

A Machine Learning Approach for identifying plant diseases and provide cure

Submitted in partial fulfillment of the requirements for
the award of
Bachelor of Engineering degree in computer Science and Engineering

by
Tumula Nikhil (38110366)
P Vishnu Teja (38110390)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

MAY – 2022



SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Tumula Nikhil (38110366), P. Vishnu Teja (38110390)** who carried out the project entitled **"A Machine Learning Approach for identifying plant diseases and provide cure"** under my supervision from Nov 2021 to March 2022

Internal Guide

Dr. G. Kalaiarasi, M.E., Ph.D

Head of the Department

Dr.S.Vigneshwari M.E., Ph.D.,

Dr.L.Lakshmanan M.E., Ph.D.,

Submitted for Viva voice Examination held on _____

Internal Examiner

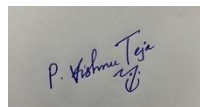
External Examiner

DECLARATION

I **P. Vishnu Teja(38110390)** hereby declare that the Project Report entitled A “**Machine Learning Approach for identifying plant diseases and provide cure**” done by me under the guidance of **Dr. G. Kalaiarasi ,M.E.,Ph.D** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:10-04-2022

PLACE: CHENNAI

A rectangular box containing a handwritten signature in blue ink. The signature appears to read 'P. Vishnu Teja' with a stylized flourish at the end.

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph.D, Dean**, School of Computing **Dr. L. Lakshmanan M.E., Ph.D. ,** and **Dr.S.Vigneshwari M.E., Ph.D. Heads** of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. G. Kalaierasi ,M.E.,Ph.D**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

To prevent losses in the yield one of the best ways is identification of plant diseases. In general, the studies of plant diseases observe visually the pattern on the plant. To get the accurate treatment for appropriate disease it is very hard to identify the patterns manually. This is a very time-consuming process and should require an expert with lot of experience. In this proposed application both the disease and treatment is being found by inputting various patterns which is physically observed on any kind of plant. For any plant there are 5 levels of disease occurrences: Stem Level, Leaves Level, Seed Level, Lessions Level, Plant Level. All the physical inputs are collected and gathered according to each level of plants and are substituted in the medical directory which has been created. Analysis results of the proposed method are written using Machine Learning approach.

	TABLE OF CONTENTS	
CHAPTER No.	TITLE	PAGE No
	ABSTRACT	i
	LIST OF FIGURES	ii
	LIST OF TABLES	iii
1	INTRODUCTION	1
	1.1 AIM OF THE PROJECT	2
	1.2 Methodology	4
	1.3 Significance of Work	4
	1.4 Report Organization	4
2	LITERATURE SURVEY	5
	2.1 Introduction	6
	2.2 Related Work	6
3	DESIGN ASPECTS	8
	3.1 Feasibility Study	9
	3.2 Existing System	10
	3.3 Proposed System	10
	3.4 System Design	11
	3.5 Data flow Diagram	13
	3.6 UML Diagram	15
	3.7 Database Design	22
4	IMPLEMENTATION ASPECTS	27
	4.1 Software Requirement Specification	28
	4.2 Functional Requirement	29
	4.3 Non Functional requirement	30
	4.4 Model Used	21
	4.5 Hardware Specification	34
	4.6 Software Specification	34
	4.7 Theoretical Background	34

	4.8 Software Project Modules	41
5	EXPERIMENTAL RESULTS	43
	5.1 Results	44
	5.2 Testing and test cases	52
	5.3 Testing for unit testing and Validation Testing	55
6	CONCLUSION AND FUTURE WORK	57
	REFERENCES	59
	APPENDICES	60
A	SOURCE CODE	60
B	SCREENSHOTS	70

List Of Figures

3.1	Steps for system design	12
3.2	DFD elements	14
3.3	Dataflow diagram for proposed application	15
3.6.1	Use case diagram	17
3.6.2	Class Diagram	18
3.6.3	Sequence Diagram	19
3.6.4	Collobration Diagram	19
3.6.5	Activity Diagram	20
3.6.6	Component Diagram	21
3.6.7	Deployment Diagram	22
4.1	Software Requirement Analysis	29
4.5	Spiral Model	32
4.7.4	Steps For Java	37

List Of Tables

3.7.3	Natural Disease Table	25
4.6	Hardware Specifications	34

CHAPTER - I

INTRODUCTION

1. INTRODUCTION

People care deeply about their health and want to be, now more than ever, in charge of their health and health care. Life is more hectic than has ever been, the medicine that is practiced today is an Evidence-Based Medicine (here after, EBM) in which medical expertise is not only based on years of practice but on the latest discoveries as well. Tools that can help us manage and better keep track of our health such as Google Health and Microsoft Health Vault are reasons and facts that make people more powerful when it comes to health care knowledge and management. The traditional health care system is also becoming one that embraces the Internet and the electronic world. Electronic Health Records (here after, EHR) are becoming the standard in the healthcare domain. Researches and studies show that the potential benefits of having an EHR system are

- Health information recording and clinical data repositories— immediate access to patient diagnoses, allergies, and lab test results that enable better and time-efficient medical decisions.
- Medication management—rapid access to information regarding potential adverse drug reactions, immunizations, supplies, etc.
- Decision support—the ability to capture and use quality medical data for decisions in the workflow of healthcare
- Obtain treatments that are tailored to specific health needs—rapid access to information that is focused on certain topics.
- First the symptoms provided by the user are processed by a expert system for identifying the diseases. If the rules required for processing the data by the above are not present in the database, then the system automatically calls the machine learning algorithm technique.

1.1 Aim Of The Project

In order to embrace the views that the EHR system has, we need better, faster, and more reliable access to information. In the medical domain, the richest and most used source of information is Medline, a database of extensive life science published articles. All research discoveries come and enter the repository at high rate, making the process of identifying and disseminating reliable information a very difficult task. The work that we present in this paper is focused on two tasks: automatically identifying sentences published in medical abstracts (Medline) as containing or not information about diseases and treatments, and automatically identifying semantic relations that exist between diseases and treatments, as expressed in these texts. The second task is focused on three semantic relations: Cure, Prevent, and Side Effect

The tasks that are addressed here are the foundation of an information technology framework that identifies and disseminates healthcare information. People want fast access to reliable information and in a manner that is suitable to their habits and workflow. Medical care related

Information is a source of power for both healthcare providers and laypeople. Studies reveal that people are searching the web and read medical related information in order to be informed about their health. Show how a new outbreak of the influenza virus can be detected from search engine query data. Our objective for this work is to show what Natural Language Processing (NLP) and Machine Learning (ML) techniques—what representation of information and what classification algorithms—are suitable to use for identifying and classifying relevant medical information in short texts.

We acknowledge the fact that tools capable of identifying reliable information in the medical domain stand as building blocks for a healthcare system that is up-to-date with the latest discoveries. In this research, we focus on diseases and treatment information, and the relation that exists between these two entities. Our interests are in line with the tendency of having a personalized medicine, one in which each patient has its medical care tailored to its needs. It is not enough to read and know only about one study that states that a treatment is beneficial for a certain disease. Healthcare providers

need to be up-to-date with all new discoveries about a certain treatment, in order to identify if it might have side effects for certain types of patients.

We envision the potential and value of the findings of our work as guidelines for the performance of a frame work that is capable to find relevant information about diseases and treatments in a medical domain repository. The results that we obtained show that it is a realistic scenario to use NLP and ML techniques to build a tool, similar to an RSS feed, capable to identify and disseminate textual information related to diseases and treatments. Therefore, this study is aimed at designing and examining various representation techniques in combination with various learning methods to identify and extract biomedical relations from literature.

1.2 Methodology

The present work deals with the concepts of Machine learning algorithms and the development of Web based online expert systems. An expert system follows the methodology of task-based specification and independent problem solving techniques. Here Machine learning technique is considered to find a good matching for the symptoms in the database. The new algorithm mainly focuses on the determination of the diseases affected to the Natural plants. First the symptoms provided by the user are processed by a rule based expert system for identifying the diseases. If the rules required for processing the data by the above are not present in the database, then the system automatically calls the machine learning algorithm technique. As a whole, the system results global solution for recognizing the diseases in Natural plants. And corresponding treatments to the diseases may also be suggested to the users.

1.3 Significance Of The Work

In proposed System, we try to implement ML Approach in data mining to identify the plant diseases. The proposed method tries to create a medical dictionary for the end users in order to predict and identify the diseases very accurately on plant.

1.4 Report Organization

In this report we try to organize the project into chapter wise and arranged as per software development life cycle. The chapter 1 contains the introduction about the project and in chapter 2 we try to discuss about the literature survey. In chapter 3 we try to arrange the design aspects which are discussed about this project. In chapter 4,we discuss about implementation aspects present in our application. In chapter 5,we discuss about experimental results and finally end with reports and conclusion and references as last topics.

CHAPTER-II

LITERATURE SURVEY

2. LITERATURE SURVEY

2.1 Introduction

Literature survey is the most important step in software development process. Before developing the tool, it is necessary to determine the time factor, economy and company strength. Once these things are satisfied, the next steps are to determine which operating system and language used for developing the tool. Once the programmers start building the tool, the programmers need a lot of external support. This support is obtained from senior programmers, from books or from websites. Before building the system, the above considerations are taken into account for developing the proposed system.

2.2 Related Work

In 2011, an innovative approach was presented [1] to automatically grade the disease on plant leaves. According to that, plant pathologists mainly rely on naked eye prediction and a disease scoring scale to grade the disease. That leads to some problems associated with manual grading. This manual grading is not only time consuming but also not feasible. Hence an image processing-based approach to automatically grade the disease spread on plant leaves by employing Fuzzy Logic had been proposed. The results are proved to be accurate and satisfactory in contrast with manual diseases which are inevitable.

in plants. The proposed methodology aims to model a promising disease grading system for plant leaves. The system was divided into the following steps:

- (1) Image acquisition
- (2) Image Pre-processing
- (3) Color image segmentation
- (4) Calculating AT and
- (5) Disease grading by Fuzzy Logic.

In 2014, an survey report was published[2], based on different classification techniques that could be used for plant leaf disease classification. A classification technique deals with classifying each pattern in one of the distinct classes. A classification is a technique where leaf is classified based on its different morphological features. There are so many classification techniques such as k-Nearest Neighbor Classifier, Probabilistic Neural Network, Genetic Algorithm, Support Vector Machine, and Principal Component Analysis, Artificial neural network, Fuzzy logic. Selecting a classification method is always a difficult task because the quality of result can vary for different input data. Plant leaf disease classifications have wide applications in various fields such as in biological research, in Agriculture etc. The paper provides an overview of different classification techniques used for plant leaf disease classification.

In 2012, an article was published[3] having a detailed description on definition of disease, types of diseases, symptoms and causes of most commonly observed plant diseases. One article was published by Michigan University[4] regarding the threats caused due to diseases. Various conditions for disease development had been discussed there. An overview of major disease-causing organisms and the effect of diseases caused by them was given.

Deep neural networks have recently been successfully applied in many diverse domains as examples of end to end learning. Neural networks provide a mapping between an input—such as an image of a diseased plant—to an output—such as a crop disease pair. The nodes in a neural network are mathematical functions that take numerical inputs

from the incoming edges, and provide a numerical output as an outgoing edge. Deep neural networks are simply mapping the input layer to the output layer over a series of stacked layers of nodes. The challenge is to create a deep network in such a way that both the structure of the network as well as the functions (nodes) and edge weights correctly map the input to the output. Deep neural networks are trained by tuning the network parameters in such a way that the mapping improves during the training process.

CHAPTER - III

DESIGN ASPECTS

PRESENT WORK

3.1 Feasibility Study

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are :

- ◆ ECONOMICAL FEASIBILITY
- ◆ TECHNICAL FEASIBILITY
- ◆ SOCIAL FEASIBILITY

3.1.1 Economical Feasibility:

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

3.1.2 Technical Feasibility:

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

3.1.3 Social Feasibility:

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it.

His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

3.2 Existing System

In Existing System, there is no concept like identifying the plant diseases based on ML Approach or using data mining algorithms. All the existing methods try to utilize manual method for identifying the plant disease and those disease identification may sometimes accurate and sometimes in efficient.

3.2.1 Limitation Of Existing System

The following are the limitation of the existing system :

- 1) All the existing methods try to use manual method to identify the disease and provide cure for that disease.
- 2) A lot of time is consumed in identifying and verifying the type of disease on the plant.
- 3) Outcome may be accurate or sometimes in accurate.
- 4) A lot of research knowledge should be their in order to judge the disease on that plant based on the symptoms observed on that plant.
- 5) There is no system to identify and predict the disease using ML approach in the existing system
- 6) All the existing methods try to exhaust a lot of time in identifying and provide cure for the infected plant.

3.3 Proposed System

In proposed System, we try to implement ML Approach in data mining to identify the plant diseases. The proposed method tries to create a medical dictionary for the end users in order to predict and identify the diseases very accurately on plant.

3.3.1 Advantages of the Proposed System

The following are the advantages of the proposed system :

- 1) The proposed system try to use ML approach to identify the disease and provide cure for that disease.
- 2) It will not exhaust lot of time for identifying and predicting the disease of that infected plant.
- 3) Here we can able to categorize the plant into 5 levels for disease prediction and based on physical appearance of that plant, we can able to choose appropriate part to identify the disease name and provide cure according to that.

- 4) Here the outcome or end result will always be accurate
- 5) There is no need to having any previous research knowledge to judge the disease of that plant, the user just need to know how to find the problem on that plant by physical appearance.
- 6) The proposed system is very accurate in identifying the type of disease in one of the 5 levels.

