Applied Information Technology And Computer Science Vol. 4 No. 2 (2023) 1091-1109
© Universiti Tun Hussein Onn Malaysia Publisher's Office

AITCS

Homepage: http://publisher.uthm.edu.my/periodicals/index.php/aitcs
e-ISSN :2773-5141

Plant Disease Detection Application Using Deep Learning (PLANTSCARE)

Mohammed Fuad Mohammed Ahmed Saif [1], **Nureize Arbaiy**[1*]

[1]Fakulti Sains Komputer dan Teknologi Maklumat,
Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA

DOI: https://doi.org/10.30880/aitcs.2023.04.02.062
Received 20 June 2023; Accepted 07 November 2023; Available online 30 November 2023

Abstract: Plant leaf diseases are conditions that harm a plant's leaves. One of the main problems that every farmer encounters is the diagnosis of plant diseases, which can be brought on by a variety of causes. The development of advanced computing technology has allowed for the improvement of agricultural decision-making. Deep learning technology is then useful in this circumstance. because it outperforms earlier conventional approaches in terms of dependability, accuracy, speed, and cost-effectiveness. As a result, the goal of this project is to develop an Android application that will allow farmers to identify the presence of plant diseases by taking a photo of the plant and using deep learning. This project will use the deep learning technique of convolutional neural networks (CNN). The model has been trained by Google Teachable machine platform on a dataset of images of diseased and healthy plants and is able to accurately identify various plant diseases. Without assistance from agronomists, the algorithm will provide correct answers about the type of plant disease. The prototyping model has been employed to assist the development of the web application system. The use of this application could improve crop productivity and contribute to the overall development of a country's agriculture industry.

Keywords: Plant disease, convolutional neural network, deep learning

1. Introduction

Plants are very essential in our life as they provide a source of energy [1]. A plant is a living thing with a stem, leaves, and roots that grow in the ground. Plants are vital to the planet and all living things. Plants absorb carbon dioxide and release oxygen through their leaves, which humans and other animals require to breathe [2]. Plants provide many products for human consumption or industry, including food, medicines, fibres, timber, firewood, oils, dyes, and so on like other living things, plants are also susceptible to disease and damage [3]. The development of a country's economy depends on the mass of agricultural land and its productivity, whereas most of the population depends on agriculture. Farmers grow different crops based on soil fertility and resource availability. Plants are affected by many diseases such as they cause devastating economic, social, and ecological losses and many more.

Due to changes in environmental conditions, such as temperature, rainfall, and soil fertility, crops can be infected with viruses, fungi, and bacteria [3]. The detection of diseases in plants is one of the

*Corresponding author: nureize@uthm.edu.my
2023 UTHM Publisher. All rights reserved.

main issues that every farmer faces. One of the most traditional methods used by farmers in the detection of plant diseases is through visual observation where farmers look at changes and effects in plant leaves and through that farmer predicts the type of disease nowadays this method with technological advances is considered a rudimentary method with a reason that is not very precise because some diseases are similar in symptoms.

Detecting plant disease at an early stage is advantageous because the disease can be controlled. Because farmers rely on agricultural experts, some farmers are unaware of how to contact the experts. Due to other work commitments, agricultural experts are not always available to assist. When expert knowledge resides in the expert's mind [4]. It is difficult to obtain and distribute to farmers. When farmers use their own knowledge [5]. However, the observed results are not always accurate. This is because they learn through experience, which can be misleading at times. Therefore, some techniques have been developed to help farmers learn more about plant diseases. One of the it is an expert system which becomes a tool for generating information from knowledge, and the advice produced by the expert system is used for those in need. At the same time, expert systems allow information from experts to be more widely used and easily accessible, because expert knowledge has been encoded into the computer [6] – [8].

Advanced computing technology was developed, and this technology can help farmers make better decisions during crop growth. In addition, there is another technology that can solve this problem by continuously monitoring crops using video processing, cloud computing and robotics [9], [10]. Deep learning technology then comes in handy in this situation. Because it is more dependable, accurate, fast, and cost-effective than previous traditional methods. Many inventions have been developed to improve plant disease detection. Farmers should always monitor the growth of plants to maintain control and prevent the situation from worsening when it is not serious.

Therefore, this project aims to develop an Android application for farmers to detect plant leaves disease. In addition, the types of plants disease that can be detected in this project are divided into two categories the first category is vegetables which include tomato, potato, and bell pepper also corn, the second category is fruits, and includes apple, orange peach, and grape. Farmers do not have to pay fees or costs when conducting the detection of plant diseases. the farmers need only to open the application and go to the Disease detector page this page includes two sections camera or upload photo. The two methods send the photo that will be detected to the system, and the system will transmit the photos to the database where the machine learning model is located, after that, the model will back the photo results to the system and immediately display to the farmers. The app also gives information about the types of common diseases that users can view and read.

