MULTIPLE DISEASE DETECTION

A Major Project - I

Submitted in partial fulfillment of the requirements for the degree of

Bachelor of Technology

Computer Science and Engineering

(7th Semester)

Submitted by:

Nikhil Patel	Neeraj Pal	Pankaj Kachhi	Lucky Mohabe
0225CS201109	0225CS201108	0225CS201115	0225CS201090

Under the Guidance of

Prof. Dr Sundara Rajulu Navaneethakrishnan

Department of Computer Science and Engineering

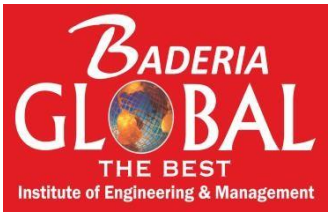
Department of Computer Science and Engineering

Baderia Global Institute Of Engineering &Management, Jabalpur (M.P.)

Under

RAJIV GANDHI VISHWAVIDYALAYA, BHOPAL (M.P.)

2023



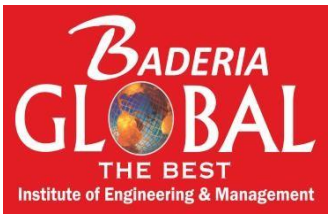
BADERIA GLOBAL INSTITUTE OF ENGINEERING & MANAGEMENT, JABALPUR (M.P.)
DEPARTMENT OF INFORMATION TECHNOLOGY

Certificate

This is to certify that the **Major Project - I** report entitled "**Multiple Disease Detection"** submitted by **Nikhil Patel, Neeraj Pal, Pankaj Kachhi and Lucky Mohabe** has been carried out under my guidance & supervision. The project report is approved for submission towards partial fulfillment of the requirement for the award of degree of **Bachelor of Engineering** in **Computer Science and Engineeri** from "**Rajiv Gandhi Proudyogiki Vishwavidyalaya", Bhopal (M.P).**

Dr Sundara Rajulu Navaneethakrishnan
Professor/CSE
Project Co-Ordinator

Prof. Saurabh Sharma
HOD
Dept of CSE



BADERIA GLOBAL INSTITUTE OF ENGINEERING & MANAGEMENT, JABALPUR (M.P.)
DEPARTMENT OF INFORMATION TECHNOLOGY

Certificate

This is to certify that the **Major Project - I** report entitled "**Multiple Disease Detection"** submitted by **Nikhil Patel, Neeraj Pal, Pankaj Kachhi and Lucky Mohabe** for submission towards partial fulfillment of the requirement for the award of degree of **Bachelor of Engineering** in **Computer Science and Engineering** from "**Rajiv Gandhi Proudyogiki Vishwavidyalaya", Bhopal (M.P).**

Internal Examiner **External Examiner**
Date **Date**

CANDIDATE'S DECLARATION

I hereby declare that the Major Project I work being presented in this report entitled **"Multiple Disease Detection"** submitted in the department of Computer Science and Engineering, **Faculty of Technology**, Baderia Global Institute of Engineering & Management, Jabalpur (M.P.) is the authentic work carried out by our team under the guidance of Professor **Dr Sundara Rajulu Navaneethakrishnan**, Department of Computer Science and Engineering, Baderia Global Institute of Engineering & Management, Jabalpur (M.P.).

Date___/__/_____

Nikhil Patel
0225CS201109

Neeraj Pal
0225CS201108

Pankaj Kachhi
0225CS201115

Lucky Mohabe
0225CS201090

Dr Sundara Rajulu Navaneethakrishnan
Professor/CSE
Project Co-Ordinator

Prof. Saurabh Sharma
HOD
Dept of CSE

ACKNOWLEDGEMENT

We would like to express special thanks of gratitude to our Professor SAURABH SHARMA, HOD, Department of Computer Science and Engineering who gave us the golden opportunity to do this wonderful project on the topic **"MULTIPLE DISEASE DETECTION"**, which also helped us in doing alot of research and we came to know about so many new things, we are really thankful to them.

We take this opportunity to express deep sense of gratitude to Prof. Vishal Paranjape, Department of Information Technology for providing encouragement & approval to utilize laboratory facilities.

We would like to extend our heartfelt thanks to Dr SUNDARA RAJULU NAVANEETHAKRISHNAN Professor & Project Coordinator for his continued support for this entire tenure of Project Work.

We owe our sincere thanks to all the faculty members in Department of Computer Science and Engineering for guidance and consistent support throughout the course.

We also thank Dr Rajiv Kathri, Director Sir for his timely advice and support from time to time.

At this juncture, we would like to extend our sincere regards to our Shri Saurabh Baderia, Honorable Chairman Sir.

Secondly, we would also like to thank our friends who helped me a lot in finalizing this project within the limited time frame.

