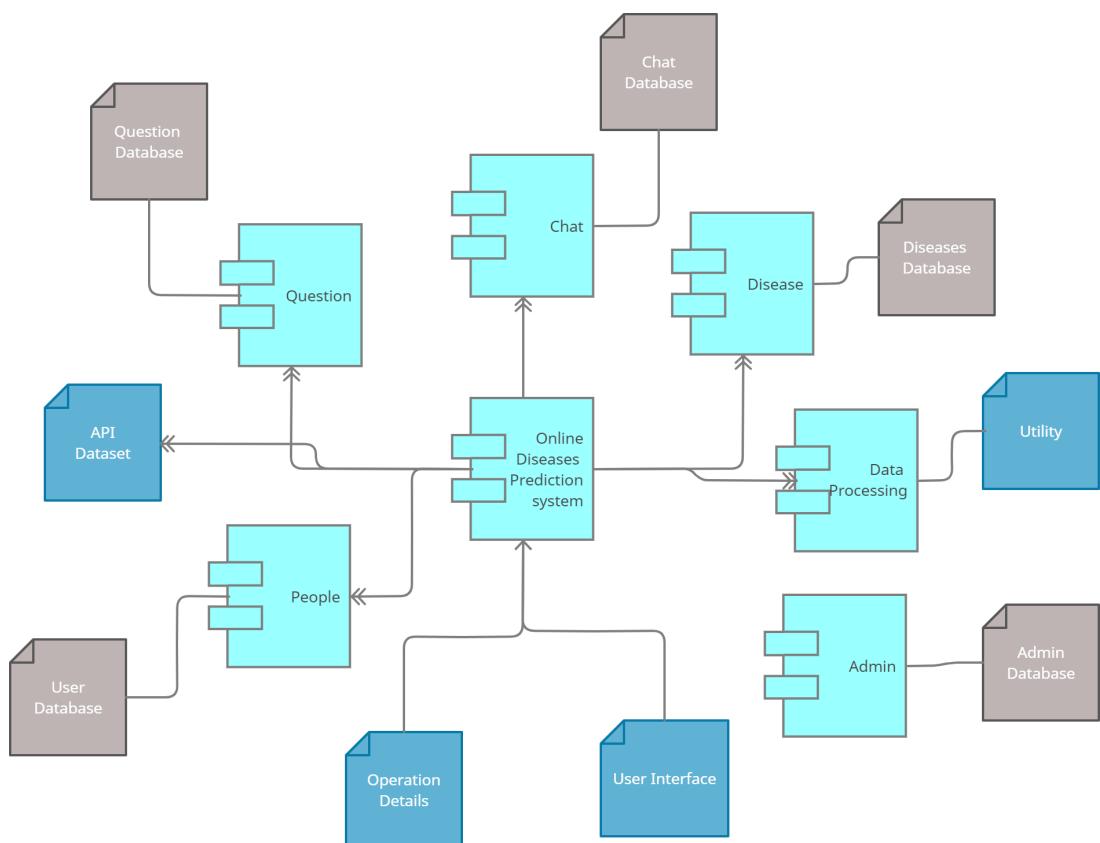


Figure 4.14: Component Diagram

4.4.7 Deployment Diagram

A deployment diagram shows the configuration of run time processing nodes and the components that live on them. Deployment diagram is a kind of structure diagram used in modeling the physical aspect of an object-oriented system. Here the deployment diagram shows the final stage of the project and it also shows how the module looks after doing all the process and deploying in the machine. Starting from the system, how it processes the people entering information and then comparing that information with the help of datasets, then training and testing that data using the algorithms such as decision tree, naive Bayes, random forest. Then finally processing all those data and the information the system gives the desired result in the interface.

Figure 4.15. Shows the deployment behaviour.

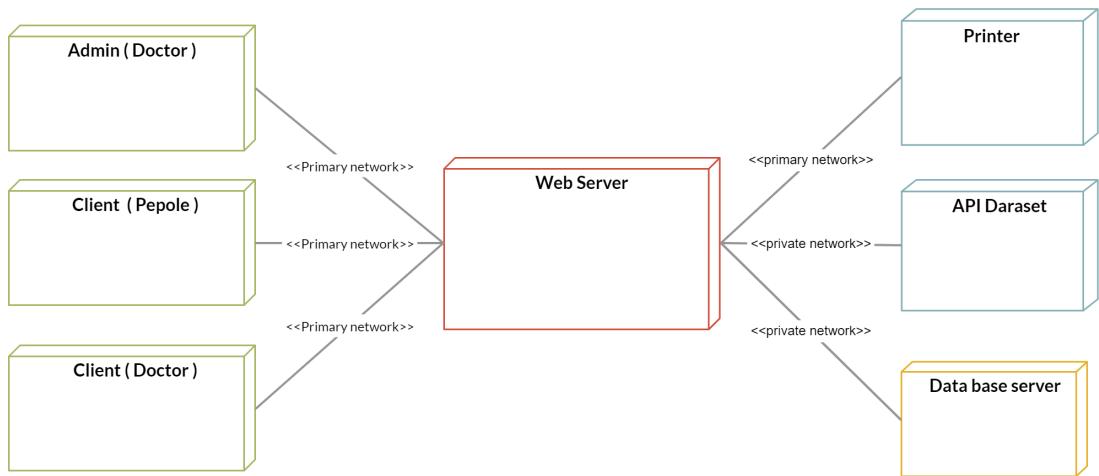


Figure 4.15: Deployment Diagram

4.5 Summary

In this chapter, System Design is presented. In the next chapter. Conclusion and Future Work Discussed.

Chapter 5

Implementation

The Implementation chapter describes how the system is implemented, and various algorithms used to build the system. Implementation is the realization of an application, or execution of a plan, idea, model, design, specification, standard, algorithm, or policy. Implementation is the carrying out, execution, or practice of a plan, a method, or any design for doing something. As such, implementation is the action that must follow any preliminary thinking in order for something to actually happen. In an information technology context, implementation encompasses all the processes involved in getting new software or hardware operating properly in its environment, including installation, configuration, running, testing, and making necessary changes.

Chapter is of four sections. First section 5.1 describes implementation details of the project, implementation environment is described in section 5.2, system development flow is described in section 5.3, section 5.4 gives summary.

5.1 Implementation Details

System implementation is the important stage of a project when the theoretical design is turned into a practical system. The main stages in the implementation are planning, training, and system testing.

5.2 Implementation Environment

An implementation detail is the decision that is left to be made by the developers and is not specified at entry level. Such as a requirement document or depending on the context of the architectural document. In order to implement an automated system, the relational database must be designed first. Conceptual design can be divided into two parts: The data model and the process model. The data model focuses on what data should be stored in the database while the process model deals with how the data is processed. To put in the

context of the relational database, the data model is used to design the relational tables. The process model is used to design the queries that will access and perform operations on those tables.