There are five sections in this article. Section 1 describes the project's background, while Section 2 discusses works that are related to it. The method is detailed in Section 3, and Section 4 provides a summary of the findings and discussion. Section 5 provides the conclusion.

2. Related Work

One of the most crucial industries for an economy's long-term viability is agriculture. It is one of the most significant generators of structural change and long-term economic growth [11]. However, it may differ by nation. The production of crops and food was the exclusive focus of agriculture in the previous ten years [12]. However, in more recent years, agriculture has extended to include the production, marketing, processing, and distribution of crops. Agricultural pursuits are regarded as the primary source of subsistence, and they boost GDP and the economy [13].

Today, plant diseases are not only a threat to global food security, but they also have disastrous effects on smallholder farmers whose livelihoods depend on their crops. In developing nations, smallholder farmers account for more than 80% of agricultural production [14]. reports of crops losing more than 50% of their production to common pests and diseases [15]. Estimated annual losses in fiber

and food production systems caused by plant diseases and pests are expected to amount to billions of dollars from an economic perspective [16]. Fungi are the cause of plant diseases. Other dangerous diseases in food and food crops, however, are carried by bacterial and viral species. Infectious plant diseases can exist [17]. This shows that this disease can spread from one plant to another, therefore it must be detected and treated as soon as possible.

Plants affected by this disease usually show clear signs or lesions on leaves, flowers, stems, or fruits. Generally, each disease or pest condition exhibits a unique observable pattern that can be used to uniquely diagnose the abnormality [17]. Nowadays, farmers still uses traditional farming methods, especially in developing countries, Due to the lack of suitable infrastructure and modern agricultural technology, Farmers have to deal with many problems including farmers struggling to take care of crops due to lack of sufficient experience, Coping with soil erosion, climate change and Especially the problem of plant diseases And they have no experience of the right way to take care of diseased plants, they use the traditional method to detect these plant diseases is done with eyes only. It is important to provide farmers with the best techniques and methodologies. Due to the ability of artificial intelligence to recognize problems, develop appropriate causes for these, and find optimal solutions for them. artificial intelligence can act as a great helper for farmers in dealing with plant diseases.

Deep learning technology, which falls under the umbrella of artificial intelligence, is a subset of machine learning that gives computers the ability to learn on their own and activate neurons in the brain to digest data and create patterns that can be used for decision-making. Deep learning allows the model to train directly to carry out the classification task from text, images, or sounds. Additionally, deep learning models can achieve excellent accuracy, occasionally outperforming human ability. Large collections of labelled data and a multi-layer neural network architecture are used to train deep learning models so they can automatically extract characteristics from the data [18]. Convolutional neural networks (CNN or ConvNets) will be employed as the deep learning algorithm in this project. It is one of the deep learning methods frequently used in object identification, picture categorization, face image recognition, etc [19]. On massive amounts of data, the algorithm performs well. This is because the neural network learns features that aid in classifying or recognizing things as the size of the training data increases. The primary layer of CNN is made up of several layers. Each layer has nodes based on input data that provide categorization outputs.

The development of the proposed system will be based on some studies on existing systems. **Table 1** gives the summary of comparison.

Table 1: System's Comparison

Features/System	Plant Disease Identification	Plants disease & their treatment	PLANTSCARE
User login and signup function	X	X	√
plant disease detection using CNN	√	X	√
Method uses for disease detection	CNN	Not mentioned	CNN
plant disease information	√	√	√
weather	X	X	√
contain advertisements	X	√	X
contain subscription fees	X	√	X

Table 2: System Development Workflow

Phase	Activity	Deliverables
Planning	Identifying the problem, scope, and objectives. And then find out how to build the app and its features based on the data collected	The proposal and Gantt chart
Analysis	Gather information and analyze them	UML diagram (Use case, activity diagram, sequence diagram) Class diagram and flowchart
Design	Using the appropriate programming language, create the overall system's interface	System architecture, database schema and data dictionaries, system user interface.
Implementation	Develop the system to include all the functions and features.	Code program and device setting. Test cases
Documentation	Discussion on what documentation should be created for future reference and maintenance.	Final report.

3. Methodology

This section explains the project's methodology as well as the results of the analysis and design phases.

3.1 Prototyping Method

The prototyping method has been employed in this project. **Table 2** gives the summary of system development workflow according to the prototype method and object-oriented approach of system analysis.