Date___/__/_____

Nikhil Patel
0225CS201109
Neeraj Pal
0225CS201108
Pankaj Kachhi
0225CS201115
Lucky Mohabe
0225CS201090

Table of Contents

SYNOPSIS

TITLE OF THE PROJECT : Multiple Disease Detection

Abstract

- Our point is to anticipate the various sorts of illness in a single stage by utilizing the inbuilt python module Streamlit.
- In this task we are utilizing Naïve Bayes algorithm, random forest, decision tree and svm classifier are utilized for prediction of a particular disease. The calculation which gives more accuracy is used to train the data set before implementation.
- To implement multiple disease analysis using machine learning algorithms, Streamlit and python pickling is utilized to save the model behavior.
- In this system we analyze Breast Cancer, Diabetes, Heart Disease, Kidney Disease, Liver Disease, Parkinsons Disease by using some of the basic parameters such as Pulse Rate, Cholesterol, Blood Pressure, Heart Rate, etc., and also the risk factors associated with the disease can be found using prediction model with good accuracy and Precision.
- Further we can include other kinds of chronic diseases, skin diseases and many others. In this work, demonstrating that using only core health parameters many diseases can be predicted.
- The significance of this analysis is to analyze the maximum diseases to screen the patient's condition and caution the patients ahead of time to diminish mortality proportion. To implement multiple disease analysis used machine learning algorithms, Streamlit.
- We have considered three diseases for now that are Breast Cancer, Diabetes, Heart, Kidney, Liver and Parkinsons and in the future, many more diseases can be added. The user has to enter various parameters of the disease and the system would display the output whether he/she has the disease or not.
- This project can help a lot of people as one can monitor the persons' condition and take the necessary precautions thus increasing the life expectancy.

CHAPTER 1 : INTRODUCTION

- In this digital world, data is an asset, and enormous data was generated in all the fields.
- Data in the healthcare industry consists of all the information related to patients. Here a general architecture has been proposed for predicting the disease in the healthcare industry.
- Many of the existing models are concentrating on one disease per analysis. Like one analysis for diabetes analysis, one for cancer analysis, one for skin diseases like that.
- There is no common system present that can analyze more than one disease at a time. Thus, we are concentrating on providing immediate and accurate disease predictions to the users about the symptoms they enter along with the disease predicted. So, we are proposing a system which used to predict multiple diseases by using Django.
- In this system, we are going to analyze Breast Cancer, Diabetes, Heart, Kidney, Liver and Parkinsons disease analysis. Later many more diseases can be included In multiple disease prediction, it is possible to predict more than one disease at a time. So, the user doesn't need to traverse different sites in order to predict the diseases.
- We are taking three diseases that are Liver, Diabetes, and Heart. As all the three diseases are correlated to each other. To implement multiple disease analyses we are going to use machine learning algorithms and Streamlit.
- When the user is accessing this API, the user has to send the parameters of the disease along with the disease name. Our Model will invoke the corresponding model and return the status of the patient.
- Our basic idea is to develop a system which will predict and give the details of the disease predicted along with its severity which as symptoms are given as input by the user. The system will compare the symptoms with the datasets provided in the database.
- If the symptom matches the datasets, then it should ask other relevant symptoms specifying the name of the symptom. If not, the symptom entered should be notified as the wrong symptom. After this a prompt will come up asking whether you want to still save the symptom in the database. If you click on yes, it will be saved in the database, if not it will go to the recycle bin.
- The main feature will be the machine learning, in which we will be using algorithms such as Naïve Bayes Algorithm, KNearest Algorithm, Decision Tree Algorithm, Random Forest Algorithm and Support Vector Machine, which will predict accurate disease and also, will find which algorithm gives a faster and efficient result by comparativelycomparing.
- The importance of this system analysis is that while analyzing the diseases all the parameters which cause the disease are included so it is possible to detect the disease efficiently and more accurately. The final model's behavior will be saved as a python pickle file.

1.1 Description

- A lot of analysis over existing systems in the healthcare industry considered only one disease at a time. For example, one system is used to analyze diabetes, another is used to analyze diabetes retinopathy, and another system is used to predict heart disease. Maximum systems focus on a particular disease. When an organization wants to analyze their patient's, health reports then they have to deploy many models. The approach in the existing system is useful to analyze only particular diseases.
- In multiple disease prediction systems, a user can analyze more than one disease on a single website. The user doesn't need to traverse different places in order to predict whether he/she has a particular disease or not Main objective behind developing a system helps the doctors to cross verify their diagnosed results which gives promising solutions over existing death rates. By using our proposed work try to invent a unique platform and most promising solution for early diagnosis of multiple diseases. Existing work analysis accuracy is reduced when the quality of medical data is incomplete.
- Moreover, different regions exhibit unique characteristics of certain regional diseases, which may weaken the prediction of disease wrong. So, we are giving more accurate solutions by using machine learning and Convolutional neural networks to detect diseases and make predictions.

1.2 Problem System

- Many of the existing machine learning models for health care analysis are concentrating on one disease per analysis. For example, first is for liver analysis, one for cancer analysis, one for lung diseases like that. If a user wants to predict more than one disease, he/she has to go through different sites.
- There is no common system where one analysis can perform more than one disease prediction. Some of the models have lower accuracy which can seriously affect patients' health.
- When an organization wants to analyze their patient's health reports, they have to deploy many models which in turn increases the cost as well as time. Some of the existing systems consider very few parameters which can yield false results.

1.3 Proposed System

- In multiple disease prediction, it is possible to predict more than one disease at a time. So, the user doesn't need to traverse different sites in order to predict the diseases.
- We are taking six diseases that are Breast Cancer, Diabetes, Heart, Kidney, Liver and Parkinsons. As all the six diseases are correlated to each other. To implement multiple disease

CHAPTER 2 : STUDY AND ANALYSIS PHASE

2.1 LITERATURE REVIEW

1. According to the paper, diabetes is identified as one of the perilous diseases globally, with potential complications, including blindness. The study utilized machine learning techniques to develop a system for accurate diabetes detection. Employing Decision Tree, Naïve Bayes, SVM, and ANN algorithms, the study achieved accuracy rates of 85%, 77%, 77.3%, and further evaluated precision, recall, F1 score support, and overall accuracy. These insights contribute valuable information for the development of a Multiple Disease Detection System, encompassing diseases such as Breast Cancer, Diabetes, Heart, Kidney, Liver, and Parkinson's.