While there are numerous technologies for building web applications that serve dynamic content, the one that has really caught the attention of the development community is MERN. And not without ample reason either. MERN is open-source . We develop our project front end using React Js (HTML, CSS, JavaScript), and the Back end Part Node js (Mongoose, Express) database uses MongoDB (NoSQL Database). And finally building this we host our project free hosting platform Netlify and back end hosting heroku.

5.2.1 React Js

React. js is an open-source JavaScript library that is used for building user interfaces specifically for single-page applications. It's used for handling the view layer for web and mobile apps. React also allows us to create reusable UI components.

5.2.2 HTML

HTML means Hypertext Markup Language. The HTML provides the front end creation of ui in the help of this language create a whole website ui.

5.2.3 CSS

CSS means Cascading Style Sheet This language helps for creation of front end ui with addition of colors, borders,shadows and also add responsiveness using this language.

5.2.4 JavaScript

JavaScript is used to develop interactive web applications. JavaScript can power features like interactive images, carousels, and forms.

5.2.5 Node Js

Node. js is primarily used for non-blocking, event-driven servers, due to its single-threaded nature. It's used for traditional web sites and back-end API services, but was designed with real-time, push-based architectures in mind.

5.2.6 Mongoose

Mongoose is an Object Data Modeling (ODM) library for MongoDB and Node. js. It manages relationships between data, provides schema validation, and is used to translate

between objects in code and the representation of those objects in MongoDB. MongoDB is a schema-less NoSQL document database

5.2.7 Express Js

Express. js is a free and open-source web application framework for Node. js. It is used for designing and building web applications quickly and easily

5.2.8 MongoDB

MongoDB is an open source NoSQL database management program. NoSQL is used as an alternative to traditional relational databases. NoSQL databases are quite useful for working with large sets of distributed data. MongoDB is a tool that can manage document-oriented information, store or retrieve information.

5.2.9 Netlify

Netlify is a web developer platform that multiplies productivity. By unifying the elements of the modern decoupled web, from local development to advanced edge logic, Netlify enables a 10x faster path to much more performant, secure, and scalable websites and apps.

5.2.10 Heroku

Heroku is a container-based cloud Platform as a Service (PaaS). Developers use Heroku to deploy, manage, and scale modern apps. Our platform is elegant, flexible, and easy to use, offering developers the simplest path to getting their apps to market.

5.2.11 PWA (Progressive Web App)

A progressive web app (PWA) is a website that looks and behaves as if it is a mobile app. PWAs are built to take advantage of native mobile device features, without requiring the end user to visit an app store, make a purchase and download software locally.

5.3 Flow of system development

The flow of system development describes How you performed and developed the proposed methods and procedure.

The project is done as follows:-

- **Planning Phase:** Requirements are gathered during the planning phase. Requirements like BRS that is Business Requirement Specifications and SRS that is System Requirement specifications.
- **Risk Analysis:** In the risk analysis phase, a process is undertaken to identify risk and alternate solutions. A prototype is produced at the end of the risk analysis phase. If any risk is found during the risk analysis then alternate solutions are suggested and implemented.
- **Engineering Phase:** In the phase software is developed, along with testing at the end of the phase. Hence the development and testing is done in the phase.
- **Evaluation phase:** The phase allows the customer to evaluate the output of the proposed system to date before the system continues to the next spiral.

5.4 Summary

In this chapter, the implementation details, implementation environment are described. In the next chapter, System Testing, is discussed.

Chapter 6

System Testing

The System Testing chapter describes whether the result is correct or not by using some test cases. Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design, and code generation. The increasing visibility of software as a system element and the attendant "costs" associated with a software failure are motivating forces for well planned, thorough testing. Once source code has been generated, software must be tested to uncover and correct as many errors as possible before delivery to the customer.

Chapter is of three sections. First section 6.1 describes how to implement testing, describe black box testing in section 6.2, section 6.3 describe white box testing. Section 6.4 describes test cases and test results. section 6.5 gives a summary.

6.1 How To Implement Testing

A goal of testing is to design a series of test cases that have a high likelihood of finding errors. The technique provide systematic guidance for designing tests that :-

- Exercise the internal logic of software components.
- Exercise the input and output domains of the program to uncover errors in program function, behavior and performance.

Software is tested from two different perspectives: In both cases, the intent is to end the maximum number of errors with the minimum amount of effort and time. A set of test cases designed to exercise both internal logic and external requirements is designed and documented, expected results are denied, and actual results are recorded. While testing, change point of view. Try hard to "break" the software. Design test cases in a disciplined fashion and review the test cases you do create for toughness.

6.2 Black Box Testing

Black box testing involves testing a system with no prior knowledge of its internal workings. A tester provides an input, and observes the output generated by the system under test. This makes it possible to identify how the system responds to expected and unexpected user actions, its response time, usability issues and reliability issues.

Black box testing is a powerful testing technique because it exercises a system end-to-end. Just like end-users “don’t care” how a system is coded or architected, and expect to receive an appropriate response to their requests, a tester can simulate user activity and see if the system delivers on its promises. Along the way, a black box test evaluates all relevant subsystems, including UI/UX, web server or application server, database, dependencies, and integrated systems.

6.3 White Box Testing

White box testing techniques analyze the internal structures, the used data structures, internal design, code structure and the working of the software rather than just the functionality as in black box testing. It is also called glass box testing or clear box testing or structural testing.

6.4 Test Cases And Test Results

A test case, in software engineering, is a set of conditions under which a tester will determine whether an application, software system or one of its features is working as it was originally intended. Test results are the results obtained after testing the software. Results can be of the various forms such as numeric value or any statement that denotes the result of a test case. Test case is the set of inputs along with the output and some additional information like:

- Test Case ID number Number identifying the test case.
- Test Case Name Name of the test case.
- Test Case Description Details/ purpose of the test case.
- Steps: Sequence of steps that the tester must follow to perform a test case.
- Test Data Input given to the function.
- Expected Results What the tester should see when the test case is executed.

- Actual Result What the tester actually witnesses when the test case is executed.
- Test Result (P/F) Indicates whether the test case passed, failed, etc.

