Garden Buddy

A
Project Report
Submitted in partial fulfilment of the requirement for the award of degree of

Bachelor of Technology
In
Computer Science & Engineering

Submitted to
RAJIV GANDHI PROUDYOGIKI VISHWAVIDYALAYA,
BHOPAL (M.P.)

Guided by
Prof. Shailendra Kumar Mishra

Submitted By

Kanak Gupta	0832CS191083
Rudresh Koranne	0832CS201141
Siddharth Singh Bais	0832CS201161

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
CHAMELI DEVI GROUP OF INSTITUTIONS
INDORE (M.P.) 452020

2022-23

Garden Buddy

A Minor Project

Software Requirement Specification Report submitted to

Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal

Bachelor of Technology

in

Computer Science and Engineering

by

Kanak Gupta **0832CS191083**

Rudresh Koranne **0832CS201141**

Siddharth Singh Bais **0832CS201161**

Under the guidance of

Prof. Shailendra Kumar Mishra

Associate professor

Session: 2022-23

Department of Computer Science & Engineering

Chameli Devi Group of Institutions, Indore

452 020 (Madhya Pradesh)

DECLARATION

We certify that the work contained in this report is original and has been done by us under the guidance of my supervisor(s).

a. The work has not been submitted to any other Institute for any degree or diploma.
b. We have followed the guidelines provided by the Institute in preparing the report.
c. We have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.
d. Whenever we have used materials (data, theoretical analysis, figures, and text) from other sources, we have given due credit to them by citing them in the text of the report and giving their details in the references.

Name and Signature of Project Team Members:

S. No.	Enrollment No.	Name of students	Signature of students
1.	0832CS191083	Kanak Gupta	
2.	0832CS201141	Rudresh Koranne	
3.	0832CS201161	Siddharth Singh Bais	

CHAMELI DEVI GROUP OF INSTITUTIONS, INDORE

CERTIFICATE

Certified that the minor project report entitled, **"Garden Buddy"** is a bonafide work done under my guidance by **Kanak Gupta, Rudresh Koranne and Siddharth Singh Bais** in partial fulfillment of the requirements for the award of degree of Bachelor of Technology in **Computer Science and Engineering**.

Date: 21 November 2022

_____

(Prof. Shailendra Kumar Mishra)

Guide

_____ _____

(Prof. Shailendra Kumar Mishra) (Prof. Ankit Chakrawarti)

Head of the Department Project Coordinator

_____

(Dr. Manish Shrivastava)

(Principal, CDGI)

CHAMELI DEVI GROUP OF INSTITUTIONS , INDORE

ACKNOWLEDGEMENT

We have immense pleasure in expressing our sincerest and deepest sense of gratitude towards our guide **Mr**. **Shailendra Kumar Mishra** for the assistance, valuable guidance and co-operation in carrying out this Project work. We are developing this project with the help of Faculty members of our institute and we are extremely grateful to all of them. We also take this opportunity to thank Head of the Department **Mr**. **Shailendra Kumar Mishra**, and the Principal of Chameli Devi Group of Institutions, **Manish Shrivastava**, for providing the required facilities for the project work . We are greatly thankful to our parents, friends and faculty members for their motivation, guidance and help whenever needed.

Name and signature of team Members:

1. Kanak Gupta .....
2. Rudresh Koranne .....
3. Siddharth Singh Bais .....

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Garden Buddy is a web application. Here we are try to make gardening smart to people who are enthusiast toward gardening. We saw problem that people regularly faced in gardening, is that plant is not grow properly, spots on leaf, yellowish color of leaf etc. These all are symptoms of poor health. So, we decide to built this system.

Plant leaf disease has been one of the major threats to food security since long ago because it reduces the yield and compromises its quality. Health monitoring and disease detection on plant is very critical for sustainable agriculture. It is very difficult to monitor the plant diseases manually. It requires tremendous amount of work, expertize in the plant diseases, and also require the excessive processing time.

Diagnosis of accurate diseases has been a major challenge and the recent advances in computer vision made possible by deep learning has paved the way for camera assisted disease diagnosis for plant leaf. It described the innovative solution that provides efficient disease detection and deep learning with convolutional neural networks (CNNs) has achieved great success in the classification of various plant leaf diseases. A variety of neuron-wise and layer-wise visualization methods were applied and trained using a CNN, with a publicly available plant disease given image dataset. So, it observed that neural networks can capture the colors and textures of lesions specific to respective diseases upon diagnosis, which can act like human decision-making.