3.4 System Design

Software System design is an interactive process through which the requirements are translated into “blue print” for constructing the software. The three characteristics that are the evaluation of a good design of the software are as follows:

1. The design must implement the entire explicit requirement contained in the analysis mode and it must be accommodate the entire implicit requirement desired by the user.
2. The design must reliable and act as an understandable guide for those, who generate code and text subsequently maintain the software.
3. The design should provide a complete picture of the software, addressing the data functional and behavioral domain from an implementing perspective

The design-phase helps in providing the procedural details necessary for implementing the system recommended in the feasibility study. The emphasis is on translating the performance requirements into design specification. Design phase goes through logical and physical phases of the development.

System Design is a process in which the end users requirements are transformed into software/package and a specification for a computer based information-processing system. System is a solution, on “how to” approach to the creation of a new system. It describes the solution to the problem and serves as the starting point for programming.

Software design is both a process and a model the design process is a set of interactive Steps that enable the designer to describe all aspects of the software to be built. Figure 3.1 as shown above describes the interactive steps for system design for

constructing the software. Basic design principles enable the software's engineer to navigate the design process.

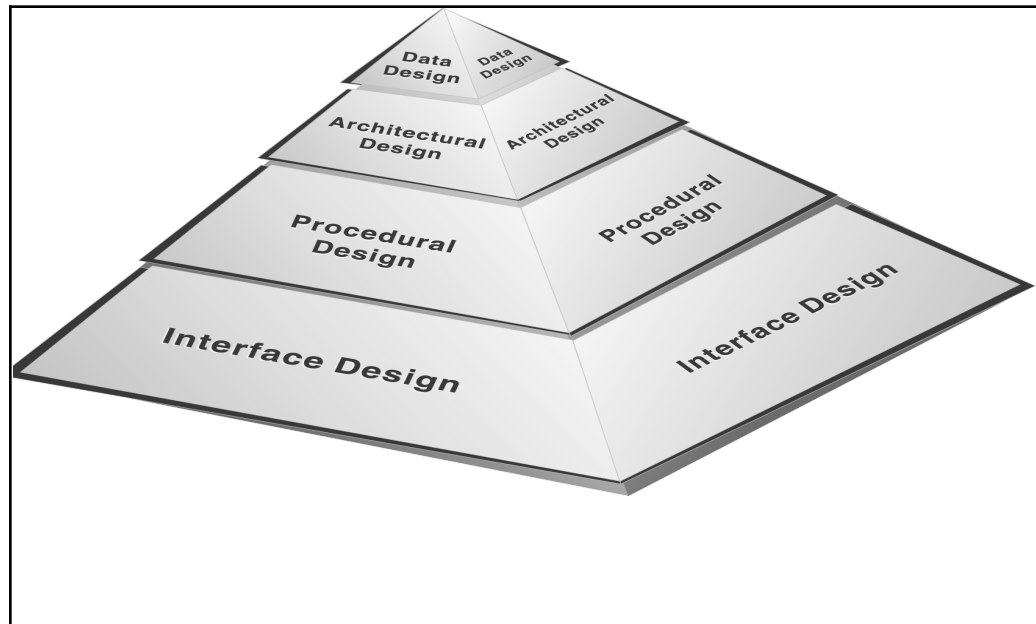


Fig.3.1 Steps for System Design

Software design is both a process and a model the design process is a set of interactive Steps that enable the designer to describe all aspects of the software to be built. Figure 3.1 as shown above describes the interactive steps for system design for constructing the software. Basic design principles enable the software's engineer to navigate the design process.

Some of the design principles are:

- The design should be traceable.
- The design should not reinvent the wheel.
- The design should “minimize the intellectual distance “between the software’s and the problem as it exist in the real world”.
- The design should exhibit uniformly and integration.

3.5 Data Flow Diagram

A Data Flow Diagram (DFD) is a diagram that describes the flow of data and the processes that change or transform data throughout a system. It's a structured analysis

and design tool that can be used for flowcharting in place of, or in association with, information oriented and process oriented system flowcharts. When analysts prepare the Data Flow Diagram, they specify the user needs at a level of detail that virtually determines the information flow into and out of the system and the required data resources. This network is constructed by using a set of symbols that do not imply a physical implementation. The Data Flow Diagram reviews the current physical system, prepares input and output specification, specifies the implementation plan etc.

A data flow diagram is graphical tool used to describe and analyze movement of data through a system. These are the central tool and the basis from which the other components are developed. The transformation of data from input to output, through processed, may be described logically and independently of physical components associated with the system. These are known as the logical data flow diagrams. The physical data flow diagrams show the actual implements and movement of data between people, departments and workstations. A full description of a system actually consists of a set of data flow diagrams.

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.

4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

Figure 3.2 show four basic symbols that are used to construct data flow diagrams. They are symbols that represent data source, data flows, and data transformations and data storage. The points at which data are transformed are represented by enclosed figures, usually circles, which are called nodes. The figure 3.3 shows the data flow diagram of our project.

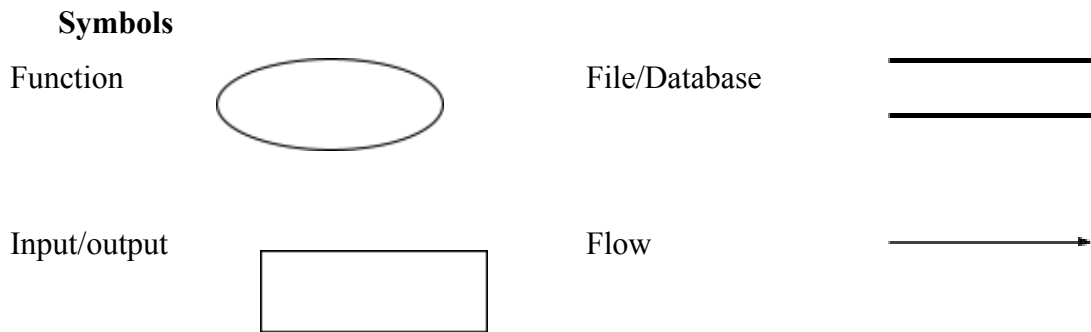


Fig. 3.2 DFD Elements

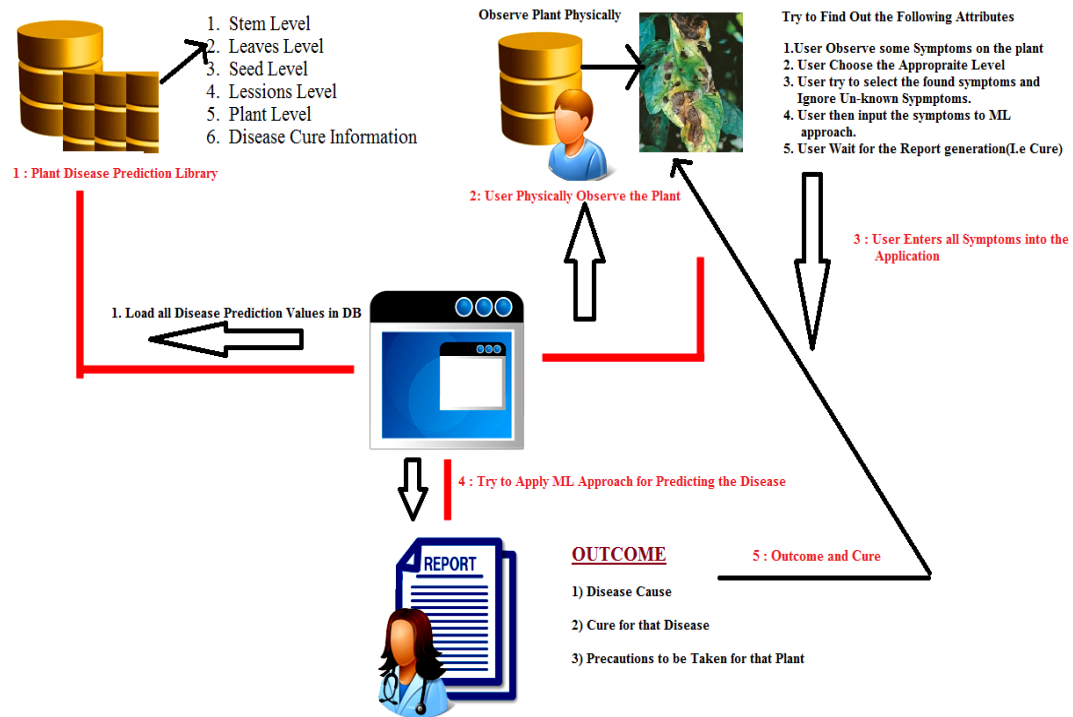


Fig.3.3 Dataflow Diagram for Proposed Application

3.6 UML Diagrams

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML. The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems. The UML is a very important

