**Chapter 1**

**Project Overview**

**1.1: Introduction**

India, being an agricultural country, depends on the production of crops to feed its people and maintain the economy. Around 70 percent population of India live in rural zones. More than 56 percent of India's workforce is directly employed by agriculture and it contributes 17.4% to the country's Gross Value Added. A greater part of the rural population depends on agriculture for most of their earnings. The country needs a huge supply of crops every year. Plant diseases inhibit the production of crops often. As a result, the price of food gets higher and poor people have to stay half-fed or unfed. Hence, Plant diseases have become a great threat to crops and the community. Moreover, due to the illiteracy of farmers, most of the time they cannot understand what’s going with the crops. This causes them great sorrow. With the increase in the human populace and the decrease in croplands, the production rate of crops decresed. Controlling plant diseases is one of many ways to keep the crop production rate above the required margin. Recent improvements in technology have created remarkable opportunities in developing countries like India. Android mobile phones are now cheap and affordable for lower-earning people. An automated system can greatly help the farmers to diagnose crop diseases easily and take actions accordingly to avoid the waste of crops. Hence, we got motivated to create an android application that can detect the diseases of plants from captured images of leaves. Researchers around the world have taken the plant disease problem seriously and are searching for ways to prevent plant diseases and detect them at an early stage. Many researchers have approached various techniques to identify plant diseases. Image Processing, Machine Learning, Deep Learning are some of the techniques being used to try to solve the problem. By using TensorFlow, the models trained using deep convolutional neural network (CNN) can be used in wide varieties of mobile devices, even low-end devices. TensorFlow lite models use very little hardware resources which is why it is supported in most mobile devices. This study aims to develop a system for the detection of grape plant diseases through deep CNN and image processing. Furthermore, the study aims to use the developed model to create a user-friendly android application named ‘APP NAME’ to detect plant disease in real time. Plants show a range of symptoms such as colored spots, or streaks occurring on the leaves when they are diseased. As the disease advances, the visual symptoms change their color, shape, and size. With the help of deep CNN, we can train these patterns and create a model to recognize them. By using the application, farmers can capture the images of plant leaves using any mobile camera of average quality. The image is then processed and cross-matched with the integrated model to identify the disease and provide solutions based on the detected disease. Thus, the farmers can save their time and money as well as their crops.

**1.2 Problem Statement**

This project aims to develop an android application to detect and identify plant diseases through deep convolutional neural network. Plant disease is a critical issue in agricultural countries like India. Every year production of crops sustains heavy loss due to diseases. It is difficult to detect plant diseases with human eyes. So it is essential to build an automated system to detect the diseases. The proposed disease detection model takes an image of a plant leaf as input, processes it and uses deep convolutional neural network to detect and identify the disease. The model is trained with a large dataset of more than 10,000 images consisting of 3 different species and 12 different diseases. The trained model is lightweight and is used to build an android application. The application also provides solutions based on detected disease. The developed mobile application is user-friendly and can be used by farmers without much technical knowledge.

**1.3 Product Scope**

This project spans lots of grounds starting from business concept to real world, and required to perform several researches to achieve the project objectives. The areas include:

* Agriculture Industry: This includes study on how the disease detection system business is running in market, and also involves the process of how to improve the existence of the business.
* Firebase (No-SQL) Technology used for the development of the application.
* General users as well as the different researchers, companies will be able to use the system.
* App-platform means that the app will be available for access 24/7 except when there is an App maintenance or Internet issues which is minimal.

**1.4 Aims & Objectives**

Our project is based on finding the different diseases of potato, grape, and apple plant leaf using Deep Learning (CNN) and implementation in Android.

The android application is developed for the purpose to provide the users with an easy way to interact with the app to detect what is going on in their plant leaf.

This app basically helps farmers to protect their crops. This app comes with a handy camera integrated to allow the farmer to click an image of the affected crop that he wishes to diagnose.

This image is processed in the backend using a machine learning model (tflite model) to classify the leaf disease (make sure we take the image of the affected leaf, otherwise it will give a hazardous result). Further, it gives information about the steps that need to follow on how to improve the plant health.

The main objectives of this application are:-

1. To help the farmer to identify the diseases that affect their crops.
2. Make a user friendly platform to detected diseases from the crop leaves.
3. To ease the task of the farmer to identify the disease with the help of technology which is better than naked eye.

**1.5 Features:**

* **Admin:** Admin is the one who administers the system and input updates.
* **User login:** Users have to create an account into the system by registering themselves. Then they may login into the system and can utilize services. User’s data is stored in the Firebase Cloud Firestore. And user’s data is stored in the Firebase Storage.
* **Detection:** The system contains a TensorFlow lite model that can detect the disease from the leaf image that is taken by the camera using the app. After taking the image from the camera it process the image using deep learning algorithm then from the model we get three best results. Using the Priority queue method from the three best result we picked the highest percentage and show it to the user.
* **Cure and Precautions:** After getting the result of the image, user can check the cure and precautions of the diseases.
* **Check Weather:** User can check the weather of the current location as well as the different places by searching. User can get the information of the current date as well as the next 5 days weather report also.

**Chapter 2**

**Project Analysis**

**2.1 Gantt Chart**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Task Name | July 1 | July 16 | July 29 | Aug 15 | Aug 24 | Sep 20 |
| Planning |  |  |  |  |  |  |
| Research |  |  |  |  |  |  |
| Design |  |  |  |  |  |  |
| Implementation |  |  |  |  |  |  |
| Follow Up |  |  |  |  |  |  |

**2.2 Project Analysis**

**SWOT** (Strengths, Weaknesses, Opportunities, and Threats)

**User’s Service:**

Our user service is good. User can easily access the app from the google play store and install it in the mobile phone (Android). And with minimum steps they can detect the disease from the leaves of the crop and get the precaution at the time of detection.