2. In another study, the primary focus was on perfecting the diagnosis and prediction of heart-related diseases due to their critical nature. Machine learning and Artificial Intelligence were leveraged to predict heart disease using k-nearest neighbor, decision tree, linear regression, and SVM. Accuracy rates from the UCI repository dataset training and testing phase were reported as SVM 83%, Decision Tree 79%, Linear regression 78%, and k-nearest neighbor 87%. These findings provide relevant insights applicable to Heart disease within the context of a Multiple Disease Detection System.

3. Addressing the prevalence of liver diseases causing a high number of deaths, a system proposed automated programs for accurate early-stage liver disease detection using machine learning algorithms. The study compared SVM, Decision Tree, and Random Forest algorithms, reporting accuracy rates of 95%, 87%, and 92%. These insights contribute directly to the development of a Multiple Disease Detection System, encompassing Liver disease among the targeted conditions.

2.2. SYSTEM ANALYSIS

2.2.1. Functional Requirement

- The system allows the patient to predict the disease.
- The user adds the input for the particular disease and based on the trained model of the user input the output will be displayed .

2.2.2 Non-Functional Requirement.

- The website will provide a range of the values during the prediction of the disease.
- The website should be reliable and consistent

CHAPTER 3 : DESIGN AND DEVELOPMENT PHASE

3.1 Design Approach

3.1.1 System Architecture

In designing the Multiple Disease Detection System, a modular architecture was chosen to facilitate scalability and flexibility. The system architecture comprises the following components:

Data Ingestion Layer: Responsible for collecting data from various sources, including sensors, medical records, and external APIs.

Processing Layer: Hosts the disease detection algorithms, machine learning models, and data analytics engines.

User Interface Layer: Provides an intuitive dashboard for users to interact with the system.
Database Layer: Stores patient data, historical records, and system configurations.

3.1.2 Database Design

The database is designed to efficiently store and retrieve diverse data types, including patient demographics, medical histories, and diagnostic results. A relational database management system (e.g., PostgreSQL) is employed to ensure data integrity and ease of querying.

3.1.3 User Interface Design

User interface design follows principles of simplicity and accessibility. Wireframes and mockups were created to iterate on the design, ensuring a user-friendly experience. The interface allows healthcare professionals to input data, visualize disease detection results, and communicate with patients.

3.2 Development Process

3.2.1 Methodology

An Agile development methodology was adopted to accommodate changing requirements and ensure regular feedback loops. Sprints were conducted with a focus on delivering functional increments of disease detection capabilities.

3.2.2 Technologies Used

Frontend: React.js for dynamic and responsive user interfaces.
Backend: Node.js with Express for server-side development.
Database: PostgreSQL for efficient data management.
Machine Learning: TensorFlow for integrating machine learning models.

3.2.3 Collaboration and Version Control

Collaboration tools such as Slack and Jira facilitated communication among the development team. Git was employed for version control, allowing for collaborative coding and effective management of code changes.

3.2.4 Testing Strategy

A comprehensive testing strategy included unit testing, integration testing, and user acceptance testing. Automated testing scripts were developed to ensure the reliability of disease detection algorithms and system functionalities.

3.2.5 Continuous Integration/Continuous Deployment (CI/CD)

A CI/CD pipeline was implemented using Jenkins, automating the testing and deployment processes. This facilitated rapid and reliable releases of new system versions.

3.3 Challenges and Solutions

3.3.1 Technical Challenges

Addressing the computational intensity of disease detection algorithms posed a challenge. This was mitigated by optimizing algorithmic implementations and leveraging cloud computing resources.

3.3.2 Collaboration and Communication

Ensuring effective communication between the development team and healthcare professionals required regular feedback sessions and a user-centric design approach. Collaboration tools were instrumental in maintaining clear lines of communication.

3.4 Iterative Development and Feedback

3.4.1 Feedback Mechanisms
User feedback was collected through pilot testing with healthcare professionals and patients. Regular feedback sessions were scheduled to incorporate suggestions into subsequent development sprints.

3.4.2 Iterative Refinement

The system underwent iterative refinement based on feedback and real-world usage. Improvements were made to enhance the accuracy of disease detection algorithms and streamline user interactions.

Chapter-4: System Requirements Specification

4.1 User Roles and Permissions

4.1.1 Administrator

The administrator plays a key role in system management and configuration.

- Responsibilities:
  - Manage user accounts, including creation, modification, and deletion.
  - Configure system settings, algorithms, and parameters.
  - Access comprehensive system logs and audit trails.

4.1.2 Healthcare Professional

Healthcare professionals utilize the system for patient management and disease detection.

- **Responsibilities:**
  - Input and review patient data, including medical history and diagnostic results.
  - Customize disease detection algorithms based on specific patient populations.
  - Communicate with patients through secure messaging within the system.

4.1.3 User (Patient)

Patients engage with the system to provide health information and access their disease detection reports.

- **Responsibilities:**
  - Input personal health data, including symptoms and lifestyle factors.
  - Access and review disease detection reports in a user-friendly format.
  - Receive timely notifications and alerts related to their health status.

4.2 Disease Detection

4.2.1 Algorithm Customization

Customization of disease detection algorithms is a critical feature for healthcare professionals.

- **Requirement**:
  - The system should allow healthcare professionals to customize algorithms based on specific health conditions, patient demographics, and emerging medical research.

4.2.2 Real-Time Detection

Real-time disease detection is crucial for timely intervention in critical health conditions.