TABLE OF CONTENTS

Chapter 1
Introduction

Developed Technologies have provided new methods. But still, the safety and security of gardening plant remained unattained. Factors like change in climate, the decline in pollinators, Plant disease, and others are challenging to the gardening. An important foundation for these factors needs to be attained on a priority basis. Making use of analysis and detection processes using present technology helps the gardener to get rid of such problems. During pandemic situations like COVID 19 the nation is dependent on the recent technologies to prevent address the issues to reduce the transmission of the diseases. As plant diseases are a significant threat to human life as they may lead to droughts and famines. The use of technologies like Computer vision and Machine Learning (ML) helps to fight against diseases .In this project, we are using ML to give a solution to Plant Diseases. In this method, we have divided the process into three stages Identity, Analyse and Verify with the Available database and then provide cure for that disease.

The key issues and challenges are identified by the researchers and the scientists, while analysing the leaf diseases of plant. Some of them are as follows:

1. The quality of the leaf image must be high.
2. Publicly available Dataset requirement.
3. Noisy data affecting the leaf samples.
4. Through the process of segmentation, diseases may be identified but the samples must undergo training and testing.
5. Classification is one more challenge, in the stage of detecting the leaf diseases.
6. Color of the leaves may be varied due to environmental effect.
7. Variety of diseases can be seen in various kinds of plants, so detection of disease is quite difficult.

Based on the challenges discussed above and combined techniques using image processing and machine learning, the proposed model provide better accuracy. Keeping all these things in mind, in this project an algorithm based on image processing and machine learning tools to automatically detect leaf diseases is proposed.

1.1 Rationale

Identification of the plant diseases is the key to preventing the losses in the yield and quantity of the agricultural product. The studies of the plant diseases mean the studies of visually observable patterns seen on the plant. Health monitoring and disease detection on plant is very critical for sustainable gardening. The main aim of this project is to provide help people in gardening. By detecting disease in plant and provide them cure for that.

1.2 Goal

Lot of people love gardening in daily life but due to lack of knowledge and time they cannot properly take care of their plants. Plants get infected with disease and not get proper treatment which cause slow growth of plant or some time plant wither. It requires tremendous amount of work, expertize in the plant diseases, and also require the excessive processing time. Hence, garden buddy is creating system to deal with this problem by image processing and Machine learning techniques used for the detection of plant diseases.

1.3 Objective

We can reduce the attack of pests by using proper pesticides and remedies .We can reduce the size of the images by proper size reduction techniques and see to it that the quality is not compromised to a great extent. We can expand the projects of the earlier mentioned authors such that the remedy to the disease is also shown by the system. The main objective is to identify the plant diseases using image processing. It also, after identification of the disease, suggest the name of pesticide to be used. It also identifies the insects and pests responsible for epidemic.

Thus the main objectives are:

1) To design such system that can detect plant disease and pest accurately.
2) Create database of insecticides for respective pest and disease.
3) To provide remedy for the disease that is detected

1.4 Methodology:

We have followed the Software Development Life Cycle shown in **fig 1.1** and utilizing which, ended up with the final product. The very first part was to figure out what exactly is the problem, what are the existing solutions, and where they fail. Following the analysis part, we design the basic architecture and visualize the product with diagrams and flow charts. Moving on to the implementation part, we have followed the industry standard for projects of such level and the proper Tech Stack Conventions. From writing a production ready ReactJS JavaScript frontend library to easy-to-understand CSS using bootstrap, font-awesome, toast, sweet-alert etc. For backend python and Django framework. And use machine learning algorithm. Once the coding is done, then comes the testing part where we tried testing this project under various circumstances like responsiveness for mobile-design, slow-internet connection, healthy relationship between API calls. For the deployment part we have used the most popular hosting service known as AWS for easy and fast builds. Once this is done, the next stage is to keep it up with the industry standards as Tech keeps evolving like the UI/UX part, providing faster API Calls and more.

Fig 1.1 Software Development Life Cycle

Diseases of tomato plants database:

Symptoms	Disease	Remedy
Tomato early blight	Alternaria Solani	Bacillus Subtilis, Hydroperoxyl
Tomato late blight	Phytophthora Infestance	Actinovate, copper

Table 1.1: Diseases

Dataset:

The database of tomato leaf is considered, the plants which are affected from variety of diseases. The images of tomato leaf having several disorders are taken to carry out the experiments for detection of leaf disease. The samples of leaf images in the database are shown in **fig 1.2.**

Early Blight

Late Blight

Fig 1.2 Leaf Dataset

1.5 Role

S. No.	Enrollment No.	Name of Member	Responsibility	Role
1.	0832CS191083	Kanak Gupta	Back-End and ML	Connecting Frontend with backend and database. Applying ML algorithm.
2.	0832CS201141	Rudresh Koranne	Back-End	Designed backend and database structure.
3.	0832CS201161	Siddharth Singh Bais	Front-End	Building User Interface using ReactJS and flow- design.

Table 1.2: Role

1.6 Contribution of Project

1.6.1 Market Potential

- The India gardening market is estimated to witness rise in revenue from about 25-30% years on years.
- The market is registering a CAGR of 4.19% during the forecast period 2022-2030.