Test Case 1	
Purpose:	Start Services
Pre-requisite:	React Js Server and Node Js Server needs to start state
Test data:	Go to Home Page
Steps:	Type the following on web browser http://localhost:3000
Expected result:	Process evaluation should start
Actual result:	PASS

Table 6.1: Text Case 1

Test Case 2	
Purpose:	Navigation Bar
Pre-requisite:	React Js Server and Node Js Server needs to start state
Test data:	Properly redirect one page to another page
Steps:	Click to navigation specific menu
Expected result:	Properly redirect one page to another page
Actual result:	PASS

Table 6.2: Text Case 2

Test Case 3	
Purpose:	Display Diseases
Pre-requisite:	React Js Server and Node Js Server needs to start state
Test data:	Display All diseases.
Steps:	Click to diseases menu
Expected result:	Properly display data all diseases
Actual result:	PASS

Table 6.3: Text Case 3

Test Case 4	
Purpose:	Specific disease details
Pre-requisite:	React Js Server and Node Js Server needs to start state
Test data:	Display specific disease details.
Steps:	<ul style="list-style-type: none"> • Click to diseases menu • Click to any one disease display read more button
Expected result:	Properly display data disease
Actual result:	PASS

Table 6.4: Text Case 4

Test Case 5	
Purpose:	Search Disease
Pre-requisite:	React Js Server and Node Js Server needs to start state
Test data:	Display Specific Disease after search
Steps:	<ul style="list-style-type: none"> • Click to diseases menu • Click to search box as well as click to mic icon • Enter any disease name as well as speak any disease name
Expected result:	Properly display sepcific disease after search
Actual result:	PASS

Table 6.5: Text Case 5

Test Case 6	
Purpose:	SignUp / Login
Pre-requisite:	React Js Server and Node Js Server needs to start state
Test data:	Signup Successfully
Steps:	<ul style="list-style-type: none"> • Click to Signup / Login menu • Enter all fields properly
Expected result:	Signup / Login Successfully
Actual result:	PASS

Table 6.6: Text Case 6

Test Case 7	
Purpose:	Contact Up Form Submitting
Pre-requisite:	React Js Server and Node Js Server needs to start state
Test data:	Contact details submitted properly
Steps:	<ul style="list-style-type: none"> • Click to Contact menu • Enter all fields properly
Expected result:	Contact details submitted Successfully
Actual result:	PASS

Table 6.7: Test Case 7

Test Case 8	
Purpose:	Ask Question
Pre-requisite:	React Js Server and Node Js Server needs to start state
Test data:	Question submitted successfully
Steps:	<ul style="list-style-type: none"> • Click to diseases menu • Click to specific disease read more button • Disease details page is open • Click to Any Query Ask here button. • In popup enter all filled properly.
Expected result:	Question details submitted Successfully
Actual result:	PASS

Table 6.8: Test Case 8

Test Case 9	
Purpose:	Gives Feedback
Pre-requisite:	React Js Server and Node Js Server needs to start state
Test data:	Feedback submitted successfully
Steps:	<ul style="list-style-type: none"> • Click to diseases menu • Click to specific disease read more button • Disease details page is open • Click to Feedback here button. • In popup enter all filled properly.
Expected result:	Feedback submitted Successfully
Actual result:	PASS

Table 6.9: Test Case 9

Test Case 10	
Purpose:	Symptoms Checker working
Pre-requisite:	React Js Server and Node Js Server needs to start state
Test data:	Display all possibility cases
Steps:	<ul style="list-style-type: none"> • Click to Symptom Checker menu • Click to Next button • Answers all question properly • Last Click to Finish button
Expected result:	Display all possibilities properly
Actual result:	PASS

Table 6.10: Test Case 10

Test Case 11	
Purpose:	Logout
Pre-requisite:	React Js Server and Node Js Server needs to start state and login user
Test data:	Redirect home page
Steps:	<ul style="list-style-type: none"> • Click to your name display drop-down menu • Click to logout menu
Expected result:	Redireact home page and display login and signup buttons
Actual result:	PASS

Table 6.11: Test Case 11

6.5 Summary

In this chapter, the system testing details, testing environment are described. In the next chapter, result and analysis are discussed.

Chapter 7

Results and Analysis

Results and Analysis chapter shows the actual result of the project. Data analysis is an ongoing process in a project. It gives succinct data to draw a meaningful conclusion. It can be wise to do some data analysis, even while collection of data is ready. Chapter is of two sections.

First section describes outputs in the database structure ,section 7.2 describes deployment links means overall system structure section 7.3 gives a summary.

7.1 Database Structure

7.1.1 User Module Document Object



The screenshot shows the MongoDB query results interface with the title "QUERY RESULTS 1-6 OF 6". It displays two user documents. The first document has the following fields:

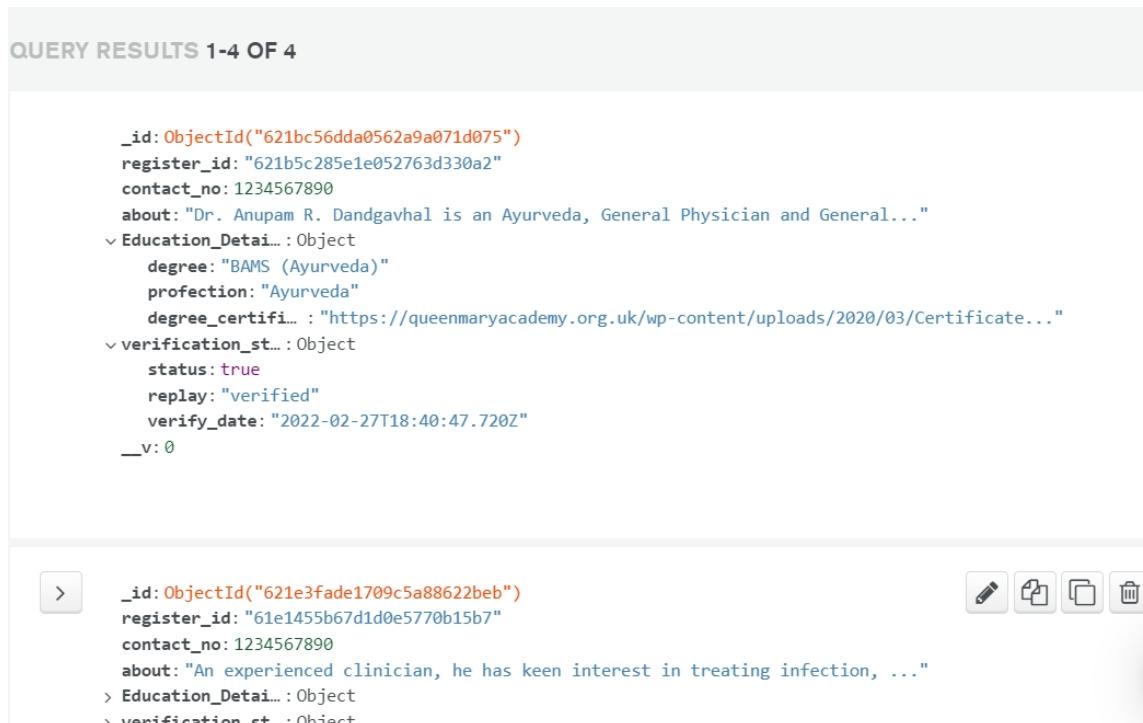
```
_id: ObjectId("61e1455b67d1d0e5770b15b7")
name: "Project Id 09"
email: "projectid09@gmail.com"
password: "$2a$12$n.s8VqW5Fie9NDlURETsZ049uZJcOvnbhZcUBCGmArkUH63EdKtLq"
cpassword: "$2a$12$gG2W2wGY.vR9cSCJtKvKyULfu9JLSpuPTDKXp6um9KBfpLf1PHM9i"
createdAt: 2022-01-14T09:41:47.853+00:00
> address: Object
role: "admin"
__v: 0
```