part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

GOALS: The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

3.6.1 Use Case Diagram

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

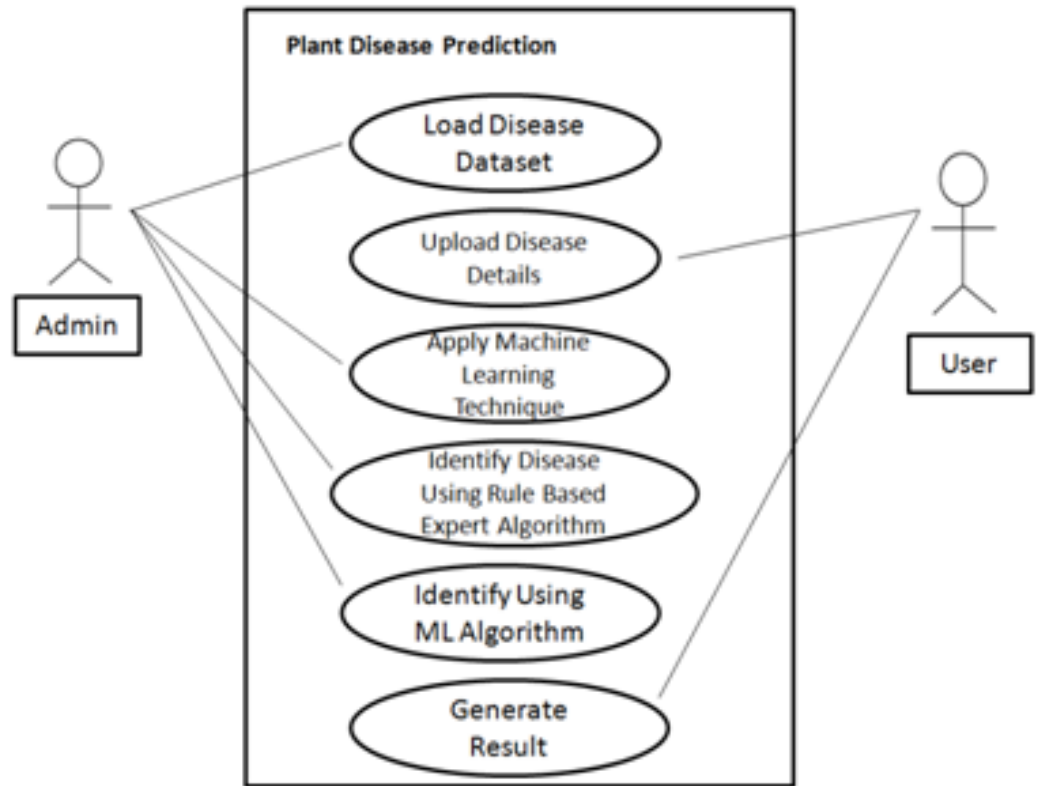


Fig.3.6.1 Use Case Diagram

3.6.2 Class Diagram

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

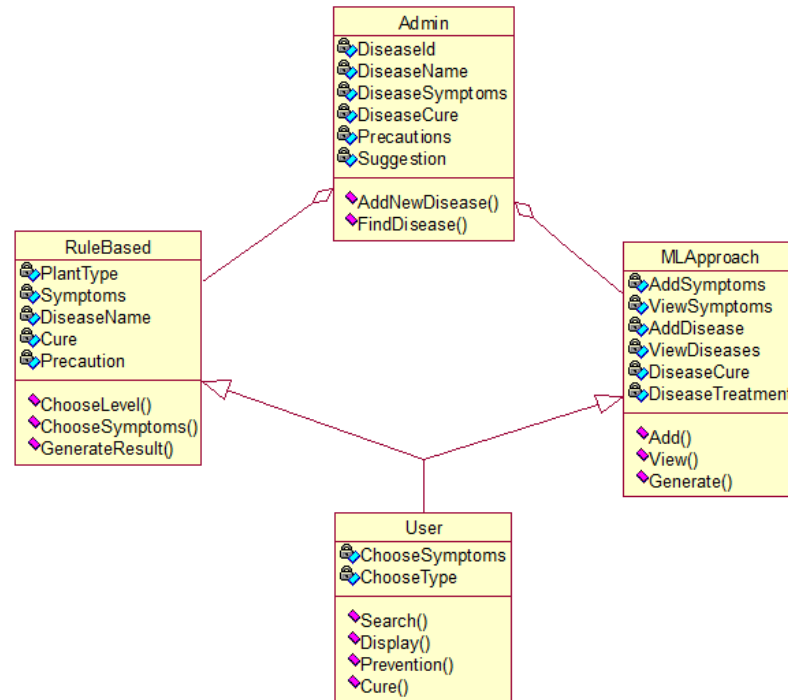


Fig.3.6.2 Class Diagram

3.6.3 Sequence Diagram

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

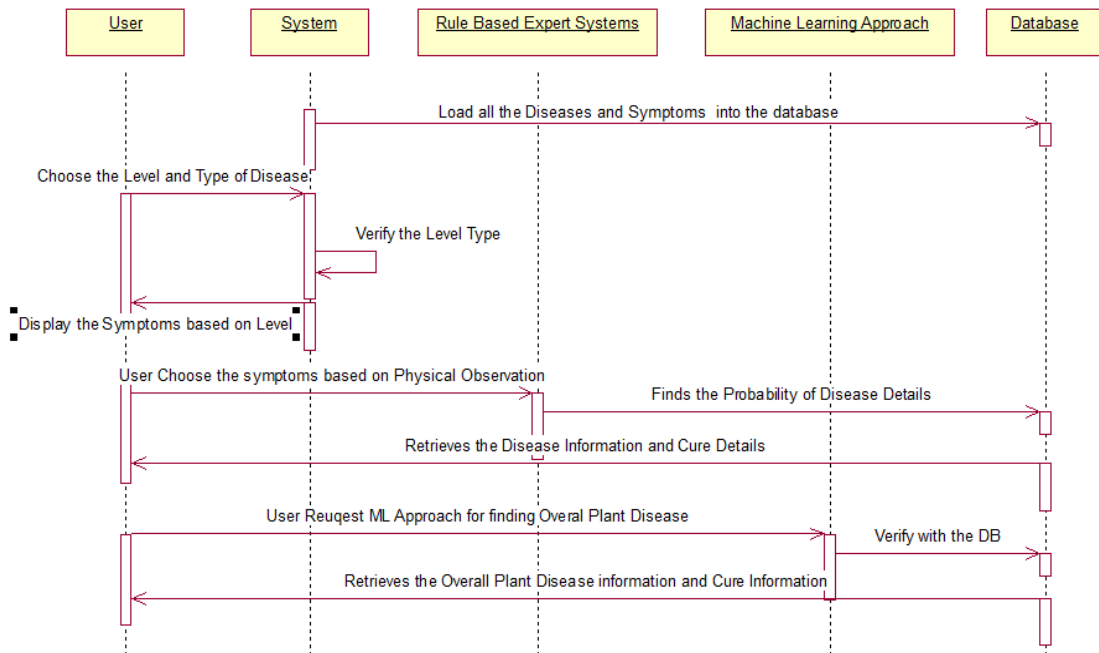


Fig.3.6.3 Sequence Diagram

3.6.4 Collaboration Diagram

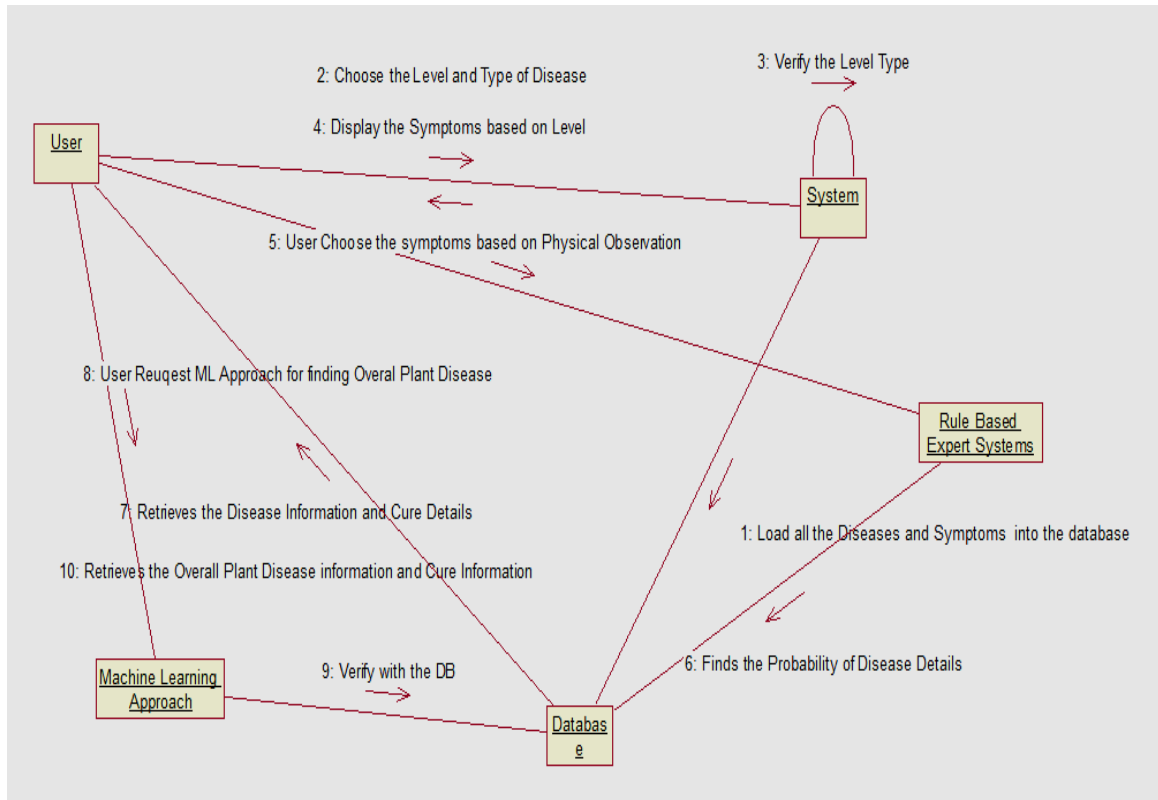


Fig.3.6.4 Collaboration Diagram

3.6.5 Activity Diagram

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

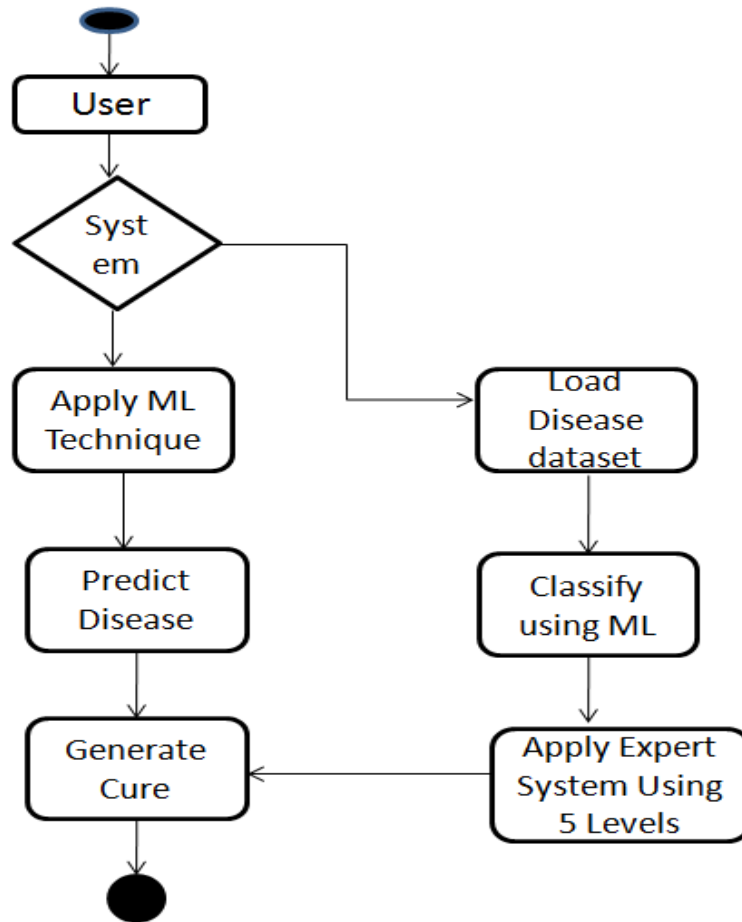
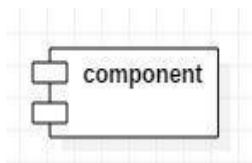


Fig.3.6.5 Activity diagram

3.6.6 Component Diagram

Component diagrams represent a set of components and their relationship. These components consist of classes, interfaces, or collaborations. It is a special kind of diagram in UML. The purpose is also different from all other diagrams, It does not describe the functionality of the system but it describes the components used to make those functionalities. Component diagrams represent the implementation view of system.



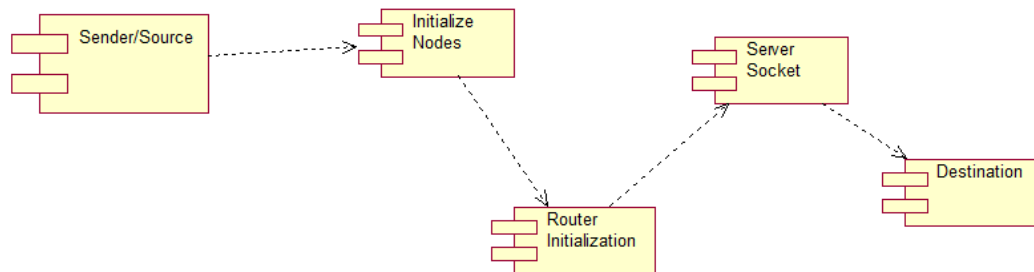


Fig 3.6.6 Component Diagram

3.6.7 Deployment Diagram

A deployment diagram in the Unified Modeling Language models the physical deployment of artifacts on nodes.

The nodes appear as boxes, and the artifacts allocated to each node appear as rectangles within the boxes. Nodes may have sub-nodes, which appear as nested boxes. A single node in a deployment diagram may conceptually represent multiple physical nodes, such as a cluster of database servers.

There are two types of Nodes

- **Device Node:** Device nodes are physical computing resources with processing memory and services to execute software, such as typical computers or mobile phones.
- **Execution Environment Node:** An execution environment node (EEN) is a software computing resource that runs.

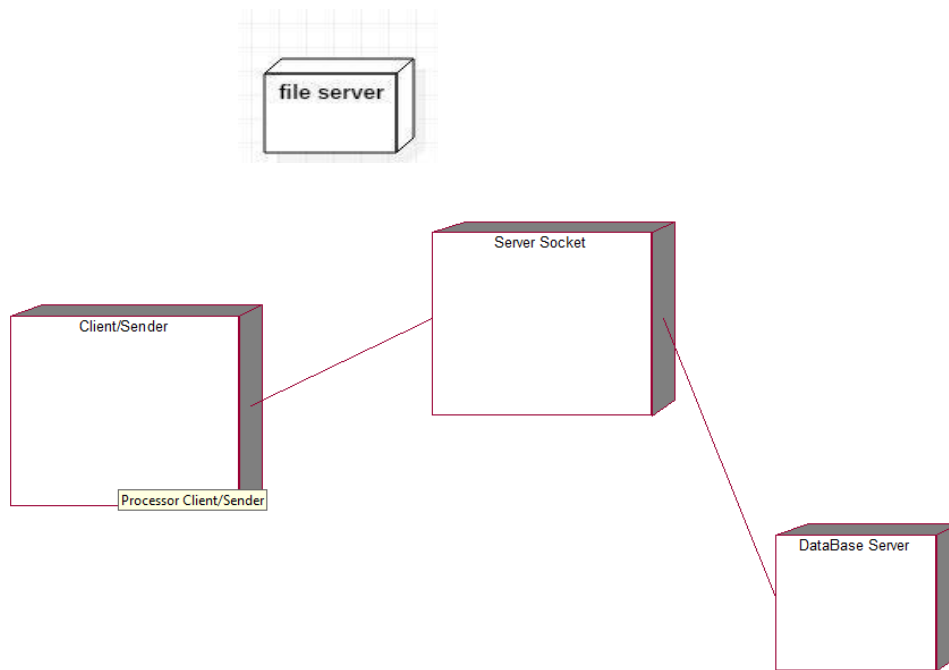


Fig 3.6.7 Deployment Diagram

3.7 Database Design

A database is an organized mechanism that has the capability of storing information through which a user can retrieve stored information in an effective and efficient manner. The data is the purpose of any database and must be protected.

The database design is a two level process. In the first step, user requirements are gathered together and a database is designed which will meet these requirements as clearly as possible. This step is called Information Level Design and it is taken independent of any individual DBMS.

In the second step, this Information level design is transferred into a design for the specific DBMS that will be used to implement the system in question. This step is called Physical Level Design, concerned with the characteristics of the specific DBMS that will be used. A database design runs parallel with the system design. The organization of the data in the database is aimed to achieve the following two major objectives.

- Data Integrity

- Data independence

3.7.1 Entity Relation (ER-Diagram)

An entity-relationship (ER) diagram is a specialized graphic that illustrates the interrelationships between entities in a database. ER diagrams often use symbols to represent three different types of information. Boxes are commonly used to represent entities. Diamonds are normally used to represent relationships and ovals are used to represent attribute also called an entity-relationship model, a graphical representation of entities and their relationships to each other, typically used in computing in regard to the organization of data within databases or information systems. An entity is a piece of data, an object or concept about which data is stored. A relationship is how the data is shared between entities. Several types of attributes occur in the ER model:

- Simple Attribute
- Composite Attributes
- Single-Valued Attributes
- Multivalve Attributes
- Stored Attributes
- Derived Attributes

3.7.2 Normalization

Normalization is the process of decomposing the attributes in an application, which results in a set of tables with very simple structure. The purpose of normalization is to make tables as simple as possible. Normalization is carried out in this system for the following reasons.

- To structure the data so that there is no repetition of data, this helps in saving.
- To permit simple retrieval of data in response to query and report request.
- To simplify the maintenance of the data through updates, insertions, deletions.
- To reduce the need to restructure or reorganize data which new application requirements arise?

