**NATIONAL COLLEGE OF COMPUTER STUDIES (NCCS)**

**Tribhuvan University**

**Institute of Science and Technology**

**Project Report On:**

**MEDCARE: Symptom Based Disease Prediction**

**Submitted To:**

**National College of Computer Studies**

**Department of Computer Science and Information Technology**

**Paknajol, Kathmandu**

**Submitted By:**

**Pratik Barakoti**

**Pratik Dhakal**

**Aayush Gyawali**

**14th Jan 2024**

# Acknowledgement

I would like to express my sincere gratitude to Mr. Teksan Gharti Magar, our esteemed mentor, for his invaluable guidance and unwavering support throughout the development of our symptoms-based disease prediction system. His expertise and encouragement have been instrumental in shaping the project, and his insightful feedback has significantly enhanced our understanding of the complex domains of both healthcare and machine learning. Mr. Gharti Magar's dedication to our academic and professional growth has been truly inspiring. We are deeply appreciative of his time, patience, and commitment to ensuring the success of this college project. His mentorship has been a beacon, illuminating our path and contributing immensely to the fulfillment of our academic endeavors.

In addition, I would like to extend my heartfelt appreciation to Mr. Teksan Gharti Magar for fostering an environment of intellectual curiosity and continuous learning. His encouragement to explore innovative solutions and his emphasis on collaborative problem-solving have been instrumental in shaping not only our technical skills but also our approach to real-world problem-solving. His passion for the subject matter and commitment to our academic success have created a motivating and enriching learning experience. We are truly fortunate to have had Mr. Gharti Magar as our mentor, and we look forward to applying the knowledge and skills acquired under his guidance in our future endeavors.

We are honored:

**Pratik Barakoti**

**Aayush Gyawali**

**Pratik Dhakal**

Date: 2024/01/14

# Abstract

The abstract of this report encapsulates the development and implementation of a Symptom-Based Disease Prediction System. Utilizing machine learning algorithms integrated with a Django backend and MySQL database, the system processes user-input symptoms to predict potential diseases, thereby contributing to early detection and healthcare management. The report details the challenges encountered during the project's development, encompassing data collection, model training, and system integration. By seamlessly blending technologies, the project represents an innovative approach to healthcare solutions. The abstract distills the essence of our work, highlighting the practical application of machine learning in the crucial realm of public health.

The main objective of this project is to provide the users facility to get idea of what disease have they been diagnosed to and get doctor assistance digitally having not to continuously go to Medicals and hospitals. We want to make Hospital 1 time stop for the users.

Keywords:

Disease Prediction, Symptoms based prediction, Doctor Recommendation, Medcare, Doctify

**Table of Content**

[Acknowledgement i](#_Toc156071922)

[Abstract ii](#_Toc156071923)

[List of Figures vi](#_Toc156071924)

[List of Tables vii](#_Toc156071925)

[Chapter 1 : Introduction 1](#_Toc156071926)

[1.1 Introduction 1](#_Toc156071927)

[1.2 Problem Statement 1](#_Toc156071928)

[1.3 Objective 2](#_Toc156071929)

[1.3.1 Major Objectives 2](#_Toc156071930)

[1.3.2 Project as a solution 2](#_Toc156071931)

[1.4 Scope and Limitation 2](#_Toc156071932)

[1.4.1 Project Scope 2](#_Toc156071933)

[1.4.2 Project Limitation 2](#_Toc156071934)

[1.5 Methodology 3](#_Toc156071935)

[Chapter 2 : Background Study and Literature Review 4](#_Toc156071936)

[2.1 Fundamental Theories 4](#_Toc156071937)

[2.1.1 Description of ‘Random Forest’ Algorithm 4](#_Toc156071938)

[2.1.2 Description of ‘Naive Bayes’ Algorithm 5](#_Toc156071939)

[2.2 Related Works 7](#_Toc156071940)

[2.3 Existing Systems 8](#_Toc156071941)

[2.4 Existing Research Works 8](#_Toc156071942)

[Chapter 3 : System Analysis 10](#_Toc156071943)

[3.1. System Analysis 10](#_Toc156071944)

[3.1.1. Requirement Analysis 10](#_Toc156071945)

[i. Functional Requirement: 10](#_Toc156071946)

[ii. Non-Functional Requirements: 11](#_Toc156071947)

[3.2 Feasibility Study: 11](#_Toc156071948)

[3.2.1 Technical Feasibility: 11](#_Toc156071949)

[3.2.2 Economic Feasibility: 12](#_Toc156071950)

[3.2.3 Operational Feasibility: 12](#_Toc156071951)

[3.2.4 Schedule Feasibility: 12](#_Toc156071952)

[3.3. Analysis 13](#_Toc156071953)

[3.3.1 Sequence Diagram 13](#_Toc156071954)

[3.3.2 State Diagram 16](#_Toc156071955)

[3.3.3 User Activity Diagram 17](#_Toc156071956)

[3.3.4 Doctor Activity Diagram 18](#_Toc156071957)

[3.3.4 Class Diagram 19](#_Toc156071958)

[3.3.5 Object Diagram 20](#_Toc156071959)

[Chapter 4 : System Design 21](#_Toc156071960)

[4.1 System Architecture 21](#_Toc156071961)

[4.1.1 System Flow 21](#_Toc156071962)

[4.1.2 Disease Prediction Module 23](#_Toc156071963)

[4.2 Use Case Diagram 24](#_Toc156071964)

[4.3 Deployment Diagram 25](#_Toc156071965)

[4.4 Component Diagram 26](#_Toc156071966)

[Chapter 5 : Implementation and Testing 27](#_Toc156071967)

[5.1 Implementation 27](#_Toc156071968)

[5.1.1 Tools Used 27](#_Toc156071969)

[5.1.2 Technology Used 27](#_Toc156071970)

[5.2 Description of disease prediction model 27](#_Toc156071971)

[5.2.1 Data gathering 27](#_Toc156071972)

[5.2.2 Data Preparation for Classification Modeling 28](#_Toc156071973)

[5.2.3 Building the Gaussian Naive Bayes Model 28](#_Toc156071974)

[5.2.4 Building Random Forest Model 28](#_Toc156071975)

[5.3 Testing 29](#_Toc156071976)

[5.3.1 Unit Testing 29](#_Toc156071977)

[5.3.2 System Testing 30](#_Toc156071978)

[5.3.3 Performance Testing 31](#_Toc156071979)

[Chapter 6 : Conclusion and Future Recommendations 32](#_Toc156071980)

[6.1 Conclusion 32](#_Toc156071981)

[6.2 Future Improvement 32](#_Toc156071982)

[References 33](#_Toc156071983)

[Screenshots 34](#_Toc156071984)

# List of Figures

[Figure 1.1 Agile Methodology of Medcare 11](file:///C:\Users\Modern\Documents\MedCare%20Report.docx#_Toc156048118)

[Figure 2.1 Algorithm Structure 14](file:///C:\Users\Modern\Documents\MedCare%20Report.docx#_Toc156048119)

[Figure 3.1 User sequence diagram 21](file:///C:\Users\Modern\Documents\MedCare%20Report.docx#_Toc156048120)

[Figure 3.2 Admin sequence Diagram 22](file:///C:\Users\Modern\Documents\MedCare%20Report.docx#_Toc156048121)