1.6.2 Innovativeness:

Plant disease identification by visual way is more laborious task and at the same time, less accurate and can be done only in limited areas. Whereas if automatic detection technique is used it will take less efforts, less time and become more accurate. In plants, some general diseases seen are brown and yellow spots, early and late scorch, and others are fungal, viral and bacterial diseases. Image processing is used for measuring affected area of disease and to determine the difference in the color of the affected area.
Image segmentation is the process of separating or grouping an image into different parts. There are currently many different ways of performing image segmentation, ranging from the simple thresholding method to advanced color image segmentation methods. These parts normally correspond to something that humans can easily separate and view as individual objects. Computers have no means of intelligently recognizing objects, and so many different methods have been developed in order to segment images. The segmentation process is based on various features found in the image. This might be color information, boundaries or segment of an image. Image is segmented using the K-means clustering technique. Then unnecessary part (green area) within leaf area is removed. After that we calculate the texture features for the segmented infected object. Finally, the extracted features are passed through a pre trained neural network

1.6.3 Usefulness

We planned to design the module so that a person with no knowledge about programming can also be able to use and get the information about the plants disease. It proposed system to predicting leaf diseases. It explains about the experimental analysis of our methodology. Different number of images is collected for each disease that was classified into database images and input images. The primary attributes of the image are based upon the shape and texture oriented features.

Chapter 2
System Features

2.1 Functional Requirement

- The software will be designed to run on regular machines which have mid specs like office Computers, laptops, basic home systems, etc.
- Admin can monitor the whole system.
- Registration of user can done by itself.
- The user will maintain few features and overview his profile.
- User can easily upload picture.

2.2 Non-Functional Requirement

- Try to put no bloat and unnecessary details that might confuse the user.
- Keep everything straight forward still attractive to us

2.2.1 Performance Requirement

- The login, search, logout should not conflict and not reduce the overall performance.
- The database shall be able to accommodate a minimum of 1,000 records of websites.
- The software shall support the use of multiple tabs at a time.
- Image should be of high resolution for better performance.
- There is no other specific performance requirement that will affect development.

2.2.2 Safety Requirement

- The user must not be forced to provide permission for their data unnecessarily.
- The requested login credentials with access should only be stored temporarily to provide services.
- All other details such as IP Addresses, etc., should also be removed as soon as the user finishes their task

2.2.3 Security Requirement

- System needs to control the user access and session.
- It needs to store the data in a secure location and stored in a secure format.
- It requires a secure communication channel for the data.

2.3 External Interface Requirement

2.3.1 Hardware Interface

S. No.	Hardware	Requirement
1.	Processor	2.5 gigahertz (GHz) frequency or above
2.	RAM	A minimum of 4 GB of RAM.
3.	Hard disk	A minimum of 20GB of available space.
4.	Input devices	High resolution camera
5.	Monitor	Minimum Resolution 1024 X 768

Table 2.1: Hardware Requirement

2.3. Software Interface

S. No.	Software	Requirement
1.	Operating Software	Windows 8 and above
2.	Technology used	Machine learning
3.	Platform	Jupyter or PyCharm
4.	Supporting libraries	Numpy, Pandas, Tensflow, Matplotlib

Table 2.2: Software Requirement

2.3 Communication Interface

- The Customer must connect to the Internet to access the Extension.
- Dialup or Broadband Connection (52 kbps min) with Internet Service.

Chapter 3
Analysis & Design

UML Diagram

UML stand for **Unified Modeling Language.** Unified Modeling Language is a general purpose modelling language. The aim of UML is to define a standard way to evoke the way a system has been designed. It is similar to blueprints used in other fields of engineering.
UML is not a programming language, it is rather a visual language. We use UML diagrams to draw the behavior and structure of a system. UML benefit software engineers, businessmen and system architects with modelling, design and analysis.

3.1 System Architecture

A software is designed architecture is a set of principles that define the way and developed. An architecture defines the structure of the software system and how it is organized. It also describes the relationships between components, levels of abstraction, and other aspects of the software system. An architecture can be used to define the goals of a project, or it can be used to guide the design and development of a new system. A software architecture is a set of principles that define the way software is designed and developed. An architecture defines the structure of the software system and how it is organized.