**Keep updating the model:**

This is one of the weakness in the model is that we need to keep updating the model with new images.

**Marketing campaign:**

In order to improve the download we’ll run ads on YouTube, Facebook, and Instagram. Also, we’ll add ads on the local news, colleges, schools, village panchayats, etc.

**New Competitors:**

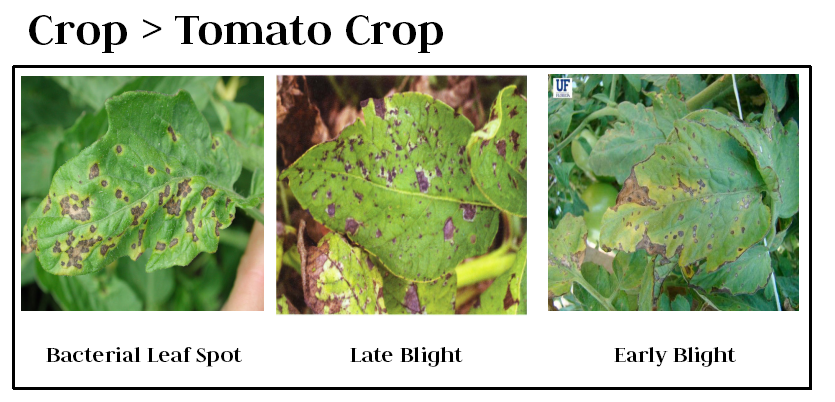
This is a new and exponentially increasing market. Every year a new competitor add in the market which would decline our customers.

**Chapter 3**

**Project Lifecycle**

**3.1 Product Perspective**

Since before, farmers usually detect the crop diseases with their naked eye that makes them take a tough decisions on which fertilisers to use, how much amount of fertiliser should use. It requires detailed knowledge the types of diseases and lot of experience needed to make sure the actual disease detection. Some of the diseases look almost similar to farmers often leaves them confused. Look at the below image for more understanding.

***(Use grape leaves or potato. Remove tomato.)***

As you see, three of them looks similar. In the case like this, the farmer makes wrong predictions and uses the wrong fertilizers or more than the normal dose, it will mess up the whole plant (or) soil and cause enough damage to plant and fields.

## So, How to prevent this from happening?

To prevent this situation we need better and perfect guidance on which fertilizers to use, to make the correct identification of diseases, and the ability to distinguish between two or more similar types of diseases in visuals.

This is where **Convolutional Neural Networks**comes handy. In short CNN

**3.2 Benefits**

* This app is fully functional and flexible.
* It is very easy to use.
* This app will help users to handle the different diseases precautions.
* It saves a lot of time, money, and work.
* Eco-friendly: The app includes the least paperwork, handy to use, easy to detect disease.
* The application acts as an office that is open 24/7.
* It also provides us the weather report of the current location.

**3.3 Users and Characteristics:**

**3.3.1 Admin:**

* Maintain contents.
* Update the database.
* Update the Crops Details.

**3.3.2 User:**

* User can log in to the system using Email/Password authentication sign-in method.
* Detect the crop disease.
* Get the cure of the diseases.
* Watch weather report.

**Chapter 4**

**Project Design**

**4.1 Activity Diagram**

**4.1.1 User App (User Registration)**

End

Start

No

Yes

Go to the Home Page

If valid

Enter valid Email and Password

Enter Email and Password

**4.1.2 Classification**

End

Start

Display the classification result

Add the image

Model run in background

**4.1.3 Detect Page**

End

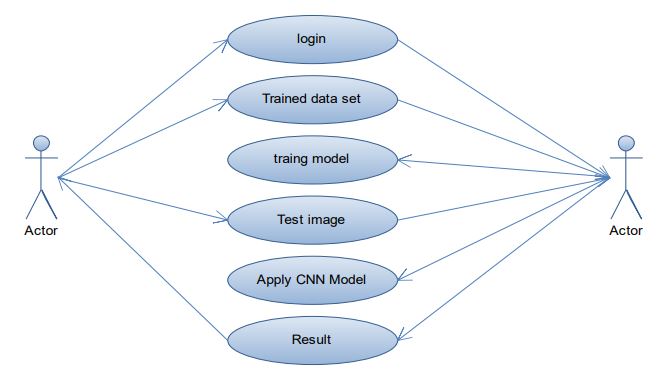
Start

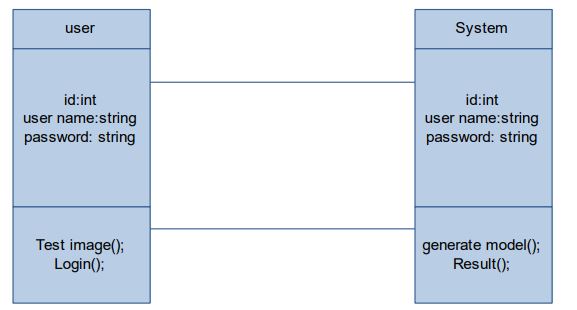
Get the Precaution

Or cure

Take Image from Camera & Gallery

Get the Result

**4.2 Use Case Diagram**

**4.3 Class Diagram**

**4.3 Sequence Diagram**

Sequence diagrams are used to demonstrate the behaviour of objects in a use case by describing the objects and the messages they pass. It provides a graphical representation of object interactions over time. Sequence diagrams show an actor, the objects and components they interact with in the execution of a use case. One sequence diagram represents a single Use Case 'scenario' or events.