[Figure 3.3 Doctor Sequence Diagram 23](#_Toc156048122)

[Figure 3.4 State Diagram of Medcare 24](file:///C:\Users\Modern\Documents\MedCare%20Report.docx#_Toc156048123)

[Figure 3.5 User Activity Diagram of Medcare 25](file:///C:\Users\Modern\Documents\MedCare%20Report.docx#_Toc156048124)

[Figure 3.6 Doctor Activity Diagram of Medcare 26](#_Toc156048125)

[Figure 3.7 Class Diagram of Medcare 27](file:///C:\Users\Modern\Documents\MedCare%20Report.docx#_Toc156048126)

[Figure 3.8 Object Diagram of Medcare 28](file:///C:\Users\Modern\Documents\MedCare%20Report.docx#_Toc156048127)

[Figure 4.1 System Flow Diagram of Medcare (User) 29](file:///C:\Users\Modern\Documents\MedCare%20Report.docx#_Toc156048128)

[Figure 4.2 System Flow Diagram of Medcare (Doctor) 30](file:///C:\Users\Modern\Documents\MedCare%20Report.docx#_Toc156048129)

[Figure 4.3 Disease Prediction Model of Medcare 31](file:///C:\Users\Modern\Documents\MedCare%20Report.docx#_Toc156048130)

[Figure 4.4 Use Case Diagram of Medcare 32](file:///C:\Users\Modern\Documents\MedCare%20Report.docx#_Toc156048131)

[Figure 4.5 Deployment Diagram of Medcare 33](#_Toc156048132)

[Figure 4.6 Component Diagram of Medcare 34](file:///C:\Users\Modern\Documents\MedCare%20Report.docx#_Toc156048133)

# List of Tables

[Table 3.1 Functional Requirements 18](#_Toc156065545)

[Table 3.2 Non-Functional Requirements 19](#_Toc156065546)

[Table 5.1 Tools Used 35](#_Toc156065547)

[Table 5.2 Technologies Used 35](#_Toc156065548)

[Table 5.3 Scores Using Naïve Bayes 36](#_Toc156065549)

[Table 5.4 Scores using Random Forest 36](#_Toc156065550)

[Table 5.5 Testing of Disease Prediction 37](#_Toc156065551)

[Table 5.6 Testing Request for Appointment 38](#_Toc156065552)

[Table 5.7 Testing for Loading Application 39](#_Toc156065553)

[Table 5.8 Testing for moderate users 39](#_Toc156065554)

[Table 5.9 Testing for peak users 39](#_Toc156065555)

# : Introduction

## 1.1 Introduction

"Medcare" is a cutting-edge disease recommendation and prediction system that harnesses the power of advanced technology to revolutionize healthcare and improve patient outcomes. Our platform integrates a range of algorithms and data analysis techniques to offer personalized insights and predictive recommendations for a wide spectrum of medical conditions. With a strong focus on patient well-being and proactive healthcare management, "Medcare" serves as a beacon of innovation in the medical field. There are some key points related to our project:

* Timely disease Identification and prediction
* Improved patient care
* Real-time updates
* Personalized Recommendation

## 1.2 Problem Statement

With the rise in number of patient and disease every year medical system is overloaded and with time have become overpriced in many countries. Most of the disease involves a consultation with doctors to get treated. In today's fast-paced world, timely and accurate diagnosis of health issues is of paramount importance. However, individuals often face challenges in promptly identifying and addressing potential health concerns. This project aims to develop a machine learning-based system that predicts potential health problems based on user-provided symptoms. The existing healthcare infrastructure often experiences high patient volumes, leading to extended waiting times and delayed diagnoses. This solution seeks to alleviate this strain by providing a preliminary assessment of health conditions based on user-reported symptoms.

The primary objective of this project is to create an intuitive and user-friendly interface where individuals can input their symptoms, leading to the generation of a probabilistic prediction regarding the most likely health problems they might be facing. This predictive model will be trained on a comprehensive dataset encompassing a wide range of medical conditions and their associated symptoms. Additionally, the model will continuously learn and adapt based on user feedback and emerging medical research.

The successful implementation of this system will revolutionize the way individuals engage with their own health, empowering them to make informed decisions and seek appropriate medical attention in a timely manner. Furthermore, it has the potential to act as a valuable tool for healthcare providers, enabling them to prioritize and optimize patient care based on initial symptom assessments.

## 1.3 Objective

### 1.3.1 Major Objectives

The main objectives of this project are as follows:

* Identify potential health issues at an early stage by analyzing symptoms through predictive analysis
* Provide recommendations for Drugs and doctor appointment

### 1.3.2 Project as a solution

Our project aims to accomplish the above-mentioned goal by effectively implementing a module on the unique web-based application to solve the issue.

## 1.4 Scope and Limitation

### 1.4.1 Project Scope

This portion of our project describes the project's objectives, delivery strategy, and level of effectiveness**.** Create algorithms to recommend appropriate medications based on individual patient profiles, disease predictions, and known drug interactions.

### 1.4.2 Project Limitation

Although this initiative achieves some of its objectives, there is still room for improvement. The primary limitation is the system's size, which was intended to accommodate a limited number of users. Furthermore, the models' accuracy could differ because so little data was utilized to train them. It's possible that the model won't adjust effectively to emerging illnesses or shifting epidemiological trends. Resource-intensive models may be more difficult to use in settings with limited resources since they demand a large amount of infrastructure and processing power.

## 1.5 Methodology

The selection of the Scrum framework for Agile development plays a pivotal role. Scrum is chosen due to its iterative and incremental approach, which is particularly well-suited for projects like "Medcare” Scrum divides the project into manageable iterations called sprints, typically 2 to 4 weeks long, during which specific tasks and features are developed and tested. This iterative and incremental process ensures that at the end of each sprint, there's a working portion of the project, allowing for continuous feedback from stakeholders, healthcare professionals, and patients. This feedback is used to adapt and refine the project, ensuring it aligns closely with user needs and the evolving healthcare landscape. Scrum's focus on flexibility, adaptability, and collaboration with users makes it ideal for "Medcare” enabling a dynamic development process that can respond effectively to changing requirements and incorporate valuable feedback.

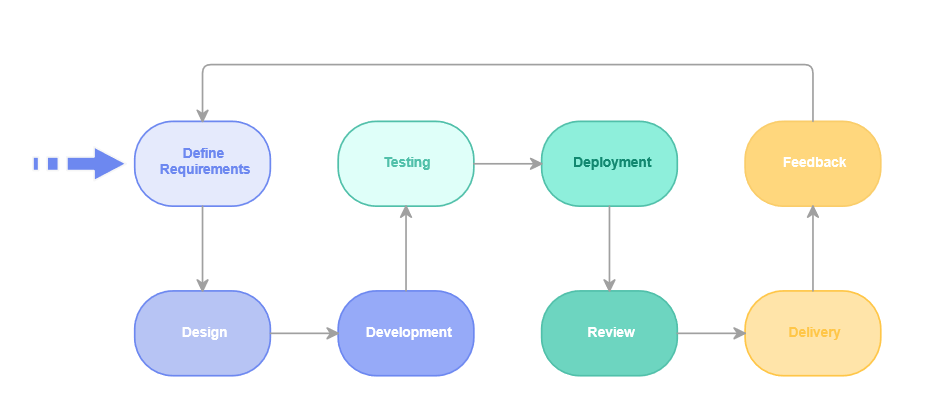
**

Figure . Agile Methodology of Medcare

# : Background Study and Literature Review

## 2.1 Fundamental Theories

Medcare is a website that provides facility to the general public who is sick/ill. It provides help to the people digitally without having to go in physical facility. The users can simply use the website to login and input the symptoms that they are facing and then have the drugs recommended and they can also take appointment of the doctor for further checkup. Our facility will be provided in form of mobile application and website.