This project develops machine learning models with Teachable Machines. Teachable Machine is a platform created by Google, And Teachable Machine uses TensorFlow.js, a library for machine learning in JavaScript, to train and run supervised models. The artificial intelligence model that used in teachable machine is TensorFlow.js models. These models use a technique called Transfer-Learning Most of the neural network architecture. The google Teachable Machine relies on a pretrained image recognition network called MobileNet.

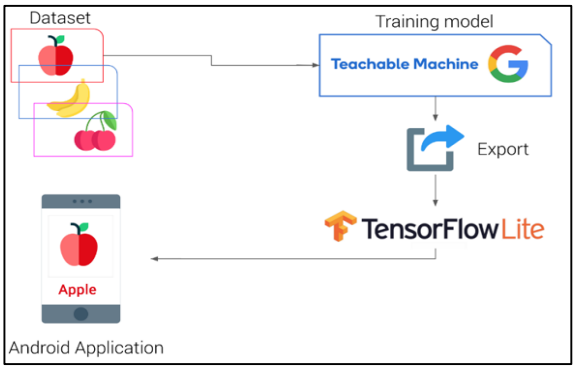

Figure 1: Machine Learning Model Development Environment

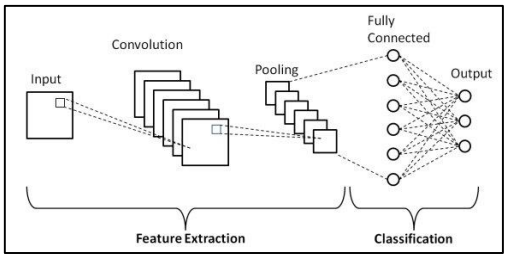

Figure 2: CNN architecture

Teachable Machine trains machine learning models using convolutional neural networks (CNN). An example of a deep learning algorithm that works particularly well for picture classification and object recognition applications is CNN. It functions by examining the patterns and features in the training data and learning to identify them in fresh photos. CNN modifies the weights and biases of its internal layers during training to maximize performance. After training is finished, CNN may be used to make predictions about new images by using the patterns and characteristics that were learned to categorize the objects in the images or to locate and identify the objects. The model will be transformed to TensorFlow Lite, a lightweight variant created especially for the mobile version.

Table 3: Modules of PLANTSCARE System

Modules	Function	User
Login and Registration Module	This module enables user registration and login	User Administrator
Common disease information	This module can display photos of common disease and provide information about common disease	User
Disease detector	The module offers the option for users to upload photos or use the camera so that the detector may use deep learning to recognize the image, diagnose the disease, and then display the results	User
Weather	The module should show the user the weather information	User
Update profile	This module is intended for user to update profile.	User
Administrative task	The module allows the administrator to verify user login, manage system's user and manage the posting made in the system.	Administrator

3.2 System Requirement Analysis

Requirement analysis is the process of determining the requirements a developed system must meet, as well as what users expect from the proposed system. Functional and non-functional requirements, user requirements, and system requirements are all part of the system requirements. The system's functional modules are summarized in **Table 3**.

The login and registration module allows application's login and account creation processes. Common disease information module enables the application's administrator to add or provide images of common diseases or information about common diseases. This material might cover the diseases' symptoms, underlying causes, and potential therapies. It allows viewers to see pictures and information.

A Disease Detector employs CNN deep learning techniques to recognize plant illnesses from photos. Users will be able to use the camera on their mobile to take a picture of a plant or choose one from the storage of their device. To evaluate the image and find any plant diseases, the system uses CNN's deep learning algorithms to recognize the image. Following that, the user will see the result faster.

The weather module informs users of the local weather conditions at any given time. Farmers can use this information to organize their activities and make knowledgeable choices regarding their crops and other agricultural endeavors. The Weather feature is intended to give farmers useful weather data

that can help them run their agricultural activities more successfully. Lastly, update profile module enables the user to edit their profiles and personal information.

Functional requirements determine the function of a developed system, while its function is described as a specific behavior that converts input into output. **Tables 4** and **5** show the functional and non -functional requirements for the proposed system.