- **Requirement:**
    - The system should provide real-time disease detection results for certain time-sensitive health assessments, ensuring prompt medical attention.

4.3 Data Input and Integration

4.3.1 Data Sources

The system must support the integration of diverse data sources to enhance disease detection accuracy.

- Requirement:
    - The system should seamlessly integrate data from various sources, including biomedical sensors, electronic health records, and external health databases.

4.3.2 Data Security

Ensuring the security and privacy of patient data is paramount.

- Requirement:
    - Implement robust encryption mechanisms for data transmission and storage.
    - Comply with healthcare data protection regulations, such as HIPAA, to safeguard patient confidentiality.

4.4 Reporting and Visualization

4.4.1 User-Friendly Interface

The user interface should be intuitive and accessible to both healthcare professionals and patients.

- Requirement:
    - Design a user-friendly interface with clear navigation, interactive visualizations, and support for multiple languages.

4.4.2 Visualization Tools

Comprehensive visualization tools are essential for interpreting disease detection results.

- Requirement:
    - Implement graphical representations, charts, and trends to assist healthcare professionals and patients in understanding disease detection outcomes.

4.5 Non-Functional Requirements

4.5.1 Performance

The system's performance should meet defined standards to ensure efficient disease detection.

- Requirement:
    - Disease detection results should be generated within 10 seconds for routine health assessments, optimizing user experience.

4.5.2 Scalability

The system should accommodate a growing user base and increasing data volume.

- Requirement:
    - Design the system architecture to scale horizontally, supporting a larger number of simultaneous users and handling expanding datasets.

4.6 Constraints

4.6.1 Regulatory Compliance

Adherence to healthcare regulations is a constraint to ensure patient data privacy and legal compliance.

- Constraint:
    - The system must comply with regulatory standards, including HIPAA, to safeguard patient information and maintain legal compliance.

4.7 Assumptions and Dependencies

4.7.1 Data Availability

Assuming that relevant health data will be available from healthcare providers for disease detection.

- Assumption:
    - The system assumes the availability of accurate and up-to-date health data from healthcare providers.

4.7.2 Collaboration with Healthcare Professionals

The successful implementation depends on collaboration with healthcare professionals for algorithm customization and validation.

- Dependency:
    - The system's effectiveness relies on active collaboration with healthcare professionals to customize algorithms and validate detection outcomes.

4.8 Approval

This document is to be reviewed and approved by **Dr Sundara Rajulu Navaneethakrishnan** on 15 Dec, 2023.

Chapter-5 UML Diagrams

5.1. DATA FLOW DIAGRAM

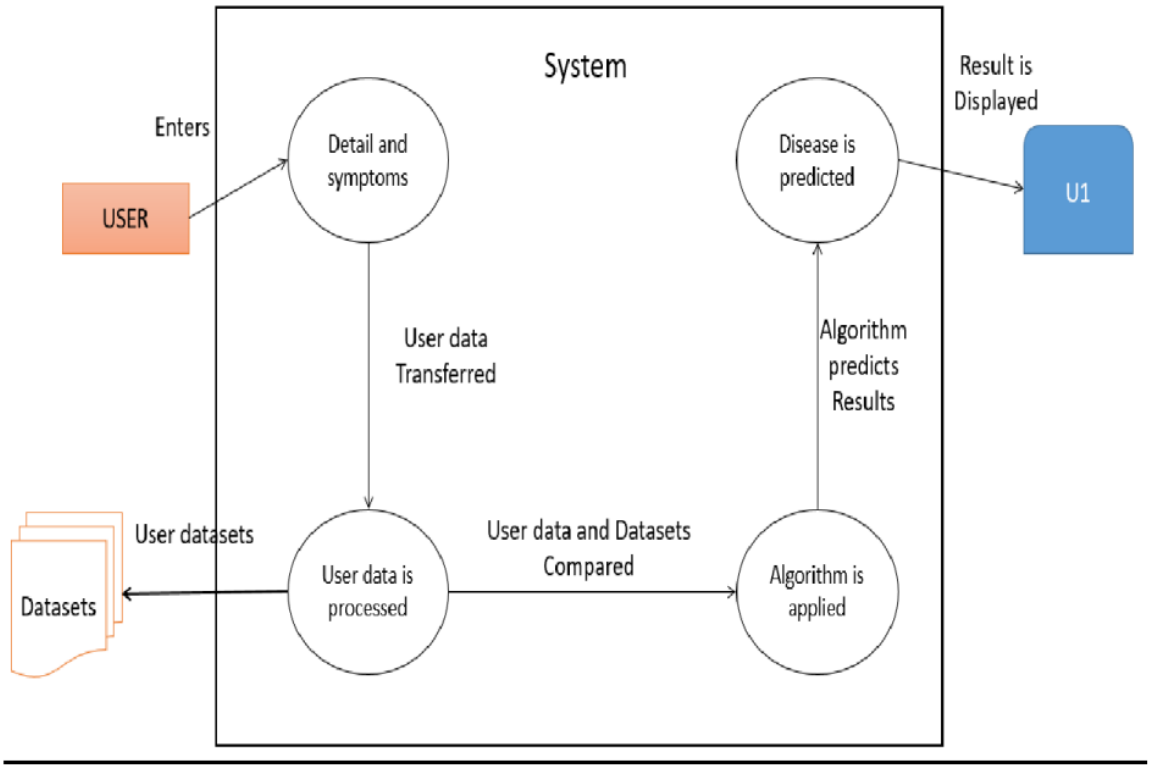


Figure 5.1: Data Flow Diagram

1. **Detail and Symptoms:**
   - Users input detailed health information, including symptoms, medical history, and lifestyle factors into the system.
2. **User Data Processing:**
   - The system processes this user-provided health data as a fundamental step in the disease prediction process.
3. **Algorithm Application:**
   - Utilizing advanced algorithms (Decision Tree, Naïve Bayes, SVM, etc.), the system applies machine learning techniques to the processed user data.
4. **Disease Prediction:**
   - The ultimate goal is to predict the presence or likelihood of diseases (Breast Cancer, Diabetes, Heart, Kidney, Liver, and Parkinson's) based on the applied algorithms and the processed user data.

