ÇANKAYA UNIVERSITY FACULTY OF ENGINEERING

COMPUTER ENGINEERING DEPARTMENT

Project Report

CENG 408

Innovative System Design and Development II

ATADAN

Özgür Bekir Uzun - 201811059

Aybüke Gökmen - 201811031

Umut Deniz Zorgül - 201811070

Gizem Akgönüllü - 201911003

Onur Kenan Maraş - 201911039

Advisor: Asst.Prof.Dr. Gül Tokdemir

TABLE OF CONTENTS

Abstract

Because plant diseases are spreading quickly, decision support systems are needed to detect them early and make wiser judgments on how to properly treat them. Growing such systems is still a difficult undertaking for researchers.

In the past two decades, a number of Computer Aided Diagnosis (CAD) methods have been suggested to improve the accuracy of plant disease diagnosis. In this project, our major goal is to provide an application for those who wish to monitor their plants when they suspect a disease threat.

Routinely sending messages to the user to scan his or her plant, the program will regularly gather photographs of plants. The program will compare fresh photographs with those that have already been posted each time.

It will analyze any changes to a plant's leaf, color, size, or body. Users have the chance to compare photos over time and quickly share their images with philologists thanks to the personal gallery. It will not provide a diagnosis.

Keywords: *Machine Learning, Image Procession, Plant Disease, Deep Learning, CNN, Classification*

1. Introduction

1.1 Problem Statement

Many plant health systems around the world have fund seriously in the cure of different illness, but they yet fall behind when it comes to exploring ways to get ahead of them. Powdery mildew disease is one of the dangerous diseases for plants and it might be deadly, if not diagnosed on time. A fungal disease called powdery mildew damages many of our ornamental plants, flowers, vegetables, and fruits.

The powdery mildew is simple to recognize. White powdery material will be present on infected plants; it is most noticeable on upper leaf surfaces but can also be seen on stems, flower buds, and even the plant's fruit. This fungus thrives in environments with low soil moisture levels and high levels of humidity on the top surfaces of the plants. Plants housed in shaded regions typically experience it more than those in full sunlight. People don't take powdery mildew illness on their plants seriously, even if they are unsure that it is present. They delay therapy and prevent early discovery in this way. The ATADAN app makes it possible to detect powdery mildew illness in its earliest stages when it is most curable and offers less expensive treatment alternatives.

1.2 Background or Related Work

The ATADAN is not an often-used application. Agrio, PlantNet, and NatureID are a few of them. Many of these programs charge a full cost for full functionality, while some don't function at all or are outdated.

1.3 Solution Statement

Early illness detection may significantly reduce both disease transmission and mortality rates. The severity of the condition, which is directly connected to how long it has been advancing unnoticed, directly influences the probability of dying from the sickness. As a result, earlier discoveries reduce illnesses and save plants. Fortunately, unlike most other illnesses, plant diseases that are present on the leaf are typically visible to the patient and the examiner.

You may quickly reduce plant disease signs on a leaf with the ATADAN application. It gives you the ability to take pictures and track changes over time, making it easier for you to monitor the health of your plants over the long term. The efficient and user-friendly solution helps to make plant monitoring a regular practice and will be made accessible for Android via the App store.

By allowing you to be aware of other people's plants, find a potentially dangerous symptom early, and follow up and check your skin over time, we want to raise awareness of the need of early plant disease identification.

1.4 Contribution

Describe how your solution improves and advances the available technology. The following sections: Background, Related Work, and Motivation, may occasionally be included in the introduction as subsections or separate sections.

1.5 Conclusion

As the project team, we tried to explain the purpose and scope of our project and to introduce how we implemented it. The main goal of our project is to create an application to guide people. Our research has shown us that people who think that their plants have a disease can reduce the death rate in plants thanks to the use of our application and that people who use our application may be more conscious about plant diseases. Also, our studies have shown that the SVM classification algorithm and the CNN deep learning algorithm get the best results.

We will photograph a plant and then use an algorithm to display the user the likelihood that the plant will contract a disease. The user may then make a personal album about a particular plant so that they can keep track of any changes, such as the plant becoming infected or changing color.

2. Literature Review

2.1 Abstract

Agriculture, which forms the basis of our diet, is becoming increasingly important for humanity. The quality of agriculture has an important place in our lives when the increasing human population, global warming, and diseases that occur in humans due to genetic and environmental factors are considered. In this literature review paper, we have benefited from x different conferences, journals, and articles such as the Google Scholar search tool, IEEEXplore, and Springer.

2.2 Introduction

Based on the information we received from the Turkish Journal of Agriculture and Forestry *[1]*, factors such as high humidity and increased temperature caused by global climate change cause an increase in plant diseases. One of the most important steps in preventing the rapid increase in vegetative diseases is the early diagnosis of diseases before they spread. In this respect, we decided to make an agriculture and farmer-friendly mobile application that can easily reach farmers and all segments of agriculture. Our aim with this application is to determine the type of plant first. Then we would like to analyze the health status of the mentioned plant. One of the important and basic functions of our application is to archive photos of the plant and to compare the disease/health status of the plant-based on old data. We aim to recognize leaf-based plant diseases. The primary objective of this research is to identify plant illnesses and to do this, we will combine machine learning techniques like K-Means, Decision Trees, CNN, and Support Vector Machines with image processing.

It is very difficult to monitor plant diseases manually. It requires a tremendous amount of work, and expertise in plant diseases, and also requires excessive processing time. Hence, image processing is used for the detection of plant diseases. Disease detection involves the steps like image acquisition, image pre-processing, image segmentation, feature extraction, and classification.[2]

2.3 Image Acquisition

Our main goal in developing this application is to give the user access to our dataset. The user will be able to upload a leaf-based photo of his own plant to the system at the same time using a camera, tablet, phone, or computer. To make the next user more efficient, the user's added images will be preserved in the system's image database.

2.4 Image Pre-Processing & Segmentation

In this process, the image must go through some methods to reach the required efficiency. This process is a basic process that we will consider using our application efficiently. Algorithms are to be used at this stage.

- Otsu's Algorithm

  a. According to the threshold, Separate pixels into two clusters b. Then find the mean of each cluster. c. Square the difference between the means. d. Multiply the number of pixels in one cluster times the number in the other *[3]*

- K-means clustering After this algorithm our image will be more efficient.

2.5 Data Pre-Process

There are some algorithms that try to make the data smooth by putting our incoming data into some methods before creating the model. Among the ways, we will use Outlier detection, Missing value, Scaling, and Standardize. Since we want to create an instance-based model, scaling and standardized methods are of great importance to us.

2.6 Classification

After the feature extraction phase, we will use image classification to identify the disease. Our artificial intelligence will be ready through the model we have created to identify plant diseases.

The most accurate classification algorithms will be used in our mobile application. For this, the classification method that gives the most accurate result will be decided according to the determined data.

2.7 The Classification Techniques That We Will Planned to Use

2.7.1 KNN Classifier

The KNN classifier method, used for regression and classification, is of great importance in our application. KNN performs plant phenotype clustering and filtering and we will use that algorithm.

Pseudocode for kNN[4]

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn import metrics
#Train Model and Predict
k = 3
neigh = KNeighborsClassifier(n_neighbors = k).fit(X_train,y_train)
Pred_y = neigh.predict(X_test)
print("Accuracy of model at K=3 is",metrics.accuracy_score(y_test, Pred_y))
```

2.7.2 Support-Vector Machine

The purpose of the SVM method is an algorithm used to minimize classification error. It usually performs statistical classification and regression analysis.

Pseudocode for Support-Vector Machine[5]