As the name implies, it denoted putting things in the normal form. The application developer via normalization tries to achieve a sensible organization of data into proper

tables and columns and where names can be easily correlated to the data by the user. Normalization eliminates repeating groups at data and thereby avoids data redundancy which proves to be a great burden on the computer resources. This includes:

- o Normalize the data.
- o Choose proper names for the tables and columns.
- o Choose the proper name for the data.

It is a process of converting a relation to a standard form. The process is used to handle the problems that can arise due to data redundancy i.e. repetition of data in the database, maintain data integrity as well as handling problems that can arise due to insertion, updating, deletion anomalies.

Decomposing is the process of splitting relations into multiple relations to eliminate anomalies and maintain anomalies and maintain data integrity. To do this we use normal forms or rules for structuring relation.

Insertion anomaly: Inability to add data to the database due to absence of other data.

Deletion anomaly: Unintended loss of data due to deletion of other data.

Update anomaly: Data inconsistency resulting from data redundancy and partial update

Normal Forms: These are the rules for structuring relations that eliminate anomalies.

First Normal Form: A relation is said to be in first normal form if the values in the relation are atomic for every attribute in the relation. By this we mean simply that no attribute value can be a set of values or, as it is sometimes expressed, a repeating group.

Second Normal Form:

A relation is said to be in second Normal form is it is in first normal form and it should satisfy any one of the following rules.

- Primary key is a not a composite primary key
- No non key attributes are present

- Every non-key attribute is fully functionally dependent on full set of primary key.

Third Normal Form: A relation is said to be in third normal form if there exists no transitive dependencies. Transitive Dependency

If two non-key attributes depend on each other as well as on the primary key then they are said to be transitively dependent.

The above normalization principles were applied to decompose the data in multiple tables thereby making the data to be maintained in a consistent state.

3.7.3 Schema

The description of a database is called the database schema, which is specified during the database design and is not expected to change frequently. Most data module has certain.

Conventions for displaying for the schema's as diagram. A displayed schema is called a schema diagram. Figure below show the schema diagram for our database:

A Data Dictionary is a collection of metadata that is data about data. In addition to storing catalog information about schemas and constraints, the data dictionary stores other information, such as design decisions, usage standards,

TABLE 3.7.3 NATURAL_DISEASE TABLE

FIELD	DATATYPE
SNO	VARCHAR2(100)
YELLOWING OF THE TIPS OF THE LOWER LEAVES(S1)	VARCHAR2(100)

LAMINA POSITION REMAINS GREEN&THE MARGIN BECOMES YELLOW (S2)	VARCHAR2(100)
LEAVES OF THE PLANT FROM BOTTOM UPWARDS(S3)	VARCHAR2(100)
LEAVES OF THE PLANT ARE DROPPING WITHERING AND DRYING(S4)	VARCHAR2(100)
OVER TO IRREGULAR WATER SOAKED SPOTS ON THE LEAVES(S5)	VARCHAR2(100)
SURFACE OF THE LEAVES CONSISTING OF BLACK DOTS(S6)	VARCHAR2(100)
LOWER LEAVES OF THE PLANTS AS YELLOWS AS MARGINS(S7)	VARCHAR2(100)
LEAVES OF THE NATURAL PLANT TURN YELLOW AND WILTING(S8)	VARCHAR2(100)
LEAVES FLACCID WITH INTENSE YELLOWISH BRONZE COLOURED(S9)	VARCHAR2(100)
LEAF SHEATHS APPEAR YELLOWISH TO DULL GREEN(S10)	VARCHAR2(100)
LEAF IRREGULAR WATER SOAKED SPOTS IN THE MID REGION(S11)	VARCHAR2(100)
LEAVES FLACCID AND HANG DOWN AND LEAVES DRY UP(S12)	VARCHAR2(100)
MOSAIC STREAKS APPEAR ON THE LEAVES(S13)	VARCHAR(100)
GROWN PSEUDO-STEM WITHERS AND DIES(S14)	VARCHAR(100)
BASE PORTION OF THE PSEUDO STEM BECOMES SOFT AND WATERY(S15)	VARCHAR(100)
STEM TRANSLUCENT BROWN COLOR TO WATER SOAKED(S16)	VARCHAR(100)
DARK STREAKS ARE OBSERVED ON THE PSEUDO STEMS(S17)	VARCHAR(100)

RHIZOME IS DISCOLORED(S18)	VARCHAR(100)
RHIZOME SHOWING CREAMY BROWN DISCOLORATION(S19)	VARCHAR(100)
INFECTED RHIZOME IS KEPT WATER IT GIVES BACTERIAL OOZE(S20)	VARCHAR(100)
LEAVES BECOME PAPERY AND SHOT HOLES APPEAR(S21)	VARCHAR(100)
WATER-SOAKED LINEAR STREAKS(S22)	VARCHAR(100)
DIEASE AND CURE	VARCHAR(1000)

CHAPTER-IV

IMPLEMENTATION ASPECTS

4. SYSTEM IMPLEMENTATION

4.1 Software Requirement Specification

A software requirement specification is developed as consequence of analysis. Review is essential to ensure that the developer and customer have the same perception. Software requirements specification (SRS) is the starting point of the software development activity. The software requirements specification is produced at culmination of the analysis task. The introduction of the software requirement specification states the goals and objectives of the software, describing it in the context of the computer based system.

The software requirements specification includes an information description, functional description, behavioral description. There are several models for such processing, each describing approaches to a variety of tasks or activities that take place during the process. Some people consider a life cycle model a more general term and a software development process a more specific term. For example, there are meant specific software development processes that 'fit' the spiral life cycle model. It aims to be the standard that defines all the tasks required for developing and maintaining software.

The introduction of the software requirement specification states the goals and objectives of the software, describing it in the context of the computer based system.

The purpose of this document is present the software requirement in a precise and easily understood manner. This document provides the functional, performance design and verification requirement of the software to be developed this is the only document that describes the requirement of the system. This is meant for use by the developers and will also be the basis for validating the final delivered system. Figure 4.1 describes software requirement analysis in a diagrammatic manner which is easily understandable.

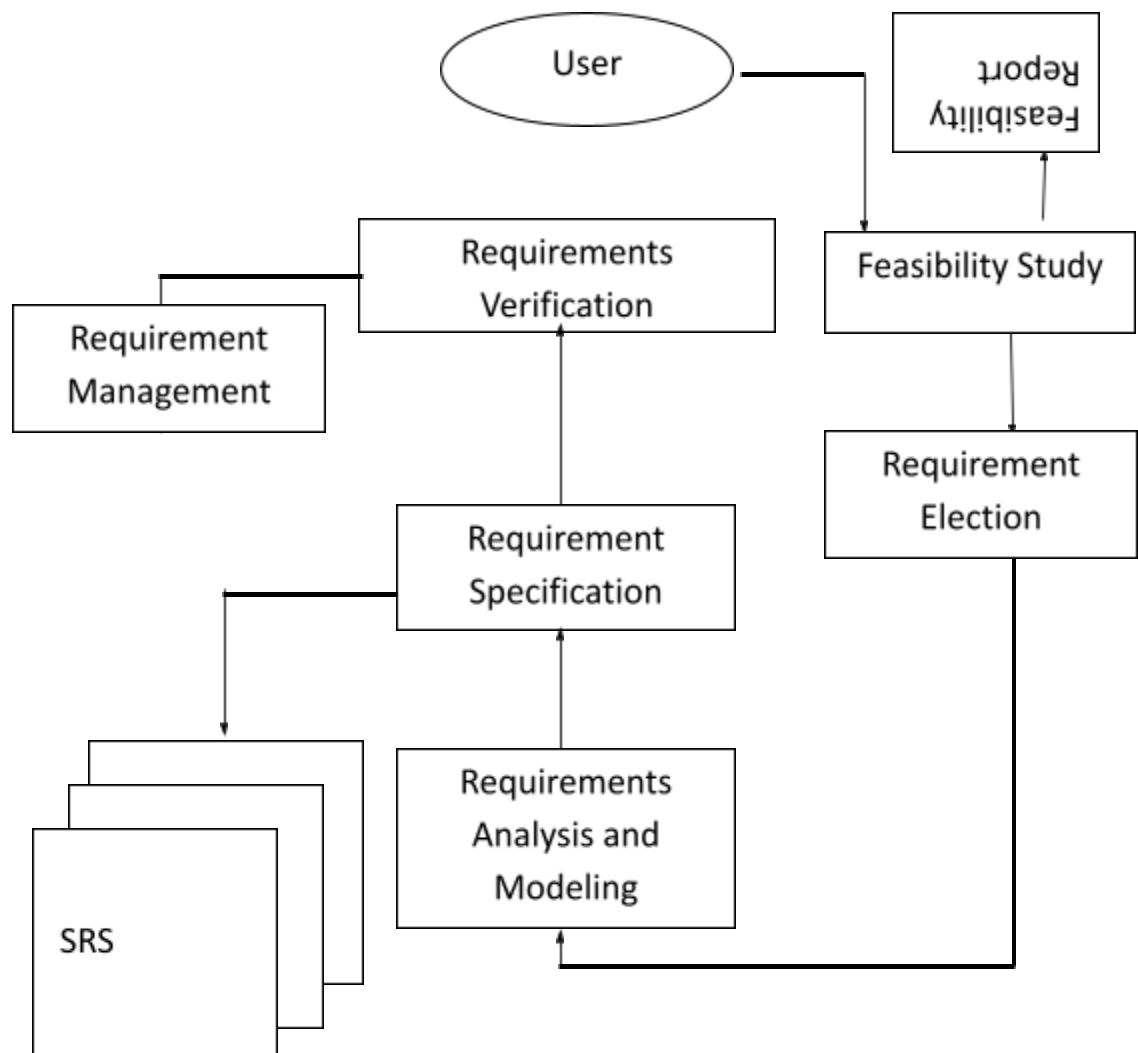


Fig. 4.1 Software Requirement Analysis

4.2 Functional Requirement

Functional requirements describe what the system should do, i.e. the services provided for the users and for the other systems.

INPUT

The Admin/System Load all the symptoms and diseases about plant into database.

The user need to choose the level of disease

The user need to input the disease symptoms based on physical appearance.

The user need to input them to the ML approach.

The system will calculate the probability of disease based on symptoms.

The System sends the result to the end user.

OUTPUT

1. The User gets an output window as” Plant Disease Identified”
2. The User gets an output window as” Report Generated Successfully”
3. The Admin gets an output window as” Disease Details added Successfully”
4. The user gets an output window as” No Disease Found”
5. The user gets a message like” Disease Found and Cure and Precautions is Listed”

Data Storage

Here we use MY Sql as back end database because of various advantages, they are as follows:

- 1) It has featured like auto commit.
- 2) It is GUI in nature (Common line Interface).
- 3) It occupies very less size for installation.
- 4) It is cross platform in nature

4.3 Non-Functional Requirement

Non-Functional requirement are constraints that must be adhered during development. They limit what resource can be used and set bounds on aspects of the software's quality. One of the most important things about non-functional requirement is to be making them verifiable. The verification is normally done by measuring various aspects of the system and seeing if the measurement confirm to the requirements.

4.3.1 Performance Requirements

Some Performance requirements identified is listed below:

1. The database shall be able to accommodate a minimum of 10,000 records of students.
2. The software shall support use of multiple users at a time.
3. There are no other specific performance requirements that will affect development.

4.3.2 Safety Requirements

The database may get crashed at any certain time due to virus or operating system failure. Therefore, it is required to take the database backup.

4.3.3 Security Requirements

Some of the factors that are identified to protect the software from accidental or malicious access, use, modification, destruction, or disclosure are described below. Keep specific log or history data sets

1. Assign certain functions to different modules
2. Restrict communications between some areas of the program.
3. Check data integrity for critical variables.
4. Later version of the software will incorporate encryption techniques in the user/license authentication process.

In non-functional requirements the following are the things that come under .They are as follows:

- 1) **Reusability:** As we developed the application in java, the application can be re-used for any one without having any restrictions in its usage. Hence it is re-Usable.
- 2) **Portability:** As the application is designed with java as programming language, we know java can be run on any operating system. Hence the application is portable to run on any operating system.
- 3) **Extensibility:** The application can be extended at any level if the user wish to extend that in future this is done because java is a open source medium which doesn't have any time limits for expiry or renewal. The same application can be re-used by any one for further extent without any time restrictions.

4.4 Model Used: Spiral Model

4.4.1 Introduction

The spiral model combines the idea of iterative development (prototyping) with the systematic, controlled aspects of the waterfall model. It allows for incremental releases of the product, or incremental refinement through each time around the spiral as shown below in figure 4.2

The spiral model also explicitly includes [risk management](#) within [software development](#). Identifying major risks, both technical and managerial, and determining how to lessen the risk helps keep [software development process](#) under control.

The spiral model is based on continuous refinement of key products for requirements definition and [analysis](#), [system](#) and [software design](#), and [implementation](#) (the code). At each iteration around the cycle, the products are extensions of an earlier product.