The second document has the following fields:

```
_id: ObjectId("621b5c285e1e052763d330a2")
name: " Dr. Anupam R. Dandgavhal"
email: "anupam.dandgavhal@gmail.com"
password: "$2a$12$U6IHLuqHE2du44QgFM6iW.24fdL3y252QhXcTrHqIdzDSB/egLE5q"
cpassword: "$2a$12$0Kj63jmfrj1AXRBZnznU3.N.WtXrhH/ZdT4kHnvBmwUpT4muzUdnm"
createdAt: 2022-02-27T11:10:32.508+00:00
> address: Object
role: "doctor"
__v: 0
```

Figure 7.1: User Document Object

7.1.2 Doctor Module Document Object



The screenshot shows the MongoDB Compass interface displaying two documents from a collection named 'Doctor'. The first document, highlighted in blue, contains the following fields:

```
_id: ObjectId("621bc56dda0562a9a071d075")
register_id: "621b5c285e1e052763d330a2"
contact_no: 1234567890
about: "Dr. Anupam R. Dandgavhal is an Ayurveda, General Physician and General..."
< Education_Detail : Object
  degree: "BAMS (Ayurveda)"
  profession: "Ayurveda"
  degree_certificate: "https://queenmaryacademy.org.uk/wp-content/uploads/2020/03/Certificate..."
< verification_status : Object
  status: true
  replay: "verified"
  verify_date: "2022-02-27T18:40:47.720Z"
__v: 0
```

The second document, shown in grey, has the following fields:

```
_id: ObjectId("621e3fade1709c5a88622beb")
register_id: "61e1455b67d1d0e5770b15b7"
contact_no: 1234567890
about: "An experienced clinician, he has keen interest in treating infection, ..."
> Education_Detail : Object
> verification_status : Object
```

On the right side of the interface, there are four icons: a pencil (Edit), a thumbs up (Like), a square (Copy), and a trash can (Delete).

Figure 7.2: Doctor Document Object

7.1.3 Disease Module Document Object

QUERY RESULTS 1-2 OF 2

```

> _id: ObjectId("e21dc2c59584ad558877addf")
doctor_id: "621bc28se1e052763d330a2"
disease_name: "Malaria"
detail: Object
  image: "https://sahyadrihospital.com/wp-content/uploads/2021/04/Be-aware-of-ma..."
  description: "Malaria is a disease caused by a parasite. The parasite is spread to h..."
  symptoms: Array
    0: "Fever"
    1: "Chills"
    2: "General feeling of discomfort"
    3: "Headache"
    4: "Nausea and vomiting"
    5: "Diarrhea"
    6: "Abdominal pain"
    7: "Muscle or joint pain"
    8: "Fatigue"
    9: "Rapid breathing"
    10: "Rapid heart rate"
    11: "Cough"
    12: "Some people who have malaria experience cycles of malaria 'attacks.' A..."
    13: "Malaria signs and symptoms typically begin within a few weeks after be..."
  precautions: Array
    0: "If you live in or are traveling to an area where malaria is common, ta..."
    1: "Cover your skin. Wear pants and long-sleeved shirts. Tuck in your shir..."
    2: "Apply insect repellent to skin. Use an insect repellent registered wit..."
    3: "Apply repellent to clothing. Sprays containing permethrin are safe to ..."
    4: "Sleep under a net. Bed nets, particularly those treated with insectici..."
  medicine: Array
    0: "If you'll be traveling to a location where malaria is common, talk to ..."
    1: "In general, the drugs taken to prevent malaria are the same drugs used..."
  vaccine: Array
    0: "The World Health Organization has recommended a malaria vaccine for us..."
    1: "Researchers are continuing to develop and study malaria vaccines to pr..."
  verification_status: Object
    status: true
    verify_date: "2022-03-01T06:57:46.296Z"
    replay: "Verified"
  most_predicated_area: Object
  visibility: true
  created_date: 2022-03-01T06:52:53.207+00:00

```

Figure 7.3: Disease Document Object

7.1.4 Question Modules Document Object

```

> _id: ObjectId("6221c9a5f33892f6ea3acc13")
user_id: "6221c557f33892f6ea3acbf1"
disease_id: "621dc2c59584ad558877addf"
user_name: "Pruthviraj Rajput"
user_email: "pruthviraj.rajput011@gmail.com"
question: "Please tell me detailed information about this disease?"
ask_date: 2022-03-04T08:11:17.488+00:00
answer: "NA"
__v: 0

```

Figure 7.4: Question Document Object

7.1.5 Feedback Module Document Object

QUERY RESULTS 1-2 OF 2

```
_id: ObjectId("621f0f96e1e145c10ea05574")
user_id: "NA"
disease_id: "621de03dd930a1065fd6fe54"
name: "Rajesh Patil"
email: "rajkpatil915@gmail.com"
feedback: "This information is very helpful for me, this is helping me a lot. tha..."
feedback_date: 2022-03-02T06:32:54.141+00:00
status: true
visibility: true
__v: 0

_id: ObjectId("621f43807ddfcae7022131a5")
user_id: "621f42fc7ddfcae702213199"
disease_id: "621dc2c59584ad558877addf"
name: " Pruthviraj Rajput"
email: "pruthvirajrajput575@gmail.com"
feedback: "Your platform provided information is easy to understand and very best..."
feedback_date: 2022-03-02T10:14:24.910+00:00
status: true
visibility: true
```

Figure 7.5: Feedback Document Object

7.1.6 Contact Module Document Object

The screenshot shows a MongoDB Compass interface with the title "QUERY RESULTS 1-1 OF 1". It displays a single document object with the following fields and values:

```
_id: ObjectId("6221c5b1f33892f6ea3acbfe")
user_id: "6221c557f33892f6ea3acbf1"
name: "Pruthviraj Rajput"
email: "pruthviraj.rajput011@gmail.com"
subject: "Information"
message: "Please Tell Me About Your Platform?"
answer: "NA"
answer_date: "NA"
contact_date: 2022-03-04T07:54:25.655+00:00
__v: 0
```

On the right side of the document, there are four icons: a pencil (edit), a checkmark (check), a square (copy), and a trash can (delete).

Figure 7.6: Contact Document Object

7.1.7 Chat Module Document Object

The screenshot shows a MongoDB Compass interface with the title "QUERY RESULTS 1-1 OF 1". It displays a single document object with the following fields and values:

```
_id: ObjectId("6221c581f33892f6ea3acbfa")
user_id: "6221c557f33892f6ea3acbf1"
name: "Pruthviraj Rajput"
message: "Welcome To E Health Care"
timestamp: "2022-03-04T07:53:34.615Z"
__v: 0
```

Figure 7.7: Chat Document Object

7.2 Overall System Structure

This project is managed using git and github. That's why we used the github organization for implementing this project.