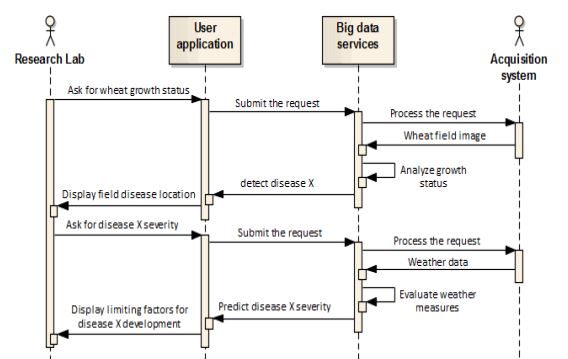
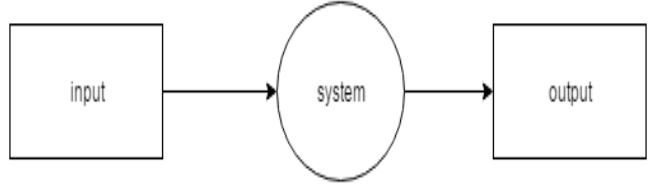


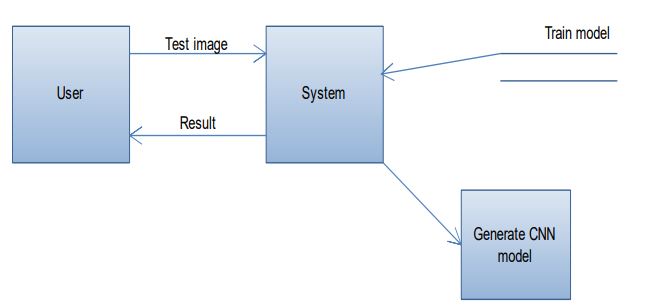
Figure: Sequence Diagram

**4.4 E-R Model**

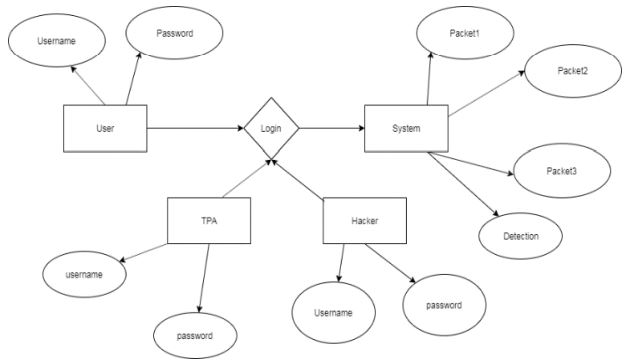
The entity relationship diagram describes the relationship between entities, cardinality, and their attributes. Entity–relationship model (ER model) is a data model for describing the data or information aspects of a business domain or its process requirements, in an abstract way that lends itself to ultimately being implemented in a database such as a relational database. The main components of ER models are entities (things) and the relationships that can exist among them. In here we describe entities with all their attributes.



**Figure: In Data flow diagram, DFD Level 0 DFD Level 1**

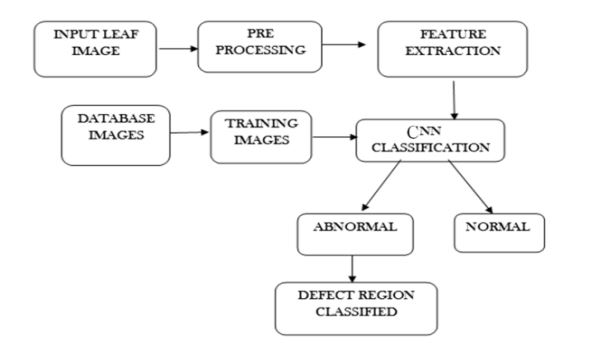


**Figure: DFD Level 1**



**Figure: Entity Relationships Diagrams**

**4.5 System Architecture**



**Figure: System Architecture**

**Chapter 5**

**Project Coding**

**5.1 Coding**

This project is design in the Android Studio. Java and XML used as the frontend language. And in the backend Firebase (NoSQL) database is used to store various data. Firebase Authentication is used to verification of the user using Email/Password sign-in method. All the images are stored in the Firebase Storage.

**5.1.1 Android Project Structure**

Here, Model View ViewModel (MVVM) project structure is used to create the app.