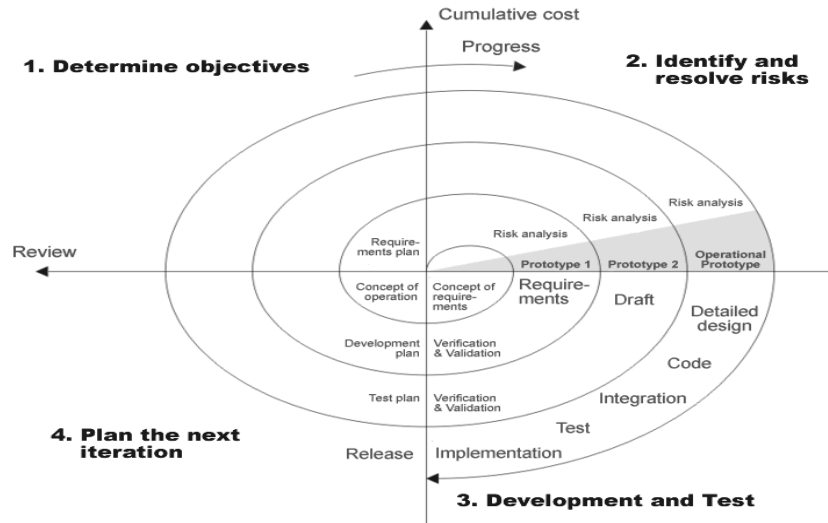


Fig.4.5.Spiral Model

Documents are produced when they are required, and the content reflects the information necessary at that point in the process. All documents will not be created at the beginning of the process, nor all at the end (hopefully). Like the product they define, the documents are works in progress. Starting at the center, each turn around the spiral goes through several task regions determine the objectives, alternatives, and constraints on the new iteration.

- Evaluate alternatives and identify and resolve risk issues.
- Develop and verify the product for this iteration.
- Plan the next iteration.

Each phase is split in 4 quadrants, which are:

- Determine objective.
- Identify and resolve risk.
- Develop next level product.
- Review and planning.

4.4.2 Advantages

The primary advantage of the spiral model is that its range of options accommodates the good features of existing software process models, while its risk-driven approach avoids many of their difficulties. In appropriate situations, the spiral model becomes equivalent to one of the existing process models. In other situations, it provides guidance on the best mix of existing approaches to a given project; for example, its application to the TRW-SPS provided a risk-driven mix of specifying, prototyping, and evolutionary development.

The primary conditions under which the spiral model becomes equivalent to other main process models are summarized as follows:

- If a project has a low risk in such areas as getting the wrong user interface or not meeting strength performance requirements, and if it has a high risk in budget and schedule predictability and control, then these risk considerations drive the spiral model into equivalence to the waterfall model.
- If a software product's requirements are very stable (implying a low risk of expensive design and code breakage due to requirements changes during development), and if the presence of errors in the software product constitutes a high risk to the mission it serves, then these risk considerations drive the spiral model to resemble the two-leg model of precise specification and formal deductive program development.
- If a project has a low risk in such areas as losing budget and schedule predictability and control, encountering large-system integration problems, or coping with information sclerosis, and if it has a high risk in such areas as getting the wrong user interface or user decision support requirements, then these risk considerations drive the spiral model into an equivalence to the evolutionary development model.
- If automated software generation capabilities are available, then the spiral model accommodates them either as options for rapid prototyping or for application of the transform model, depending on the risk considerations involved.

If the high-risk elements of a project involve a mix of the risk items listed above, then the spiral approach will reflect an appropriate mix of the process models above (as

exemplified in the TRW-SPS application). In doing so, its risk avoidance features will generally avoid the difficulties of the other models.

4.5 Hardware Specification

Table 4.6. Hardware Specification

Description	Minimum	Recommended
System type	P IV 2.8 GHz and above	Core Processors
RAM	256 MB	1 GB MB
Cache	256 MB	1 GB
Storage	40 GB	500 GB
Disk Drive	Floppy or CD-RW (back up purpose)	CD-RW or DAT(back up purpose)
Display	15” VGA	17” LED
Output media	DMP Printer	Laser Printer

4.6 Software Specification

Platform (both side): Window 9x, ME, 2000, XP/UNIX/Solaris.

Software:-

- Operating system : Windows Family
- Front End : JSP,HTML,CSS
- Back End : My Sql
- Tools Used : HeildSql,
- IDE : Netbeans 7.1
- Web Server : Apache Tomcat 7.0
- Softwares : JAVA 7

4.7 Theoretical Background

The following languages are used built the required software. They are:

4.7.1 HTML

HTML or Hyper Text Markup Language is the standard markup language used to create web pages. HTML is written in the form of HTML elements consisting of tags enclosed in angle brackets. HTML tags most commonly come in pairs like `<h1>` and `</h1>`, although some tags represent empty elements and so are unpaired, for example `<img>`. The first tag in a pair is the start tag, and the second tag is the end tag (they are also called opening tags and closing tags). The purpose of a web browser is to read HTML documents and compose them into visible or audible web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page. HTML describes the structure of a website semantically along with cues for presentation, making it a markup language rather than a programming language. HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive forms.

4.7.2 CSS

CSS is designed primarily to enable the separation of document content from document presentation, including elements such as the layout, colors and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple pages to share formatting and reduce complexity and repetition in the structural content (such as by allowing for tables web design). CSS can also allow the same markup page to be presented in different styles for different rendering methods, such as on-screen, in print, by voice and on Braille-based, tactile devices. It can also be used to allow the web page to display differently depending on the screen size or devices on which it is being viewed. While the author of a document typically links that document to a CSS file, readers can use a different style sheet, perhaps one on their own computer, to override the one the author has specified. However if the author or the reader did not link the document to a specific style sheet the default style of the browser will be applied.

CSS specifies a priority scheme to determine which style rules apply more than one rule matches against a particular element. In this so-called cascade, priorities or weights are calculated and assigned to rules, so that the results are predictable.

4.7.3 MYSQL

The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation. MySQL is a popular choice of database for use in web applications, and is a central component of the widely used LAMP open source web application software stack (and other 'AMP' stacks). LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python." Free-software-open source projects that require a full-featured database management system often use MySQL.

MySQL is a relational database management system (RDBMS), and ships with no GUI tools to administer MySQL databases or manage data contained within the databases. Users may use the included command line tools, or use MySQL "front-ends", desktop software and web applications that create and manage MySQL databases, build database.

4.7.4 JAVA

Java technology is both a programming language and a platform.

The Java Programming Language

The Java programming language is a high-level language that can be characterized by all of the following buzzwords:

- Simple
- Architecture neutral
- Object oriented
- Portable
- Distributed
- High Performance
- Interpreted
- Multithreaded

- Robust
- Dynamic
- Secure

The Java programming language is unusual in that a program is both compiled and interpreted. With the compiler, first you translate a program into an intermediate language called *Java byte codes* —the platform-independent codes interpreted by the interpreter on the Java platform. The interpreter parses and runs each Java byte code instruction on the computer. Compilation happens just once; interpretation occurs each time the program is executed. The following figure 4.3 illustrates how this works.

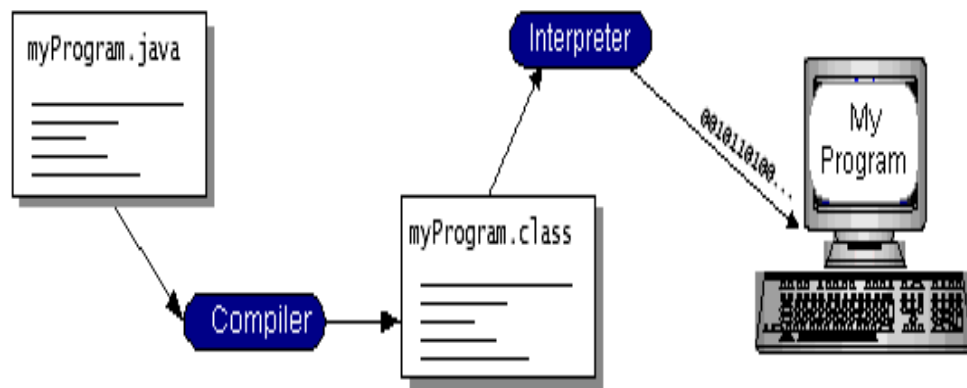


Fig.4.7.4 Steps of Java

You can think of Java byte codes as the machine code instructions for the *Java Virtual Machine* (Java VM). Every Java interpreter, whether it's a development tool or a Web browser that can run applets, is an implementation of the Java VM. Java byte codes help make “write once, run anywhere” possible. You can compile your program into byte codes on any platform that has a Java compiler. The byte codes can then be run on any implementation of the Java VM. That means that as long as a computer has a Java VM, the same program written in the Java programming language can run on Windows 2000, a Solaris workstation, or on an iMac.

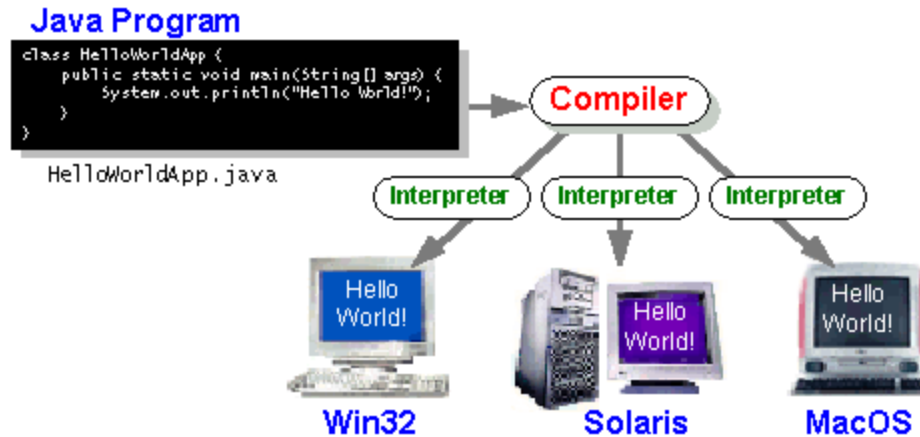


Fig.4.7.4 Java Execution

The Java Platform

A **platform** is the hardware or software environment in which a program runs. We've already mentioned some of the most popular platforms like Windows 2000, Linux, Solaris, and MacOS. Most platforms can be described as a combination of the operating system and hardware. The Java platform differs from most other platforms in that it's a software-only platform that runs on top of other hardware-based platforms.

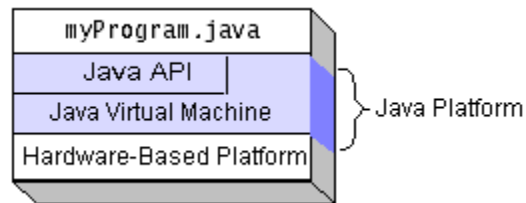
The Java platform has two components:

- The **Java Virtual Machine** (Java VM)
- The **Java Application Programming Interface** (Java API)

You've already been introduced to the Java VM. It's the base for the Java platform and is ported onto various hardware-based platforms.

The Java API is a large collection of ready-made software components that provide many useful capabilities, such as graphical user interface (GUI) widgets. The Java API is grouped into libraries of related classes and interfaces; these libraries are known as **packages**. The next section, What Can Java Technology Do? Highlights what functionality some of the packages in the Java API provide.

The following figure depicts a program that's running on the Java platform. As the figure shows, the Java API and the virtual machine insulate the program from the hardware.



Native code is code that after you compile it, the compiled code runs on a specific hardware platform. As a platform-independent environment, the Java platform can be a bit slower than native code. However, smart compilers, well-tuned interpreters, and just-in-time byte code compilers can bring performance close to that of native code without threatening portability.

ODBC

Microsoft Open Database Connectivity (ODBC) is a standard programming interface for application developers and database systems providers. Before ODBC became a *de facto* standard for Windows programs to interface with database systems, programmers had to use proprietary languages for each database they wanted to connect to. Now, ODBC has made the choice of the database system almost irrelevant from a coding perspective, which is as it should be. Application developers have much more important things to worry about than the syntax that is needed to port their program from one database to another when business needs suddenly change.

Through the ODBC Administrator in Control Panel, you can specify the particular database that is associated with a data source that an ODBC application program is written to use. Think of an ODBC data source as a door with a name on it. Each door will lead you to a particular database. For example, the data source named Sales Figures might be a SQL Server database, whereas the Accounts Payable data source could refer

to an Access database. The physical database referred to by a data source can reside anywhere on the LAN.

The ODBC system files are not installed on your system by Windows 95. Rather, they are installed when you setup a separate database application, such as SQL Server Client or Visual Basic 4.0. When the ODBC icon is installed in Control Panel, it uses a file called ODBCINST.DLL. It is also possible to administer your ODBC data sources through a stand-alone program called ODBCADM.EXE. There is a 16-bit and a 32-bit version of this program and each maintains a separate list of ODBC data sources.

From a programming perspective, the beauty of ODBC is that the application can be written to use the same set of function calls to interface with any data source, regardless of the database vendor. The source code of the application doesn't change whether it talks to Oracle or SQL Server. We only mention these two as an example. There are ODBC drivers available for several dozen popular database systems. Even Excel spreadsheets and plain text files can be turned into data sources. The operating system uses the Registry information written by ODBC Administrator to determine which low-level ODBC drivers are needed to talk to the data source (such as the interface to Oracle or SQL Server). The loading of the ODBC drivers is transparent to the ODBC application program. In a client/server environment, the ODBC API even handles many of the network issues for the application programmer.

The advantages of this scheme are so numerous that you are probably thinking there must be some catch. The only disadvantage of ODBC is that it isn't as efficient as talking directly to the native database interface. ODBC has had many detractors make the charge that it is too slow. Microsoft has always claimed that the critical factor in performance is the quality of the driver software that is used. In our humble opinion, this is true. The availability of good ODBC drivers has improved a great deal recently. And anyway, the criticism about performance is somewhat analogous to those who said that compilers would never match the speed of pure assembly language. Maybe not, but the compiler (or ODBC) gives you the opportunity to write cleaner programs, which means you finish sooner. Meanwhile, computers get faster every year.