### 2.1.1 Description of ‘Random Forest’ Algorithm

A Random Forest is an ensemble learning algorithm that is used for both classification and regression tasks. It is a versatile and powerful machine learning method that operates by constructing a multitude of decision trees during training and outputs the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.

Here's a step-by-step description of how the Random Forest algorithm works:

1. **Data Collection**: Gather a dataset that includes information on symptoms, diseases, and corresponding drug recommendations.
2. **Feature Extraction**: Identify relevant features from the dataset, such as various symptoms and other relevant patient information.
3. **Training the Random Forest**: Utilize the Random Forest algorithm to train a model on the dataset. During training, the algorithm will create an ensemble of decision trees based on subsets of the data and symptoms.
4. **Symptom-Based Decision Trees**: Each decision tree in the Random Forest focuses on subsets of symptoms and their relationships with diseases. The randomness introduced during tree construction helps capture diverse patterns in symptom-disease associations.
5. **Prediction Process**: When a new set of symptoms is input into the system, the ensemble of decision trees collectively predicts the likely disease. For drug recommendation, this prediction can be further used to suggest appropriate medications associated with the predicted disease.
6. **Ensemble Voting for Robust Predictions**: The Random Forest ensemble provides robustness by aggregating predictions from multiple decision trees. The final disease prediction is often determined by majority voting among the trees.
7. **Drug Recommendation**: Based on the predicted disease, the system can recommend drugs commonly prescribed or associated with the identified condition. This recommendation can be further refined based on additional patient-specific factors if available in your dataset.

### 2.1.2 Description of ‘Naive Bayes’ Algorithm

Naive Bayes is particularly suitable for text classification tasks, making it effective for symptom-based disease prediction where symptoms can be treated as features. It is computationally efficient and can provide quick predictions, making it a valuable component in a symptom-based disease prediction system with drug recommendation

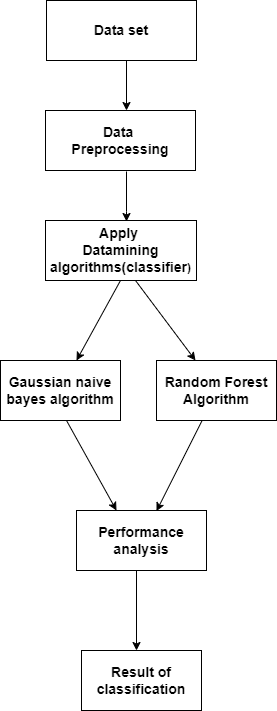
1. **Data Representation**: Collect a dataset with symptom-disease pairs, where each instance consists of a set of symptoms associated with a particular disease. Include drug information corresponding to each disease in the dataset.
2. **Preprocessing:** Preprocess the data by representing symptoms and diseases in a format suitable for Naive Bayes. This may involve converting categorical data into a numerical format.
3. **Training:** Use the preprocessed data to train a Naive Bayes classifier. The algorithm estimates the probability of a disease given a set of symptoms by calculating the conditional probabilities of each symptom given the presence of a disease.
4. **Classification:** Given a set of symptoms for a patient, use the trained Naive Bayes model to predict the probability of each disease. The model assigns the patient to the disease with the highest probability.
5. **Drug Recommendation**: Once the disease is predicted, associate it with the corresponding drug information from the dataset. Provide drug recommendations based on the predicted disease.
6. **Handling Independence Assumption**: Naive Bayes assumes that symptoms are conditionally independent given the disease, which may not be entirely accurate in real-world scenarios. However, this simplifying assumption often works well in practice.
7. **Feedback Loop:** Implement a feedback loop where user feedback on the accuracy of predictions and drug recommendations can be used to update and improve the model over time.

Figure . Algorithm Structure

## 2.2 Related Works

There has been a significant amount on disease prediction system in last decade. Some of them are listed below:

1. "A Survey on Deep Learning in Medical Image Analysis" (2017)

This survey provides insights into the applications of deep learning, including neural networks, in medical image analysis. While it may not focus explicitly on symptoms, understanding image-based disease diagnosis is crucial in the broader context of healthcare prediction systems. [1]

1. "Prediction of Heart Disease Using Classification Algorithms" (2018)

Some studies focus on specific diseases, such as heart disease. This paper discusses the use of classification algorithms for predicting heart disease based on symptoms and patient data. [2]

1. "Predictive Modeling of Hospital Readmission Rates Using Electronic Medical Record-Wide Machine Learning: A Case-Study Using Mount Sinai Heart Failure Cohort" (2018)

This work explores predictive modeling of hospital readmission rates using machine learning. While not specifically symptom-based, it demonstrates the use of machine learning in predicting healthcare outcomes. [3]

1. "Disease Prediction by Machine Learning Over Big Data from Healthcare Communities" (2017)

This paper discusses the use of machine learning over large healthcare datasets for disease prediction. It may provide insights into handling diverse data sources and predicting diseases based on symptoms. [4]

1. "A Review on Heart Disease Prediction using Data Mining Techniques" (2018)

Reviews like this one summarize existing literature on disease prediction using data mining techniques, including methods that use symptoms as input features. [5]

## 2.3 Existing Systems

Several systems and platforms have been developed for symptom-based disease prediction. These systems often leverage machine learning, artificial intelligence, and data analytics to enhance accuracy in disease prediction.

**Ada Health:**

Ada Health is an AI-powered health platform that uses a symptom-checker to help users assess their symptoms and understand potential health issues. The system considers various symptoms to provide personalized health information and advice.

**Buoy Health:**

Buoy Health is an AI-driven platform that assists users in understanding their symptoms and guides them to appropriate healthcare resources. The system uses a chat-based interface to gather information about symptoms and offers potential diagnoses.

**Google's Symptom Search:**

Google has implemented a symptom search feature that provides information about possible conditions based on entered symptoms. It aims to help users understand potential health issues and encourage informed discussions with healthcare professionals.

**Adastra Health:**

Adastra Health offers a symptom-checking tool that uses AI to analyze symptoms and provide information on potential health conditions. It aims to assist individuals in understanding their symptoms and deciding when to seek medical attention.