Table 4: Functional requirements of the proposed system

Modules	Function	User
1. Login and Registration	• The system should allow the user to login into the system and create new account to use the system.	User Administrator
2. Common disease information	• The system should display photos and provide information about common disease.	User
3. Disease detector	• The system should allow users to use the camera. • The system should allow users to upload a photo. • The system should be able to detect the image using deep learning algorithm to identify the disease. • The system should be able to display the result.	User
4. Weather	• The system should display the weather to user.	User
5. Update profile	• The system should allow user to Update profile.	User
6. Administrative task	• The system should allow the administrator to verify user login, and manage posts	Administrator

Table 5: Non-functional requirements of the proposed system

Requirements	Description
1. Operational	• The system should be user friendly, easily maintained and updated. • The system should be able to work on Android operating system
2. Performance	• The system should be available 24 hours per day with a good Internet access. • The system should be able to handle a certain level of workload or volume of requests without degrading performance. • The system should be stable and not prone to crashing or other issues.
3. Security	• Users can only access their own account with user email and password. • The system should be secure against unauthorized access or data breaches. • The system should use encryption to protect data and user passwords
4. Usability	• The system should be easy for users to learn and understand. • The system should allow users to complete tasks quickly and without unnecessary steps or effort. • The system should be flexible and allow users to customize it to meet their specific needs. • The system should have a well-designed and intuitive user interface that is easy for users to navigate and use.

3.3 System Analysis

The results of the system analysis, which is a Unified Modelling Language (UML) diagram and class diagram, are defined in this sub-section. An object-oriented approach is used to perform system analysis. **Figure 3** demonstrates the use case diagram for this system. The two primary actors are the

administrator and the user (farmer). Use cases include sign-up, login, user list, disease detector, and information on the weather and diseases.

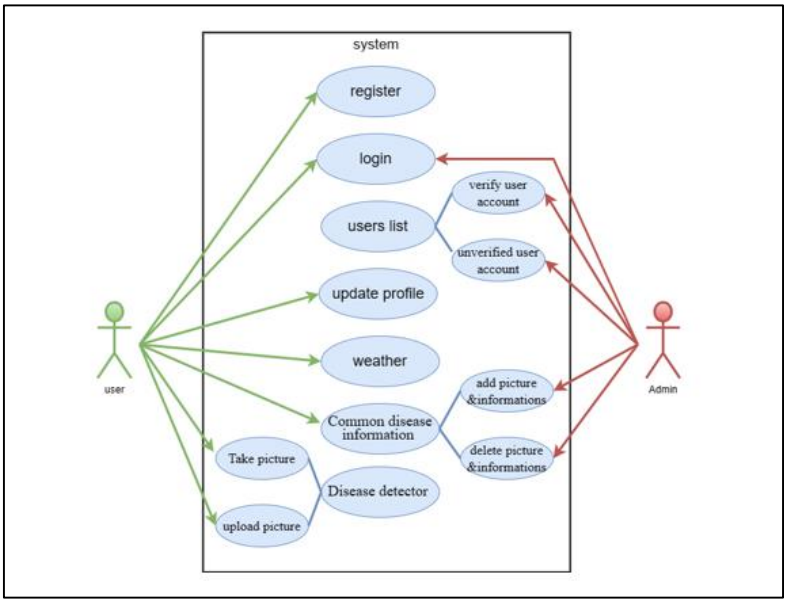

Figure 3: Use Case Diagram

Class diagrams are one type of diagram that illustrate the connections between classes and objects to depict a system's structure. **Figure 4** shows the class diagram. It represents the application's static view.

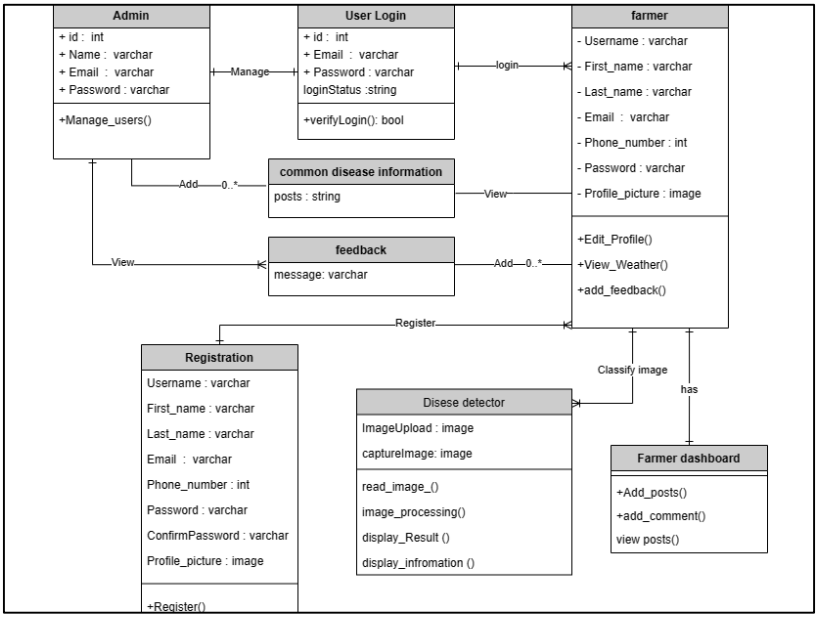

Figure 5: Class diagram

Flowchart is graphical representation to represent specific steps from the start of an algorithm to its end that are part of the system where it displayed the overall steps in a process. **Figure 6** illustrates the process flow for the user.