```
from sklearn.svm import SVC #Building a Support Vector Machine on train
data svc_model = SVC(C= .1, kernel='linear', gamma= 1) svc_model.fit(X_train,
y_train) prediction = svc_model .predict(X_test) #check the accuracy on the
training set print(svc_model.score(X_train, y_train))
print(svc_model.score(X_test, y_test))
```

2.7.3 Decision Tree

The Decision tree algorithm is an algorithm that allows us to arrive at a probabilistic result. The reason why we think of using this algorithm is to act according to the possibility of reaching the direct result since it is a Leaf-based application. Also, it has a fast prediction.

2.7.4 Convolutional Neural Networks (CNN)

Convolutional neural networks (CNN) are widely used in deep learning to solve various computer vision problems. There are several CNN architectures and each has a unique effect. Some pre-trained models trained on more than 1 million native photos in the dataset are still viable today. As a result, the weights of these previously taught models can be used in structures. Alternative architectures will be designed in such a way that they do not affect the weights or deviations of visual information, which may change according to plant diseases. Convolutional neural networks have entered critical attention in recent times as they've proven able of outperforming former records in image recognition challenges. utmost noticeable is the work by Krizhevsky, Sutskever, and Hinton (2012), who in 2012 set the record in the ImageNet Large Scale Visual Recognition Challenge_[7]_ with a periphery of 10.9 compared to the alternate-stylish entry.

Pooling returns the form of restatement invariance [8]*; works singly on each depth slice of the input and resizes it spatially. Lapping pooling is usefully enforced to reduce overfitting. Also, to reduce overfitting, a drop subcaste*[9]_ is used on the first two completely clicked layers. still, the strike is that it increases the training time by 2- 3 times compared to a standard neural network with the same armature [10]*. Bayesian optimization trials also proved that ReLUs and release have a community effect, which means they're profitable when used together*[11].

The advance of CNNs refers to their ability to learn rich mid-level image representations as opposed to hand-designed low-level features used in other image classification methods *[12]*.

2.8 Related Works

• *Agrio App* program called Agrio uses artificial intelligence to identify plant diseases. The app delivers correct information about herbal illness therapy after using user-uploaded photos to match data from the app's database. [13]

• *CropsAI* There are presently just a few goods that this app supports, but more are planned in the future. Use the camera on your phone to take pictures of the infected crop, then upload them to CropsAI. With the use of artificial intelligence, the app will provide you with the finest guidance for treating plant problems. [14]

• *Plantix* Upload a picture of the impacted plant to the app. The app will automatically detect the illness and provide you with all relevant information on the illness and its treatment (biologically and chemically). By providing information on how to grow a new crop, the app also helps. Calculating how much fertilizer to use also helps. In general, farmers can benefit from using it. APK is accessible for Android, but not on iPhone yet. [15]

• _Pasific & Pests Pathogens _ This app asks the farmer questions regarding the crop that has been afflicted with pests. It is whittled down after a series of inquiries until the ideal match is discovered. The app's offered photographs may then be compared to the crop that has been infected. The Australian Center for International Agricultural Research, or ACIAR, is a sponsor of the Pests and Pathogens of the Pacific application. It is available for iPhone or Android download.[16]

• _Blossom - Plant Identification _ Identify plants, flowers, succulents, and trees by photo and get how-tos and useful plant care tips. Features: accurate plant recognition, plant disease identification, virtual care assistant (chatbot), garden for edible plants, botanist hotline, plant care reminders, personal plant collection, customized water calculator, useful information and smart tips, green blog, notes, light meter. [17]

2.9 Conclusion

The methods and algorithms we have described above will be used in the most appropriate way according to the data we have and the model creation process will be started.

3. Software Requirement Specification (SRS)

3.1 Introduction

We are making an application for the analysis and detection of herbal diseases. We will create leaf-based plant recognition system and disease recognition system with image recognition and machine learning models. Although there are applications configured for this purpose in the market, we want to add different and additional functions to it. While doing this, we will stay in one-on-one communication with potential users and take their needs as a basis. We also want to make this accessible and easy-to-use for farmers and people at all levels, who are involved with plants.

3.1.1 Purpose

ATADAN SRS is a document created to give a detailed description of the requirements for the "ATADAN (Smart Advisor for Smart Agriculture)" project. In addition to defining the purpose, this document will provide information about interfaces and system restrictions. A comprehensive description of the project to be developed will be made. Therefore, this document will guide the development process.

3.1.2 Scope of Project

Plants have different kinds of diseases just like humans. If the people who take care of the plants do not have the necessary information, the diseases can spread among the plants and progress to a point where they cannot be treated. It is important to detect these diseases early, because the earlier they are detected, the faster the treatment process and at the same time, thanks to early diagnosis, the disease that develops in the plant prevents the transmission of the disease to other plants around it. The application we have developed helps people dealing with plants in any field to reach the information they need about plant diseases. Its main purpose is to reduce and correct plant diseases as much as possible. The uploaded plant photos will be scanned in detail and compared with the data in the hands of the application, the correct determination will be reached. The use of the application by more people will increase the amount of data owned, and thus the consistency of the determinations will be more. Thanks to the mobile application of ATADAN, those who need it can be easily accessed. It is practical and easy to use and does not give the user any trouble. It only needs a photograph to reach the necessary information. It is much easier for the app to detect in well-lit photos with high clarity. In order to take up less space on the user's phone, it reaches the necessary data via the internet and adds the diseased plant photos uploaded by the user to the database.