***(Change these screenshot. Take your android studio screenshot after running the project)***

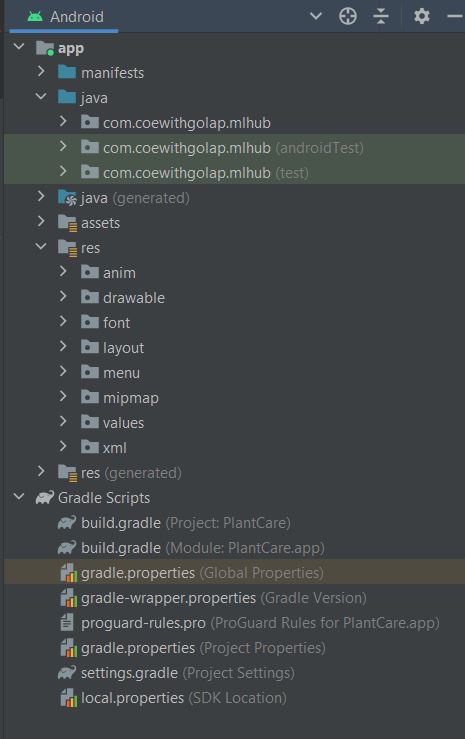


Figure 2: Project Structure

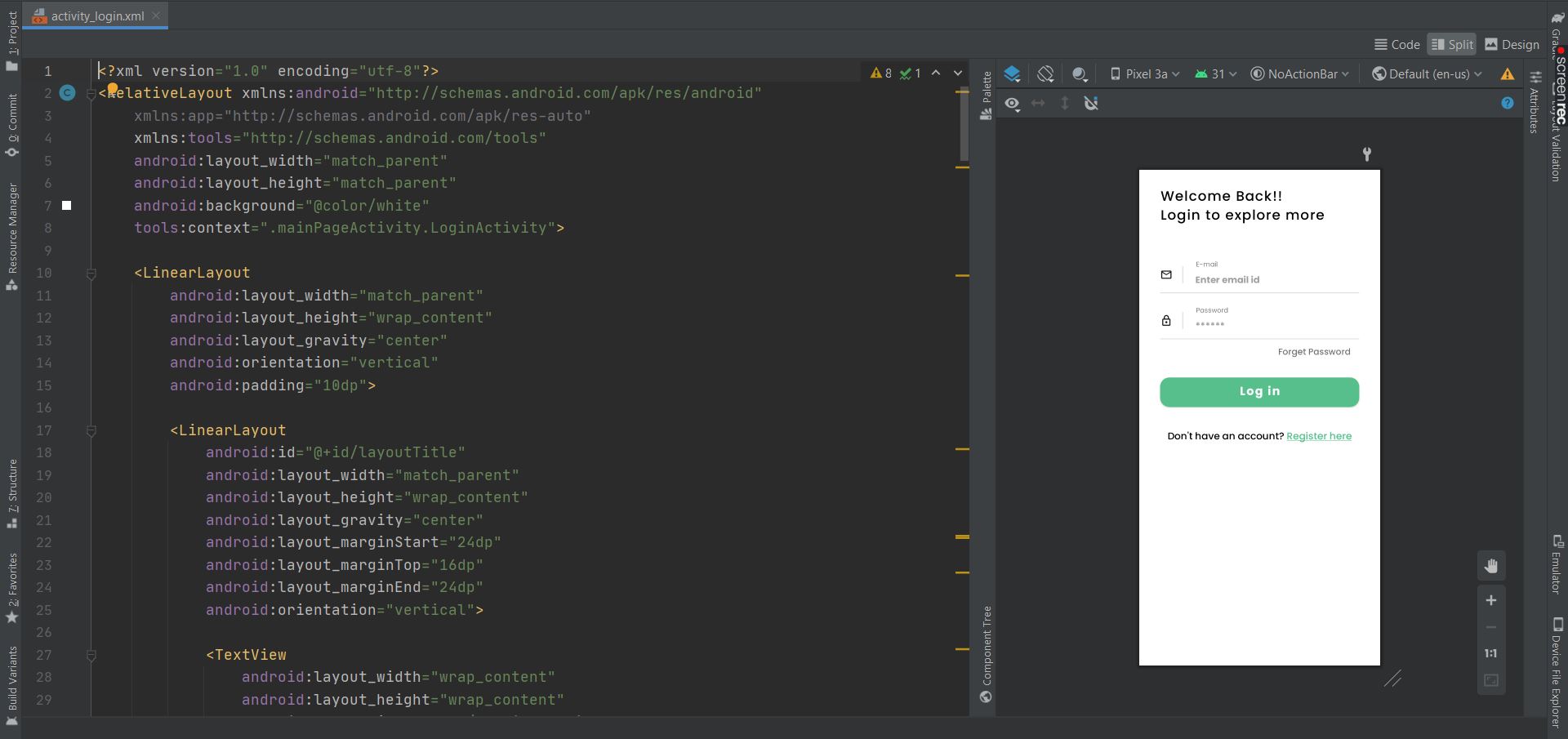


Figure 3: XML file to design the UI of the app (Login Page)