- GitHub Project organization link is given for coding <https://github.com/Full-Stack-Warriors>
- Deployment link Client Side (Preview of project) <https://ehealthcareforeveryone.netlify.app>
- Deployment link Admin Side (Preview of Project) <https://ehealthcarefordoctor.netlify.app>

7.2.1 Snap Shorts

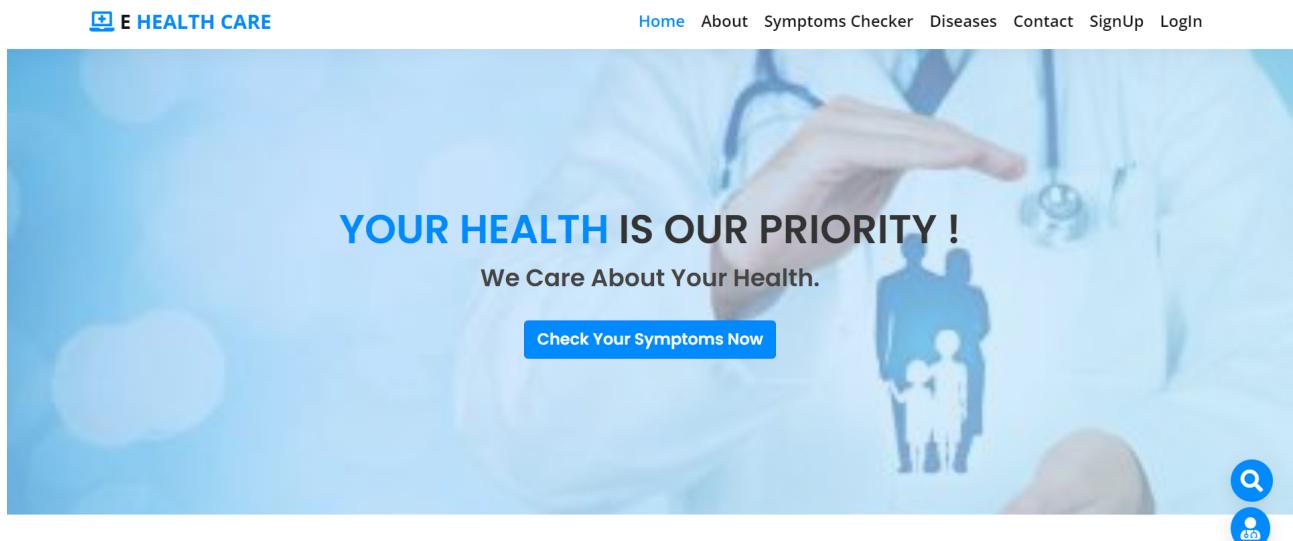


Figure 7.8: Home Page

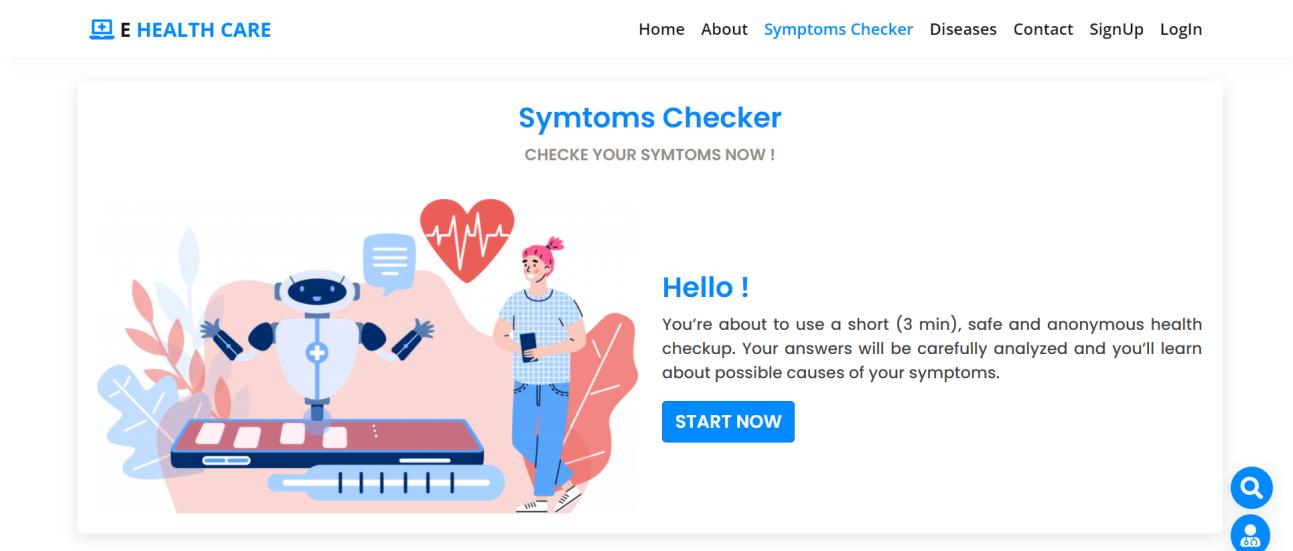


Figure 7.9: Symptom Checker

Symtoms Checker

CHECKE YOUR DISEASE POSSIBLE CASES NOW !

Before using this symptom checker, please read carefully and accept our Terms and Services:

Welcome

This checkup is not a diagnosis.

Patient

This checkup is for informational purposes and is not a qualified medical opinion.

Symptom

Information that you provide is anonymous and not shared with anyone. We also do not store any information on our server.

Disease

I agree to the **E Health Care** terms and conditions

Back **Next**



Figure 7.10: Symptom Checker Welcome Window

Symtoms Checker

CHECKE YOUR DISEASE POSSIBLE CASES NOW !

Welcome

What is your age?
18

Patient

What is your sex?
 Male
 Female

Symptom

Disease

Back **Next**



Figure 7.11: Symptom Checker Patient Age Window

Figure 7.12: Symptom Checker Symptoms Select Window

Figure 7.13: Symptom Checker Possible Cases Result

Welcome !

ABOUT US

About This Platform



Lifestyle diseases are common among the population today not only in India but also in almost every country. Lifestyle diseases are caused because of the habits that we have on a day to day basis. The way one lives his life is the major cause of it. It includes heart disease, hypertension, etc. which all may hear of. In our life also, one also comes across at least one person who is either suffering from such diseases or the diseases became the reason for his death. We also came across many such people who died because they were not aware of their disease and were left with no appropriate time for treatment.

That is why we decided to develop the portal which will analyse the data entered by the user and will give the predictions of the diseases which he or she may have chances to suffer from. This not only gives the predictions but also gives you the preventive



Figure 7.14: About Us Page

Diseases

ALL DISEASES DISPLAY HERE.

Search Disease Here...