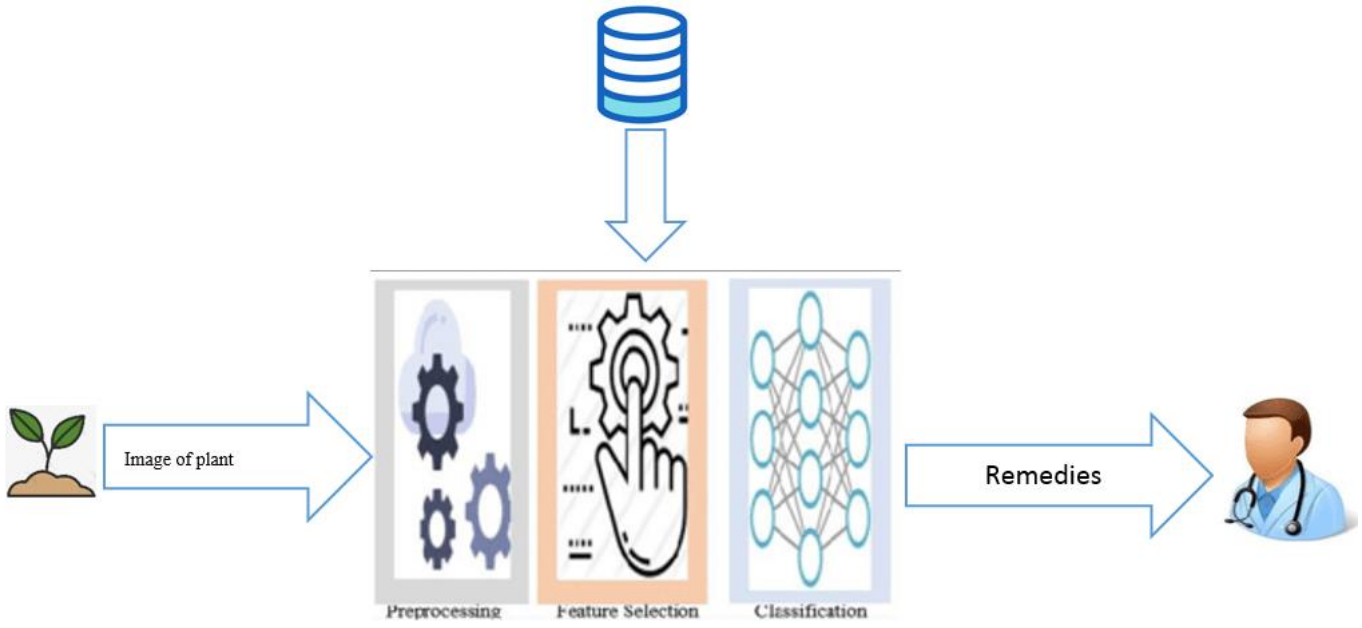

Fig 3.1: System Architecture

Figure 3.1 is system architecture of project in which srchitecture of garden buddy is briefly explained. Firstly user click the image of plant, then system preproces it after which feature extraction is done then classification. After which image compare with database image. If infected the Remaedies according to disease is provided.

3.2 Use-case Diagrams:

Use case diagram is a graphic depiction of the interactions among the elements of a system. Use cases will specify the expected behaviour, and the exact method of making it happen. Use cases once specified can be denoted both textual and visual representation.

Use case diagrams are used to specify:

- Requirements (external), required usages of a system under design or analysis – to capture what the system is supposed to do.
- The functionality offered by a subject – what the system can do.
- Requirements the specified its environment - by defining how environment should interact with the subject so that it will be able to perform its services.

When the initial task is complete, use case diagrams are modelled to present the outside view.

- **Use cases**: Horizontally shaped ovals that represent the different uses that a user might have.
- **Actors**: Stick figures that represent the people actually employing the use cases.
- **Associations**: A line between actors and use cases. In complex diagrams, it is important to know which actors are associated with which use cases.
- **System boundary boxes**: A box that sets a system scope to use cases. All use cases outside the box would be considered outside the scope of that system. For example, Psycho Killer is outside the scope of occupations in the chainsaw example found below