## 2.4 Existing Research Works

In medical domains, artificial intelligence (AI) primarily focuses on developing the algorithms and techniques to determine whether a system’s behavior is correct in disease diagnosis. Medical diagnosis identifies the disease or conditions that explain a person’s symptoms and signs. Typically, diagnostic information is gathered from the patient’s history and physical examination. According to the National Academics of Science, Engineering, and Medicine report of 2015, the majority of people will encounter at least one diagnostic mistake during their lifespan. The appropriate application of ML to these data promises to transform patient risk stratification broadly in the field of medicine and especially in infectious diseases. This, in turn, could lead to targeted interventions that reduce the spread of healthcare-associated pathogens. In this review, we begin with an introduction to the basics of ML. [6]

There has been research on Efficient Heart Disease Prediction Using Hybrid Deep Learning Classification Models. To further enhance this process, a newly developed GSA (Genetic Sine Algorithm) is proposed as it is capable of selecting optimal features and avoid getting trapped in local optima. The selected features are subjected to the classification technique by RNN (Recurrent Neural Network) integrated with LSTM (Long Short-Term Memory) algorithm. To filter out all the [invalid information](https://www.sciencedirect.com/topics/engineering/invalid-information) and emphasize only on critical information, DPA-RNN+LSTM (Deep Progressive Attention-RNN+LSTM) has been developed so as to improve the classification rate. [7]

Machine learning (ML) is used practically everywhere, from cutting-edge technology (such as mobile phones, computers, and robotics) to health care (i.e., disease diagnosis, safety). ML is gaining popularity in various fields, including disease diagnosis in health care. Many researchers and practitioners illustrate the promise of machine-learning-based disease diagnosis (MLBDD), which is inexpensive and time-efficient. Traditional diagnosis processes are costly, time-consuming, and often require human intervention. While the individual’s ability restricts traditional diagnosis techniques, ML-based systems have no such limitations, and machines do not get exhausted as humans do. As a result, a method to diagnose disease with outnumbered patients’ unexpected presence in health care may be developed.

# : System Analysis

## 3.1. System Analysis

### 3.1.1. Requirement Analysis

Requirement analysis is the essential phase of systematically gathering and examining the essential elements for a system or project. This process involves comprehending the core issue or opportunity that the system intends to address, along with identifying both functional and non-functional prerequisites crucial for achieving the desired objectives. The requirement analysis for our healthcare platform entails two primary sections: Functional Requirements and Non-Functional Requirements.

### i. Functional Requirement:

Functional requirements for symptoms-based disease prediction systems play a pivotal role in ensuring the effectiveness and reliability of such platforms. These requirements serve as a foundation for the development and implementation of robust solutions that can accurately analyze and interpret symptoms to predict potential diseases.

Table . Functional Requirements

|  |  |  |
| --- | --- | --- |
| Req no. | Req. name | Req. Description |
| FR1 | Symptom Input | Users can input their symptoms into the system. |
| FR2 | Disease Prediction | The system should predict potential diseases based on symptoms. |
| FR3 | User Profile Management | Users can create, view, and manage their profiles. |
| FR4 | User Interface | Ensure an interactive interface for smooth user experience. |

### ii. Non-Functional Requirements:

In developing a symptoms-based disease prediction system, non-functional requirements are critical to ensure its overall efficiency, reliability, and user satisfaction. These encompass aspects such as system performance, scalability, security, usability, and compliance with relevant healthcare standards.

Table . Non-Functional Requirements

|  |  |  |
| --- | --- | --- |
| Req. No. | Req. name | Req. Description |
| NFR1 | Scalability | The application will expand beyond the scope of a college project. Necessary measures, such as server upgrades and team expansion, will be implemented accordingly. |
| NFR2 | Maintainability | A comprehensive documentation of all system components will ensure easy maintenance, allowing future developers to uphold application quality effortlessly. |
| NFR3 | Usability | The website will prioritize simplicity, making features easily accessible and navigable. User interface components will be thoughtfully organized for effective usage. |
| NFR4 | Security | System administration privileges, like assigning doctors and managing permissions, will be restricted to authorized team members. Additionally, the system will be secured against SQL Injection attacks. |

## 3.2 Feasibility Study:

### 3.2.1 Technical Feasibility:

The proposed medical alert system necessitates intricate ICT solutions for continuous patient monitoring. Successful operation relies on seamless integration of hardware and software components. Once established and automated, it will efficiently gather and process patient data, utilizing specialized algorithms for timely alerts.

### 3.2.2 Economic Feasibility:

This is a web-based facility which is developed without using any physical hardware. To gather the necessary tools and implement them with the proper algorithm for hospitals, it costs the least possible. Thus, developing and maintaining Medcare is economically feasible.

### 3.2.3 Operational Feasibility:

Ensuring practicality and usability is crucial for successful adoption. Reliable models and algorithms promptly notify healthcare professionals in case of patient distress. The intuitive user interface design facilitates efficient operation, ensuring high operational feasibility.

### 3.2.4 Schedule Feasibility:

The project plan includes well-defined milestones, allowing completion within the stipulated timeline. Allocated time accounts for system development, testing, and refinement stages, ensuring adherence to the schedule.