Figure 7.15: Diseases Display Page

The screenshot shows a disease profile page. At the top, there are three red error messages: "PIN CODE : Pin Code Not Mentioned !", "CITY : City Details Not Mentioned !", and "AREA : Area Details Not Mentioned !". Below this, the disease information is listed: "Disease Added Date ~ Tue Mar 01 2022 14:28:37 GMT+0530", "Added By", "Dr. Anupam R. Dandgavhal", "BAMS (Ayurveda)", "Email On ~ anupam.dandgavhal@gmail.com", "About ~ Dr. Anupam R. Dandgavhal is an Ayurveda, General Physician and General Practitioner in Jalgaon Peth, Jalgaon and has an experience of 8 years in these fields. Dr. Anupam R. Dandgavhal practices at Atharva Speciality Clinic in Jalgaon Peth, Jalgaon. He completed BAMS from MUHS in 2007, PGFP from MUHS in 2008 and PGDEMS from WMAS England in 2008..", and "Joining Date ~ Sun Feb 27 2022 16:40:32 GMT+0530". At the bottom, there are two blue buttons: "Click Here To Ask Any Queries ?" and "Click Here To Add Your Valueable Feedback (Comment)".

02 Comments

Sumit Chauhan
a month ago
Good Information Provided



Figure 7.16: Disease Information

The screenshot shows a modal window titled "Ask Question ?". It contains three input fields: "Enter Your Full Name" with "Pruthviraj Rajput" typed in, "Enter Your Email" with "pruthviraj.rajput01@gmail.com" typed in, and a large text area labeled "Ask Your Question ?". At the bottom right of the modal is a blue button labeled "Ask Question". The background of the page shows a comment from "Jayesh Patil" with a timestamp of "a month ago" and the text "Very Good Inform".