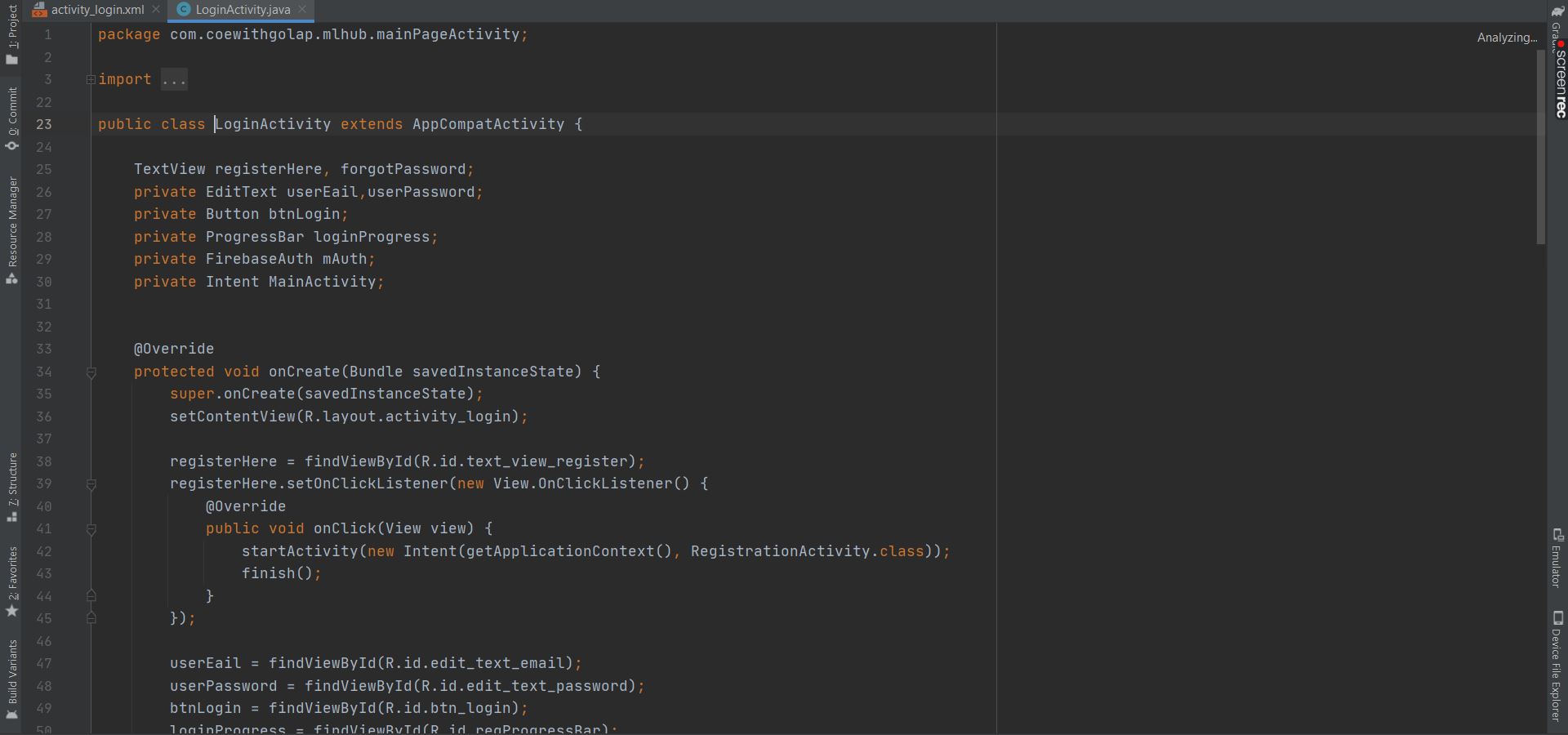


Figure 4: Using Java code add the login functionality in LoginActivity.java

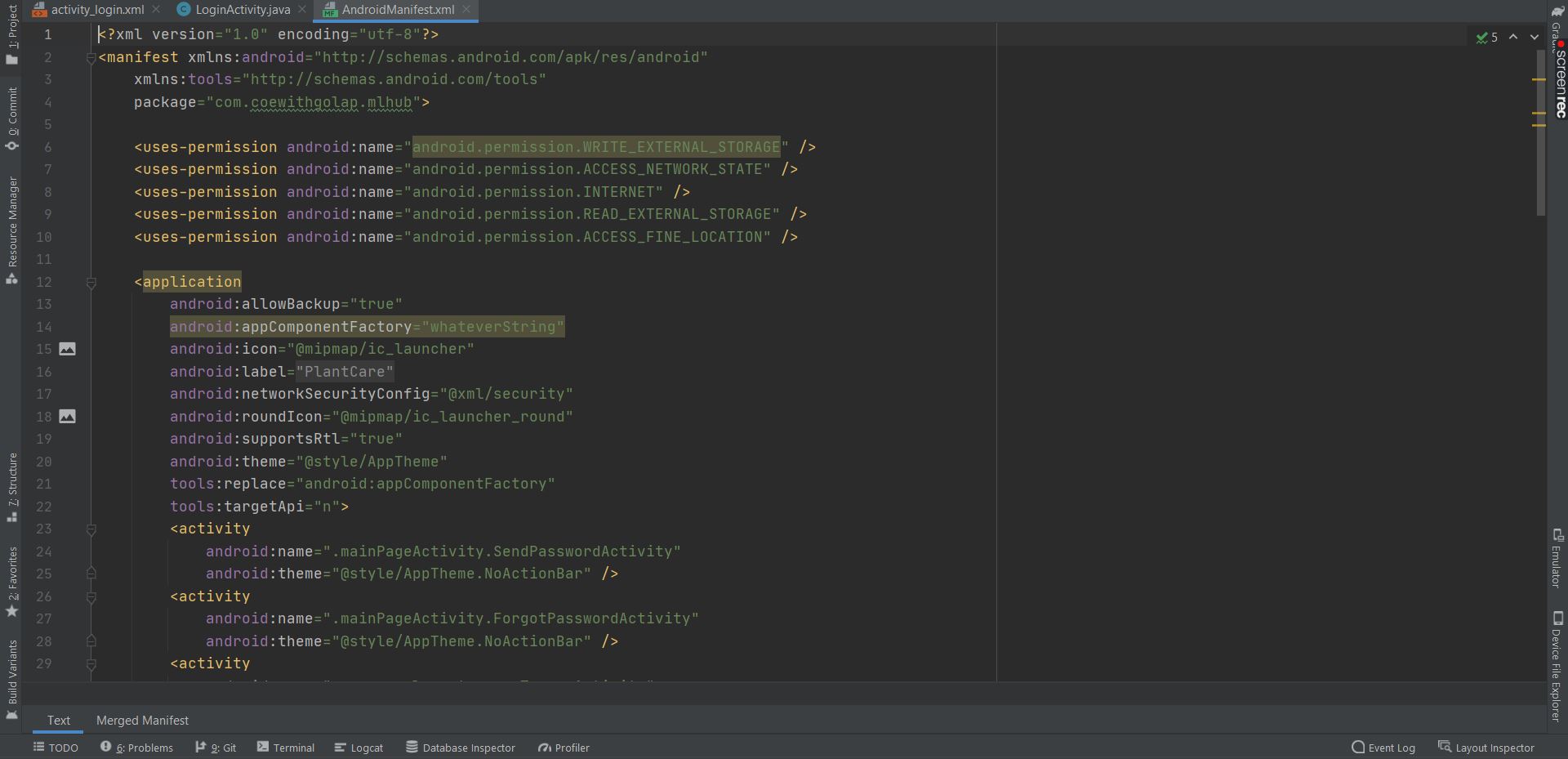


Figure 5: AndroidManifest.XML file, all the permission and activities are inside here

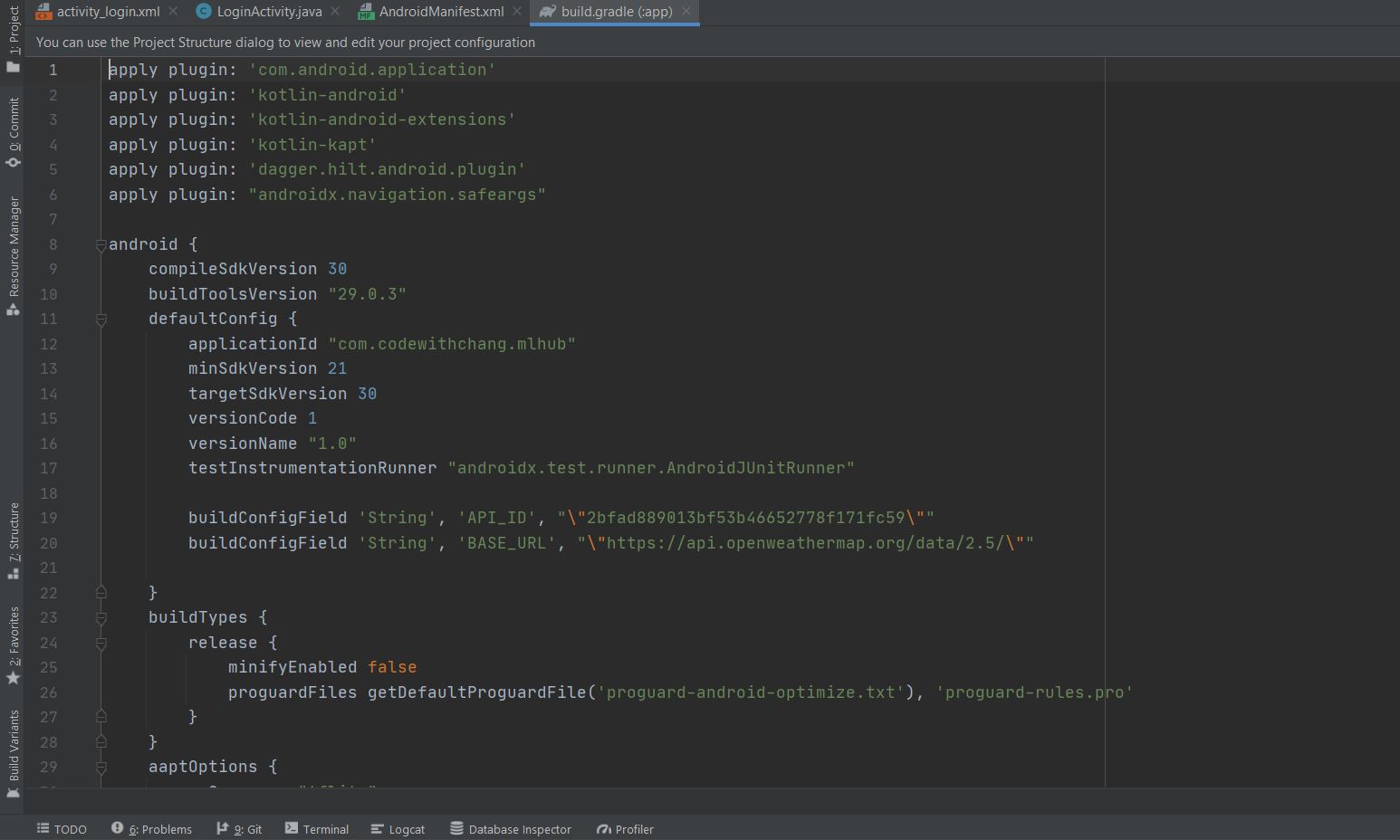


Figure 6: build.gradle app file, where all the dependencies are mention

**Chapter 6**

**Project Snapshots**

***(Add App screenshots. Every pages)***

**Chapter 7**

**Project Implementation**

**7.1 Introduction to Technology**

**7.1.1 Java and XML**

**Java** is a **programming language** and a **platform**. Java is a high level, robust, object-oriented and secure programming language.

Java was developed by Sun Microsystems (which is now the subsidiary of Oracle) in the year 1995. James Gosling is known as the father of Java. Before Java, its name was Oak. Since Oak was already a registered company, so James Gosling and his team changed the name from Oak to Java.