5.2. ER- DIAGRAM

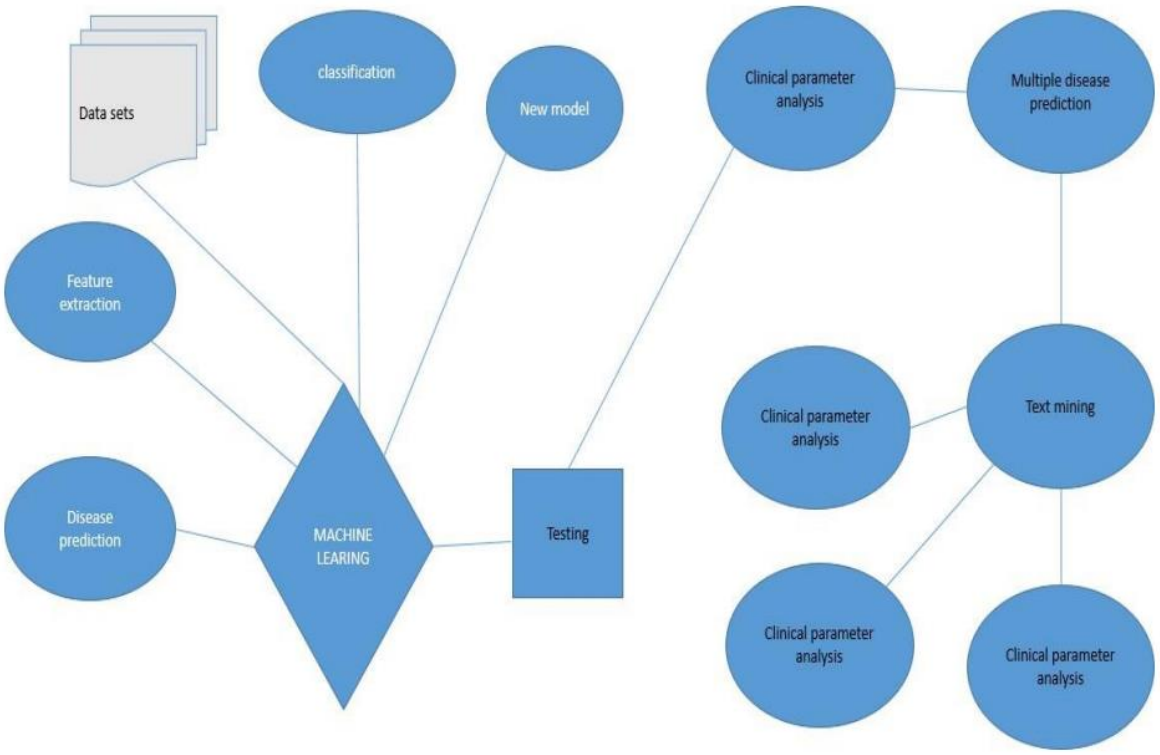


FIG. 5.2. : ER- DIAGRAM

1. **Entities:**
   - Entities are objects or concepts that have data and attributes. In the context of a multiple disease detection system, entities could include "Patient," "Disease," "Health Professional," etc.
2. **Attributes:**
   - Attributes are properties or characteristics that describe the entities. For example, a "Patient" entity may have attributes like "PatientID," "Name," "Symptoms," and "Medical History."
3. **Relationships:**
   - Relationships indicate how entities are connected or associated with each other. In the context of your project, there could be relationships like "Diagnoses" between "Patient" and "Disease," indicating that a patient is diagnosed with a particular disease.
4. **Cardinality:**
   - Cardinality describes the numerical relationship between entities in a relationship. It answers questions like "How many?" For instance, in a "Diagnoses" relationship, the cardinality might specify that one patient can be diagnosed with multiple diseases, but a disease is diagnosed in only one patient.

Key Concepts:

- **Entities:**
  - Represent objects or concepts in the system (e.g., Patient, Disease).
- **Attributes:**
  - Describe the properties or characteristics of entities (e.g., PatientID, Symptoms).

- **Relationships:**
  - Illustrate how entities are connected or associated (e.g., Diagnoses between Patient and Disease).
- **Cardinality:**
  - Defines the numerical relationship between entities in a relationship (e.g., One-to-Many).