01 Comments

Jayesh Patil
a month ago
Very Good Inform



Figure 7.17: Disease Related Ask Question

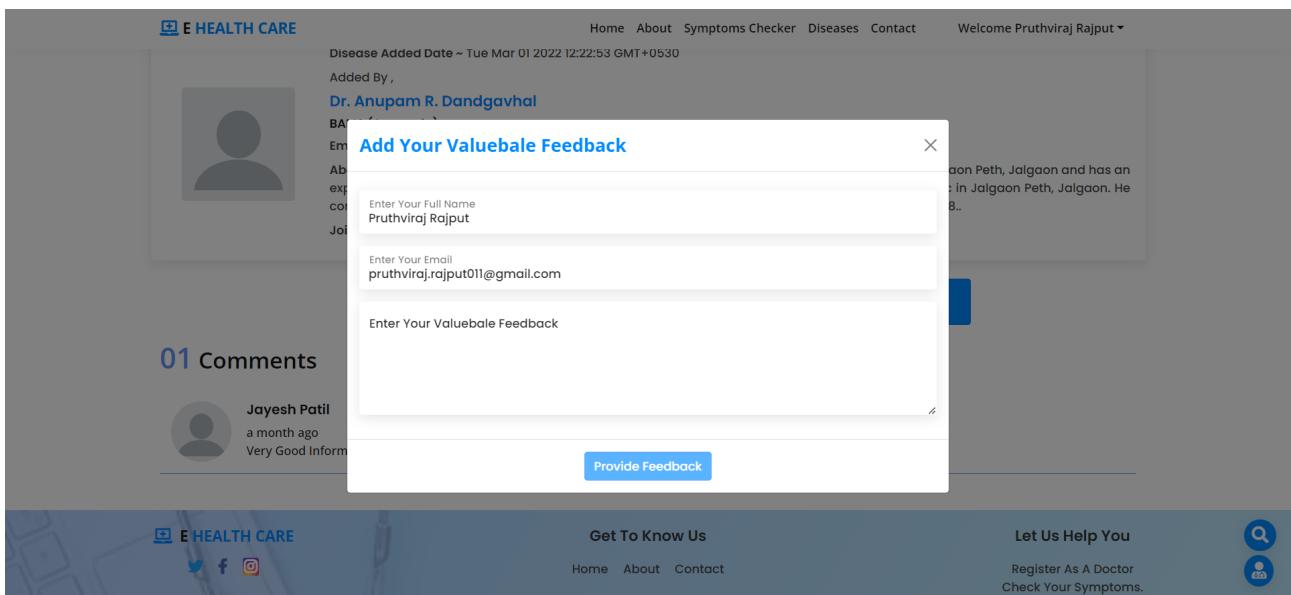


Figure 7.18: Specific Disease Add Feedback

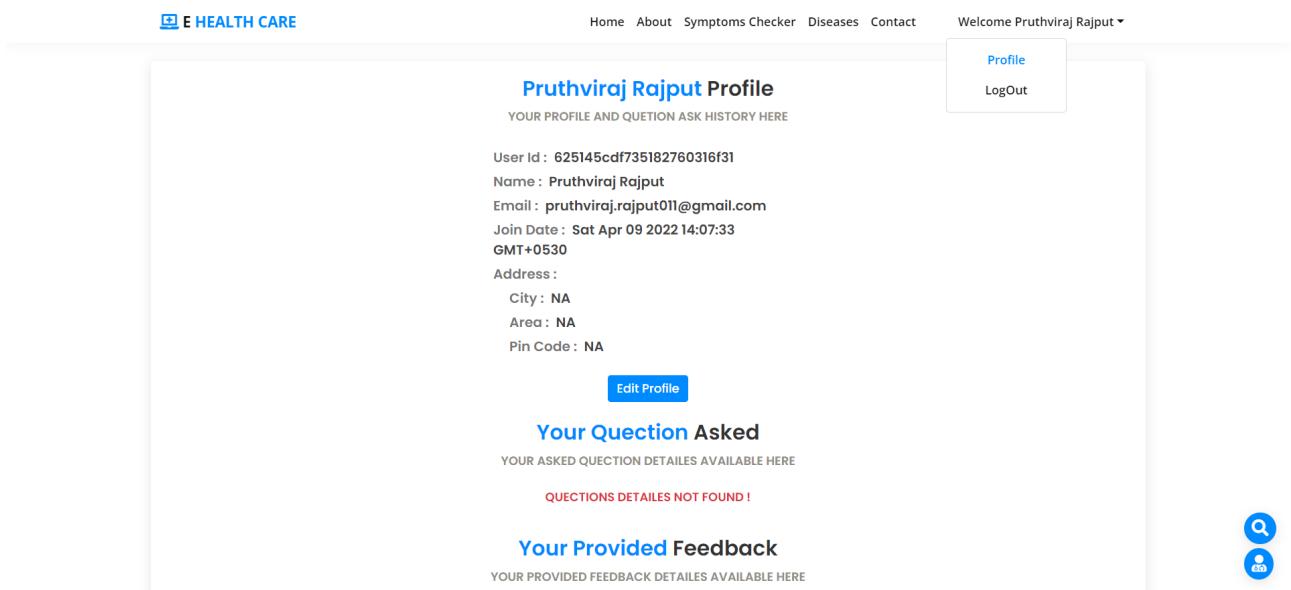


Figure 7.19: User Profile

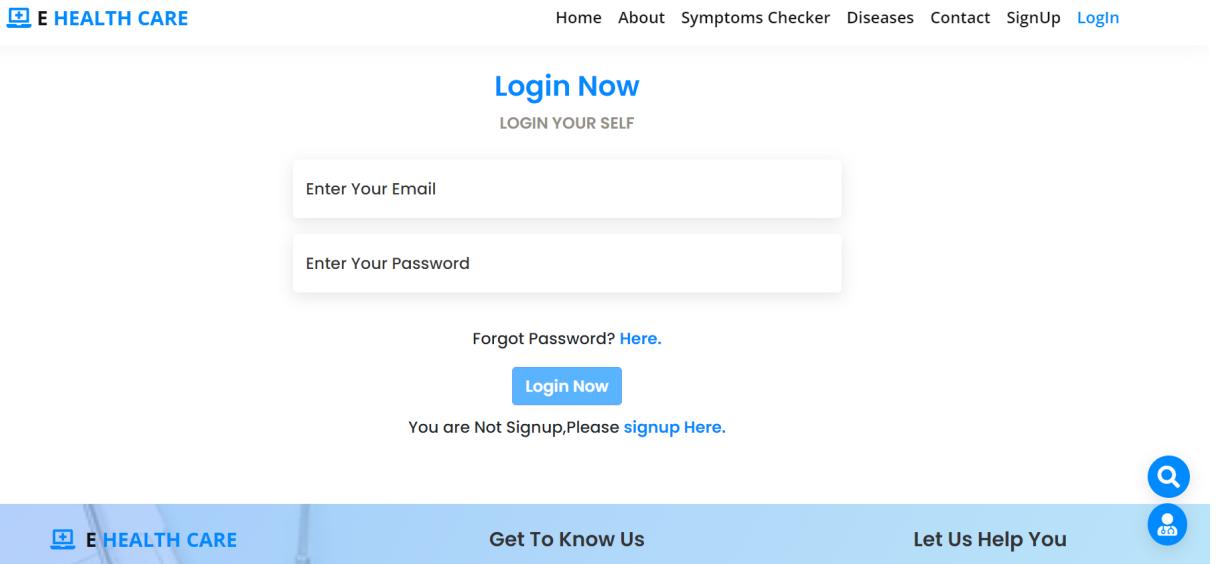


Figure 7.20: Login Page

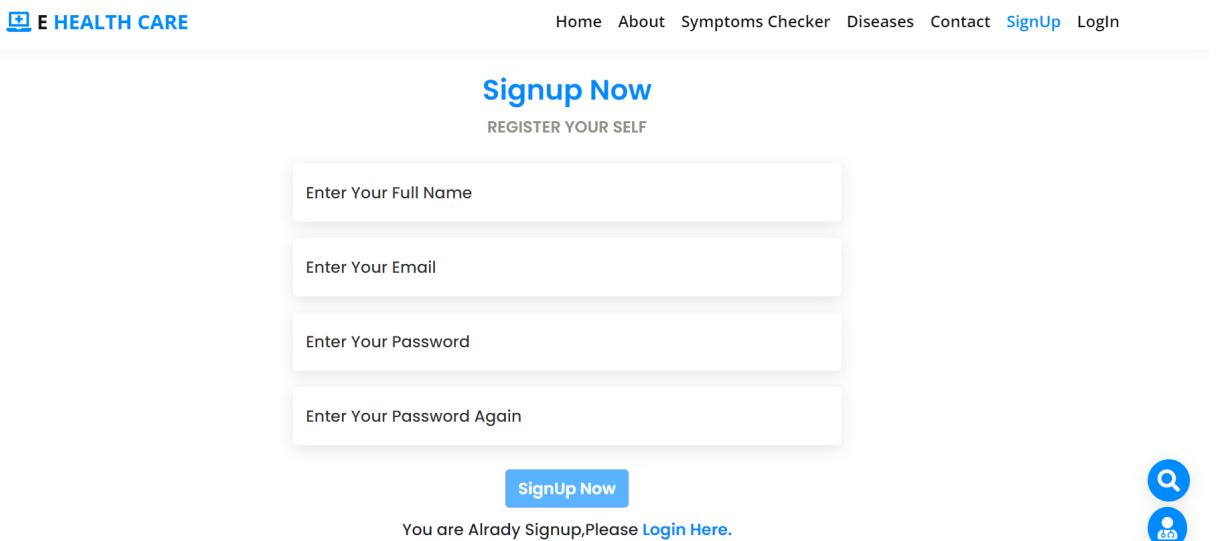


Figure 7.21: Sign up Page

The screenshot shows the login page for doctors on the E Health Care platform. At the top left is the logo 'E HEALTH CARE'. At the top right are links for 'SignUp' and 'Login'. The main heading is 'Login Now' with a sub-instruction 'LOGIN AS A DOCTOR'. Below this are two input fields: 'Enter Your Email' and 'Enter Your Password'. To the right of the password field is a link 'Forgot Password? [Here.](#)'. Below the input fields is a blue 'Login Now' button. Further down are links for 'Login As A System Admin [Login Here.](#)' and 'You are Not Register,Please [Register Here.](#)'.