* Java has been one of the most popular programming languages for many years.
* Java is Object Oriented. However, it is not considered as pure object-oriented as it provides support for primitive data types (like int, char, etc)
* The Java codes are first compiled into byte code (machine-independent code). Then the byte code runs on **J**ava **V**irtual **M**achine (JVM) regardless of the underlying architecture.
* Java syntax is similar to C/C++. But Java does not provide low-level programming functionalities like pointers. Also, Java codes are always written in the form of classes and objects.
* Java is used in all kinds of applications like Mobile Applications (Android is Java-based), desktop applications, web applications, client-server applications, enterprise applications, and many more.
* When compared with C++, Java codes are generally more maintainable because Java does not allow many things which may lead to bad/inefficient programming if used incorrectly. For example, non-primitives are always references in Java. So we cannot pass large objects (like we can do in C++) to functions, we always pass references in Java. One more example, since there are no pointers, bad memory access is also not possible.
* When compared with Python, Java kind of fits between C++ and Python. The programs are written in Java typically run faster than corresponding Python programs and slower than C++. Like C++, Java does static type checking, but Python does not.

**XML** is a software- and hardware-independent tool for storing and transporting data.

* XML stands for eXtensible Markup Language
* XML is a markup language much like HTML
* XML was designed to store and transport data
* XML was designed to be self-descriptive
* XML is a W3C Recommendation
* **Xml** (eXtensible Markup Language) is a mark up language.
* XML is designed to store and transport data.
* Xml was released in late 90’s. it was created to provide an easy to use and store self describing data.
* XML became a W3C Recommendation on February 10, 1998.
* XML is not a replacement for HTML.
* XML is designed to be self-descriptive.
* XML is designed to carry data, not to display data.
* XML tags are not predefined. You must define your own tags.
* XML is platform independent and language independent.

**Why xml**

**Platform Independent and Language Independent:** The main benefit of xml is that you can use it to take data from a program like Microsoft SQL, convert it into XML then share that XML with other programs and platforms. You can communicate between two platforms which are generally very difficult.

**7.1.2 Firebase**

Firebase is a mobile and web app development platform that provides developers with a plethora of tools and services to help them develop high-quality apps, grow their user base, and earn more profit.

The Firebase Realtime Database is a cloud-hosted NoSQL database that lets you store and sync between your users in realtime.

The Realtime Database is really just one big JSON object that the developers can manage in realtime.



Figure 7: Firebase Real-time Database Example

**7.1.2 Android Studio**

Android Studio is Android's official IDE. It is purpose-built for Android to accelerate your development and help you build the highest-quality apps for every Android device. Based on Intellij IDEA, Android Studio provides the fastest possible turnaround on your coding and running workflow.

The code editor helps you write better code, work faster, and be more productive by offering advanced code completion, refactoring, and code analysis. As you type, Android Studio provides suggestions in a dropdown list. Simply press Tab to insert the code.

The Android Emulator installs and starts your apps faster than a real device and allows you to prototype and test your app on various Android device configurations: phones, tablets, Android Wear, and Android TV devices. You can also simulate a variety of hardware features such as GPS location, network latency, motion sensors, and multi-touch input.

**7.1.3 Keras**

Keras is a high-level Python neural network library that runs on top of either TensorFlow or Theano. Other high-level Python neural networks libraries can be used on top of TensorFlow, such as TF-Slim, although these are less developed simplifies the codes used in TensorFlow by making use of a smaller code base so that the code length will reduce and make sure that the processing will run smoothly. Keras is used for a graphical representation of the models which helps to understand the structure of the model. Auto Keras, a library based on Keras, has also gained popularity and can be used to make it quicker to get results

**7.1.4 Tensorflowlite**

Tensorflow lite is a deep learning framework and is based on the tensorflow framework. It is used to reduce the size of a normally huge tensorflow model so that it can be used in modular devices such as mobile phones. We can use tensorflow lite to access the model with android studio. It is a complex procedure and is used to access a minimal reduction algorithm of the model

**7.1.5 CNN**

Convolutional Neural Networks are a complex neural network chain which work to get the features of an image from a dataset which is trained and classify them to get the required output. It trains the neural networks by using the dataset images and changing them to numerical values. The main advantage of CNN compared to its predecessors is that it automatically detects the important features without any human supervision....ConvNets are more powerful than machine learning algorithms and are also computationally efficient. These numerical values are then put into numerical arrays based on their categorized characteristics. These arrays are then put into different nodes in the network and passed through multiple iterations based on the input given. The CNN models are used for geographical classification in multiple companies which require data to be classified in a quick and secure way it almost acts like a filter removing dust and separates the features of the images

**7.2 Feasibility Report**

Feasibility analysis begins once the goals are defined. It starts by generating broad possible solutions, which are possible to give an indication of what the new system should look like.

This is where creativity and imagination are used. Analysts must think up new ways of doing things- generate new ideas. There is no need to go into the detailed system operation yet.

The solution should provide enough information to make reasonable estimates about project cost and give users an indication of how the new system will fit into the organization. It is important not to exert considerable effort at this stage only to find out that the project is not worthwhile or that there is a need significantly change the original goal.

Feasibility of a new system means ensuring that the new system, which we are going to implement, is efficient and affordable. There are various types of feasibility to be determined. They are

## 1. Economically Feasibility

Development of this application is highly economically feasible. The only thing to be done is making an environment with an effective supervision. It is cost effective in the sense that has eliminated the hard work to the vehicle agencies to verify documents to rent a car, then choose a car and do the insurance and all etc. The system is also time effective because the process take only 2 steps.

## 2. Technical feasibility

The technical requirement for the system is internet service and it does not use any other additional Hardware and software.

Technical evaluation must also assess whether the existing systems can be upgraded to use 6 the new technology and whether the organization has the expertise to use it.