3.1.3 Glossary

Term	Definition
User	People who use our application. It will generally used by people who are interested in agricultural areas.
Android	A operating system which purpose operate phone. We will make an application to Android.
Python	Python, a high-level programming language, will allow us to build our artificial intelligence model.
Stakeholders	Any person or group of people involved in a project
EmguCV	It is an OpenCV library designed for more precise image processing. EmguCV is a wrapper that provides implementation in .NET environment.
Classification	Classification is the process of taking some type of input and assigning a label to it. Classification is often used when estimates are discrete. For example, matching a plant according to its classification as diseased or healthy.
Database	A place for keeping data which provides from users.
S.R.S	It is a documentation describing what to do before starting the software. It also contains information under which conditions the necessary functions can work.

3.2 Overall Description

3.2.1 Product Perspective

ATADAN is a mobile application that aimed detects plant types and plant diseases with leaf-based image recognition. Users can get information about expected diseases in their early stages also.

3.2.1.1 User Interfaces

This mobile application is designed for farmers, agricultural engineers and anyone who has an interest in plants. For this reason, the interface should be simple and user-friendly. Users will be able to use this mobile application without any registration obligation after downloading it from the Google Play Store. After opening the application, a main menu will appear. This main menu is divided into sections such as adding new plant image (from gallery or camera), communication. After the photo is added, the user will be shown the result on the screen.

3.2.1.2 Hardware Interfaces

Access can be from any device that supports android.

3.2.1.3 Software Interfaces

Since our application is mobile, it will run on the android operating system. It will be accessible from Google play store and app store.

3.2.1.4 Communication Interfaces

It will access the data of the application through the internet. It is not possible to make accurate determinations without the internet.

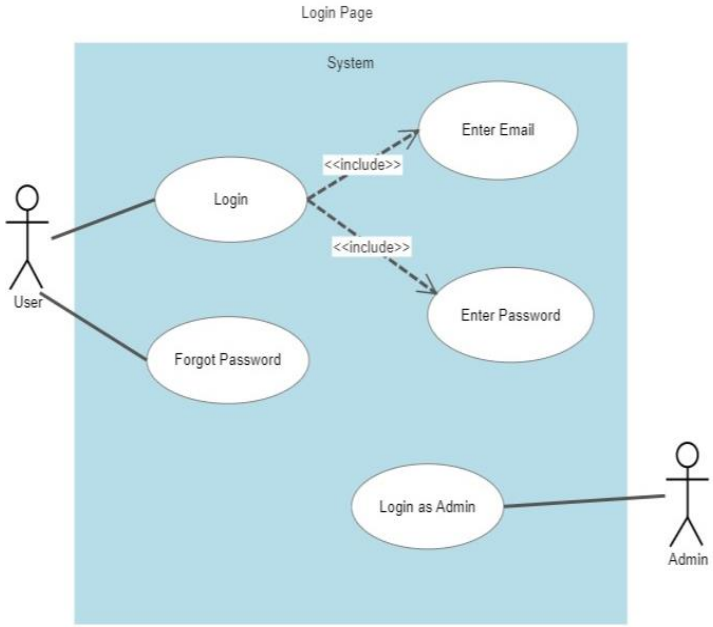
3.2.1.5 Memory

The memory space required for our application is about 40-50 mb. As the user uses the application, the application will take up more space due to the option to add photos.Minimum system requirements: above 4-gb ram, 1.6 GHz Quad-Core ARM Cortex, Android 8.0 (OREO), stable net connection for background AI Services.

3.2.2 Development Methodology

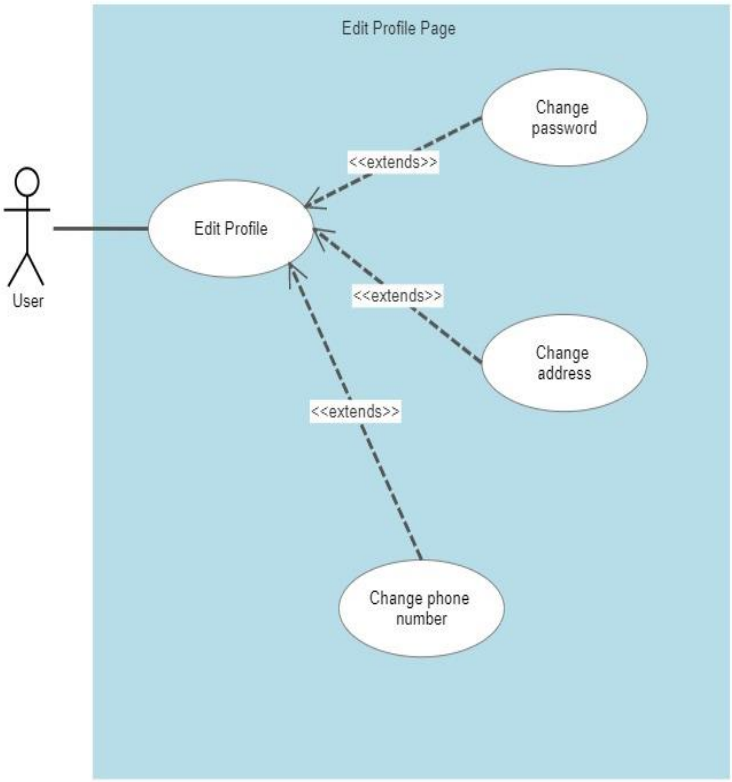
3.2.3 Product Functions

3.2.3.1 Login Page



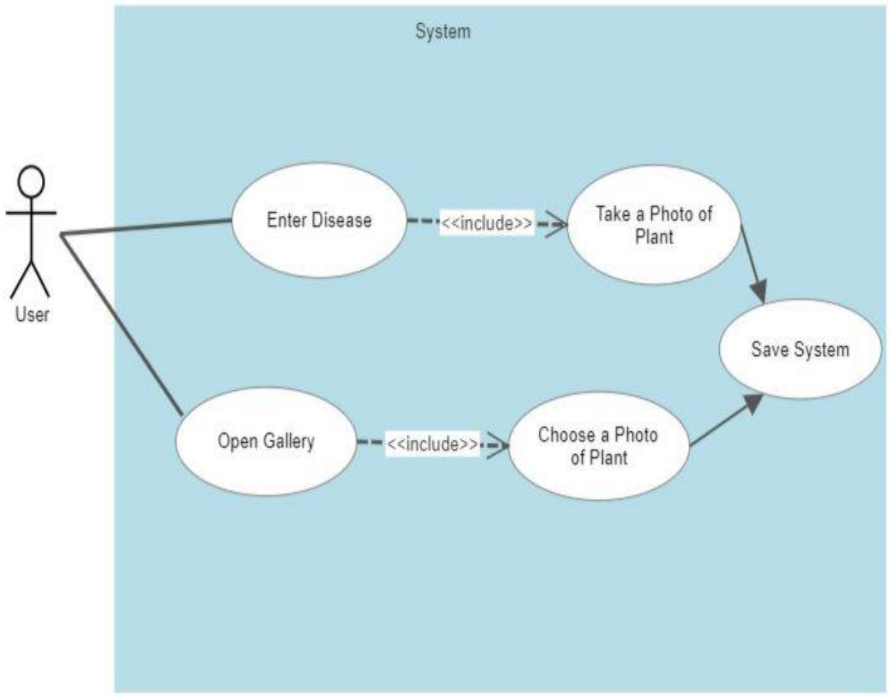
Use Case 1	Sign Up
Primary Actor	User
Goal In Context	Signing up to system.
Preconditions	None
Trigger	User clicking the "Sign Up" button.
Scenario	User enters email, password and personal information.
Exceptions	Email verification fails.

3.2.3.2 Edit Profile Page



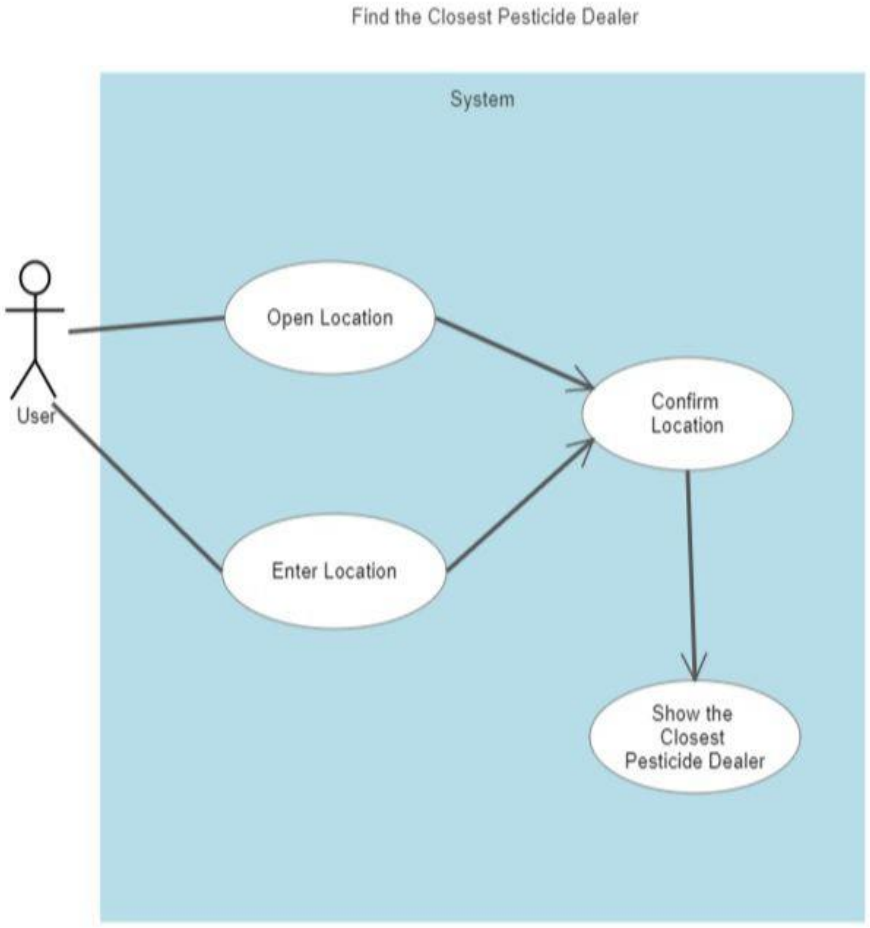
Use Case 2	Edit Profile
Primary Actor	User
Goal In Context	Change password, address and phone number.
Preconditions	None
Trigger	User clicking the "Edit Profile" button.
Scenario	User wants to change personal information.
Exceptions	Incompatible changes on information.

3.2.3.3 Add and Take Photo to Album



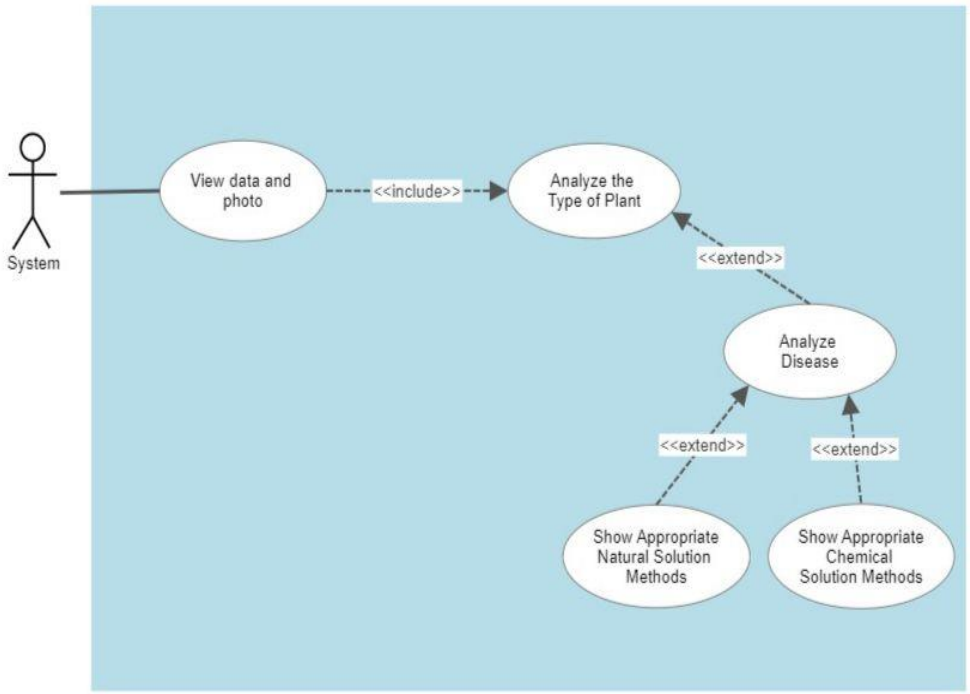
Use Case 3	Add Photo to Album
Primary Actor	User
Goal In Context	Saving system a new photo or taking one.
Preconditions	Photo must be ready to upload.
Trigger	None
Scenario	User selects a photo or takes one for specific disease.
Exceptions	Wrong format or image with inappropriate size selected.

3.2.3.4 Find the Closest Pesticide Dealer



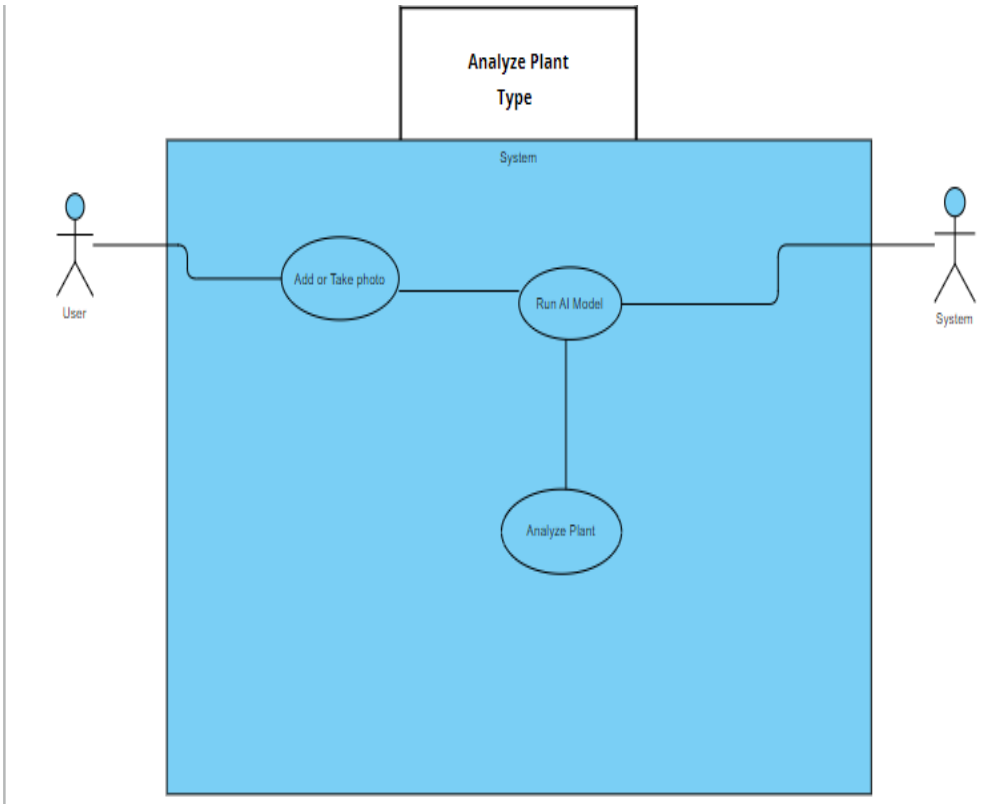
Use Case 4	Find the Closest Pesticide Dealer
Primary Actor	User
Goal In Context	Searching for nearest pesticide dealer.
Preconditions	Location information.
Trigger	User enters location or opens location and confirms.
Scenario	System shows the closest pesticide dealer.
Exceptions	Location service not works.

3.2.3.5 Show Natural and Chemical Solution Methods



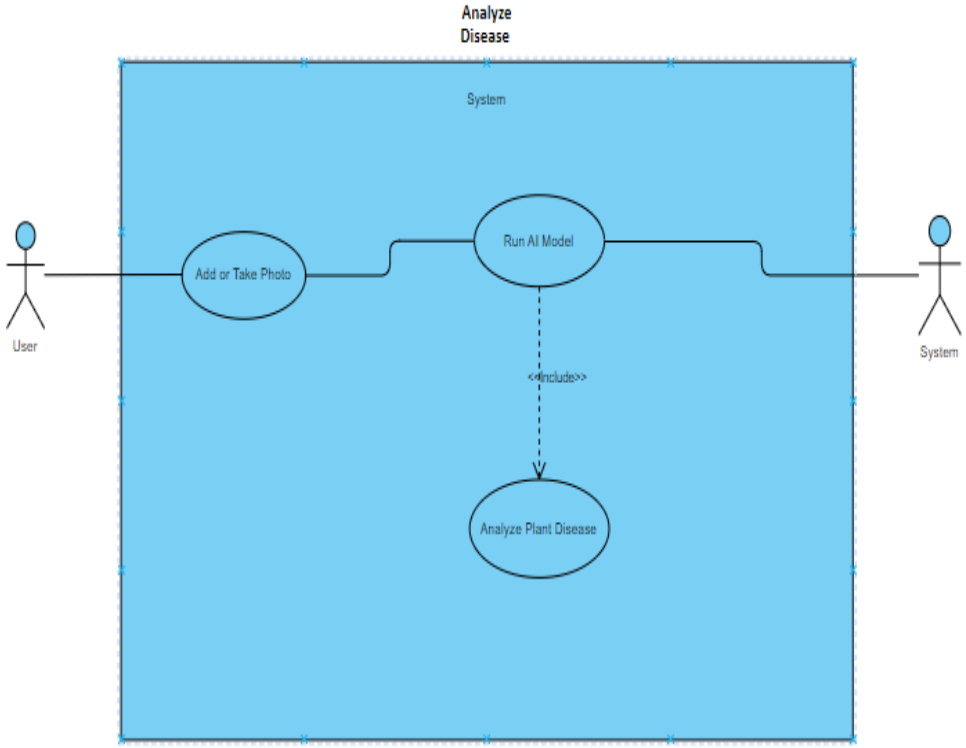
Use Case 5	Show Natural and Chemical Solution Methods
Primary Actor	System
Goal In Context	Finding solution methods for the disease.
Preconditions	None
Trigger	User views data and photo.
Scenario	System analyses disease and shows solution methods.
Exceptions	No solution method in the system.

3.2.3.6 Analyze Plant Type



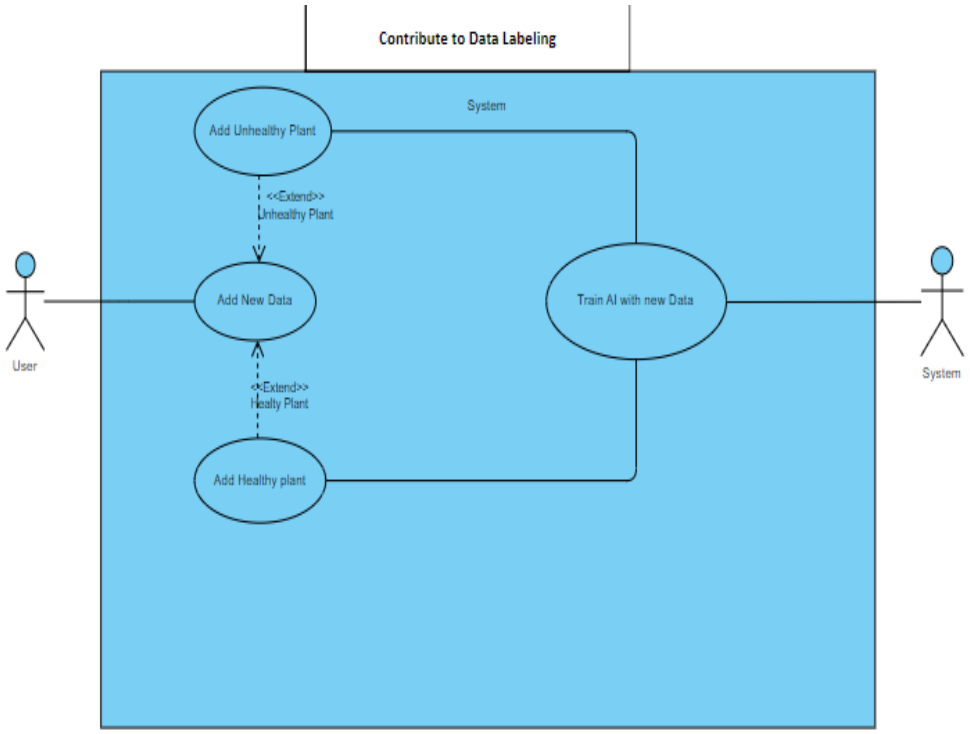
Use Case 6	Analyze Plant Type
Primary Actor	User
Goal In Context	Determine plant type.
Preconditions	Adding or taking photo.
Trigger	None
Scenario	System runs AI model and analyze type.
Exceptions	Photo is not a plant photo or resolution is low.

3.2.3.7 Analyze Disease



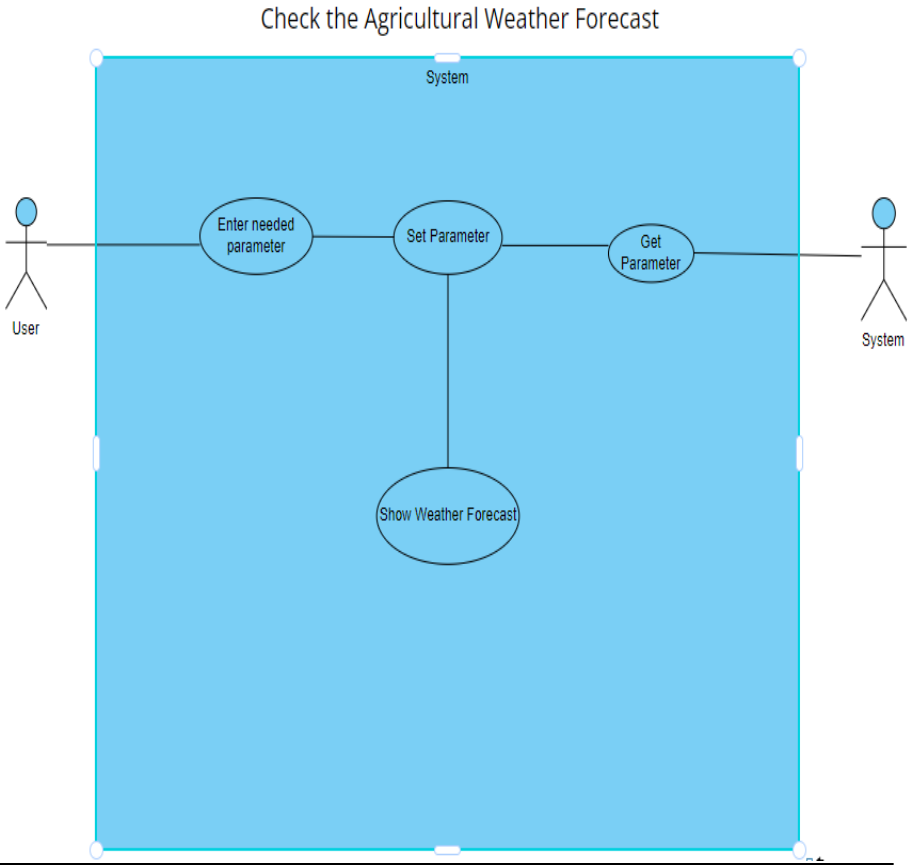
Use Case 7	Analyze Disease
Primary Actor	User
Goal In Context	Determine the disease.
Preconditions	Adding or taking photo.
Trigger	None
Scenario	System runs AI model and analyze disease.
Exceptions	There is no disease or resolution is low.

3.2.3.8 Contribute to Data Labeling



Use Case 9	Contribute to Data Labeling
Primary Actor	User
Goal In Context	Training AI with new data.
Preconditions	None
Trigger	User adds new data.
Scenario	System gets new data classify it and trains AI.
Exceptions	New data is not usable.

3.2.3.9 Check the Agricultural Weather Forecast



Use Case	Check the Agricultural Weather Forecast
Primary Actor	User
Goal In Context	Getting info about the weather forecast.
Preconditions	Needed parameter must known.
Trigger	User clicks "Check Weather Forecast" button.
Scenario	User enters and sets parameter, system shows weather.
Exceptions	Parameter is not needed or not set.

3.3 Software System Attributes

There are some non-functional requirements here. These are the constraints that the system must operate within these requirements.

3.3.1 Portability