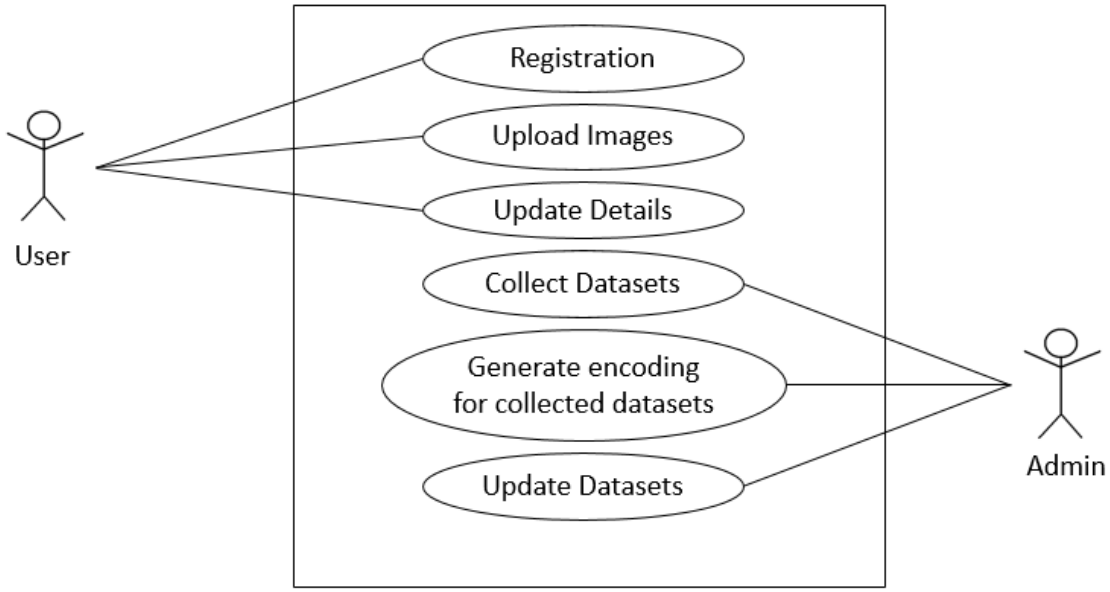

Fig 3.2: Use case

Fig 1.2 use case for garden buddy, here two actor are present one is user and another is admin. User can do registration, upload image, upload details and admin can collect datasets, generate encoding for collected datasets, update datasets.

3.3 Activity Diagrams

Activity diagram is another important diagram in UML to describe dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as in operation of the system. The control flow is drawn from one operation to another.
The only missing thing in the activity diagram is the message part. It does not show any message flow from one activity to another. The activity diagram is sometimes considered as the flowchart. Although the diagrams look like a flowchart, they are not. It shows different flows such as parallel, branched, concurrent, and single.

Activity diagrams symbols can be generated by using the following notations:

- **Initial states:** The starting stage before an activity takes place is depicted as the initial state.
- **Final states:** The state which the system reaches when a specific process end is known as a Final State.
- **State or an activity box:** Any type of particular action is defined in this box.
- **Decisionbox:** It is a diamond shape box which represents a decision with alternate paths. It represents the flow of control.

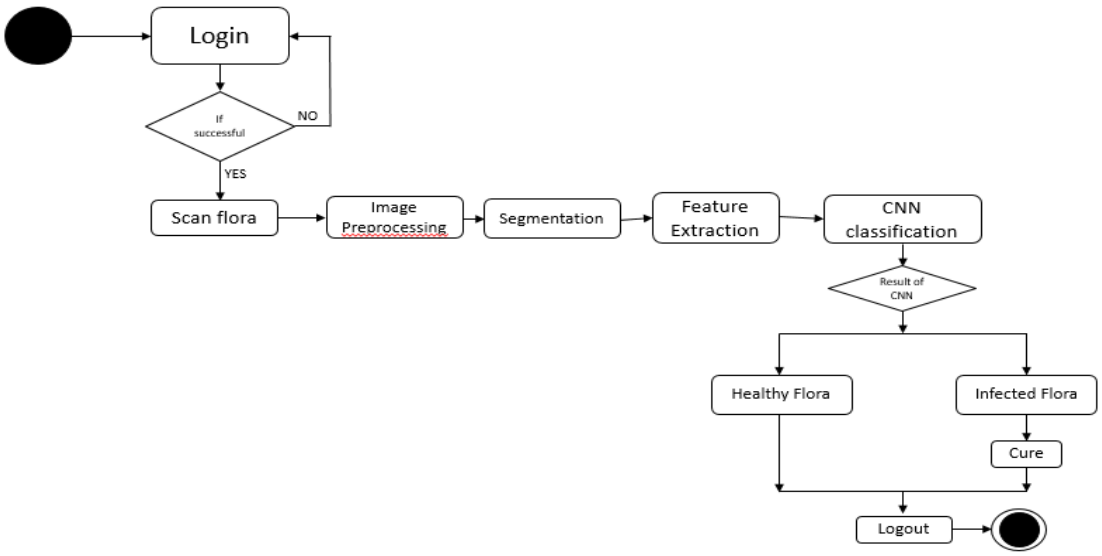

Fig 3.3: Activity Diagram

The fig 1.3 represents the flow from one activity to another activity, the activity starts from login, if successful scan flora through camera, and then input leaf is pre-processed and extract the features like color, shape, texture and so on. Then segmentation, feature extraction and CNN algorithm will run. Now, the processed image is classified as Normal or Abnormal, if Abnormal is found in the leaf, then remedies will be suggested.

3.4 Sequence Diagrams

Sequence diagrams are sometimes called event diagrams or event scenarios. A sequence diagram shows as parallel vertical lines (lifelines), different processes or objects that live simultaneously and as horizontal arrows, the messages exchanged between them in the order in which they occur.