## 3.3. Analysis

### 3.3.1 Sequence Diagram

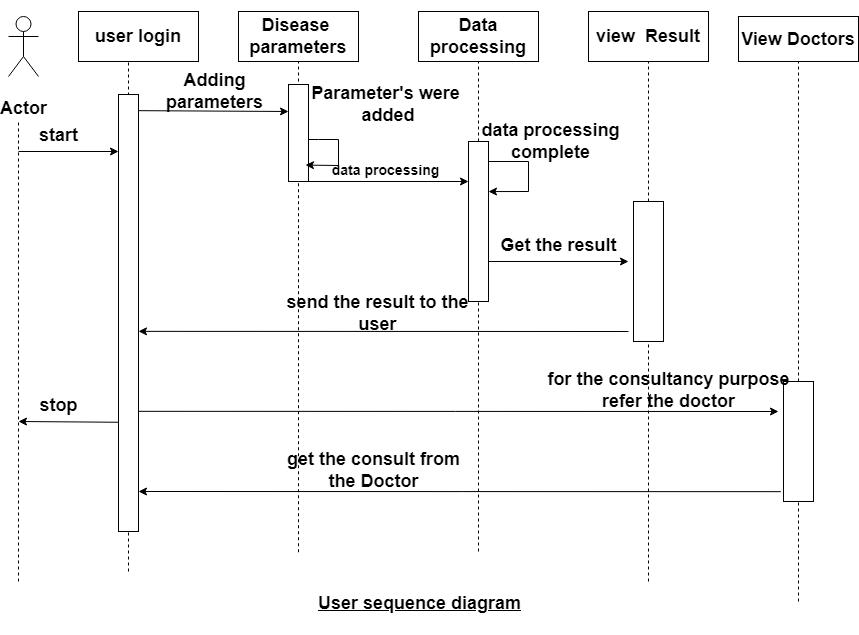


Figure . User sequence diagram

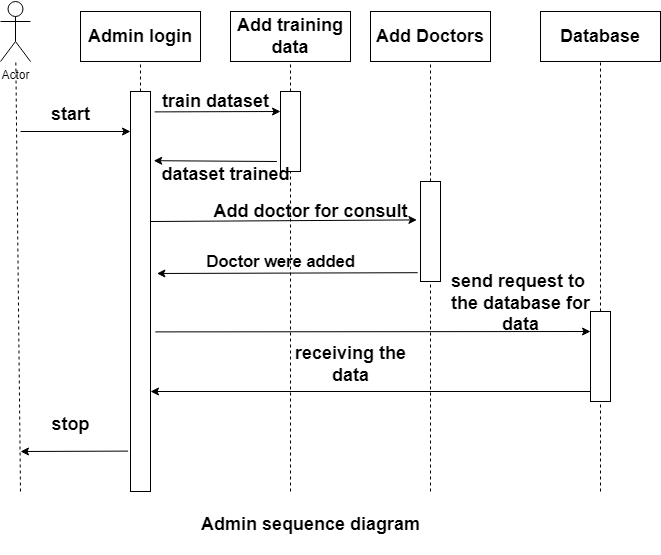


Figure . Admin sequence Diagram

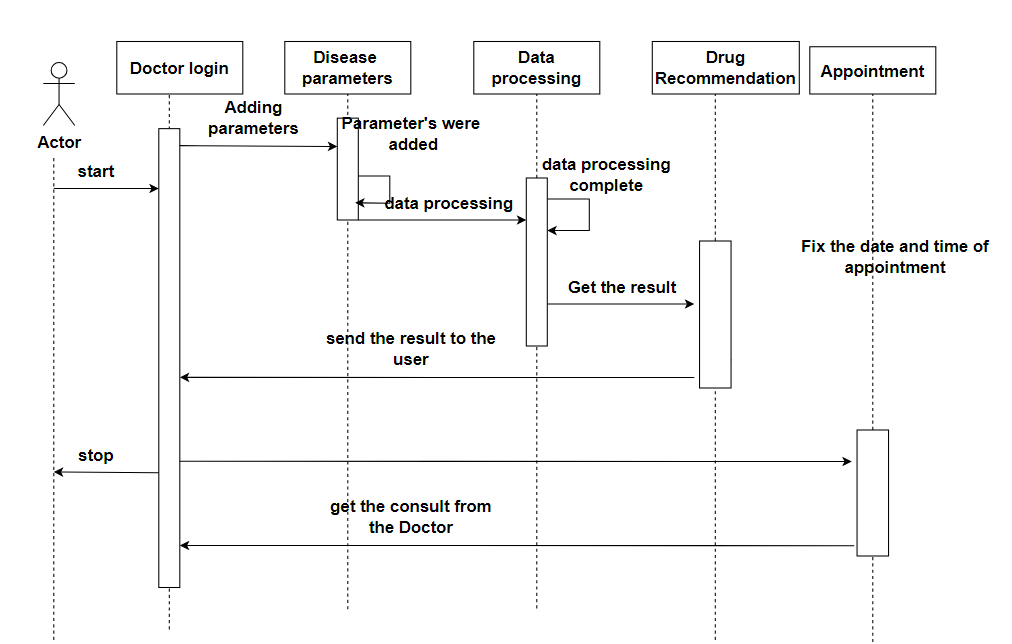


Figure . Doctor Sequence Diagram

### 3.3.2 State Diagram

Figure . State Diagram of Medcare

### 3.3.3 User Activity Diagram

Figure . User Activity Diagram of Medcare

### 3.3.4 Doctor Activity Diagram

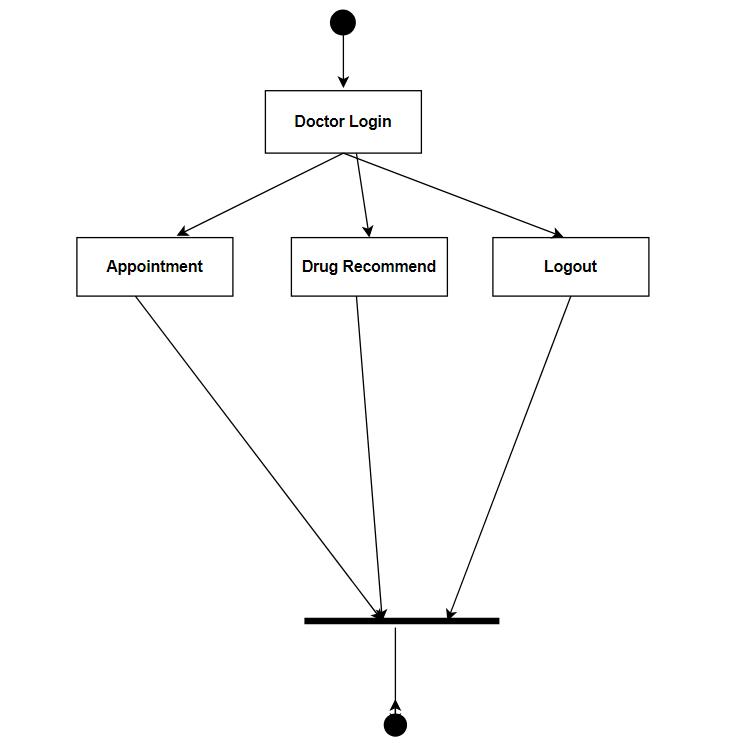


Figure . Doctor Activity Diagram of Medcare

### 3.3.4 Class Diagram

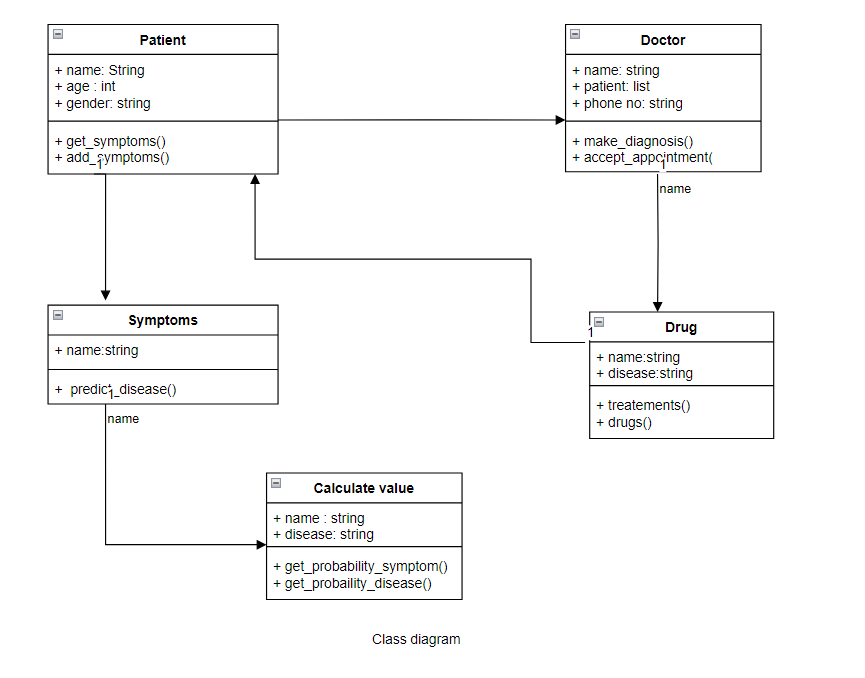


Figure . Class Diagram of Medcare

### 3.3.5 Object Diagram

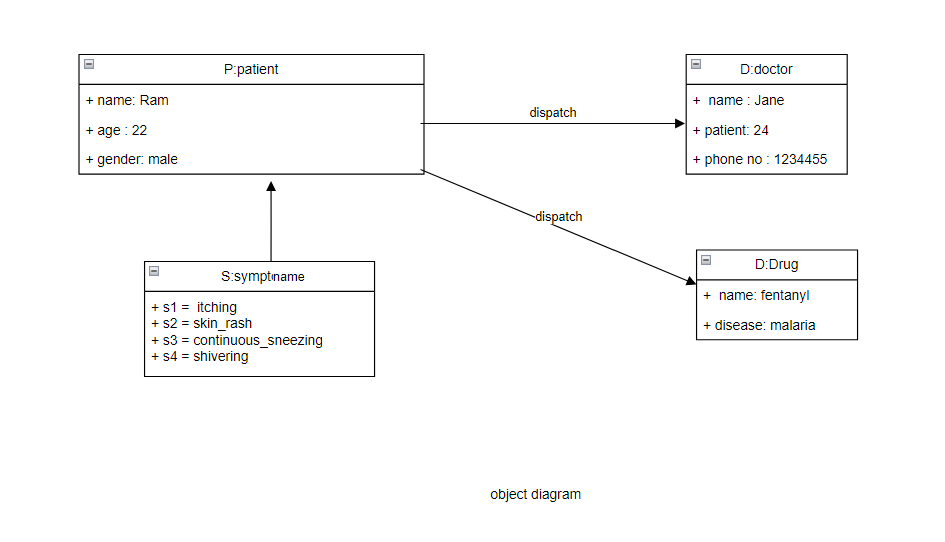


Figure . Object Diagram of Medcare

# : **System Design**

## 4.1 System Architecture

### 4.1.1 System Flow

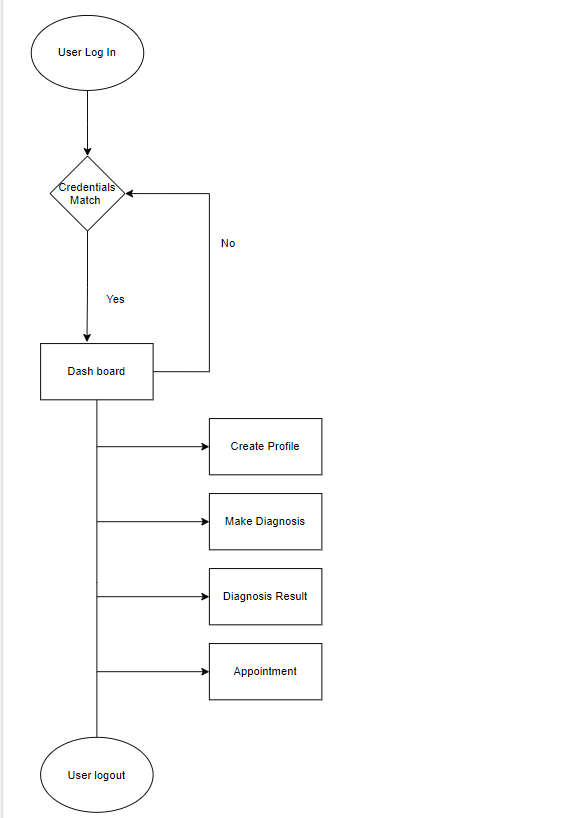
The application is layered on top of disease prediction and doctor recommendation mechanism which predicts the disease of the patient based on respective symptoms.

Figure . System Flow Diagram of Medcare (User)

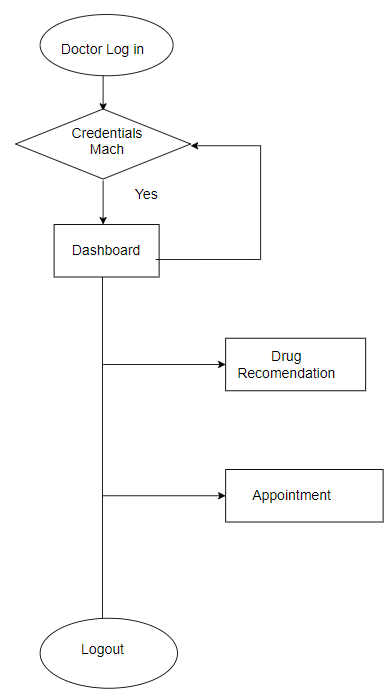


Figure . System Flow Diagram of Medcare (Doctor)