4.8 Software project Modules

Implementation is the stage where the theoretical design is converted into programmatically manner. In this stage we will divide the application into a number of modules and then coded for deployment. We have implemented the proposed concept on Java programming language with JEE as the chosen language in order to show the performance this proposed protocol. The proposed application is mainly divided into 5 modules. They are as follows:

1. Load Disease Data Set
2. User Module
3. Machine Learning Technique
4. Identify Disease using Rule Based Expert System
5. Identify Disease using Machine Learning Algorithm

Now let us discuss about each and every module in detail as follows:

4.8.1 Load Plant Dataset Module

This is a predefined task which is done by the administrator in order to maintain a proper data set to classify the plant diseases. Whenever a new plant disease is invented, then that disease details should be maintained into the database by the administrator. This is nothing but collecting training data set information for the naïve bayes classification algorithm.

4.8.2 User Module

In this module the user is one who try to verify the diseases on a plant. he try to identify all the infected areas of that plant physically and then try to substitute those symptoms on that plant library. Here those symptoms which he found will be choose as yes and remaining those symptoms which he didn't observe on that plant will be selected as No. So once after choosing all the values the corresponding inputs is send to the ML approach. Here the user try to give test inputs for the Naïve Bayes classification

algorithm. These test values should be matched with training data set and then probability of disease is identified by the ML approach.

4.8.3 Machine Learning Technique Module

In this module, the predefined disease data sets and user inputs are to be learned by the machine (computer). Machine learning is the study of how to make computers learn; the goal is to make computers improve their performance through experience. This ML is mainly used in clustering the diseases based on the type of symptoms. As we all know that all plants may not suffer with same type of disease and same level of complaints. So based on the individual problem, the ML system needs to guide the user to take cure on those conditions.

4.8.4 Rule Based Expert Module

Here the diseases can be identified based on rule based expert system which means user needs to input the symptoms based on one of the 5 levels and after the user chooses the level and enters all the symptoms then only the user will get the appropriate disease name based on expert knowledge. Here if the user chooses all symptoms as null, then the resultant output will be displayed as no disease for that plant. If the same user tries to choose appropriate inputs then based on that the disease will be predicted and cure will be provided for that user.

4.8.5 Machine Learning Based Module

Here this module clearly tells that there is no need to choose individual category and then input the symptoms. If the user finds some common symptoms which can be generally visualized and identified. Those symptoms he tries to choose from this common list of attributes and based on those fields, the ML approach will decide which type of disease the plant suffers from and how much percentage of infection occurred to the plant. We can calculate the percentage of infection on that plant.

CHAPTER-V

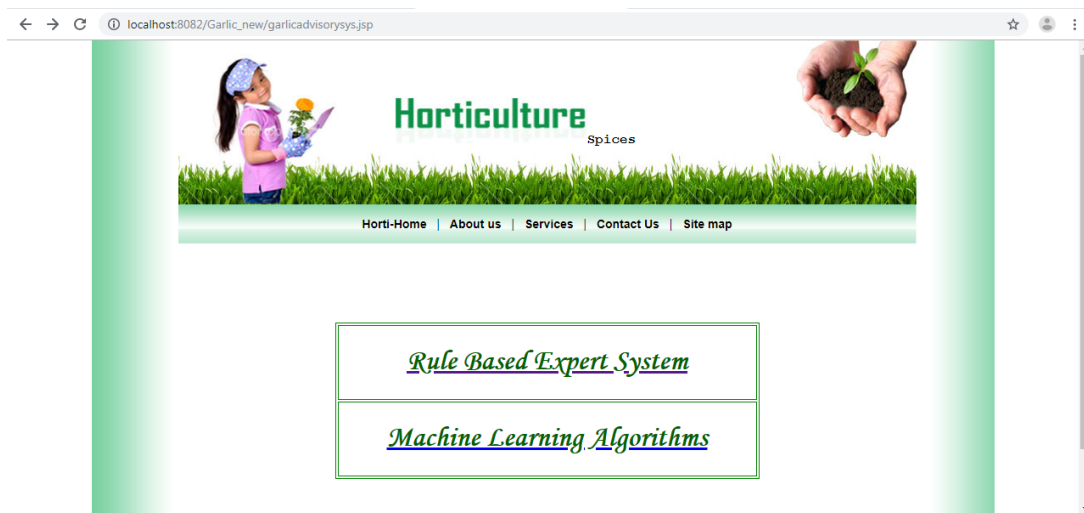
EXPERIMENTAL RESULTS

5. SYSTEM TESTING

5.1 Results (Output Screens)

Output Screens

Home Page



Rule Based Expert System to identify the plant diseases

← → ↻ localhost:8082/Garlic_new/rulbased/garlicindexroot.jsp ☆ ⚙ ⋮



[Horti-Home](#) | [About us](#) | [Services](#) | [Contact Us](#) | [Site map](#)

Rule Based System

Which Part Of the Plant is Effectd ?	
☒ Stem	
☐ Leaves	
☐ Seed	
☐ Lesions	
☐ Plant	
SUBMIT	

Here we need to choose which part of plant is effected and based on the selection this expert system will tell the type of disease

For example we choose stem level

← → ↻ localhost:8082/Garlic_new/rulbased/garlicindexroot.jsp ☆ ⚙ ⋮



[Horti-Home](#) | [About us](#) | [Services](#) | [Contact Us](#) | [Site map](#)

Rule Based System

Which Part Of the Plant is Effectd ?	
☒ Stem	
☐ Leaves	
☐ Seed	
☐ Lesions	
☐ Plant	
SUBMIT	

Then it will ask us what is the condtion about that stem

← → ↻ localhost:8082/Garlic_new/rulbased/garlicstem1.jsp ☆ ⚙ ⋮



Horti-Home | About us | Services | Contact Us | Site map

Stem

What was the condition of the STEM ?	
☒	Fungal Growth
☐	Swelling Of Stems
☐	Looking like water soaked
SUBMIT	

Here for example we choose fungal growth on the stem

← → ↻ localhost:8082/Garlic_new/rulbased/garlicstem2.jsp ☆ ⚙ ⋮



Horti-Home | About us | Services | Contact Us | Site map

Stem

What was the Temperature Condition ?	
☒	Cool
☐	Medium
☐	Warm
SUBMIT	

Then from the above window we can choose what is the temperature condition in that region

We have choosed medium temperature

← → ↻ ⓘ localhost:8082/Garlic_new/rulbased/garlicstem2.jsp ☆ ⚙ ⋮



Horti-Home | About us | Services | Contact Us | Site map

Stem

What was the Temperature Condition ?
☐ Cool
☒ Medium
☐ Warm
SUBMIT

Now the expert system gives the suggestion as below:

← → ↻ ⓘ localhost:8082/Garlic_new/rulbased/garlicdisease3.jsp ☆ ⚙ ⋮



Horti-Home | About us | Services | Contact Us | Site map

May be Effected with " <i>powdery mildew</i> "
Cure Is :using Sulphur can reduce this disease

[Back](#)

In the same way we can check other attributes of the plant like
Seed Level

localhost8082/Garlic_new/rulbased/garlicindexroot.jsp



[Horti-Home](#) | [About us](#) | [Services](#) | [Contact Us](#) | [Site map](#)

Rule Based System

Which Part Of the Plant is Effectd ?
☐ Stem
☐ Leaves
☒ Seed
☐ Lesions
☐ Plant
SUBMIT



[Horti-Home](#) | [About us](#) | [Services](#) | [Contact Us](#) | [Site map](#)

Seed

Color of the Seeds?
☒ Greyish violet
☐ White centres in color
SUBMIT

Now it will ask the temp

← → ↻ ⓘ localhost:8082/Garlic_new/rulbased/garlicseed2.jsp ☆ ⚙ ⋮



Horti-Home | About us | Services | Contact Us | Site map

Seed

What was the temperature?	
☐ Cool	
☐ Medium	
☒ High	
SUBMIT	

Now the expert system suggest as follows:



Horti-Home | About us | Services | Contact Us | Site map

May be Effected with " <i>Downy Mildew</i> "
Cure Is : Using fungicide,pesticides like Copper Hydroxide and Chlorothanol can reduce this disease.

[Back](#)

Now we can check plant level

localhost:8082/Garlic_new/rulbased/garlicindexroot.jsp



[Horti-Home](#) | [About us](#) | [Services](#) | [Contact Us](#) | [Site map](#)

Rule Based System

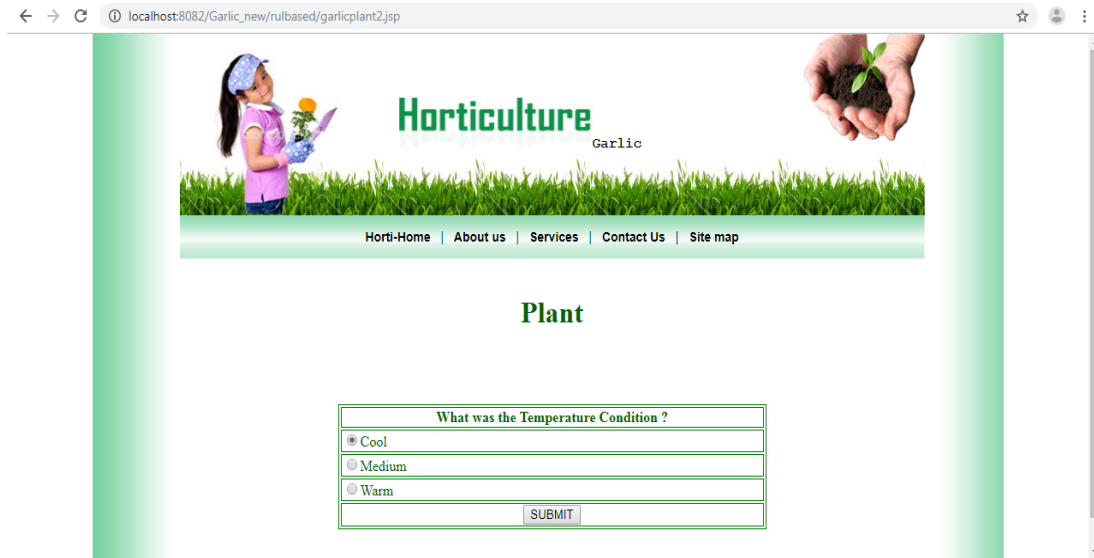
Which Part Of the Plant is Effected ?	
☐	Stem
☐	Leaves
☐	Seed
☐	Lesions
☒	Plant
SUBMIT	



[Horti-Home](#) | [About us](#) | [Services](#) | [Contact Us](#) | [Site map](#)

Plant

What was the condition of the Plant ?	
☒	Stunted
☐	Plants die suddenly
SUBMIT	



Now we can check the Machine Learning based APPRAOCH



5.2 Testing And Test Cases

5.2 Testing

Verification and Validation encompass a wide range of software quality assurance(SQA) activities that include formal, technical review and quality and configuration audits, performance monitoring,simulation,feasibility study,documentation,review algorithm analysis, development testing, qualification testing and installation testing.

5.2.1 Verification

Verification is the fundamental concept in the software design. Verification refer in the set of activities that ensure the software correctly implement a specification

function validation refer to a different set of activities that ensure that software that has been built in traceable customer. Validation is nothing but the process of using software in live environment in order to find errors verification is typically done in two steps.

- Verification of the requirement verification that the software requirement definition satisfies the customer needs.
- Verification of the design: verification that the design satisfies the requirement definition.

Software reviews are one of the most important SQA activities. Review serves as a filter for the software process, removing errors while they are relatively inexpensive to find and correct. To properly verify a system, data about engineering process should be collect, eventually and disseminated,SQA helps to improve the quality of the product and the software procedure.

5.2.2 Testing Methodology

Testing a is process, which reveals in the program. It is the condition known as text cased and the e major quality measure employed during software development. During testing, the program is executed with set of condition known as text cased and output evaluated to determine whether the program is performing as expected. In order to make sure that the system does not error, the different levels of testing strategies that are applied at different levels of testing strategies that are applied at different phases of software development are:

5.2.1.1 Unit Testing

Unit Testing is done on individual modules are they are completed and become executable. It is confined only to the designer requirement.

5.2.1.2 Black Box Testing

In this strategy some test cases are generated as input condition that execute all function requirement for the program this testing has been uses to find error in the following categories:

- Incorrect or missing function
- Interface errors

- Error in data structure or external database access
- Performance
- Initialization

In this testing the output is checked for correctness the logical flow of the data is not checked

5.2.1.3 White Box Testing

In this test cases are generated on the logic of each module by drawing flow graph of that module and logical decision are tested on all the cases. While white-box testing can be applied at the unit, integration and system levels of the software testing process, it is usually done at the unit level. It can test paths within a unit, paths between units during integration, and between subsystems during a system-level test.

- Guarantee that all independent paths have been executed.
- Executed all logic decision on their true and false sides.
- Execute all loops at their boundaries and within their operational bounds.
- Execute internal data structures to ensure their validity.

5.2.1.4 Integration Testing

Integration testing ensures that software and subsystem work together as whole. It tests the interface of the entire module to make behave properly when integrated together.

5.2.1.5 System Testing

Involves in-housing testing of entire system before delivery to the user. Its aim is to satisfy the user the system meet all requirement of the clients specification.

5.2.1.6 Validation Testing

The system has been tested and implemented successfully and thus ensured that all requirement are listed in the software requirement specification are completely fulfilled. In case of enormous input corresponding error message are displayed

5.2.1.7 Compilation Testing

It was a good idea to do our stress early on, because it gave us time to fix some unexpected deadlocks and stability problem that only occurred when components were exposed to very high transaction Volumes.