The application of the system must run on the Android operating system. With the phone application, our project will become very portable.

3.3.2Performance

All users should be able to use the app at the same time. At the same time, it should be able to upload photos to the system from the gallery or with the camera.

3.3.3 Usability

The interface of the application will be quite simple and user-friendly. Users will be able to easily upload photos of their plants and see the result.

3.3.4 Reliability

First, we make sure that no bots are logged into this app. We will also use the user data in a separate database and only the administrator can access and modify the plant database of our application.

3.3.5 Scalability

As this system uses Machine Learning, as time goes on, the system recognizes plant species and plant diseases much more easily. We will also integrate the data we receive from users into our database. Therefore, ML algorithms will work much more smoothly as time goes on.

3.3.6 Availability

We will ensure that our application can be easily accessed by everyone by placing it in a store accessible to everyone, such as the Google Play Store.

3.3.7 Security

To ensure security, we will put user data in a different database.

4. Software Design Description (SDD)

4.1 Introduction

4.1.1 Purpose

Plant diseases are one of the biggest problems faced by plant owners and agricultural people. This problem, which can be encountered in all plants, leads to more significant problems if not diagnosed and intervened in a timely manner. For example, some plant diseases are contagious, and they can quickly

spread to the surrounding plants and cause great harm, especially to those who are engaged in agriculture. A mistake made in the diagnosis leads to the wrong treatment and causes similar problems. Consulting plant diseases with botanists is costly and time-consuming, and in time lost, the disease can progress and spread. The ATADAN project aims to diagnose plant diseases early and accurately, and thus solve the problems that the plant owner and people engaged in agriculture may encounter, in an easy and cost-effective way.

4.1.2 Scope

ATADAN project offers applications that detect plant diseases to people who need them. It is enough for people who want to use the application to download it to their phone, the application is very easy to use, the user takes a photo of the plant that he thinks is diseased and uploads it to the application. The system examines the uploaded photo and detects the disease thanks to the dataset it has. After the determination, the user who uploads the photo is shared the necessary information about the disease of the plant and solutions. If a successful result is obtained, the system adds the uploaded photo with the necessary information to the dataset.

4.1.3 Glossary

1.3 Glossary

Term	Definition
SDD	Software Design Document
UI	User Interface
OpenCV	Python Library For Image/Video Recognition
SVM	Classification Algorithm
AI	Artificial Intelligence
ML	Machine Learning

4.1.4 Overview of the Document

The first part of this document contains a table explaining the purpose, scope, and terms used in the project. The second part is the architectural design section, which shows the development stages of the project. It also contains an architecture design that lists actors, exceptions, fundamental sequences, priorities, pre-conditions, and post-conditions, as well as a system class diagram. This section also contains the activity diagram for the scenario generator. The final stage is called Use Case Realization. This section presents a block diagram of the system that is created in accordance with the use cases in the SRS paper. The environment is covered in the fourth part. In this section, we've described the circumstance and shown sample environment frames from the prototype.

4.1.5 Conceptual Model for Software Design Description

In this section, basic SDD words, ideas, and context will be provided.

4.1.5.1 Software Design in Context

The task of the project is to solve plant diseases and collect data on the same diseases. ATADAN project is a mobile application. In order to prevent the progression and spread of plant diseases, those in need download and use the ATADAN app. This application aims to help people who own diseased plants and to provide necessary services. This project will be implemented with JavaScript and PHP as scripting languages and CSS and bootstrap as style sheet languages. In addition, it is planned to use Google Play Store and App Store to install the application.

4.1.5.2 Software Design Descriptions within the Life Cycle

4.1.5.2.1 Influences on SDD Preparation

Typically, the software requirements specification serves as the primary software life cycle product that informs software design. The specifications for the SRS (product viewpoint, functional and non-functional requirements, and interface requirements), as well as the expectations of the stakeholders, define the project's design.

4.1.5.2.2 Influences on Software Life Cycle Products

Some needs may change when the SDD is being prepared or as the project is being implemented. SDD also affects the Carpool System's test strategies and test documentation.

4.1.5.2.3 Design Verification and Design Role Validation

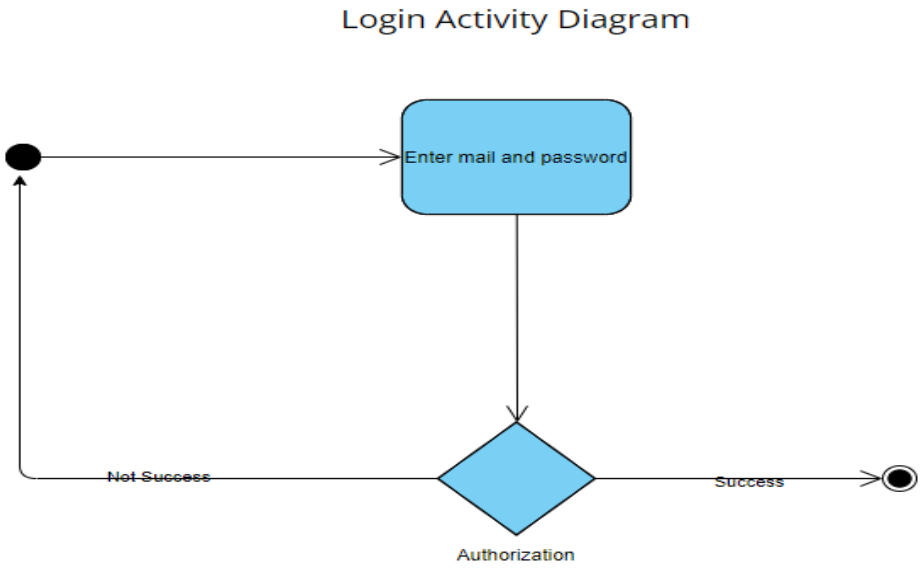
After creating the test cases, verification and validation will be tested. Those scenarios will be used to test every component of the system. The fulfillment of the conditions will be evaluated.

4.2 System Design

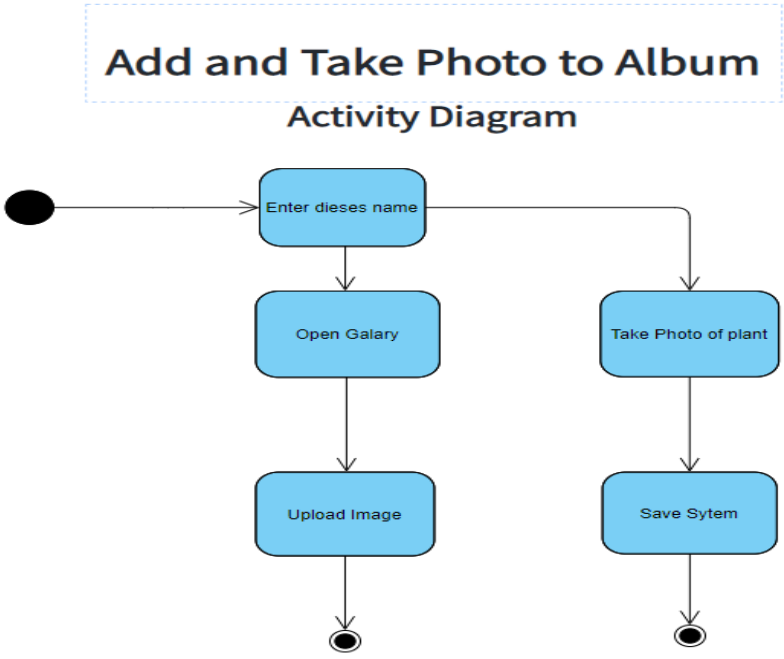
4.2.1 Architectural Design

4.2.1.1 Activity Diagram

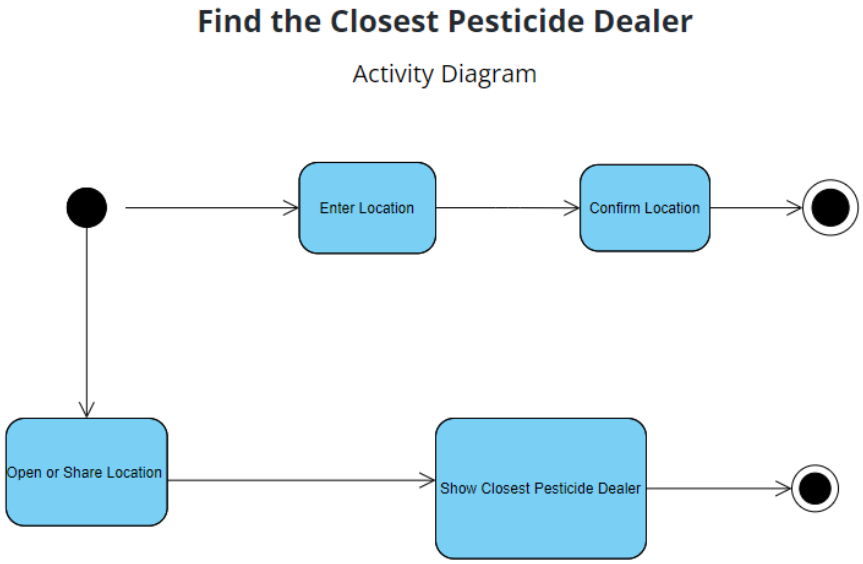
4.2.1.1.1 Login Activity Diagram



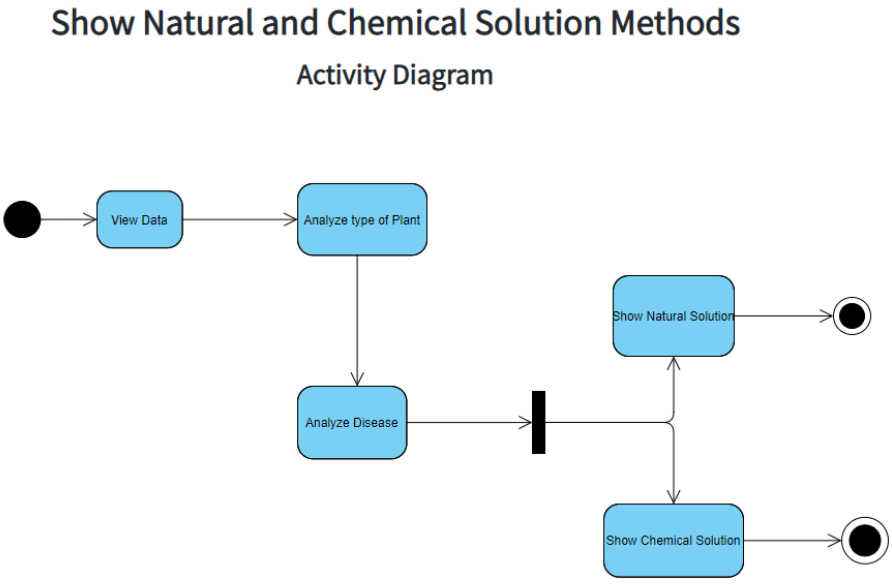
4.2.1.1.2 Add and Take Photo Diagram



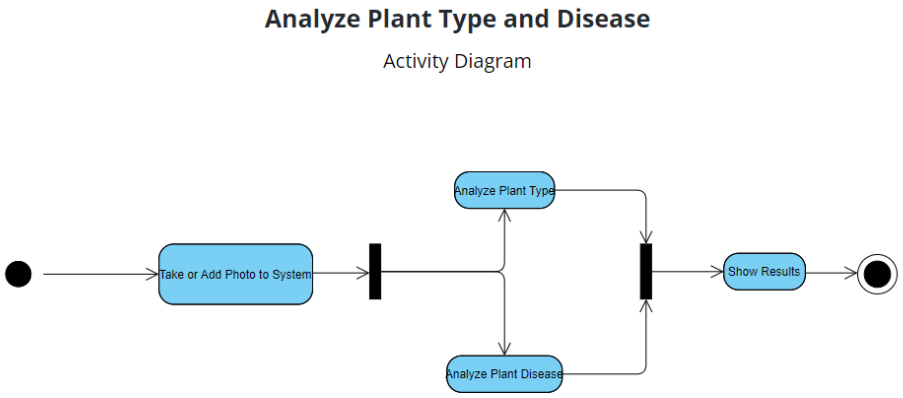
4.2.1.1.3 Find the Closest Pesticide Dealer



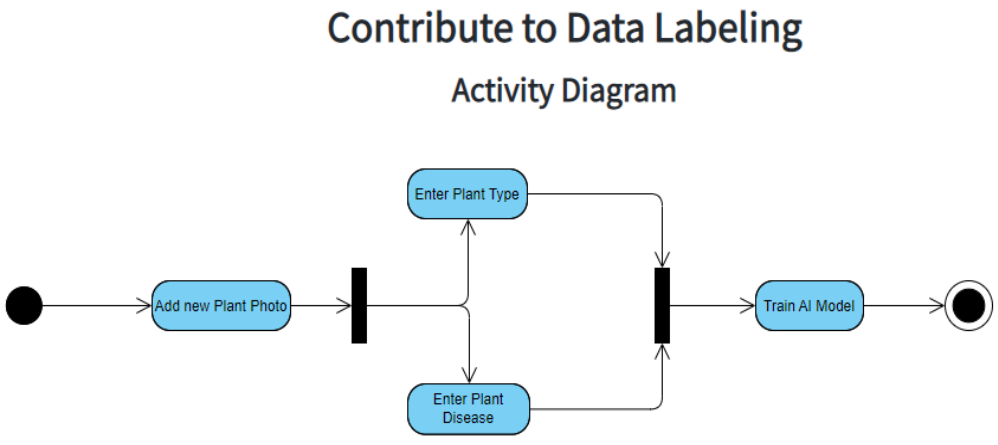
4.2.1.1.4 Show Chemical or Natural Solutions



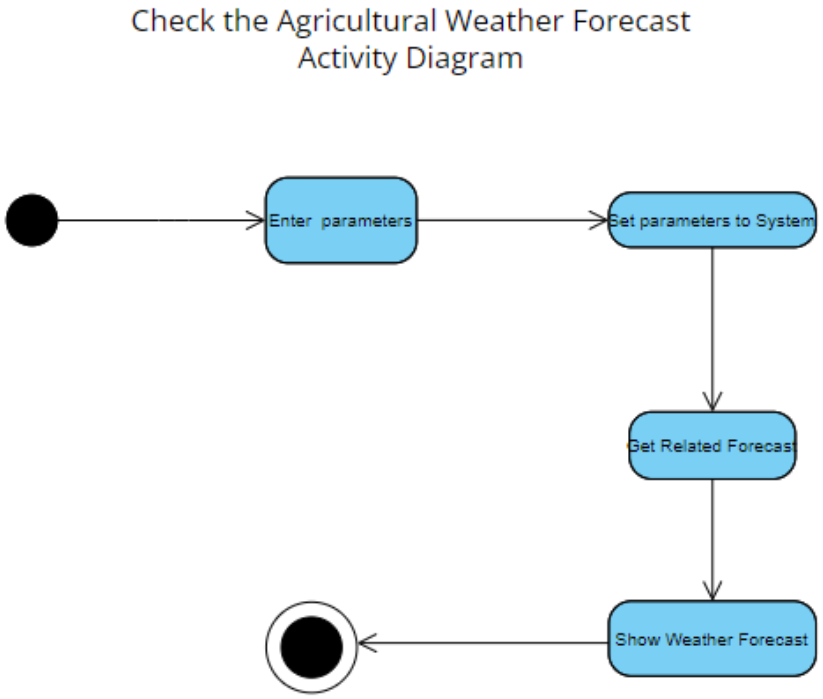
4.2.1.1.5 Analyze Plant Type and Disease



4.2.1.1.6 Contribute to Data Labeling

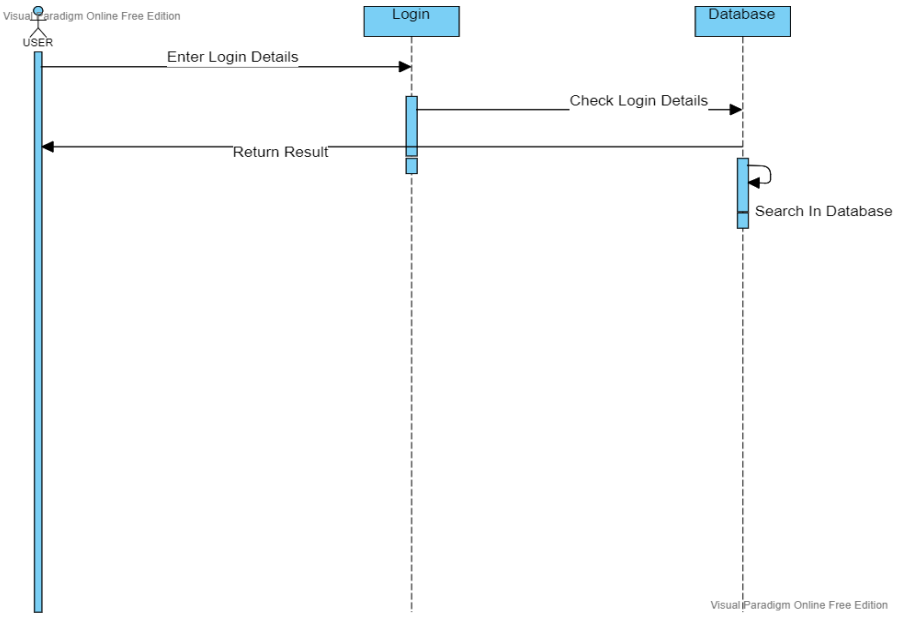


4.2.1.1.7 Check the Agricultural Weather Forecast

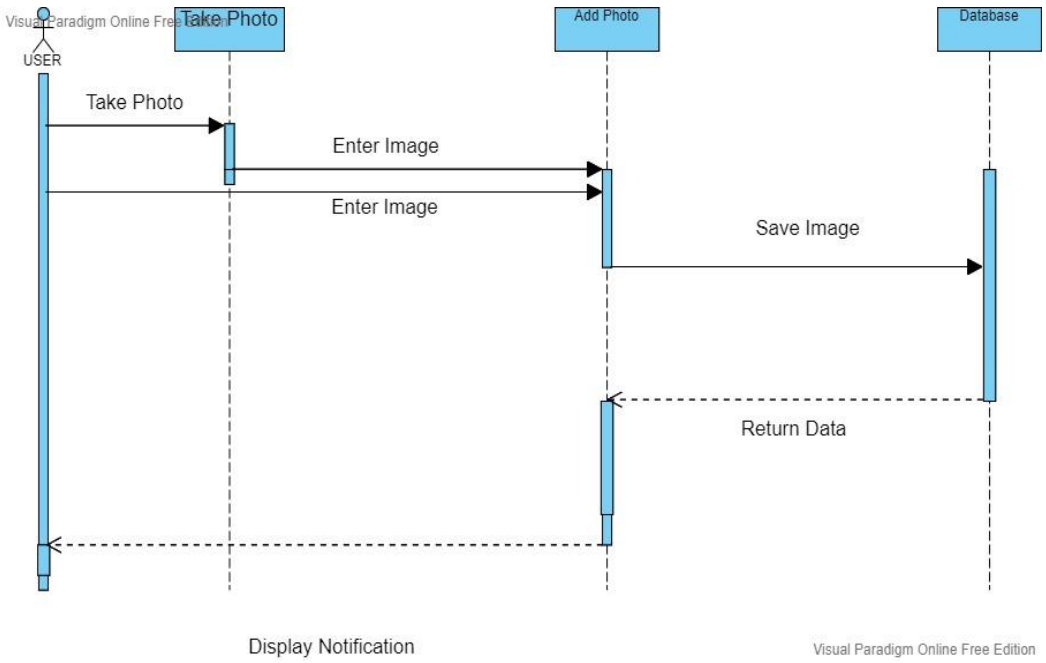


4.2.1.2 Sequence Diagram

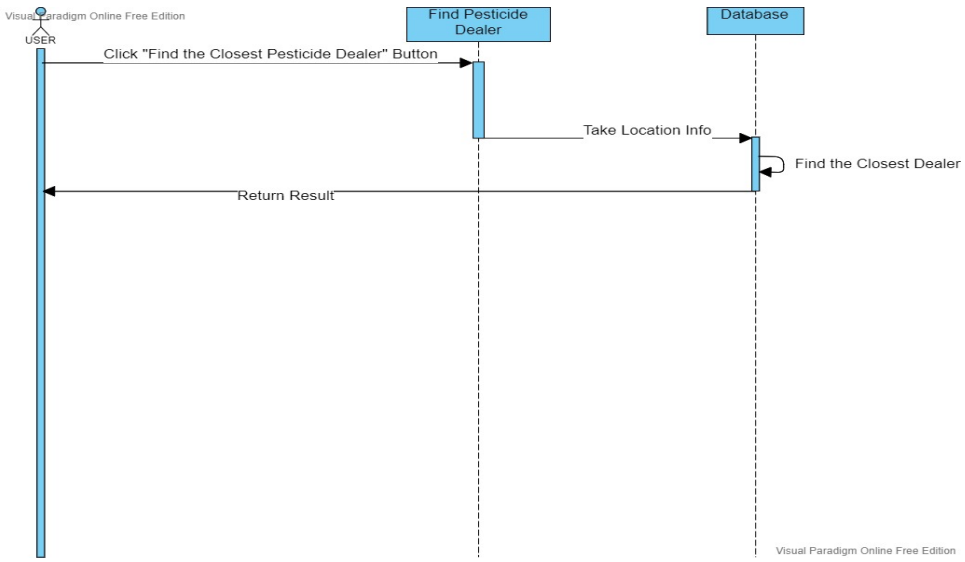
4.2.1.2.1 Login Sequence Diagram



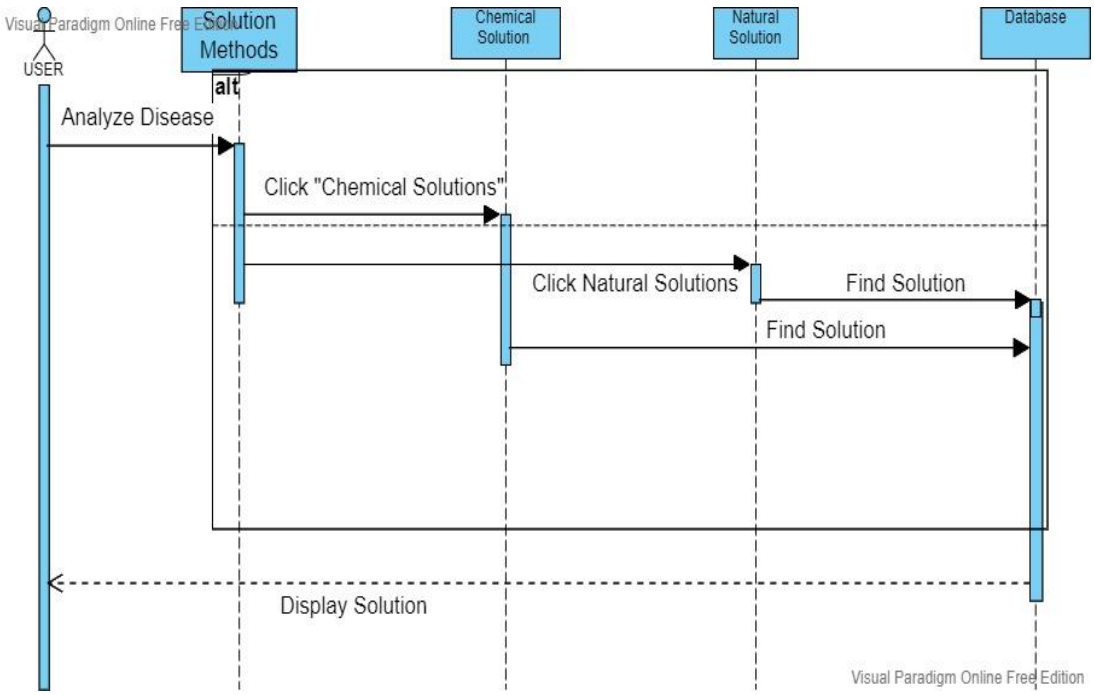
4.2.1.2.2 Add and Take Photo Sequence Diagram



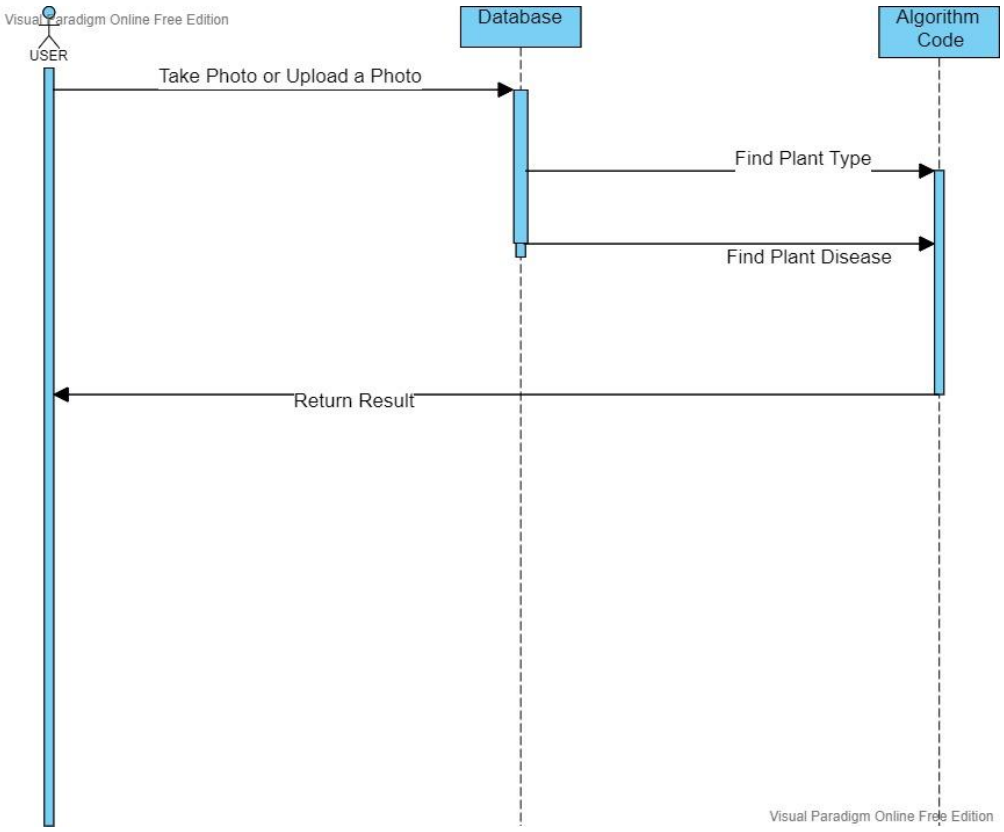
4.2.1.2.3 Find the Closest Pesticide Dealer



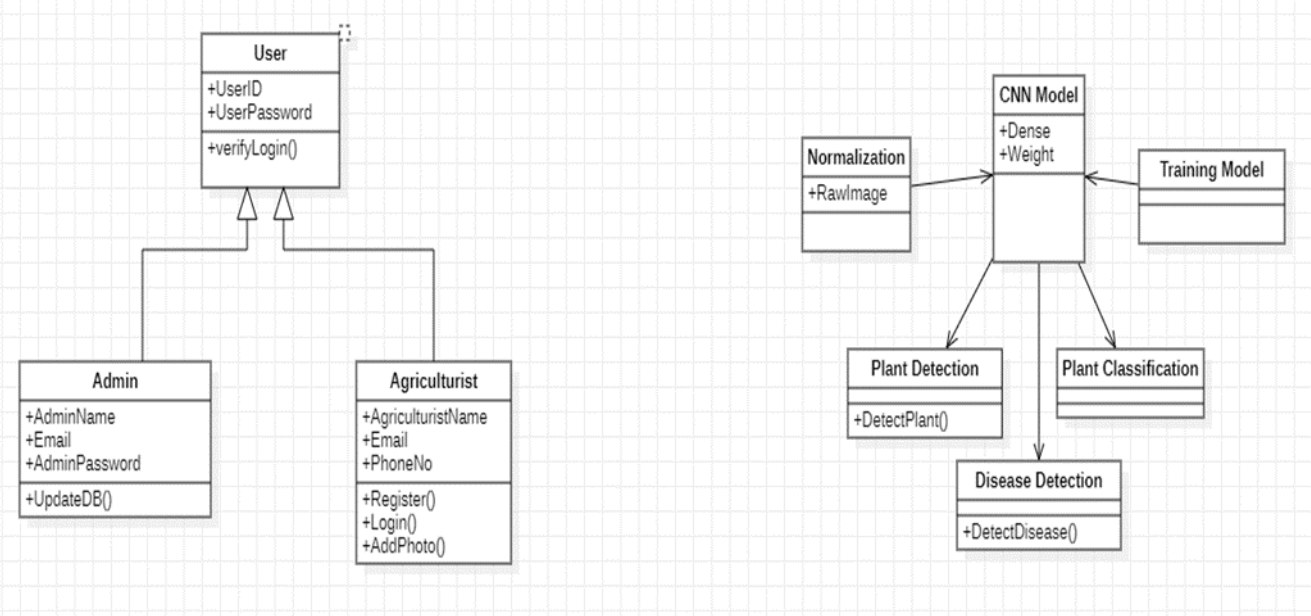