This application depends on Google’s Firebase and internet service.

## 3. Operational Feasibility

The system working is quite easy to use and learn due to its simple but attractive interface.

User requires no special training for operating the system. Technical performance include issues such as determining whether the system can provide the right information for the voter personal details, and whether the system can be organized so that it always delivers this information at the right place and on time using internet services.

## 

## 4. Behavioural Feasibility

This includes the following questions

* Is there sufficient support for the users?
* Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed.

All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible.

**The App** is behaviorally feasible and there will be no harm for the users.

**Chapter 8**

**Testing**

**8.1 Unit Testing**

**Unit Testing** is a type of software testing where individual units or components of a software are tested. The purpose is to validate that each unit of the software code performs as expected. Unit Testing is done during the development (coding phase) of an application by the developers. Unit Tests isolate a section of code and verify its correctness. A unit may be an individual function, method, procedure, module, or object.

***(ADD MOBILE SCREENSHOTS)***

***USE TWO OR THREE DIFFERENT SCEEN SIZE PHONE AND RUN THE APP AND TAKE THE SCREENSHOTS AND ADD HERE. MAKE SURE GIVE NAME OF THE MODEL OF THE PHONE. LIKE:***

***Model: Samsung galaxy s20 (6.7 inch screen)***

***(Add 2-3 screenshots)***

***Model: OnePlus Nord 2 (6.43 inch screen)***

***(Add 2-3 screenshots)***

**Chapter 9**

**Advantage and Disadvantage**

**9.1 Advantage**

* This app is fully functional and flexible.
* It is very easy to use.
* This app will help users to handle the different diseases precautions.
* It saves a lot of time, money, and work.
* Eco-friendly: The app includes the least paperwork, handy to use, easy to detect disease.
* The application acts as an office that is open 24/7.
* It comes up with a Community page, where users can rise the queries related to the crop diseases, and experts or other users will answer the queries.
* It also provides us the weather report of the current location.

**9.2 Disadvantage**

* It requires internet connection.
* Constant monitoring is required.
* Need to update the ML model.

**Chapter 10**

**Conclusion**

In this study, an android based mobile application has developed to detect and identify plant diseases using deep CNN. The model has been trained using a large public dataset of grape plant images (**Plant Village Dataset**) and has attained an accuracy of around 96% which indicates the efficiency of the model. The trained model takes an image of a plant leaf as input, processes it, and uses deep CNN algorithm to detect and identify plant disease. Then the model is exported to the android application **assests folder**. The application also provides necessary steps to prevent and cure the diseases (Pop up the precaution of the disease). Due to the availability and low price of android mobile devices in recent times, even the lower-earning people like farmers can easily afford android devices.

So, it is useful for the farmers to be able to use the application. Farmers in rural areas can use it to detect plant diseases accurately and take action accordingly. It will

**Bibliography**

* Patel, A., Joshi, M.: A Survey on the Plant Leaf Disease Detection Techniques. IJARCCE. vol. 6, pp. 229-231 (2017). doi: 10.17148/ijarcce.2017.6143
* Yunbin, D.: Deep Learning on Mobile Devices - A Review. SPIE Defense + Commercial Sensing (2019). doi: 10.13140/RG.2.2.15012.12167
* Thorat, N., Nikam, S.: Early Disease Detection and Monitoring LargeField of Crop By Using IoT. (IJCSIS) International Journal of Computer Science and Information Security. vol.15, pp. 236-248 (2017).
* D M, S., Akhilesh, Kumar, S., M G, R., C, P.: Image based Plant Disease Detection in Pomegranate Plant for Bacterial Blight. 2019 International Conference on Communication and Signal Processing (ICCSP). pp. 0645 0649. IEEE, Chennai, India (2019). doi: 10.1109/iccsp.2019.8698007
* Valdoria, J., Caballeo, A., Fernandez, B., Condino, J.: iDahon: An Android Based Terrestrial Plant Disease Detection Mobile Application Through Digital Image Processing Using Deep Learning Neural Network Algorithm. 4th International Conference on Information Technology (InCIT).pp. 94-98. IEEE, Bangkok, Thailand (2019).doi: 10.1109/incit.2019.8912053
* Ferentinos, K.: Deep learning models for plant disease detection and diagnosis. Computers and Electronics in Agriculture. vol. 145, pp. 311-318 (2018).doi: 10.1016/j.compag.2018.01.009
* Mahalakshmi, J., Shanthakumari, G.: Automated Crop Inspection and Pest Control Using Image Processing. International Journal of Engineering Research and Development. vol. 13, pp. 25-35 (2017).
* Thakre, G., More, A., Gajakosh, K., Yewale, M., Shamkuwar, D.: A Study on Real Time Plant Disease Diagnosis System. International Journal of Advance Research, Ideas and Innovations in Technology. vol. 3, pp. 1118-1124 (2017).
* Raut, S., Fulsunge, A.: Plant Disease Detection in Image Processing Using MATLAB. International Journal of Innovative Research in Science, Engineering and Technology. vol. 6, pp. 10373-10381 (2017). doi: 15680/IJIRSET.2017.0606034
* Singh, V., Misra, A.: Detection of plant leaf diseases using image segmentation and soft computing techniques. Information Processing in Agriculture. vol. 4, pp. 41-49 (2017). doi:10.1016/j.inpa.2016.10.005
* Mohanty, S., Hughes, D., Salathé, M.: Using Deep Learning for Image-Based Plant Disease Detection. Frontiers in Plant Science. vol. 7, (2016).doi: 10.3389/fpls.2016.01419