5.2.1.8 Executive Testing

This program was successfully loaded and executed, because of good programming there no execution errors.

5.2.1.9 Output Testing

The successful output screens are placed in the output screens section above.

5.2.1.10 Activities Undertaken During Integration And System Testing

It involves testing of integrating various modules and testing their overall performance due to their integration. Integration of different modules is undertaken once they have been coded and unit tested. During the integration and system testing phase, the modules are integrated in a planned manner and the goal of system testing is to ensure that the developed system conforms to its requirements laid out in the SRS document. System testing usually consists of three different kinds of testing activities:

- α – testing: It is the system testing performed by the development team.
- β – Testing: It is the system testing performed by a friendly set of customers.
- acceptance testing: It is the system testing performed by the customer himself after the

5.3 Test Cases for Unit Testing and Validation Testing's

S No. of test case	TC1
Name of test	<i>User Inputs The Symptoms</i>

Sample Input	Here the user should select each and every symptom based on physical observance
Expected output	Displays Message as “User Chosen Valid Inputs”
Actual output	Same as expected
Remarks	This component clearly tells that User will be get an acknowledgment if he choose valid inputs

S No. of test case	TC2
Name of test	<i>User Fails to Find the Disease</i>
Sample Input	Here the user leaves or choose wrong inputs
Expected output	Displays Error Message as “Disease not Found”
Actual output	Same as expected
Remarks	This component clearly tells that User cant able to get the disease name if he chose wrong entries.

S No. of test case	TC3
Name of test	<i>Disease Found</i>

Sample Input	Here the user should choose all the valid symptoms based on type or plant affected level
Expected output	Displays Message as “Disease and Cure Found for that Plant”
Actual output	Same as expected
Remarks	This component clearly tells that user can get disease and cure details if all the details are entered correctly.

S No. of test case	TC4
Name of test	<i>Please select the field</i>
Sample Input	If the user fail to choose the level correctly
Expected output	Displays Error Message as “Please choose any field”
Actual output	Same as expected
Remarks	This component clearly tells that cant get any disease details if he fail to select any level properly

CHAPTER-VI

CONCLUSION AND FUTURE SCOPE

6. CONCLUSION AND FUTURE WORK

The conclusions of our study suggest that domain-specific knowledge improves the results. Probabilistic models are stable and reliable for tasks performed on short texts in the medical domain. The representation techniques influence the results of the ML algorithms, but more informative representations are the ones that consistently obtain the best results.

The first task that we tackle in this paper is a task that has applications in information retrieval, information extraction, and text summarization. We identify potential improvements in results when more information is brought in the representation technique for the task of classifying short medical texts. We show that the simple BOW approach, well known to give reliable results on text classification tasks, can be significantly outperformed when adding more complex and structured information from various ontologism.

REFERENCES

- [1] R. Bunescu and R. Mooney, "A Shortest Path Dependency Kernel for Relation Extraction," Proc. Conf. Human Language Technology and Empirical Methods in Natural Language Processing (HLT/ EMNLP), pp. 724-731, 2005.
- [2] R. Bunescu, R. Mooney, Y. Weiss, B. Schölkopf, and J. Platt, "Subsequence Kernels for Relation Extraction," Advances in Neural Information Processing Systems, vol. 18, pp. 171-178, 2006.
- [3] A.M. Cohen and W.R. Hersh, and R.T. Bhupathi raju, "Feature Generation, Feature Selection, Classifiers, and Conceptual Drift for Biomedical Document Triage," Proc. 13th Text Retrieval Conf.(TREC), 2004.
- [4] M. Craven, "Learning to Extract Relations from Medline," Proc.Assoc. for the Advancement of Artificial Intelligence, 1999.
- [5] I. Donaldson et al., "Pre BIND and Textomy : Mining the Biomedical Literature for Protein-Protein Interactions Using a Support Vector Machine," BMC Bioinformatics, vol. 4, 2003.
- [6] C. Friedman, P. Kra, H. Yu, M. Krauthammer, and A. Rzhetsky, " GENIES: A Natural Language Processing System for the Extraction of Molecular Pathways from Journal Articles," Bio informatics, vol. 17, pp. S74-S82, 2001.
- [7] O. Frunza and D. Inkpen, "Textual Information in Predicting Functional Properties of the Genes," Proc. Workshop Current Trends in Biomedical Natural Language Processing (Bio NLP) in conjunction with Assoc. for Computational Linguistics (ACL '08), 2008.
- [8] R. Gaizauskas, G. Demetriou, P.J. Artymiuk, and P. Willett, "Protein Structures and Information Extraction from Biological Texts: The PASTA System," Bioinformatics, vol. 19, no. 1, pp. 135-143, 2003.

- [9] C. Giuliano, L. Alberto, and R. Lorenza, "Exploiting Shallow Linguistic Information for Relation Extraction from Biomedical Literature," Proc. 11th Conf. European Chapter of the Assoc. for Computational Linguistics, 2006.
- [10] J. Ginsberg, H. Mohebbi Matthew, S.P. Rajan, B. Lynnette, S.S.Mark, and L. Brilliant, "Detecting Influenza Epidemics Using Search Engine Query Data," Nature, vol. 457, pp. 1012-1014, Feb.2009.

APPENDICES

SOURCE CODE

Home.jsp

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<title>Health Care Domain</title>
<link rel="stylesheet" href="images/all11.css" type="text/css">
<style type="text/css">
<!--
.main { font-family: Verdana, Arial, Helvetica, sans-serif; font-size: 10px; line-height:
21px; color: #4F1A00; text-align: justify; font-weight: normal}
-->
</style>
</head>

<body background="images/totalbg.jpg" leftmargin="0" topmargin="0"
marginwidth="0" marginheight="0"
onLoad="MM_preloadImages('images/contactus2.jpg','images/skjc2.jpg','images/tsrd2.j
pg','images/tsrw2.jpg','images/aboutus2.jpg')">
<table width="100%" border="0" cellspacing="0" cellpadding="0">
<tr>
```

```

<td align="center">
<table border="0" cellpadding="0" cellspacing="1" bgcolor="#666666" width="409">
<tr>
<td valign="top" bgcolor="#CCCCCC">
<table width="555" border="0" cellpadding="0" cellspacing="0"
bgcolor="#CCCCCC">
<tr>
<td valign="top"><imgsrc="images/title1.png" width="800" height="90"></td>

</tr>

<tr>
<td valign="top" height="53">
<table width="77%" border="0" cellspacing="0" cellpadding="0">
<tr valign="top">
<td width="313" height="98"><table border="0" cellspacing="0" cellpadding="0">
<tr>
<td valign="top"><imgsrc="images/surgeon_home.jpg" width="200"
height="210"></td>
<td valign="top"><imgsrc="images/Img-Medical.jpg" width="200"
height="210"></td>
</tr>
</table></td>
<td width="500" height="98"><imgsrc="images/test.jpg" width="300" height="210">
<td width="500" height="98"><imgsrc="images/Img-Med.jpg" width="100"
height="210">

</table></td>
</tr>
<tr>
<td valign="top" bgcolor="#FFFFFF">

```

<table border="0" style="width: 96%;"> <tr> <td style="height: 174px; vertical-align: top;"> <p class="main"> The Machine Learning (ML) is envisioned as a tool by which computer-based systems can be integrated in the healthcare field in order to get a better, more efficient medical care. The empirical domain of automatic learning is used in tasks such as medical decision support, medical imaging, protein-protein interaction, extraction of medical knowledge, and for overall patient management care. Machine Learning (ML) field has gained Its momentum in almost any domain of research and just recently has become a reliable tool in the medical domain. </p> <p> This paper describes a ML-based methodology for building an application that is capable of identifying and disseminating healthcare information. It extracts sentences from published medical Papers that mention diseases and treatments, and identifies semantic relations that exist between diseases and treatments. The potential value of this paper stands in the ML settings that we propose and in the fact that we outperform previous results on the same data set. </p> </td> </tr> </table>	<p class="main"> The Machine Learning (ML) is envisioned as a tool by which computer-based systems can be integrated in the healthcare field in order to get a better, more efficient medical care. The empirical domain of automatic learning is used in tasks such as medical decision support, medical imaging, protein-protein interaction, extraction of medical knowledge, and for overall patient management care. Machine Learning (ML) field has gained Its momentum in almost any domain of research and just recently has become a reliable tool in the medical domain. </p> <p> This paper describes a ML-based methodology for building an application that is capable of identifying and disseminating healthcare information. It extracts sentences from published medical Papers that mention diseases and treatments, and identifies semantic relations that exist between diseases and treatments. The potential value of this paper stands in the ML settings that we propose and in the fact that we outperform previous results on the same data set. </p>
<p class="main"> The Machine Learning (ML) is envisioned as a tool by which computer-based systems can be integrated in the healthcare field in order to get a better, more efficient medical care. The empirical domain of automatic learning is used in tasks such as medical decision support, medical imaging, protein-protein interaction, extraction of medical knowledge, and for overall patient management care. Machine Learning (ML) field has gained Its momentum in almost any domain of research and just recently has become a reliable tool in the medical domain. </p> <p> This paper describes a ML-based methodology for building an application that is capable of identifying and disseminating healthcare information. It extracts sentences from published medical Papers that mention diseases and treatments, and identifies semantic relations that exist between diseases and treatments. The potential value of this paper stands in the ML settings that we propose and in the fact that we outperform previous results on the same data set. </p>	

```

<tr><td width="39%" height="79" valign="top"><a href="AdminLogin.jsp"
target="_self"><imgsrc="images/admin.jpg" width="79" height="79"
border="0"></a></td>
<td width="54%" valign="top" class="main11">
<table height="78" border="0" cellpadding="0" cellspacing="0">
<tr>
<td height="60" valign="top" bgcolor="#FFFFFF" class="main11" align="center">
<table width="79%" border="0" cellspacing="0" cellpadding="0">
<tr>
<td class="main1111" height="46"><font color="#003399"><a href="AdminLogin.jsp"
target="_self">Admin
Login</a></font></td>
</tr>
</table>
</td>
</tr>
<tr>
<td height="19" valign="bottom" bgcolor="#FFFFFF"><imgsrc="images/lib2.jpg"
width="118" height="8"></td>
</tr>
</table>
</td>
<td width="7%" align="right" valign="top"><imgsrc="images/lib3.jpg" width="6"
height="79"></td>
</tr>
</table></td>
</tr>
</table></td>
<td width="30%" align="center"><table width="203" border="0" cellspacing="0"
cellpadding="0">
<tr>

```

```

<td width="203" height="28" valign="top">&nbsp;</td>
</tr>
<tr>
<td height="79" valign="top"><table width="203" border="0" cellspacing="0"
cellpadding="0">
<tr>
<td height="19" valign="bottom" bgcolor="#FFFFFF"><imgsrc="images/lab2.jpg"
width="118" height="8"></td>
</tr>
</table></td>
<tr>
<td width="7%" align="right">&nbsp;</td>
</tr>
</table></td>
<tr>
<td width="30%" align="center"><table width="203" border="0" cellspacing="0"
cellpadding="0">
<tr>
<td valign="top">&nbsp;</td>
</tr>
<td class="main1111" height="46"><font color="#006666"><a
href="PatientLogin.jsp">Patient
Login </a></font></td>
</tr>
</table>
</body></html>

```

Search.jsp

```

<%@ page import="java.io.*,java.util.*,java.sql.*"%>
<%

```

```

Connection con=null;

//Class.forName(application.getInitParameter("ora_driver"));
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
con=DriverManager.getConnection("jdbc:odbc:medical","medical","medical");
String fetch="";
String nstr="";
String Qry="";
Statement st=null;
ResultSets=null;

//String assetid=request.getParameter("disname");
try{

    if(request.getParameter("disname")!=null ){
Qry="SELECT *from DISEASE_DETAILS WHERE dis_name
='"+request.getParameter("disname")+"'";
st=con.createStatement();
rs=st.executeQuery(Qry);%>
<%
        if(rs.next())
        {
            %>

document.cfrm.disname1.value="<%=rs.getString(3)==null)?nstr:rs.getString(3)%>";

            <%
        }
    }
}

```

```

    }catch(Exception e){
    e.printStackTrace();
    }
    %>

```

Machine Algorithm Code

```

<%@ page import="com.mysql.jdbc.Driver" %>
<%@page language="java" import="java.sql.*" %>

```

```

<%
    /*If matching disease is not there , go with the neighbor disease and display
    Scouts are the bees where theydon?t have a single match of food sources in
    hive
    If not even single symptom doesn?t match with the symptoms in the knowledge
    base then it displays knowledge is insufficient
    Display the best food source which shows the place in the hive
    Until our requirement is satisfied , loop is to be runned
    */
%>

```

```

<html>
<head>
<body bgcolor="#FFFFFF">
<table cellpadding="0" cellspacing="0" border="0" align="center">
<tr>
<td width="102" class="bgleft" style="
background-image:url(..images/bgleft.jpg)">&nbsp;&nbsp;&nbsp;</td>
<td>

```

```
<tr><td width="528" valign="top" class="style1">
```

```
<%
```

```
Connection con = null;
```

```
try {
```

```
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
```

```
con=DriverManager.getConnection("jdbc:odbc:medical","medical","medical");
```