4.2.1.2.4 Show Chemical or Natural Solutions



4.2.1.2.5 Analyze Plant Type and Disease

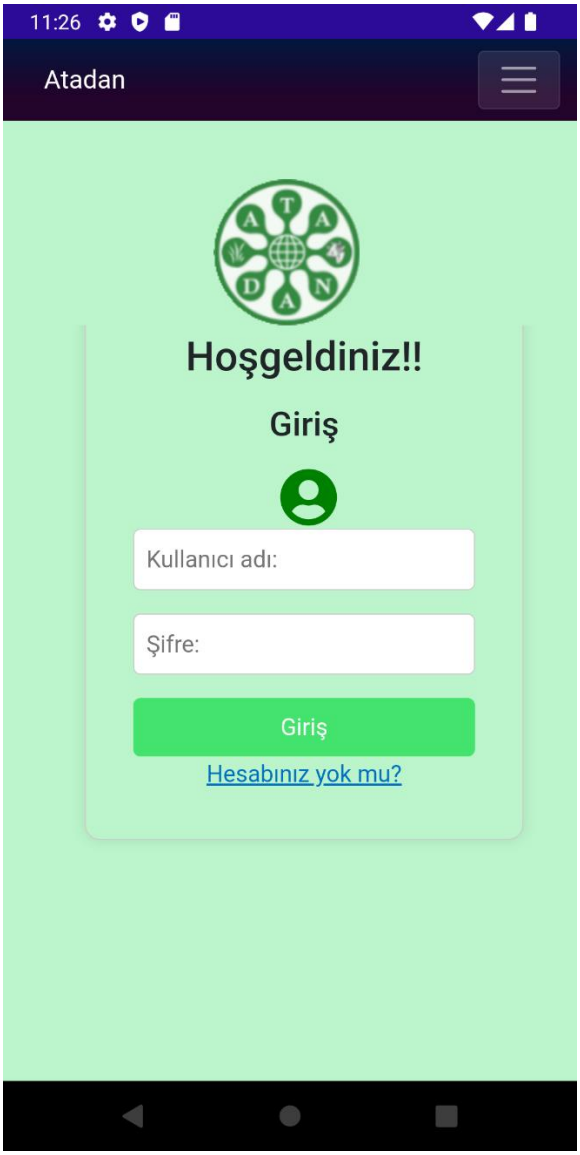


4.2.1.3 Class Diagram

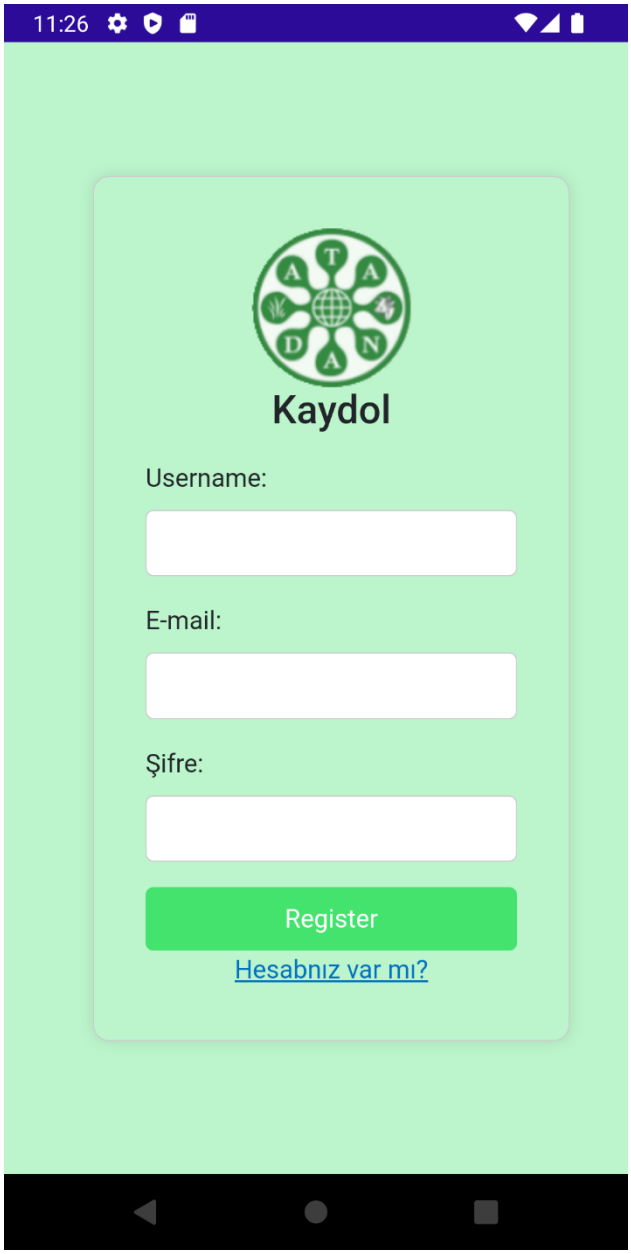


4.2.2 User Interface Design

4.2.2.1 Login Screen UI

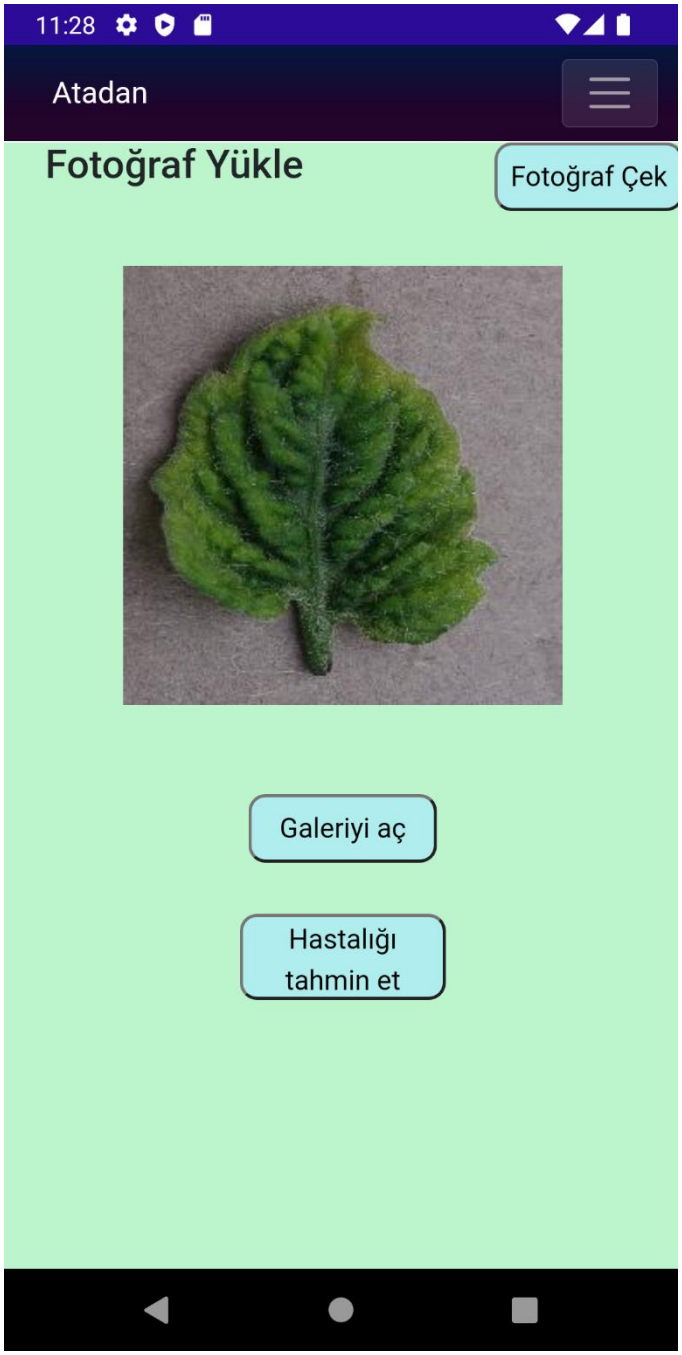


4.2.2.2 Register Screen UI



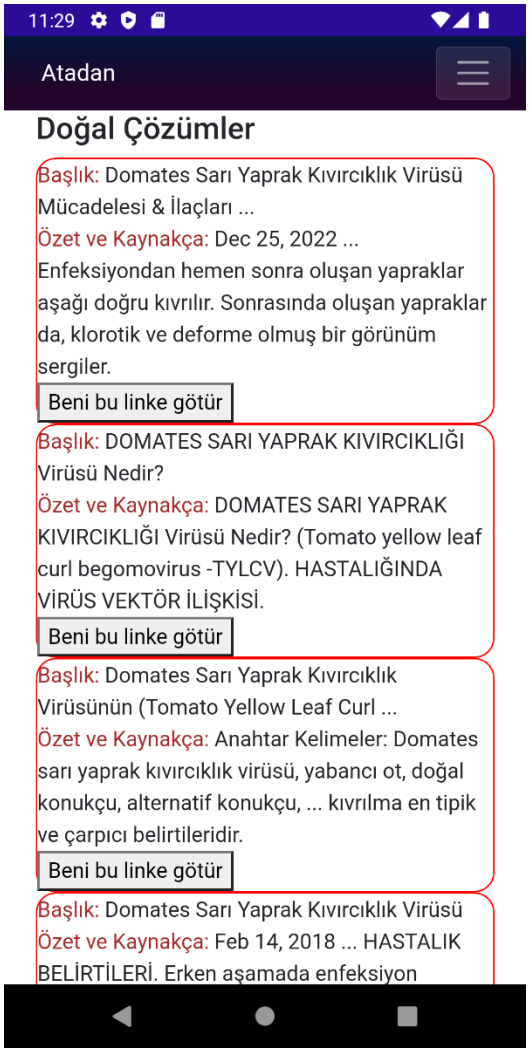
4.2.2.3 Home Page UI

4.2.2.4 Add and Take Photo UI

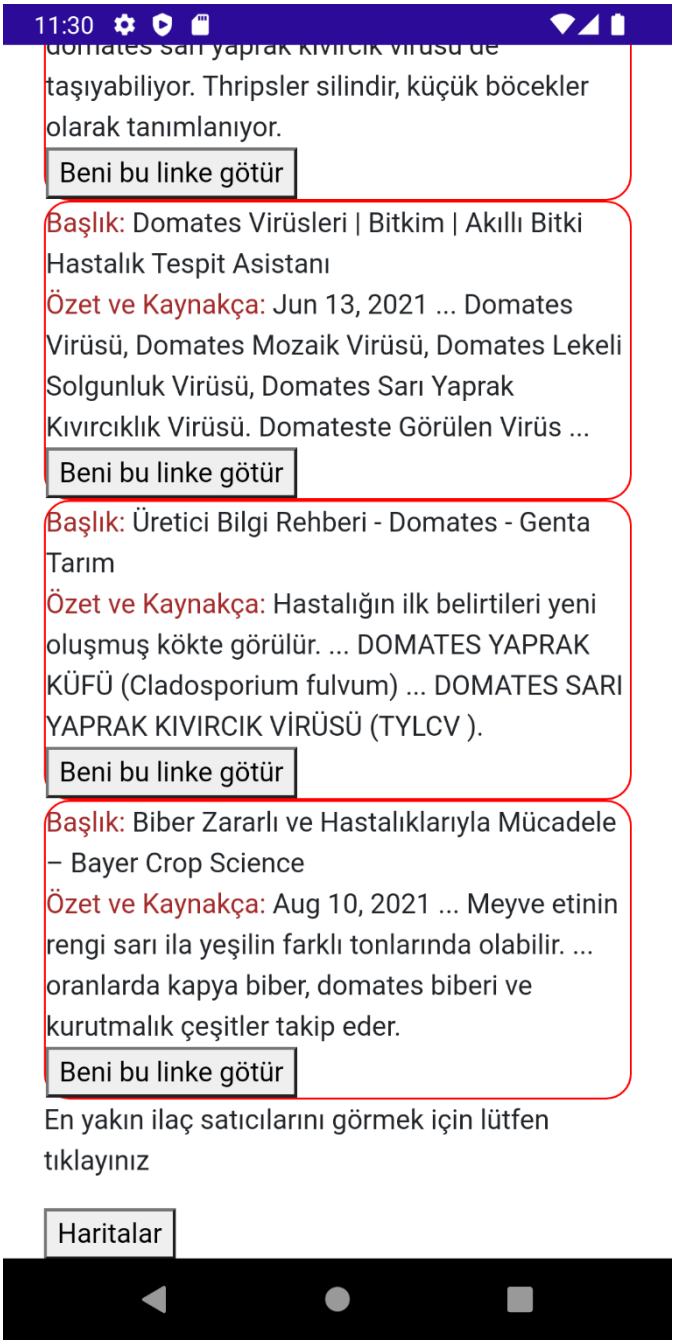


4.2.2.5 Show Result Page UI

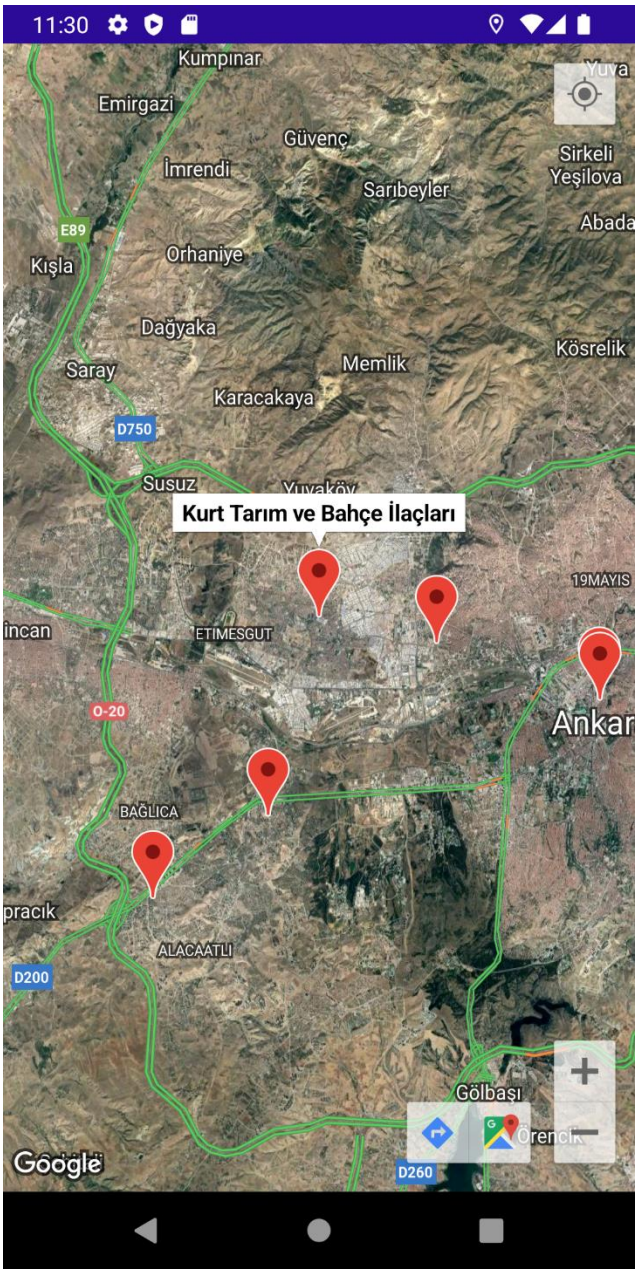
4.2.2.6 Show Natural Solution UI



4.2.2.7 Show Chemical Solution and Find the Closest Pesticide Dealer UI

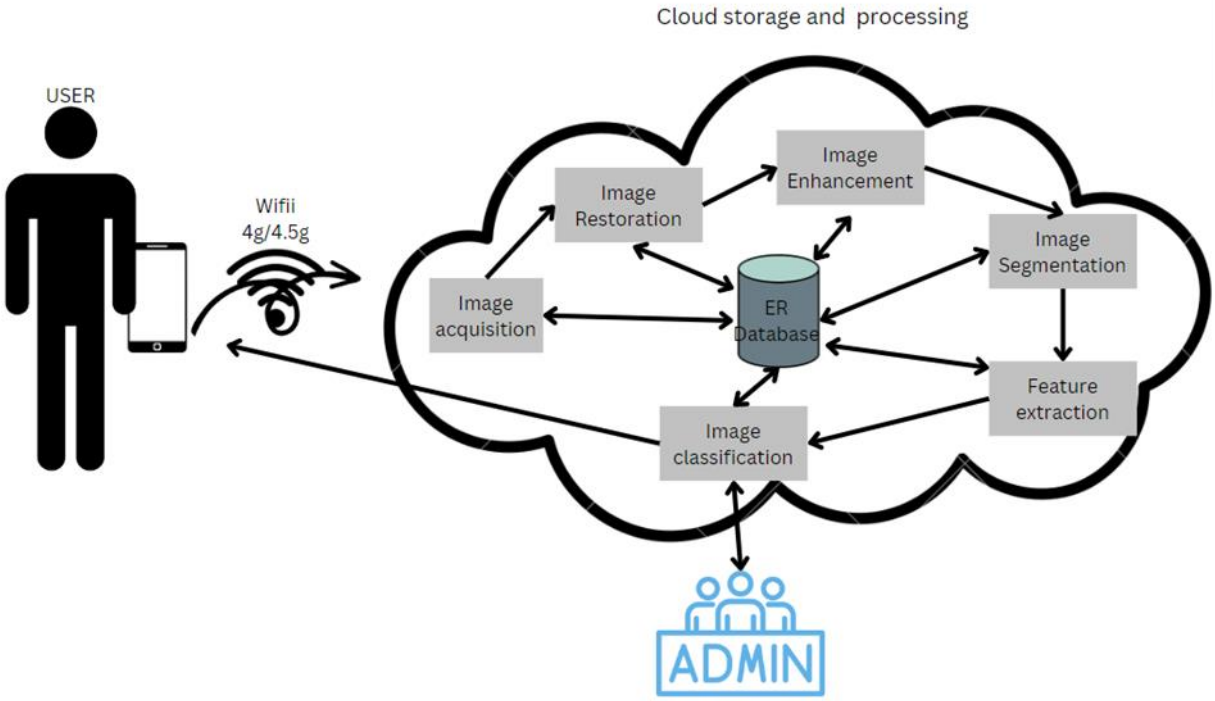


4.2.2.8 Closest Pesticide Dealer



4.2.3 Hardware

4.2.3.1 Cloud Based Architecture



4.3 Plant Recognition

4.3.1 Introduction

In the first stage, plant images are entered into the database with the reading camera. After the system reads the images, it goes to the preprocessing stage. After successfully completing the various pre-processing steps, he moves on to the training phase, where he will learn to identify the plant image. After various pieces of training are completed, the models are tested and the most correct working model is selected. The result, after the whole process has been successfully completed, the system will display the plant type and plant disease if any.

4.3.2 Datasets

We need to consider every possibility that may occur in these datasets. We need to distribute all kinds of situations evenly. In this way, our model will give more reliable and more accurate results. As in every data set, there will be

many missing errors in the data set we will use. This is why the preprocessing steps are of great importance to our dataset. In supervised learning, datasets used for machine learning research assign labels to the data. A training set and a test set are created from the preprocessed data set. The ratio of training and test sets may vary according to the models to be created.

4.3.3 Training dataset

We use this dataset to train our model. It is a compilation of both diseased plant and healthy plant information. Since we will use it in education, it is important that the distribution of the data in these two knowledge is balanced. For this reason, they are often grouped by the same number of components or features. The training dataset is one of the most important components of future feature extraction for plant detection.

Diseased plant picture example from the training set

Healthy plant picture example from training set

4.3.4 Testing dataset

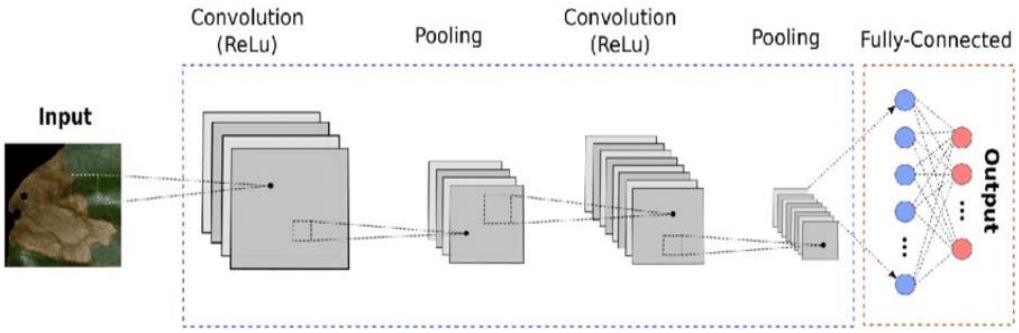
The test data set is the data that has not been used before in the training of the model to test the created model. The balanced distribution of the test data set is important for the accuracy of the test result.

4.3.5 Training and classification

4.3.5.1 OpenCV

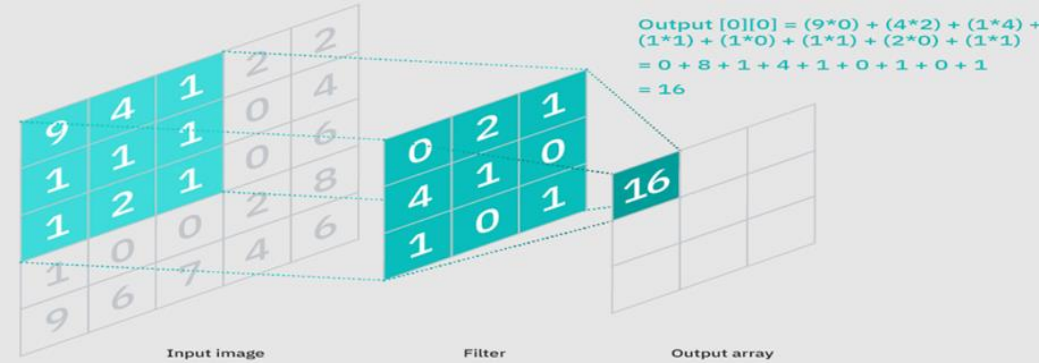
OpenCV is a huge open-source library used for applications such as computer vision, machine learning, image processing, and video analysis. It plays a very important role in real-time transactions. It contains hundreds of functions that support the capture, analysis, and processing of visual information connected to a computer by webcams, video files, or other devices, thanks to the OpenCV library. In this project, various analyzes will be made from the images of the plants and the diagnosis of plant species and diseases will be made with the OpenCV library. We will reach various information by processing the images using the OpenCV library in the same way in the diagnosis of plants taken with the camera. It has interfaces with OpenCV to C++, C, Python, Java, and MATLAB, and the API for these interfaces can be found in the online documentation.

OpenCV also supports Windows, Linux, Mac OS, iOS, and Android. Deep learning algorithms, especially convolutional networks, have quickly become the preferred method for analyzing plant images. CNN is a deep learning algorithm used in image processing that takes images as input. Visuals passing through the Convolutional Layer, Pooling, and Fully Connected layers are developed to the stage of entering the deep learning model by applying many different processes. When creating CNN models, we lose much less time in data preprocessing than machine learning algorithms, since we deal with unstructured data.



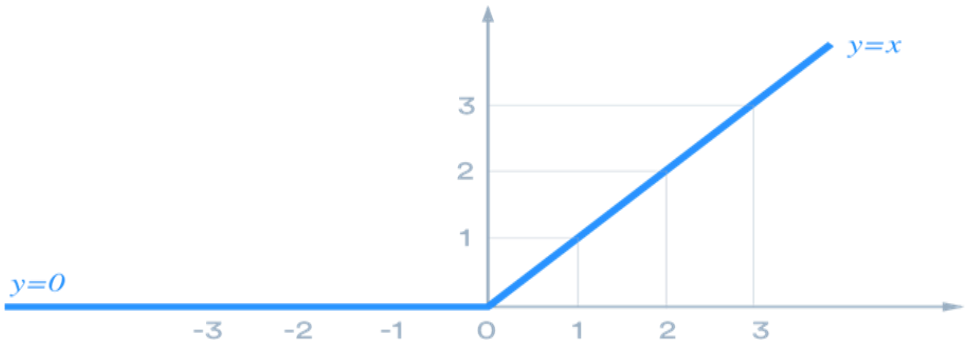
4.3.5.2 Convolutional Layer

It is the first layer that handles the image in CNN. Images are matrices of pixels with certain values in them. In the convolution layer, a matrix filter smaller than the original image size hovers over the original image and thus aims to find certain features from the original images.



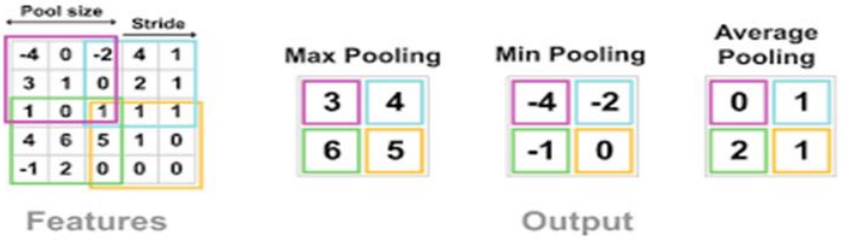
4.3.5.3 ReLU

It is a function that works as f(x) = max(0,x). It is used to prevent our model from learning negative values or failing to grasp some features due to these negative values.



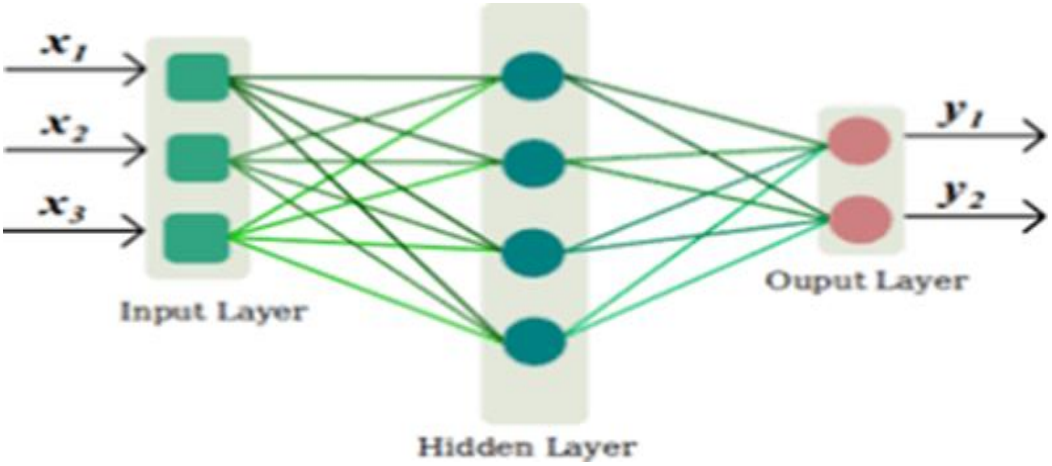
4.3.5.4 Pooling

It is applied to reduce dimensionality. By applying pooling, the required processing power is reduced and more important features are focused on instead of unnecessary features that are captured.



4.3.5.5 Fully Connected Layer

At the nodes in the layers, the features are kept and the learning process is entered by changing the weight and bias.



5. Conclusions

As the project team, we tried to explain the purpose and scope of our project and to introduce how we implemented it. The main goal of our project is to create an application to guide people. Our research has shown us that people who think that their plants have a disease can reduce the death rate in plants thanks to the use of our application and that people who use our application may be more conscious about plant diseases. Also our studies have shown that the SVM classification algorithm and the CNN deep learning algorithm get the best results.

We will photograph a plant and then use an algorithm to display the user the likelihood that the plant will contract a disease. The user may then make a personal album about a particular plant so that they can keep track of any changes, such as the plant becoming infected or changing color.

TEST PLAN

1. INTRODUCTION

2. Version Control

Version No	Description of Changes	Date
1.0	First Version	Mar 26, 2023

3. Overview

The usability, efficiency and error margins of the specified use cases of ATADAN Application will be tested.

4. Scope