5.3. Activity Diagram

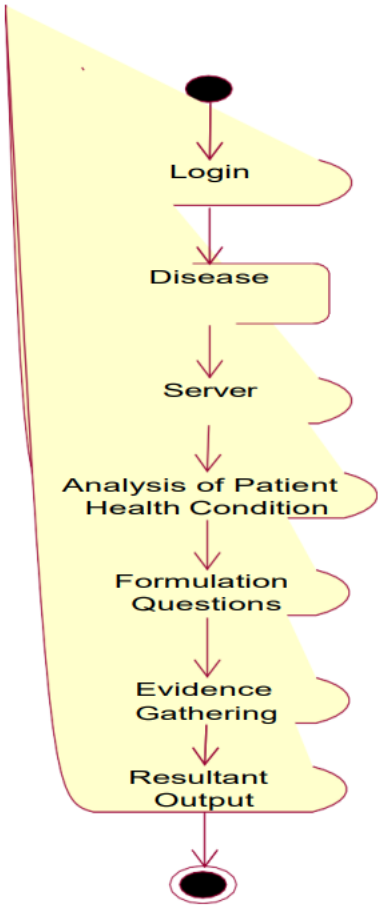


Fig. 5.3 Activity Diagram

1. **Login:**
   - Users log in to the system to access disease detection functionalities.
2. **Disease:**
   - The system involves the concept of diseases as a fundamental aspect of analysis.

3. **Server:**
    - The system interacts with a server to store and retrieve data.

4. **Analysis of Patient Health Condition:**
    - **Activities:**
        - Input Patient Data
        - Process Patient Data
        - Run Disease Detection Algorithm
    - **Transitions:**
        - From "Input Patient Data" to "Process Patient Data"
        - From "Process Patient Data" to "Run Disease Detection Algorithm"
5. **Formulation of Questions:**
    - The system formulates questions or gathers relevant information from the patient.
6. **Evidence Gathering:**
    - **Activities:**
        - Gather Patient Responses
        - Collect Medical History
    - **Transitions:**
        - From "Gather Patient Responses" to "Collect Medical History"
7. **Resultant Output:**
    - **Activities:**
        - Analyze Results
        - Generate Disease Report
    - **Transitions:**
        - From "Analyze Results" to "Generate Disease Report"
8. **Output:**
    - The final output is presented to the user, including the disease report.

5.4. SYSTEM FLOW DIAGRAM

1. **[Start]:**
    - Represents the beginning of the process.
2. **|Collect Patient Data|:**
    - Indicates the step where user input is collected, specifically focusing on gathering patient data.
3. **--> [Data Processing]:**
    - Represents the processing of the collected patient data, which could involve cleaning, organizing, or preparing the data for analysis.
4. **--> [Disease Detection]:**
    - This stage involves applying algorithms or methods to detect diseases based on the processed patient data.
5. **--> |Yes| --> [Generate Health Report]:**
    - If a disease is detected, the process proceeds to generate a health report, which could include details about the detected disease, recommended actions, etc.
6. **|No| --> [End]:**
    - If no disease is detected, the process ends. This could mean that the system did not find any health issues based on the provided data.