Figure 7.22: Login Page Doctor

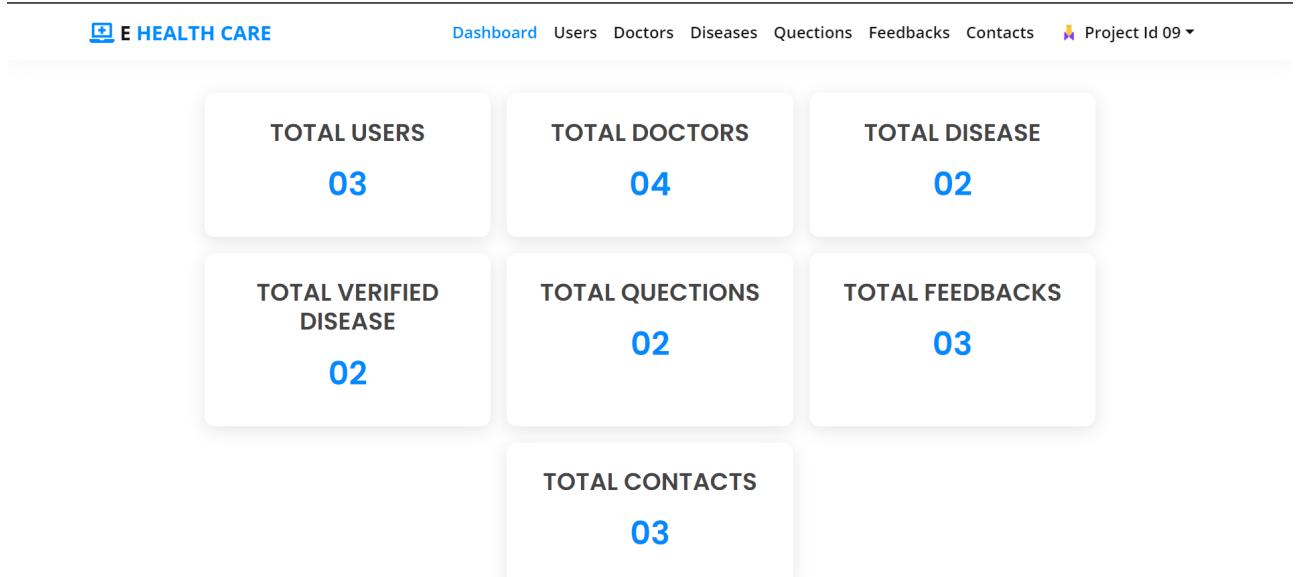


Figure 7.23: Admin Dashboard

All Doctors

ALL DOCTORS DETAILES DISPLAY HERE

#	Name	Email Id	Address	Join Date	Action
1	Kirti Ravindra Patil	rkpatil5020@gmail.com	City : Jalgaon, Area : Khote Nagar,Jalgaon, PinCode : 425001	02 March 2022	 
2	Dr. Pritesh Chaudhari	pritesh.patil@gmail.com	City : NA, Area : NA, PinCode : NA	02 March 2022	 
3	Dr. Manisha A Gosavi	manisha.gosavi@gmail.com	City : NA, Area : NA, PinCode : NA	01 March 2022	 
4	Dr. Anupam R. Dandgavhal	anupam.dandgavhal@gmail.com	City : NA, Area : NA, PinCode : NA	27 February 2022	 

Figure 7.24: All Doctor Display

All Diseases

ALL DISEASES DETAILES DISPLAY HERE



#	Disease Name	Description	Verification Status	Visibility Status	Added Date	Approved Date	Action
1	Diabetes	Diabetes is a disease that occurs when your blood glucose,...	Approved	Approved	01 March 2022	01 March 2022	   
2	Malaria	Malaria is a disease caused by a parasite. The parasite...	Approved	Approved	01 March 2022	01 March 2022	   

Figure 7.25: All Diseases Display

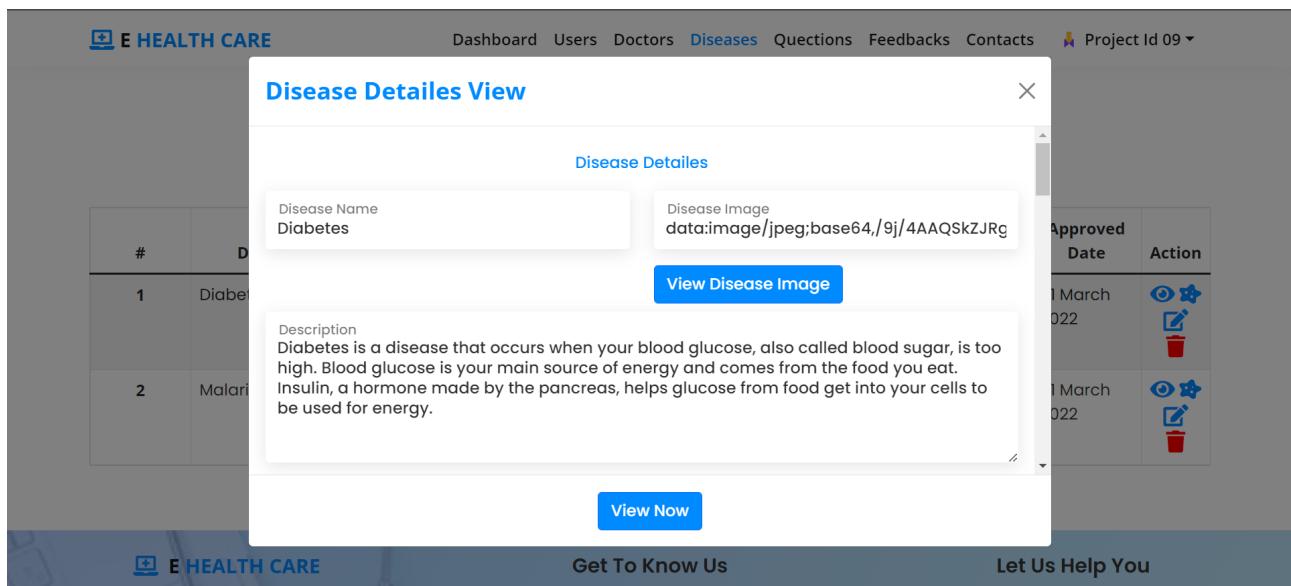


Figure 7.26: All Diseases Display

Question Details								
ALL QUESTION DETAILS DISPLAY HERE								
#	Name	Disease Id	Email Id	Question	Asked Date	Answer	Answer Date	Action
1	Pruthviraj Rajput	621dc2c59584ad558877addf	pruthviraj.rajput01@gmail.com	Please tell me detailed information about this disease?	04 March 2022	Not Answer	05 March 2022	
2	Sumit Chauhan	621de03dd930a1065fd6fe54	sumit@gmail.com	Please tell me full details of this disease?	04 March 2022	Not Answer	05 March 2022	

Figure 7.27: All Question Display

7.3 Summary

In this chapter, the result and analysis details, test cases and test results are described. In the next chapter, Conclusion and Future Scope are discussed.

Chapter 8

Conclusion and Future Work

8.1 Conclusion

So, Finally we conclude by saying that, this project early prediction of lifestyle diseases. Learning is very much useful in everyone's day to day life and it is mainly more important for the healthcare sector, because they are the one that daily uses these systems to predict the diseased of the patients based on their general information and there symptoms that they are been through. Now a day's health industry plays major role in curing the diseases of the patients so this is also some kind of help for the healthy industry to tell the people and also it is useful for the user in case he/she doesn't want to go to the hospital or any other clinics, so just by entering the symptoms and all other useful information the user can get to know the disease he/she is suffering from and the health industry can also get benefit from this portal by just asking the symptoms from the user and entering in the system and in just few seconds they can tell the exact and up to some extent the accurate diseases. If the health industry adopts this project then the work of the doctors can be reduced and they can easily predict the disease of the patient. The Disease prediction is to provide prediction for the various and generally occurring diseases that when unchecked and sometimes ignored can turn into fatal disease and cause a lot of problems to the patient and as well as their family members.

8.2 Future Work

Our team modified the overall system and added some functionality like email sending, forgetting password, SMS notification and many more.

Bibliography

1. Research paper Early Prediction and Prevention of Lifestyle Diseases by Sakshi Gaur, Sarvesh Sharma and Ayush Tripathi <https://easychair.org/publications/preprint/Dztm>
2. Research paper A Framework for Supervising Lifestyle Diseases Using Long-Term Activity Monitoring by Yongkoo Han, Manhyung Han, Sungyoung Lee, A. M. Jehad Sarkar, and Young-Koo Lee1, <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3386688/>
3. Wikipedia results https://en.wikipedia.org/wiki/Lifestyle_disease
4. Github search refer symptom checker implementation <https://github.com/topics/symptom-checker>
5. Software Development for report reference,
[https://www.tutorialspoint.com/software_engineering/software_development_life_cycle.htm](https://www.tutorialspoint.com/software_engineering/software_development_lifecycle.htm)

Index

Kilo Lines of Code, 15