This document explains the testing stages and test details of the ATADAN Application. The goal is to help the application develop and reach its most advanced state by testing.

5. Terminology

Acronym	Definition
SU	Sign Up
AT	Analyze Plant Type
AD	Analyze Disease
DL	Contribute Data Labeling
EP	Edit Profile
AP	Add and Take Photo to Album
FC	Find the Closest Pesticide Dealer
SM	Show Natural and Chemical Solution Methods
CA	Check the Agricultural Weather Forecast

6. FEATURES TO BE TESTED

This section lists and gives a brief description of all the major features to be tested. For each major feature there will be a Test Design Specification added at the end of this document.

7. Sign Up (SU)

The feature that allows the registered user to sign up to the system using mail and password.

8. Analyze Plant Type (AT)

The feature that detects the type of plant whose photo is uploaded to the system.

9. Contribute Data Labeling (DL)

The feature that allows the obtained photo data to be classified and stored.

10. Edit Profile (EP)

The feature that allows users to edit their profiles on the system.

11. Add and Take Photo to Album (AP)

The feature that allows users to edit their profiles on the system.

12. Find the Closest Pesticide Dealer (FC)

The feature that allows users to find closest pesticide dealer.

13. Show Natural and Chemical Solution Methods (SM)

The feature that allows users to see natural and chemical solution methods.

14. Check the Agricultural Weather Forecast (CA)

The feature that allows users to edit their profiles on the system.

15. ITEM PASS/FAIL CRITERIA

16. Exit Criteria

Describe under what conditions the testing of the product is considered successful. Some examples are:

- 100% of the test cases are executed
- 95% of the test cases passed
- All High and Medium Priority test cases passed

17. REFERENCES

[1] https://github.com/CankayaUniversity/ceng-407-408-2022-2023-ATADAN/wiki/SDD
[2] https://github.com/CankayaUniversity/ceng-407-408-2022-2023-ATADAN/wiki/SRS

18. TEST DESIGN SPECIFICATIONS

19. Sign Up (SU)

20. Subfeatures to be tested

21. *User (SU.US)*

Person trying to register to the server to use the application.

22. *Sign Up Button (SU.SUB)*

After providing the necessary information, the button to be clicked on to register.

23. *Input Display (SU.ID)*

Screen output that allows the user to properly enter the required information.

24. Test Cases

Here list all the related test cases for this feature

TC ID	Requirements	Priority	Scenario Description
SU.US.01	2.1	H	Enter a user id and password
SU.US.02	2.1	H	Enter a user id and blank password
SU.US.03	2.1	H	Enter a blank user id and password
SU.US.04	2.1	H	Enter a blank user id and blank password

TC ID	Requirements	Priority	Scenario Description
SU.SUB.01	2.1	M	Click button 1 time
SU.SUB.02	2.1	S	Click button more than 1 time rapidly

TC ID	Requirements	Priority	Scenario Description
SU.ID.01	2.1	S	Change display window size

25. Analyze Plant Type (AT)

26. Subfeatures to be tested

27. *Taking Photo (AT.TP)*

Taking the photo instantly and uploading it to the system for review.

28. *Uploading Photo(AT.UP)*

Uploading the previously taken photo from the device memory to the system.

29. Test Cases

Here list all the related test cases for this feature

TC ID	Requirements	Priority	Scenario Description
AT.TP.01	2.2	H	Taking a plant photo
AT.TP.02	2.2	H	Taking a non plant photo

TC ID	Requirements	Priority	Scenario Description
AT.UP.01	2.2	H	Uploading a plant photo
AT.UP.02	2.2	H	Uploading a non plant photo

30. Contribute Data Labeling (DL)

31. *Adding Data (DL.AD)*

Describe in 1-2 sentence this sub feature.

32. Test Cases

Here list all the related test cases for this feature

TC ID	Requirements	Priority	Scenario Description
DL.AD.01	2.3	H	Adding valid data
DL.AD.02	2.3	H	Adding unvalid data

33. Edit Profile (EP)

34. Subfeatures to be tested

35. *User (EP.US)*

Person trying to edit his profile in the application.

36. *Edit Profile Button (EP.EPB)*

Button that the user uses to edit their profile.

37. Test Cases

Here list all the related test cases for this feature

TC ID	Requirements	Priority	Scenario Description
EP.US.01	2.4	H	Change user name
EP.US.02	2.4	H	Change password

TC ID	Requirements	Priority	Scenario Description
EP.EPB.01	2.4	M	Click button 1 time
EP.EPB.02	2.4	S	Click button more than 1 time rapidly

38. Add and Take Photo to Album (AP)

39. Sub features to be Tested

40. *Add Photo (AP.AP)*

Add a previously taken photo to the system.

41. *Take Photo (AP.TP)*

Taking a new photo using the app.

42. Test Cases

Here list all the related test cases for this feature

TC ID	Requirements	Priority	Scenario Description
AP.AP.01	2.5	H	Add photo
AP.AP.02	2.5	H	Add another file

TC ID	Requirements	Priority	Scenario Description
AP.TP.01	2.5	H	Take photo
AP.TP.02	2.5	H	Take non plant photo

43. Find the Closest Pesticide Dealer (FC)

44. Subfeatures to be tested

45. *Enter Location (FC.EL)*

User enters his address in the system.

TC ID	Requirements	Priority	Scenario Description
FC.EL.01	2.6	M	Enter a location
FC.EL.02	2.6	M	Enter something else

46. Show Natural and Chemical Solution Methods (SM)

47. Subfeatures to be tested

48. *Show Solution Button (SM.SS)*

Button to show solution methods.

TC ID	Requirements	Priority	Scenario Description
SM.SS.01	2.7	M	Click button 1 time

SM.SS.02	2.7	S	Click button more than 1 time rapidly

49. Check the Agricultural Weather Forecast (CA)

50. Subfeatures to be tested

51. *Set Parameter (CA.SP)*

Add a previously taken photo to the system.

52. *Check Weather Button (CA.CW)*

Taking a new photo using the app.

53. Test Cases

Here list all the related test cases for this feature

TC ID	Requirements	Priority	Scenario Description
CA.SP.01	2.9	H	Set parameter
CA.SP.02	2.9	H	Set wrong parameter

TC ID	Requirements	Priority	Scenario Description
CA.CW.01	2.9	M	Click button 1 time
CA.CW.02	2.9	S	Click button more than 1 time rapidly

54. Detailed Test Cases

55. SU.US.01

TC_ID	SU.US.01
Purpose	Enter a user id and password
Requirements	2.1
Priority	High.
Estimated Time Needed	3 Minutes
Dependency	Warning test cases should pass
Setup	An user should be created.
Procedure	[A01] Go to sign up page.
	[A02] Enter a user id.
	[A03] Enter the password for this user
	[A04] Click on the “Sign Up” button.
	[V01] Observe that the signing up is successful and the user page appears
	-
Cleanup	Logout

56. SU.US.02

TC_ID	SU.US.02
Purpose	Enter a user id and blank password
Requirements	2.1
Priority	High.
Estimated Time Needed	2 Minutes
Dependency	Warning test cases should pass
Setup	A warning must be show up.