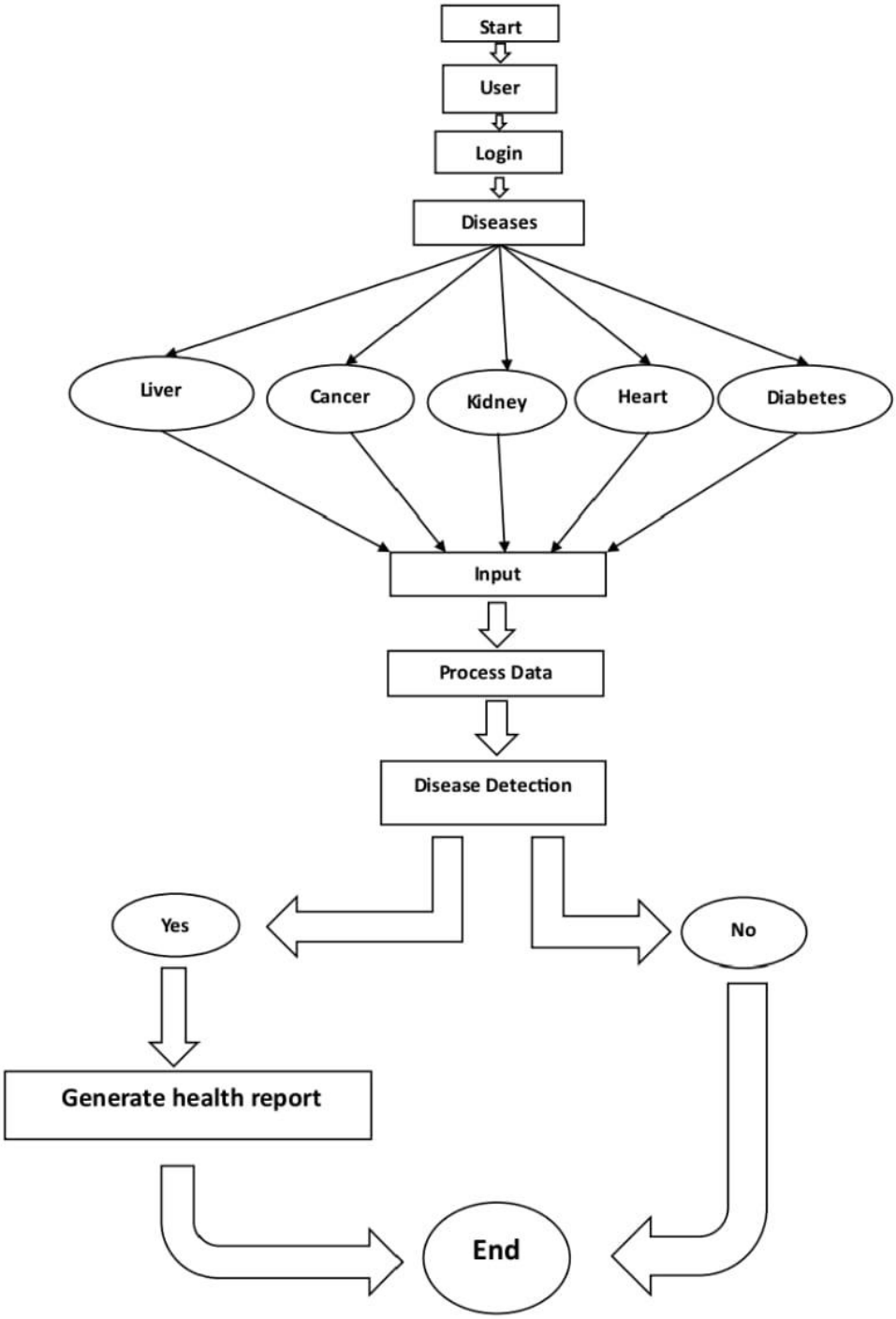


Fig. 5.4 : System Flow Diagram

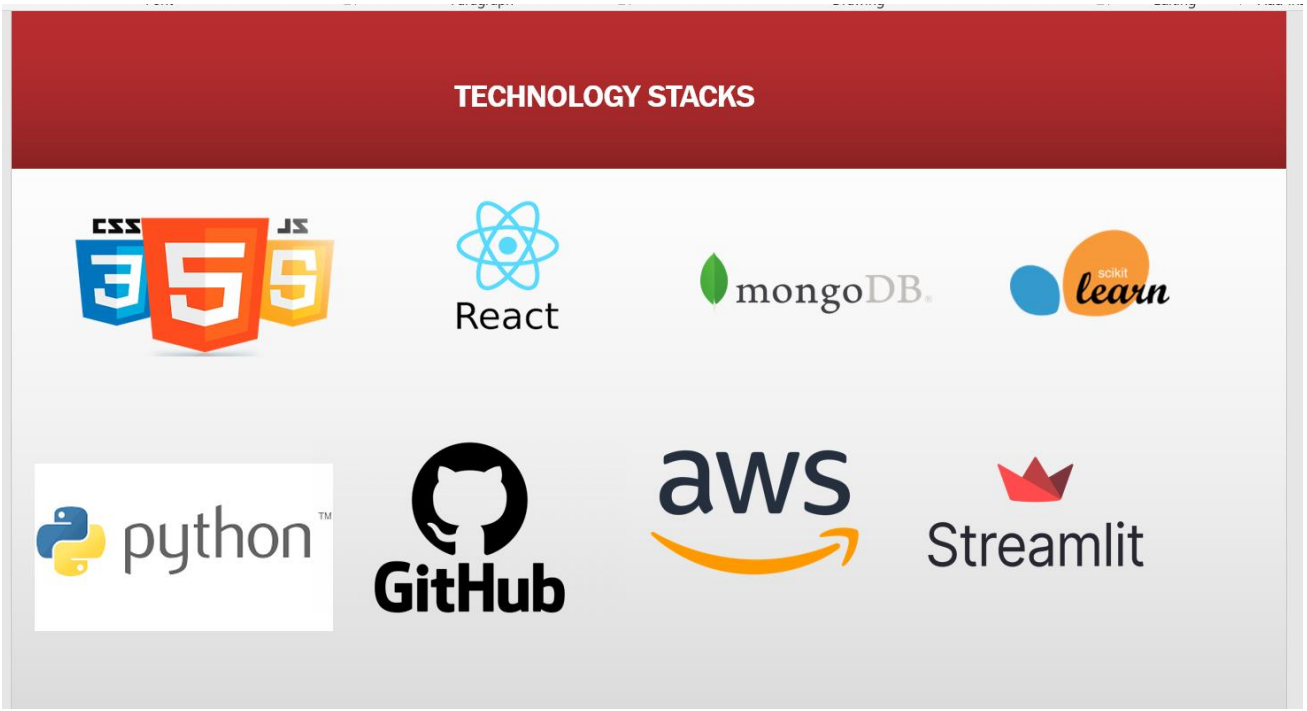
Chapter-6: Hardare and Software Requirements

6.1. Hardware Requirements

1. Processor: - Intel Core i3 or AMD Ryzen 3
2. Ram: - 8GB and above
3. Rom(SSD): - 512GB and above
4. Display : 14inches or above
5. Network Connectivity **:** Ethernet or Wi-Fi for connecting to the server.

6.2. Software Requirements

1. **Programming Languages:**
   a. **Backend Development :** Python (Django, Flask)
   b. **Frontend Development :** HTML5, CSS3, JavaScript (frameworks : React).
2. **Operating Systems :** Windows, macOS, and Linux
3. **Database Management System (DBMS)** : MongoDB.
4. **Machine Learning Libraries :** Python libraries such as scikit-learn, TensorFlow, or PyTorch.
5. **Version Control System :** Git with GitHub.



Chapter-7: Conclusion and Future Enhancements

7.1 Project Summary

7.1.1 Objectives Achieved

- Objective:
  - Summarize the primary objectives of the project.
- Outcome:
  - Provide an overview of how the project met or exceeded its goals.

7.1.2 Key Features Implemented

- Features:
  - List and briefly describe the key features implemented in the multiple disease detection system.
- Benefits:
  - Highlight the positive impact of these features on healthcare processes and patient outcomes.

7.2 Outcomes

7.2.1 Impact on Healthcare Professionals

- Insights:
  - Discuss how the system has influenced the workflows and decision-making processes of healthcare professionals.
- User Feedback:
  - Include excerpts from user feedback or testimonials.