1. Usage scenarios: A usage scenario is a description of a potential way your system is used. The logic of a usage scenario may be part of a use case, perhaps an alternate course. It may also be one entire pass through a use case, such as the logic described by the basic course of action or a portion of the basic course of action, plus one or more alternate scenarios. The logic of a usage scenario may also be a pass through the logic contained in several use cases.

2. The logic of methods: Sequence diagrams can be used to explore the logic of a complex operation, function, or procedure. One way to think of sequence diagrams, particularly highly detailed diagrams, is as visual object code

3. The logic of services: A service is effectively a high-level method, often one that can be invoked by a wide variety of clients. This includes web-services as well as business transactions implemented by a variety of technologies

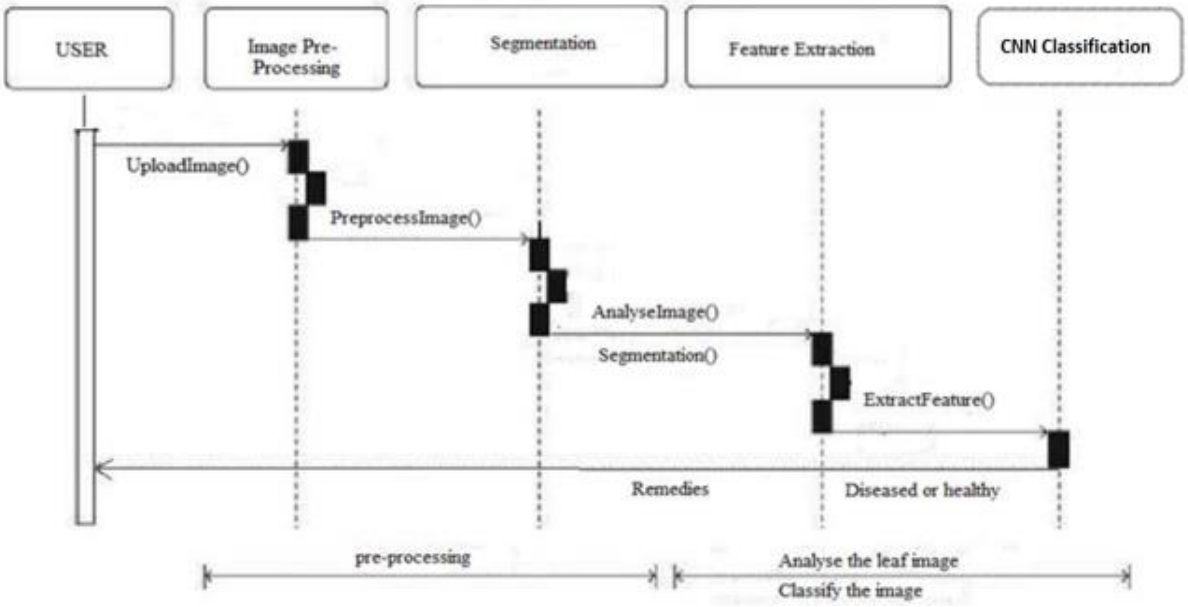

Fig 3.4: Sequence Diagram

The fig 1.4 represent the sequence of system, where user start from uploading image the image under go from preprocessing after which segmentation and feature extraction is done. Then CNN algorithm applied after which if plant is infected them remedies are suggested.

3.5 E-R Diagram:

ER Diagram stands for Entity Relationship Diagram, also known as ERD is a diagram that displays the relationship of entity sets stored in a database. In other words, ER diagrams help to explain the logical structure of databases.

ER diagrams or ERD's are composed of three main elements: entities, attributes, and relationships.

- **Entities** - typically displayed in a rectangle, entities can be represented by objects, persons, concepts, or events that contain data.
- **Attributes** - displayed in a circle or an oval, the attributes refer to the characteristics of an entity. They can be categorized as simple, composite, or derived, and an object can have one or multiple attributes.
- **Relationships** - illustrate how two or more entities interact with each other. They are displayed as labels placed on the lines connecting the objects.