Procedure	[A01] Go to sign up page.
	[A02] Enter a valid user id.
	[A03] Leave password empty.
	[A04] Click on the “Sign UP” button.
	[V01] Observe that the warning window works properly.
	-
Cleanup	Close warning

57. SU.US.03

TC_ID	SU.US.03
Purpose	Enter a blank user id and password
Requirements	2.1
Priority	High.
Estimated Time Needed	2 Minutes
Dependency	Warning test cases should pass
Setup	A warning must be show up.
Procedure	[A01] Go to sign up page.
	[A02] Leave user id empty.
	[A03] Enter random password for this user
	[A04] Click on the “Sign Up” button.

	[V01] Observe that the warning window works properly.
	-
Cleanup	Close warning

58. SU.US.04

TC_ID	SU.US.04
Purpose	Enter a blank user id and blank password
Requirements	2.1
Priority	High.
Estimated Time Needed	2 Minutes
Dependency	Warning test cases should pass
Setup	A warning must be show up.
Procedure	[A01] Go to sign up page.
	[A02] Leave user id empty.
	[A03] Leave password empty.
	[A04] Click on the “Sign Up” button.
	[V01] Observe that the warning window works properly.
	-
Cleanup	Close warning

59. SU.SUB.01

TC_ID	SU.SUB.01
Purpose	Click sign up button 1 time
Requirements	2.1
Priority	Medium.
Estimated Time Needed	1 Minutes
Dependency	Button test cases should pass
Setup	Sign up button must gets darker.
Procedure	[A01] Go to sign up page.
	[A02] Click on the “Sign Up” button.
	[V01] Observe that the sign up button works properly.
	-
Cleanup	Click blank

60. SU.SUB.02

TC_ID	SU.SUB.02
Purpose	Click sign up button more than 1 time rapidly
Requirements	2.1
Priority	Small.
Estimated Time Needed	1 Minutes
Dependency	Button test cases should pass
Setup	Sign up button must gets darker rapidly.
Procedure	[A01] Go to sign up page.
	[A02] Click on the “Sign Up” button.
	[V01] Observe that the sign up button works properly.
	-
Cleanup	Click blank

61. SU.ID.01

TC_ID	SU.ID.01
Purpose	Change display window size
Requirements	2.1
Priority	Small.
Estimated Time Needed	2 Minutes
Dependency	Display test cases should pass
Setup	Display size must change.
Procedure	[A01] Go to sign up page.
	[A02] Click on the corner of display.
	[A03] Change size
	[V01] Observe that the display size changes properly.
	-
Cleanup	Leave page

62. AT.TP.01

TC_ID	AT.TP.01
Purpose	Taking a plant photo
Requirements	2.2
Priority	High.
Estimated Time Needed	4 Minutes
Dependency	Photo take test cases should pass
Setup	A photo must taken.
Procedure	[A01] Go to camera.
	[A02] Set frame.
	[A03] Click on the circle button.
	[V01] Observe that the photo taken.
	-
Cleanup	Exit camera

63. AT.TP.02

TC_ID	AT.TP.02
Purpose	Taking a non plant photo
Requirements	2.2
Priority	High.
Estimated Time Needed	3 Minutes
Dependency	Photo take test cases should pass
Setup	A photo must taken but not accepted.
Procedure	[A01] Go to camera.
	[A02] Set frame.
	[A03] Click on the circle button.
	[V01] Observe that the photo taken and not accepted.
	-
Cleanup	Exit camera

64. AT.UP.01

TC_ID	AT.UP.01
Purpose	Uploading a plant photo
Requirements	2.2
Priority	High.
Estimated Time Needed	3 Minutes
Dependency	Photo upload test cases should pass
Setup	A photo must uploaded.
Procedure	[A01] Go to gallery.
	[A02] Select photo.
	[V01] Observe that the photo Uploaded.
	-
Cleanup	Exit gallery

65. AT.UP.02

TC_ID	AT.UP.02
Purpose	Uploading a non plant photo
Requirements	2.2
Priority	High.
Estimated Time Needed	3 Minutes
Dependency	Photo upload test cases should pass
Setup	A photo must uploaded but not accepted.
Procedure	[A01] Go to gallery.
	[A02] Select photo.
	[V01] Observe that the photo Uploaded but not accepted.
	-
Cleanup	Exit gallery

66. DL.AD.01

TC_ID	DL.AD.01
Purpose	Adding valid data
Requirements	2.3
Priority	High.
Estimated Time Needed	4 Minutes
Dependency	Data test cases should pass
Setup	A data must added.
	[A01] Select data.
	[V01] Observe that the data added.
	-
Cleanup	Go to menu

67. DL.AD.02

TC_ID	DL.AD.02
Purpose	Adding unvalid data
Requirements	2.3
Priority	High.
Estimated Time Needed	4 Minutes
Dependency	Data test cases should pass
Setup	A data must added but not accepted.
	[A01] Select data.
	[V01] Observe that the data added but not accepted.
	-
Cleanup	Go to menu

68. EP.US.01

TC_ID	EP.US.01
Purpose	Change user name
Requirements	2.4
Priority	High.
Estimated Time Needed	2 Minutes
Dependency	Warning test cases should pass
Setup	An user name should be changed.
Procedure	[A01] Go to edit profile page.
	[A02] Enter a new user id.
	[A03] Click on the “Change User Name” button.
	[V01] Observe that the user name change is successful and the user page appears
	-
Cleanup	Logout

69. EP.US.02

TC_ID	EP.US.02
Purpose	Change password
Requirements	2.4
Priority	High.
Estimated Time Needed	2 Minutes
Dependency	Warning test cases should pass
Setup	A password should be changed.
Procedure	[A01] Go to edit profile page.
	[A02] Enter a new password.
	[A03] Leave password empty.
	[A04] Click on the “Change Password” button.
	[V01] Observe that the password change is successful and the user page appears
	-
Cleanup	Logout

70. EP.EPB.01

TC_ID	EP.EPB.01
Purpose	Click change button 1 time
Requirements	2.1
Priority	Medium.
Estimated Time Needed	1 Minutes
Dependency	Button test cases should pass
Setup	Change button must gets darker.
Procedure	[A01] Go to edit profile page.
	[A02] Click on the “Change” button.
	[V01] Observe that the sign up button works properly.
	-
Cleanup	Click blank

71. EP.EPB.02

TC_ID	EP.EPB.02
Purpose	Click change button more than 1 time rapidly
Requirements	2.1
Priority	Small.
Estimated Time Needed	1 Minutes
Dependency	Button test cases should pass
Setup	Change button must gets darker rapidly.
Procedure	[A01] Go to edit profile page.
	[A02] Click on the “Change” button.
	[V01] Observe that the sign up button works properly.
	-
Cleanup	Click blank

72. AP.AP.01

TC_ID	AP.AP.01
Purpose	Adding suitable photo from gallery
Requirements	2.5
Priority	High.
Estimated Time Needed	1 Minutes
Dependency	Button test cases should pass
Setup	Sign up button must gets darker.
Procedure	[A01] Go to Home after the login page
	[A02] Click on the “Open Gallery” button
	[V01] Choose the appropriate photo from the gallery.
	-
Cleanup	Go to menu

73. AP.AP.02

TC_ID	AP.AP.02
Purpose	Adding unsuitable photo from gallery
Requirements	2.5

Priority	High.
Estimated Time Needed	1 Minutes
Dependency	Button test cases should pass
Setup	Sign up button must gets darker.
Procedure	[A01] Go to Home after the login page
	[A02] Click on the “Open Gallery” button
	[V01] Choose the unsuitable photo from the gallery.
	-
Cleanup	Go to menu

74. FC.EL.02

TC_ID	FC.EL.01