```
ResultSets;
```

```
stmt = con.createStatement();
```

```
String st1 = "";
```

```
intst[] = new int[150];
```

```
floatav[] = new float[150];
```

```
int k = 0;
```

```
intyy = 0;
```

```
char cc;
```

```
int x = 0;
```

```
int n = 0;
```

```
float v = 0, temp = 0;
```

```
int id1[] = new int[150];
```

```
float z[] = new float[150];
```

```
double q[] = new double[150];
```

```
String m[] = new String[150];
```

```
String ss = "";
```

```
ss = request.getParameter("f1");
```

```
for (int i = 0; i < 34; i++) {
```

```
cc = ss.charAt(i);
```

```
if (cc == '1') {
```

```
st[i] = 1;
```

```
} else {
```

```
st[i] = 0;
```

```

    }
}

for (int i = 0; i < 5; i++);
for (int i = 0; i < 34; i++) {
    if (st[i] == 1) {

        n = n + 1;
    }
}

st1 = "select * from garlic_disease";

int found = 0, y = 0, c = 0;
rs = stmt.executeQuery(st1);
if (rs != null) {
    while (rs.next()) {

        //out.println(rs.getInt(1));

        found = 0;

        x = 0;
        c = 0;

        for (int i = 0; i < 34; i++) {

            y = rs.getInt(i + 3);

            if (y == 1) {
                c++;

            }

            if ((y == 1) && (st[i] == 1)) {

                found++;
            }

            if (y == st[i]) {
                x++;

            }

            id1[k] = rs.getInt(1);
            if (found != 0) {
                av[k++] = (float) x / 34;

```

```

                                } else {
av[k++] = 0;
                                }
                                }
                                }

for (int i = 0; i < 12; i++) {
    st1 = "update garlic_disease set avg=" + av[i] + " where id=" + id1[i];
    int r = stmt.executeUpdate(st1);
                                }
                                st1 = "select * from garlic_disease";

rs = stmt.executeQuery(st1);

                                k = 0;

int o = 0;

                                String cure[] = new String[12];

temp = 0;
if (rs != null) {
                                %>

<%
while (rs.next()) {
    v = rs.getFloat("avg");
    if (v >= 0.1) {
        m[o] = rs.getString(2);
        q[o] = rs.getFloat("avg");
        cure[o] = rs.getString("cure");
        o++;
        yy++;

```

```

out.println(m[o-1]+"</td><td>"+(float)(q[o-1]*100));
                                                                    }
                                                                    }

if (yy == 0) {
out.print("Insufficient Data ");
                                                                    }

                                                                    %>

<%
if (yy >= 1) {
double max = 0;
int xx = 0;
for (int i = 0; i < o; i++) {
if (q[i] > max) {
max = q[i];
xx = i;
}
}
                                                                    %>

<table border="1" align="center" bordercolor="green" width="500">
<tr><td align="center"><br>
<b>Effected with <% out.print(m[xx]);%> with optimization of <%
out.print(q[xx]*100);%>%</b></td></tr>
<tr>
<td>
<b> Cure Is : </b><%= cure[xx]%></td>
</tr>
</table>
<% }
con.close();

                                                                    } catch (Exception ex) {

con.close();

```

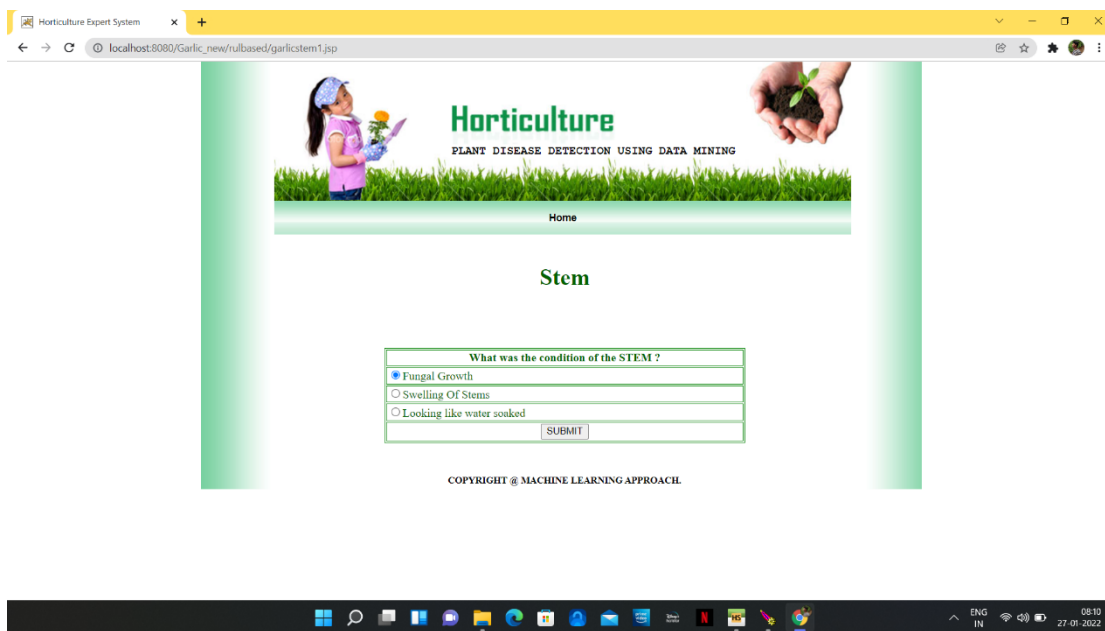
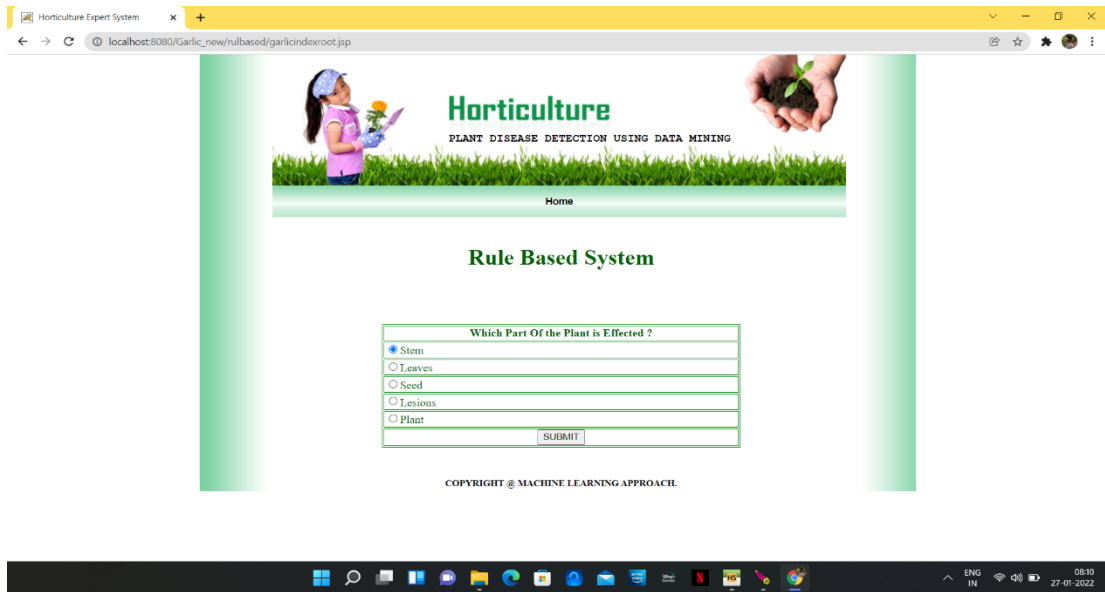
```

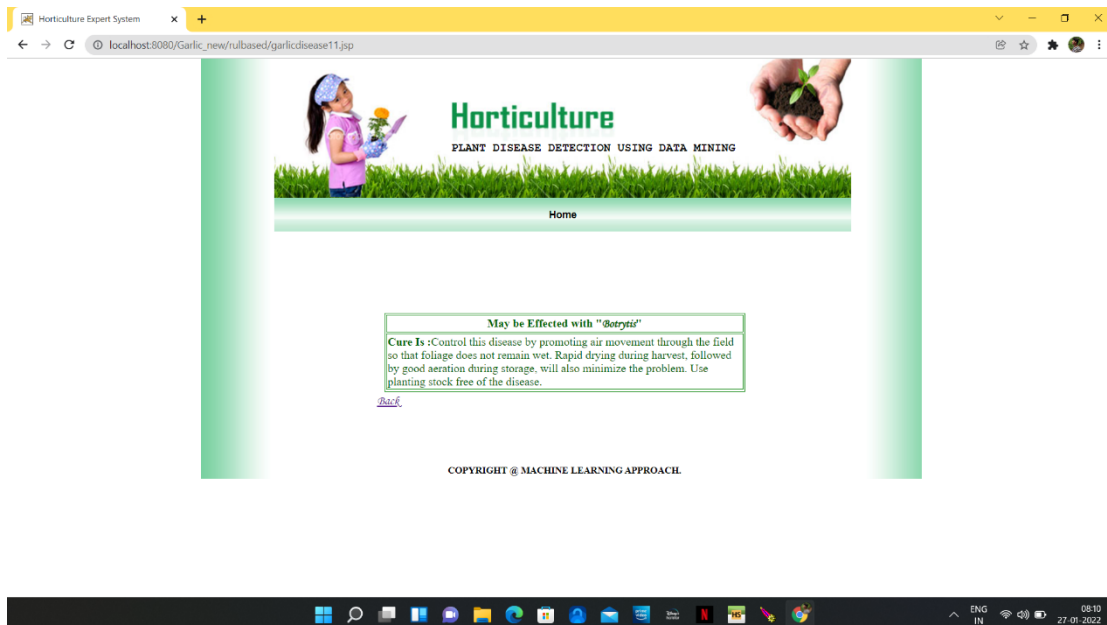
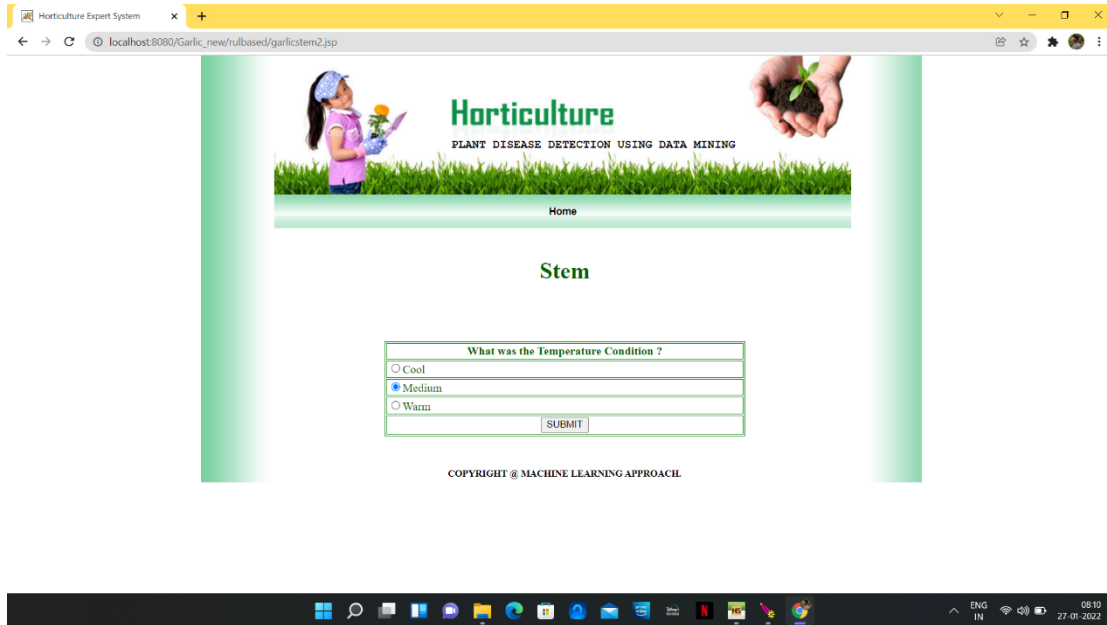
out.println(ex);
                                }
out.print("<br><br> ");
                                %>
</html>

```

SCREENSHOTS







Horticulture Expert System

localhost:8080/Garlic_new/expertsystem1/garlicexpert.jsp

Horticulture

PLANT DISEASE DETECTION USING DATA MINING

Home

Welcome to the Expert System of Whole Plant :

1. is " Foliage brown in color is found " present ?
☐ yes ☒ no
2. is " Greyish violet in color leaves are found " present ?
☐ yes ☒ no
3. is " Greyish violet in seeds " present ?
☐ yes ☒ no
4. is " Lesions violet to purple " present ?
☐ yes ☒ no
5. is " Leaves fold over " present ?
☐ yes ☒ no
6. is " Cool temperature " present ?
☐ yes ☒ no
7. is " High temperature " present ?
☐ yes ☒ no
8. is " Warm temperature " present ?
☐ yes ☒ no

Copyright © Machine Learning Approach

Horticulture Expert System

localhost:8080/Garlic_new/expertsystem1/garlicexpert.jsp

☒ yes ☐ no

25. is " Soils heavily infected " present ?
☐ yes ☒ no
26. is " Yellowing of leaves " present ?
☐ yes ☒ no
27. is " white growth of bulbs " present ?
☐ yes ☒ no
28. is " wilted and yellowing of plants " present ?
☒ yes ☐ no
29. is " reduced growth of plants " present ?
☐ yes ☒ no
30. is " yellowing of bulbs " present ?
☐ yes ☒ no
31. is " swelling of stems " present ?
☒ yes ☐ no
32. is " water soaked stem " present ?
☐ yes ☒ no
33. is " stored bulbs " present ?
☐ yes ☒ no
34. is " drying of leaves " present ?
☒ yes ☐ no

SUBMIT

Copyright © Machine Learning Approach