### 4.1.2 Disease Prediction Module

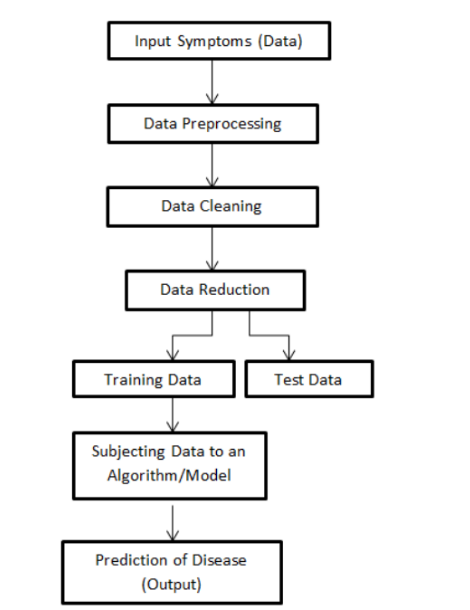


Figure . Disease Prediction Model of Medcare

## 4.2 Use Case Diagram

Figure . Use Case Diagram of Medcare

## 4.3 Deployment Diagram

The deployment diagram visualizes the physical hardware on which the software will

be deployed. It portrays the static deployment view of a system. It involves the nodes and their

relationships.

The deployment diagram visualizes the physical hardware on which the software will

be deployed. It portrays the static deployment view of a system. It involves the nodes and their

relationships.

The deployment diagram visualizes the physical hardware on which the software will be deployed. It portrays the static deployment view of a system. It involves the nodes and their relationships.

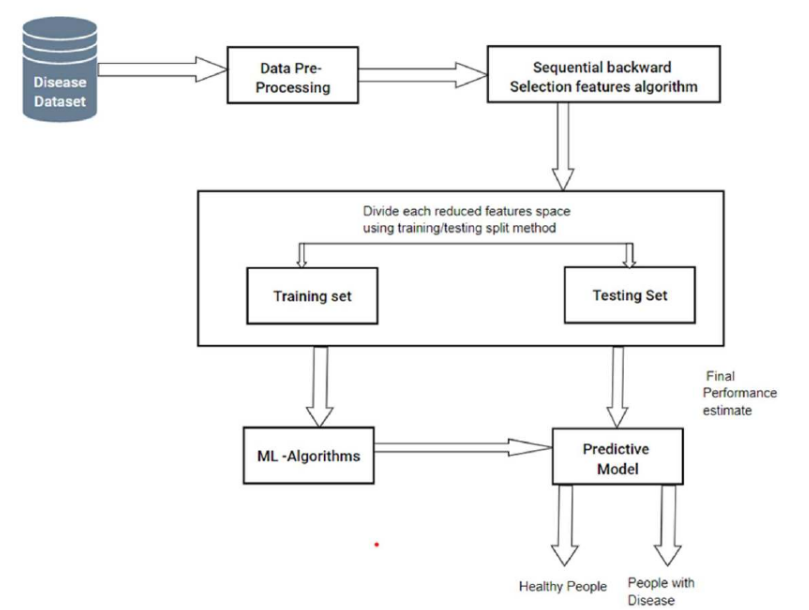


Figure . Deployment Diagram of Medcare

## 4.4 Component Diagram

A component diagram is used to break down a large object-oriented system into the smaller components, so as to make them more manageable. It models the physical view of a system such as executables, files, libraries, etc. that resides within the node.