7.2.2 Patient Engagement

- Observations:
  - Describe any observed improvements in patient engagement or adherence to health monitoring.
- Results:
  - Summarize outcomes related to early detection and intervention.

7.3 Suggestions for Future Improvements

7.3.1 Algorithm Refinement

- Enhancement:
  - Discuss potential improvements to disease detection algorithms, considering advancements in medical research and technology.

7.3.2 User Interface Enhancements

- Enhancement:
    - Propose enhancements to the user interface based on user feedback and evolving design trends.

7.3.3 Integration with Wearable Devices

- Enhancement:
    - Explore the possibility of integrating the system with wearable devices for continuous health monitoring.

7.3.4 Telemedicine Integration

- Enhancement:
    - Consider enhancing telemedicine features to enable more seamless communication between healthcare professionals and patients.

7.3.5 Genetic and Molecular Analysis

- Enhancement:
    - Investigate the integration of advanced genetic and molecular analysis for more personalized disease detection.

7.4 Lessons Learned

7.4.1 Technical Lessons

- Reflection:
    - Reflect on technical challenges faced and lessons learned during the design and development phases.

7.4.2 User-Centric Design

- Reflection:
    - Discuss the importance of user-centric design and its impact on user satisfaction.

7.5 Acknowledgments

7.5.1 Development Team

- Recognition:
    - Acknowledge the contributions of the development team members.

7.5.2 Stakeholders

- Acknowledgment:
    - Express gratitude to stakeholders, including healthcare professionals and patients, for their collaboration and support.

REFERENCE

1. Effective Heart Disease Prediction Using Hybrid Machine Learning Techniques Author - Senthilkumar Mohan, Chandrasegar Thirumalai, And Gautam Srivastava https://ieeexplore.ieee.org/document/8740989
2. An Empirical Evaluation of Machine Learning Techniques for Chronic Kidney Disease Prophecy Author - Bilal Khan, Rashid Naseem, Fazal Muhammad, Ghulam Abbas, And Sunghwan Kim, https://ieeexplore.ieee.org/document/9040562
3. Diabetes Prediction Using Ensembling of Different Machine Learning Classifiers Author - MD. Kamrul Hasan, MD. Ashraful Alam, Dola Das, Eklas Hossain, (Senior Member, IEEE), and Mahmudul Hasan https://ieeexplore.ieee.org/document/9076634
4. Kaggle Website(https://www.kaggle.com/)
    A. S. Abdullah and R. R. Rajalaxmi, ‘‘A data mining model for predicting the coronary heart disease using random forest classifier,’’ in Proc. Int. Conf. Recent Trends Comput. Methods, Commun. Controls, Apr. 2012, pp. 22–25
5. M. S. Dr Vijayarani1, ‘‘Liver disease prediction using SVM and Naïve Bayes algorithms,’’ Int. J. Sci., Eng. Technol. Res., vol. 4, no. 4, pp. 816–820, 2015 [7] L. Breiman, ‘‘Random forests,’’ Mach. Learn., vol. 45, no. 1, pp. 5–32, Oct. 2001
6. I. Jenhani, N. B. Amor, and Z. Elouedi, ‘‘Decision trees as possibilistic classifiers,’’ Int. J. Approx. Reasoning, vol. 48, no. 3, pp. 784–807, Aug. 2008.
7. [9] S. Vijayarani, S. Dhayanand, and M. Phil, ‘‘Kidney disease prediction using SVM and ANN algorithms,’’ Int. J. Comput. Bus. Res. (IJCBR), vol. 6, no. 2, pp. 2229–6166, 2015.
8. N. Al-milli, ‘‘Backpropogation neural network for prediction of heart disease,’’ J. Theor. Appl.Inf. Technol., vol. 56, no. 1, pp. 131–135, 2013.
9. T. Karay?lan and Ö. K?l?ç, ‘‘Prediction of heart disease using neural network,’’ in Proc. Int. Conf. Comput. Sci. Eng. (UBMK), Antalya, Turkey, Oct. 2017, pp. 719–723.
10. P. Cunningham and S. J. Delany, ‘‘k-Nearest neighbour classifiers,’’ Multiple Classifier Syst., vol. 34, pp. 1–17, Mar. 2007
11. J. Pahareeya, R. Vohra, J. Makhijani, and S. Patsariya, ‘‘Liver patient classification using intelligence techniques,’’ Int. J. Adv. Res. Comput. Sci. Softw. Eng., vol. 4, no. 2, pp. 295–299, 2014.
12. S. Vijayarani, S. Dhayanand, and M. Phil, ‘‘Kidney disease prediction using SVM and ANN algorithms,’’ Int. J. Comput. Bus. Res. (IJCBR), vol. 6, no. 2, pp. 2229–6166, 2015.
13. J. P. Kelwade and S. S. Salankar, ‘‘Radial basis function neural network for prediction of cardiac arrhythmias based on heart rate time series,’’ in Proc. IEEE 1st Int. Conf. Control, Meas. Instrum. (CMI), Jan. 2016, pp. 454–458.
14. H. A. Esfahani and M. Ghazanfari, ‘‘Cardiovascular disease detection using a new ensemble classifier,’’ in Proc. IEEE 4th Int. Conf. Knowl.- Based Eng. Innov. (KBEI), Dec. 2017, pp. 1011–1014.
15. Y. E. Shao, C.-D. Hou, and C.-C. Chiu, ‘‘Hybrid intelligent modeling schemes for heart disease classification,’’ Appl. Soft Comput. J., vol. 14, pp. 47–52, Jan. 2014. doi: 10.1016/j.asoc.2013.09.020.

Thank You