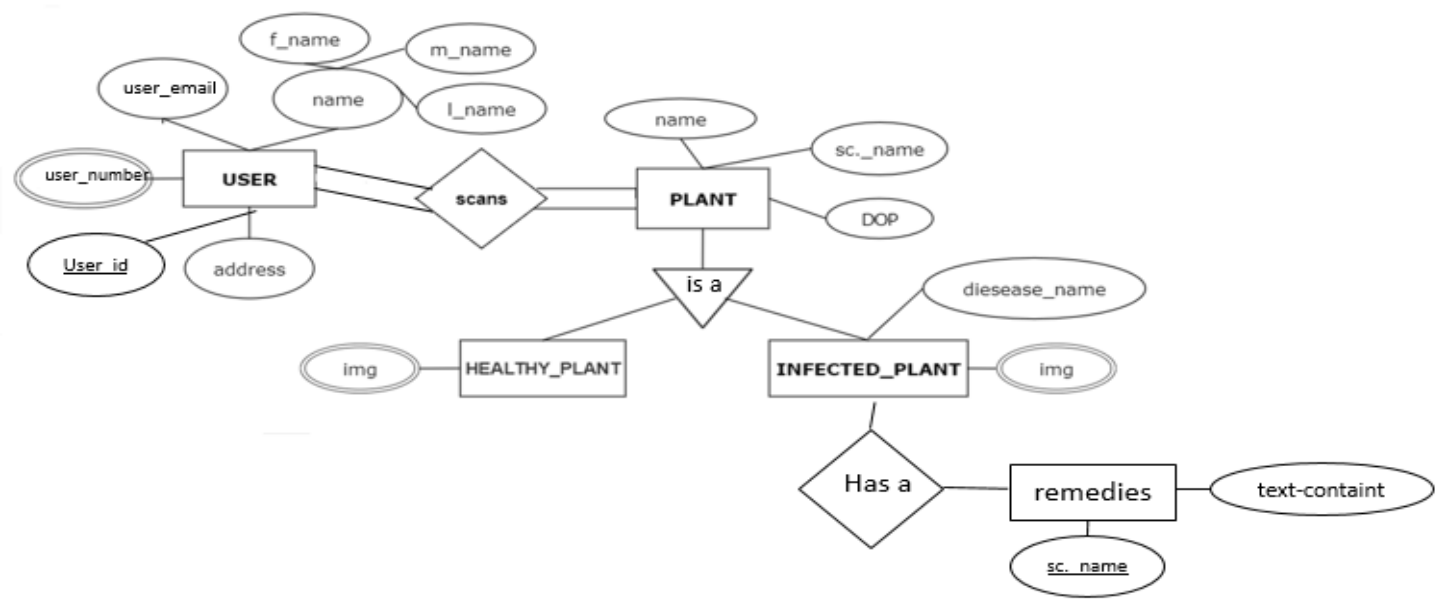

Fig 3.5: Entity-Relationship diagram

The fig 1.5 represents relation between user and plant, the relation starts from entity user which have attributes user_email, user_number as multivalued attribute, user_id as primary attribute, address, name as composite attribute and user have many to many relation with entity plant which have attributes name, DOP, sc_name and have is-a relation with healthy_plant and infected_plant. Healthy_plant have img as multivalued attribute. Infected_plant have attributes disease_name, img as multivalued attributes and remedies as composite attribute.

3.6 Class Diagram

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

The purpose of class diagram is to model the static view of an application. Class diagrams are the only diagrams which can be directly mapped with object-oriented languages and thus widely used at the time of construction.

UML diagrams like activity diagram, sequence diagram can only give the sequence flow of the application, however class diagram is a bit different. It is the most popular UML diagram in the coder community.

The purpose of the class diagram can be summarized as −

- Analysis and design of the static view of an application.
- Describe responsibilities of a system.
- Base for component and deployment diagrams.
- Forward and reverse engineering.