Purpose	Finding Closest Pesticide Dealer
Requirements	2.6
Priority	Medium.
Estimated Time Needed	1 Minutes
Dependency	Button test cases should pass
Setup	Sign up button must gets darker.
Procedure	[A01] Go to Closest Pesticide Dealer page.
	[A02] Connect to Google Maps.
	[V01] Show the suitable pesticide dealers.
	-
Cleanup	Go to menu

75. FC.EL.02

TC_ID	FC.EL.02
Purpose	Finding Closest Pesticide Dealer
Requirements	2.6
Priority	Medium.
Estimated Time Needed	1 Minutes
Dependency	Button test cases should pass
Setup	Sign up button must gets darker.
Procedure	[A01] Go to Closest Pesticide Dealer page.
	[A02] Connect to Google Maps.
	[V01] Show the unsuitable pesticide dealers.
	-
Cleanup	Go to menu

76. CA.SP.01

TC_ID	CA.SP.01
Purpose	Setting valid data
Requirements	2.9
Priority	High.
Estimated Time Needed	3 Minutes

Dependency	Data test cases should pass
Setup	A parameter must added.
	[A01] Set parameter.
	[V01] Observe that the parameter set.
	-
Cleanup	Go to menu

77. CA.SP.02

TC_ID	CA.SP.02
Purpose	Setting unvalid data
Requirements	2.9
Priority	High.
Estimated Time Needed	3 Minutes
Dependency	Data test cases should pass
Setup	A parameter must setted but not accepted.
	[A01] Set wrong parameter.
	[V01] Observe that the parameter set but not accepted.
	-
Cleanup	Go to menu

78. CA.CW.01

TC_ID	CA.CW.01
Purpose	Click sign up button 1 time
Requirements	2.9
Priority	Medium.
Estimated Time Needed	1 Minutes
Dependency	Button test cases should pass
Setup	Sign up button must gets darker.
Procedure	[A01] Go to Check the Agricultural Weather Forecast page.
	[A02] Click on the “Check Weather” button.
	[V01] Observe that the Check Weather button works properly.
	-
Cleanup	Click blank

79. CA.CW.02

TC_ID	CA.CW.02
Purpose	Click sign up button more than 1 time rapidly
Requirements	2.9
Priority	Small.
Estimated Time Needed	1 Minutes
Dependency	Button test cases should pass
Setup	Sign up button must gets darker rapidly.
Procedure	[A01] Go to Check the Agricultural Weather Forecast page.
	[A02] Click on the “Check Weather” button.

	[V01] Observe that the Check Weather button works properly.
	-
Cleanup	Click blank

9.1 User Manuel For Web Page

Technology

- Asp.net MVC
- Entity Framework(Code First)
- C# programming language
- Project Design (Html5, Bootstrap, JavaScript, Jquery)
- MSSQL

Step 1: Open Page

If you are registered to the system as an administrator, you can login to the Web Page. When you type your administrator email and password and click login, if your password and email are correct, you will be logged into the site. If not, you will get an error message.

Step 2: Login

When you log in, you will be directed to the page you will see with the date and disease names under the photos taken and uploaded by experts in the mobile application. Here you will find the photos sorted by the date they were uploaded.

Step 3: Search Image

By typing the disease you are looking for in the search box on the page, you can get the photos of that disease.

Step 4: Photo Review

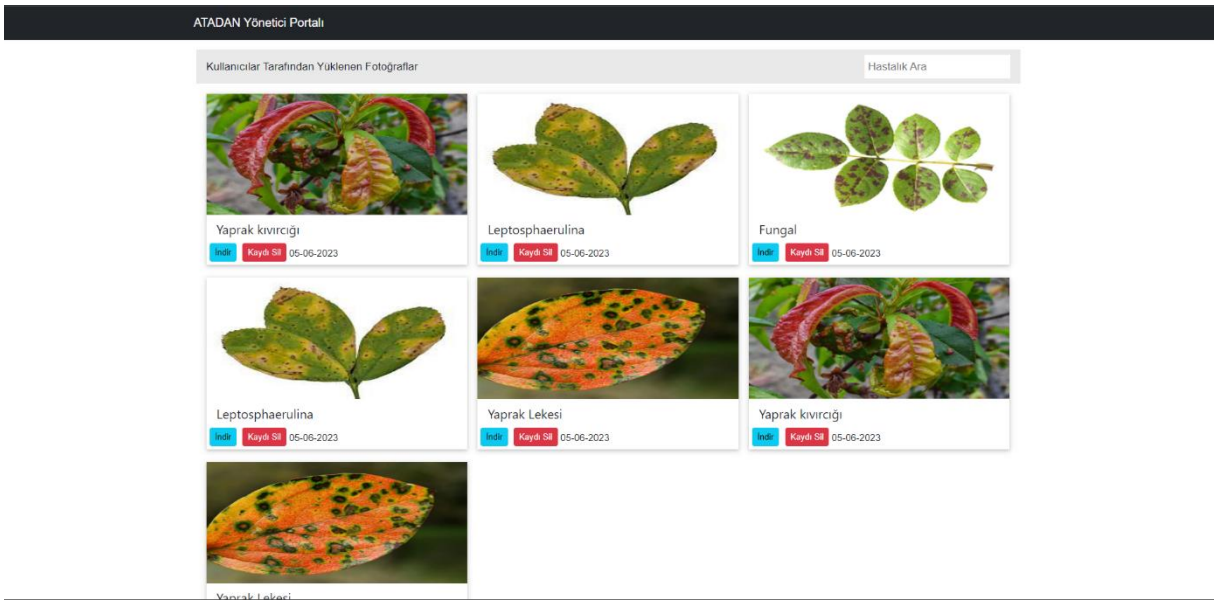
When you click on the photos, you can see the larger version of the photo in a new tab.

Step 5: Download Photo

When you click the download button under the photos, the photo will be downloaded to your computer's local memory. You can use this photo to develop your model.

Step 6: Delete Photo

When you click the delete button under the photos, the photo is deleted from the database and will not appear on the site again. This prevents a photo from being used more than once by different managers to train a model.

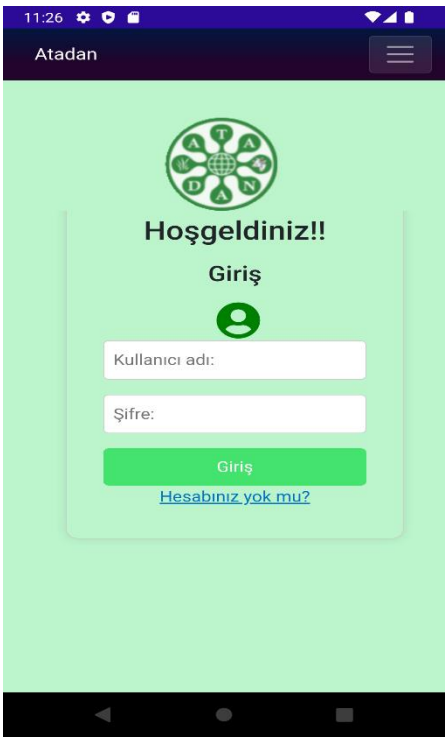


9.1 User Manuel for the Mobil Application

- Technology
- .Net Maui
- Ml Onnx
- Control Maps
- Six Labors
- Skia Sharp
- Tensorflow AI Model
- Onion

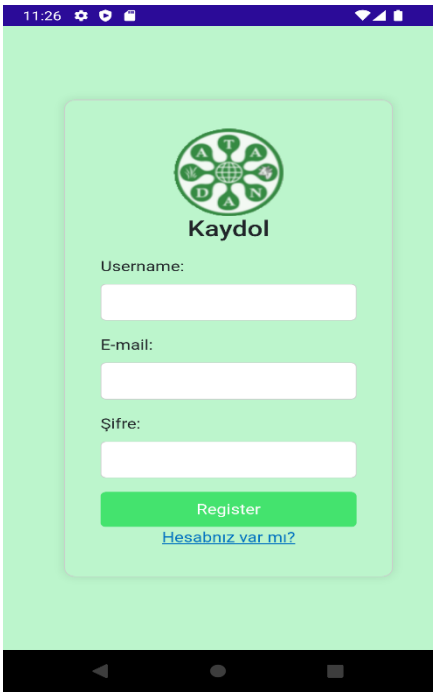
Step 1: Open Page

If you don't have an account registered, "Don't have an account?" at the bottom of the login page. By clicking on the text, you will be directed to the sign-up page. On this page, you can register by typing your username, email, and password information.



Step 2: Login

You can log in to the application with your username and password registered to the system. If the username and password are incorrect, you will get an error message.



Step 3: Navigation Bar

After logging in, you will see a navigation bar with 5 options.

These options are Homepage, Weather, Login, Sign Up, Contribute to Artificial Intelligence.

Step 4: Home

There are 3 buttons on the homepage. These are: Take photo, Open gallery and Predict disease buttons.

Step 5: Upload an Image

You can take a photo from your device's camera with the "take a photo" button on the homepage. The other option is to upload a photo of your choice from the gallery of your device with the "Open gallery" button.

Step 6: Disease Prediction

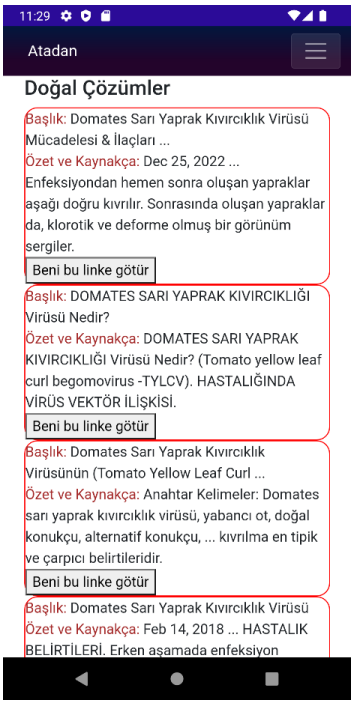
After uploading your photo, you can have the model show you the disease prediction by pressing the "guess the disease" button.

Step 7: Disease Result

On the screen that opens, the photo you upload and the detected disease are shown with the probability of accuracy.

Step 8: Solutions

As a result of the disease, there are two buttons at the bottom. When you click on the "Natural Remedies" button, the page with suggestions for natural remedies opens. When the "Chemical Solutions" button is pressed, you are directed to the map showing the closest pesticide shops.



Step 9: Weather

Clicking on the weather will show the weather of your area.

Step 10: Contribution to Artificial Intelligence

When you click on the contribution to the artificial intelligence, there are boxes where you can upload a photo and the necessary information about the disease. If you fill in this information and click the "contribute to artificial intelligence" button, the information is saved in the database.

WORK PLAN

Tasks	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7
Updated Project Work Plan / LR / SRS / SDD							
Test Plan Document							
Mobile App							
Midterm-Interactive-Demo							
User Manual							
Project Report / Test Results							
Project Poster							
Project Tracking Form							
Updated Project Webpage							
Demo Video							
Presentation							

Tasks	Week 8	Week 9	Week 10	Week 11	Week 12
Updated Project Work Plan / LR / SRS / SDD					
Test Plan Document					
Mobile App					
Midterm-Interactive-Demo					
User Manual					
Project Report / Test Results					
Project Poster					
Project Tracking Form					
Updated Project Webpage					
Demo Video					
Presentation					