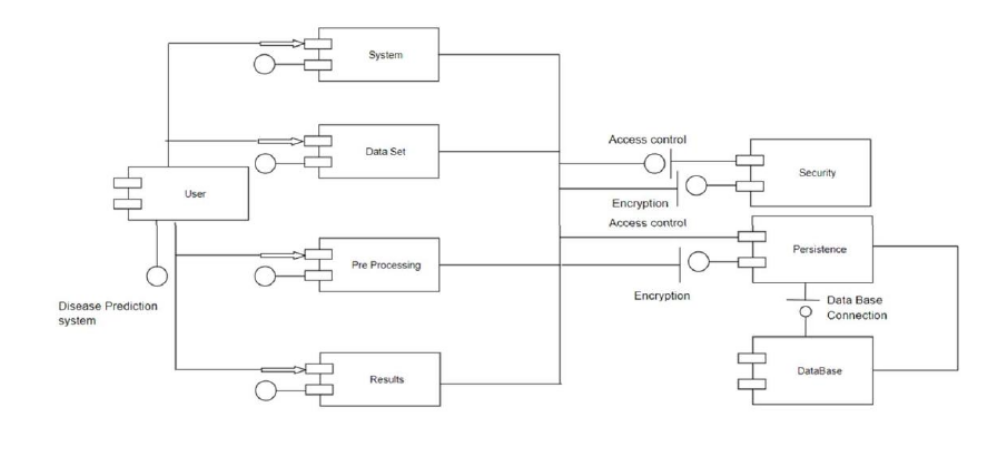


Figure . Component Diagram of Medcare

# : Implementation and Testing

## 5.1 Implementation

### 5.1.1 Tools Used

Table . Tools Used

|  |  |
| --- | --- |
| **Category** | **Tools Used** |
| Diagram | Draw.io, Photoshop |
| Communication | Discord, Google Meet, Viber |
| UI/UX | Figma |
| Code Editors | Jupyter, Sublime Text, VSCode |
| Documentation | Microsoft Office |
| Presentation | Microsoft PowerPoint |

### 5.1.2 Technology Used

Table . Technologies Used

|  |  |
| --- | --- |
| **Category** | **Technologies** |
| Frontend | Html, CSS, JS, Bootstrap |
| Backend | Django, MySQL |
| Machine Learning Libraries | SckitLearn, Pandas, Numpy, Joblib |
| Version Control | Github |

## 5.2 Description of disease prediction model

### 5.2.1 Data gathering

Data gathering is a critical phase in the development of a Disease Prediction System using Artificial Intelligence (AI). The quality and diversity of the data collected directly influence the accuracy and reliability of the predictions made by the system.

Here, in this project we used datasets from online sources like Kaggle as well we will also provide manual data for the operation of our project.

### 5.2.2 Data Preparation for Classification Modeling

The combined dataset was splitted into training and testing datasets. The dataset was splitted into around 70% training data and 30% testing data.

### 5.2.3 Building the Gaussian Naive Bayes Model

When Developing the Gaussian Naive Bayes Model, our first plan was to ensure it’s well organized with feature and target. We Split the data into features (X) and the target variable (y). After fitting the model, the following loss and accuracy scores were obtained.

Table . Scores Using Naïve Bayes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Train Dataset | Test Dataset | | Train Accuracy | Test Accuracy |
| 3936 | | 984 | 0.82 | 0.82 |

The model performance was not that great, but was a good starting point. Therefore, we worked on improving the model’s performance by increasing the number of datasets and increasing the number of epochs to 20. After the implementation of these strategies, following results were observed.

### 5.2.4 Building Random Forest Model

Table . Scores using Random Forest

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Train Dataset | Test Dataset | | Train Accuracy | | Test Accuracy |
| 3936 | | 984 | | 0.86 | 0.86 |

## 5.3 Testing

### 5.3.1 Unit Testing

Modules are tested using unit testing in comparison to the comprehensive design. The design and coding of the project have been tested at every stage. To ensure that data is correctly entering and leaving the program unit, we verify the module interface during the testing process. We verify that the data that is temporarily stored is intact while the algorithm is running by looking at the local data structure. Every path that deals with errors is tested at the end.

Table . Testing of Disease Prediction

|  |  |
| --- | --- |
| Objective | Predict disease |
| Action | User input symptoms |
| Expected result | The output should be an expected disease |
| Actual Result | The application gave the expected result |
| Conclusion | The test was successful. |