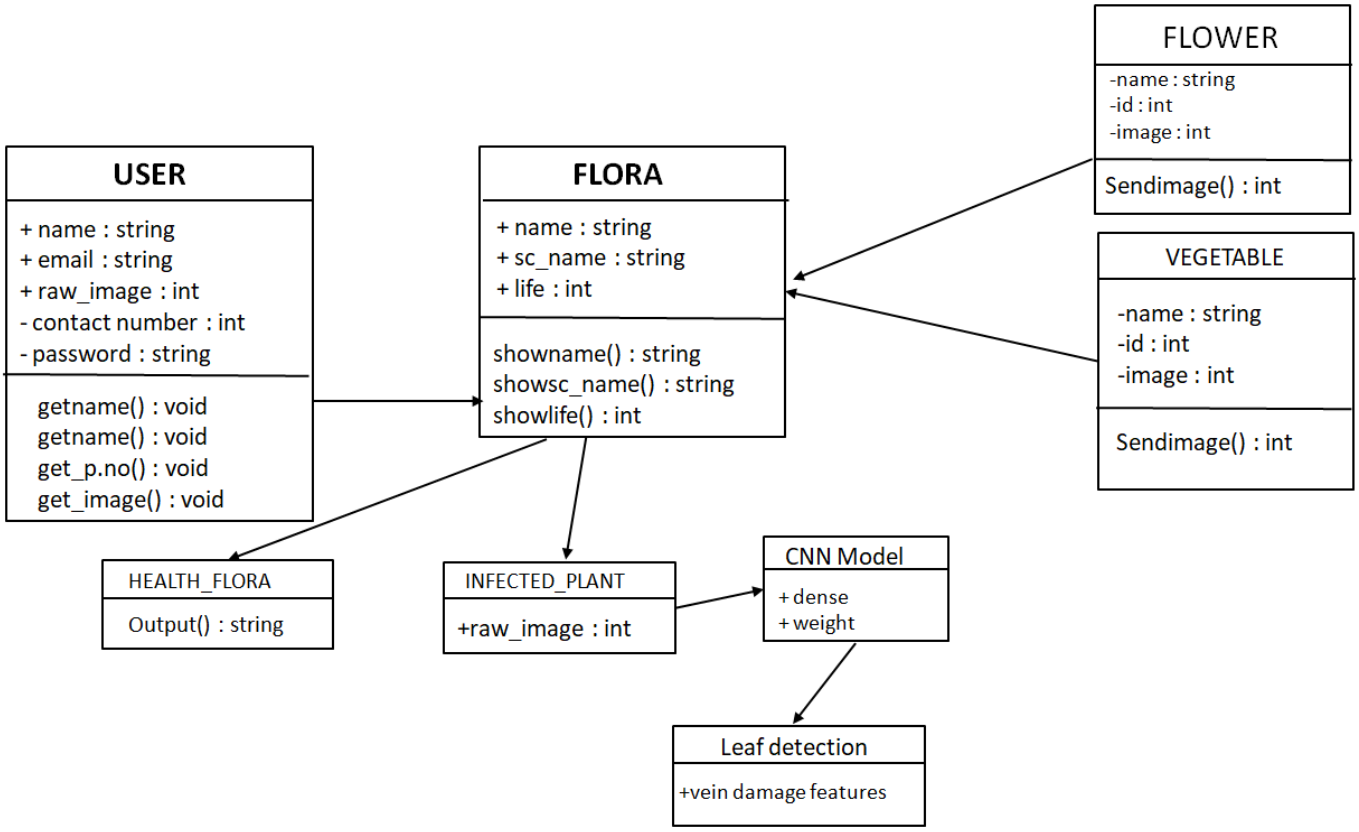

Fig 3.6: Class Diagram

The above class diagram represents different classes. In user class there is variables like name, email, raw_image, contact number, password and functions are getname(), get_p.no(), get_image(). Next, we have class flora which is inherited with flower and vegetable where flower have variables name, id, image on other hand vegetable have variables name, id, image and both have same function Sendimage(). In class flora variables are name, sc_name, life and function is showname(), showc_name(), Showlife(). Next, there is health_flora and infected_plant as class which have VariablesOutput and raw_image respectively.

Chapter 4
Conclusion

It focused how image from given dataset (trained dataset) in field and past data set used predict the pattern of plant diseases using CNN model. This brings some of the following insights about plant leaf disease prediction. As maximum types of plant leaves will be covered under this system, gardener may get to know about the leaf which may never have been cultivated and lists out all possible plant leaves, it helps the gardener in decision making.

There are number of ways by which we can detect disease of plants and suggest remedies for them. Each has some pros as well as limitations .On one hand visual analysis is least expensive and simple method, it is not as efficient and reliable. Image processing is a technique which is most spoken for very high accuracy and least time consumption are major advantages offered. The applications of K-means clustering and Neural Networks (NNs) have been formulated for clustering and classification of diseases that effect on plant leaves. Recognizing the disease accurately and efficiently is mainly the purpose of the proposed approach. The experimental results indicate that the proposed approach is a valuable approach, which can significantly support an accurate detection of leaf diseases in a little computational effort.

References

1. Sunil S. Harakannanavar, Jayashri M. Rudagi, Veena I. Puranikmath, Ayesha Siddique, R Pramodhini "**Plant leaf disease detection using computer vision and machine learning algorithms**", Department of Electronics and Communication Engineering, 2022. (https://www.sciencedirect.com/science/article/pii/S2666285X22000218#!)

2. P Krithika, S Veni
   Leaf disease detection on cucumber leaves using multiclass support vector machine
   IEEE International Conference on Wireless Communications, Signal Processing and Networking (2017), pp. 1276-1281

3. R Prakash, G P Saraswathy, G Ramalakshmi
   Detection of leaf diseases and classification using digital image processing
   IEEE International Conference on Innovations in Information, Embedded and Communication Systems (2017), pp. 1-4

4. B Mishra, S Nema, M Lambert, S Nema
   Recent technologies of leaf disease detection using image processing approach-review
   IEEE International Conference on Innovations in Information, Embedded and Communication Systems (2017), pp. 1-5

5. C. Puttamadappa, B.D. Parameshachari
   Demand side management of small scale loads in a smart grid using glow-worm swarm optimization technique
   Microprocessors Microsyst 71 (2019), Article 102886

6. V Pooja, R Das, V Kanchana
   Identification of plant leaf diseases using image processing techniques
   IEEE International Conference on Technological Innovations in ICT for Agriculture and Rural Development (2017), pp. 130-133