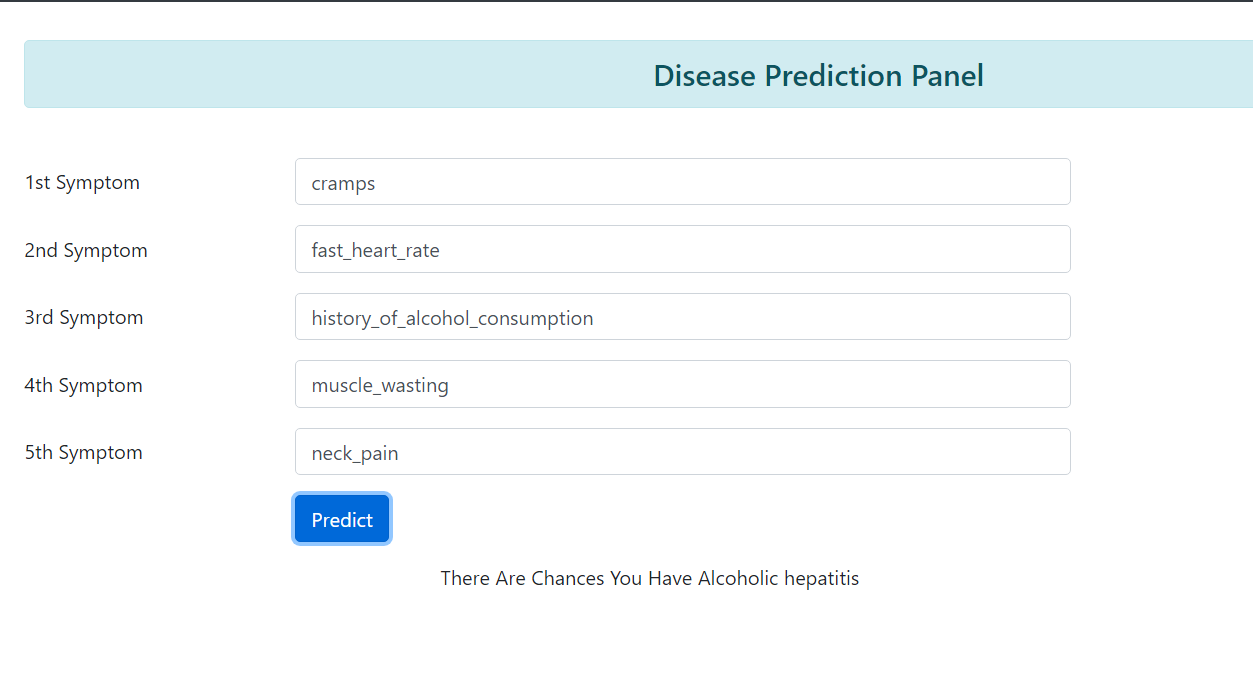
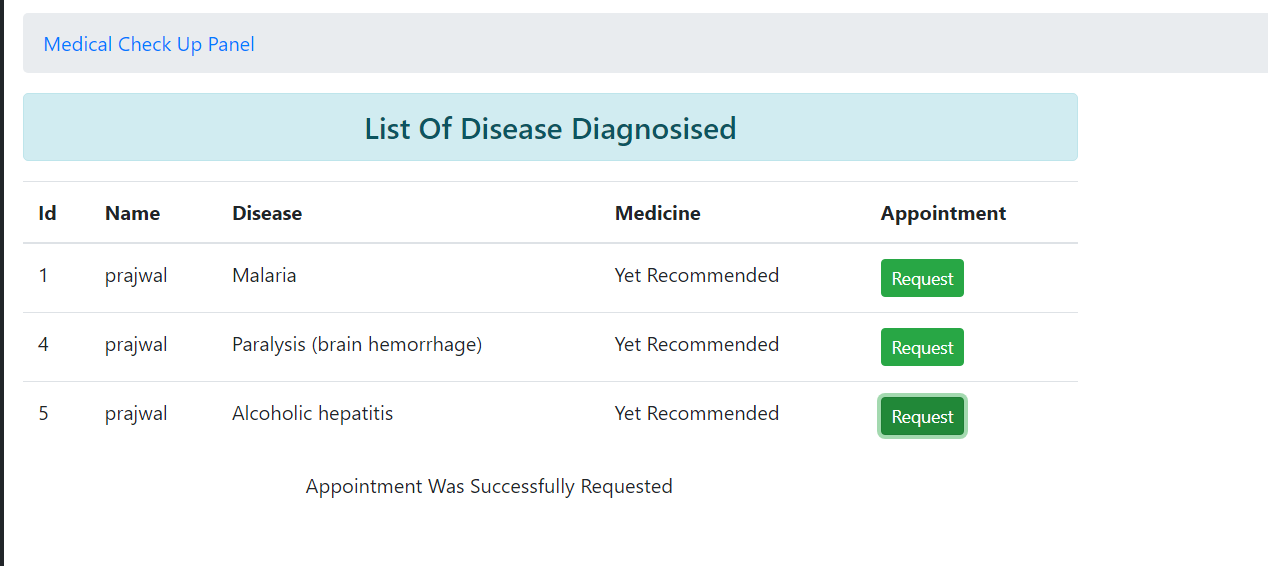
When the user inputs the given symptoms, the module gave an expected output (disease predicted).

Table . Testing Request for Appointment

|  |  |
| --- | --- |
| Objective | To make appointment. |
| Action | User request for an appointment. |
| Expected Result | The appointment is requested successfully. |
| Actual Result | The application gave a positive notification |
| Conclusion | The test was successful. |



As soon as the user request for an appointment the appointment notification is sent to doctor panel which is ready to be approved or declined.

### 5.3.2 System Testing

System testing is conducted to verify the overall functionality, integration, and performance of a software system. It aims to detect and rectify defects, ensuring the software meets specified requirements. This phase evaluates the system's reliability, usability, and compatibility across various environments. Security measures are validated, and stakeholders gain confidence in the software's readiness for deployment. System testing is a crucial step to deliver a reliable and effective product to end-users.

Table . Testing for Loading Application

|  |  |
| --- | --- |
| Objective | Opening the application |
| Action | The application was run through terminal command. |
| Expected Result | The application should load properly. |
| Actual Result | The application loaded properly. |
| Conclusion | The test was successful |

### 5.3.3 Performance Testing

In the context of an integrated system, it is done to verify the software's run-time performance. Throughout the testing process, these tests are run. For instance, unit testing uses white box testing to access the performance of each individual module.

Table . Testing for moderate users

|  |  |
| --- | --- |
| Objective | Measure system performance under normal conditions. |
| Action | Simulate a moderate number of concurrent users interacting with the system |
| Expected Result | System should handle the load with acceptable response times and accuracy. |
| Actual Result | Evaluate response times and accuracy. |
| Conclusion | The test was successful. |

Table . Testing for peak users

|  |  |
| --- | --- |
| Objective | Measure system performance under peak conditions. |
| Action | Simulate a high number of concurrent users interacting with the system |
| Expected Result | Identify the point of system saturation and measure response times. |
| Actual Result | Evaluate response times and accuracy. |
| Conclusion | The test was successful. |

# : Conclusion and Future Recommendations

## 6.1 Conclusion

In conclusion, our project has shown that technology has become so advanced that we can even digitally know about our health problems and even get appointment of doctor using Internet and a computer instead of going to hospitals physically. We all know how much of a headache it is to visit a hospital during illness and our website will provide them the doctor assistance digitally. So Medcare is efficient and will surely unburden the patient stack on the hospital.

## 6.2 Future Improvement

The system is still not completed and there can be many improvements made in order to increase efficiency and user experience.

* Improved user interfaces: Improved user interfaces can make it easier for users to navigate the system and access the information they need.
* Doctor Panel: There is still no Doctor’s panel in the given system to accept the appointment of the users. We will add it in further version.
* Improved data collection: Improved data collection and analysis can enable our website to predict more diseases with more accuracy.
* Add video appointments: We will also add Video Conferencing feature for immediate appointment with the doctor in emergency conditions.

# References

|  |  |
| --- | --- |
| [1] | G. Litjens, "Medical Image Analysis," December 2017. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S1361841517301135. |
| [2] | M. A. M. Hlaudi Daniel Masethe, "Prediction Of Heart Disease Using Classification Algorithms," 2020. [Online]. Available: https://www.semanticscholar.org/paper/Prediction-Of-Heart-Disease-Using-Classification-Masethe-Masethe/d436cf077581d4411f4f8e88b1fb781a230641bc. |
| [3] | K. W. J. A. Y. R. M. L. I. L. D. R. J. J. P. K. P. P. S. S. G. A. M. B. D. D. L. D. A. K. N. P. T. Khader Shameer 1, "PREDICTIVE MODELING OF HOSPITAL READMISSION RATES USING ELECTRONIC MEDICAL RECORD-WIDE MACHINE LEARNING: A CASE-STUDY USING MOUNT SINAI HEART FAILURE," [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/27896982/. |
| [4] | M. Chen, "Disease Prediction by Machine Learning Over Big Data From Healthcare Communities," 26 April 2017. [Online]. Available: https://ieeexplore.ieee.org/document/7912315/authors#authors. |
| [5] | E. J. G. Lamido Yahaya, "A Comprehensive Review on Heart Disease Prediction Using Data Mining and Machine Learning Techniques," 23 April 2020. |
| [6] | M. Victor J. Dzau, "ANNUAL Medicine Report 2015," 2015. |
| [7] | M. V. ,. P. C. ,. G. S. Vaishali Baviskar, "Efficient Heart Disease Prediction Using Hybrid Deep Learning Classification Models," 2023. |

# Screenshots