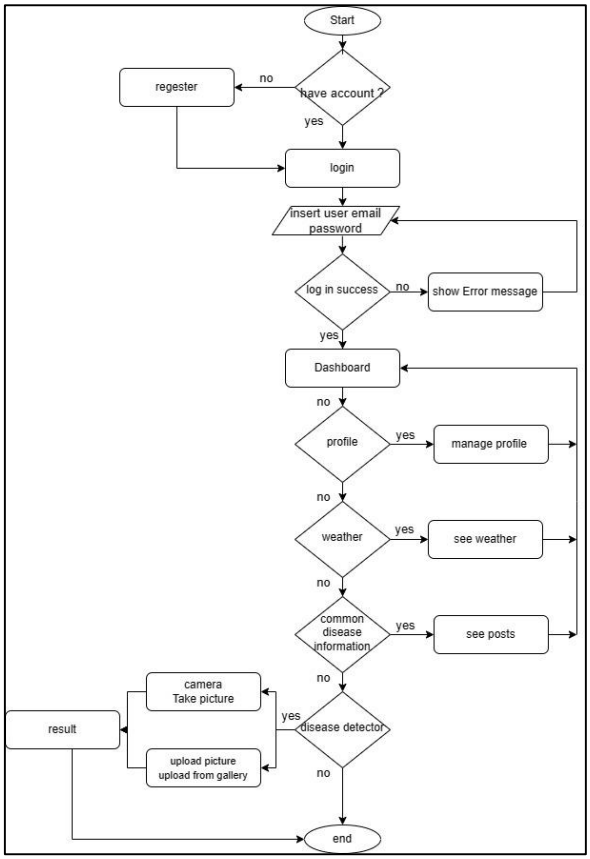

Figure 6: Flowchart

3.4 System Design

Systems design is the process of defining a system's components, including modules, architecture, and components, as well as their interfaces and data.

The architecture of the system involves a combination of frontend, backend, and database components, which work together to deliver the system functionality. The frontend communicates with the backend through APIs (application programming interfaces). The user interface of the system is referred to as the front-end. While the server-side components of the system, which conduct functions like data processing and storage and are hosted on a firebase server, are referred to as the backend. **Figure 7** shows the system architecture.

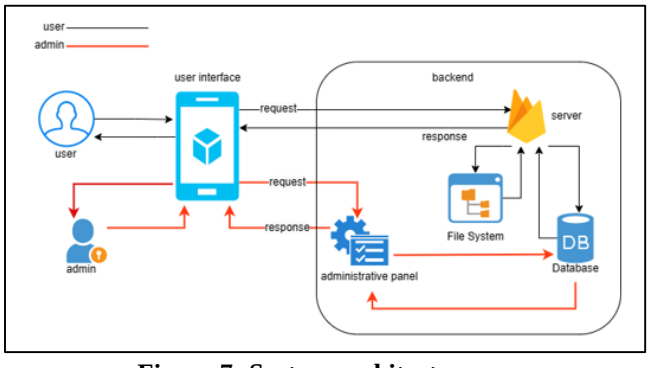

Figure 7: System architecture

Database schema are as follows:

i. `farmer(farmer_id,username,FristName,LasttName,email,phonenumber,password,)`
ii. `admin(id,username,FristName, LasttName, email, password,)`

iii. `Common_disease(Common_diseaseID ,ImageURL,post_info)`

The process through which designers produce software is known as user interface (UI) design. **Figures 8** to **11** demonstrate the design.

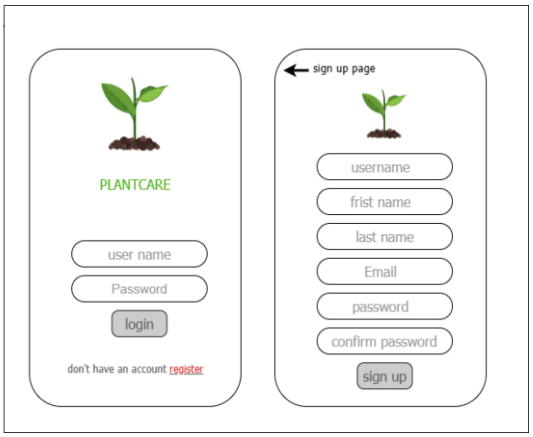

Figure 8: User login and sign up

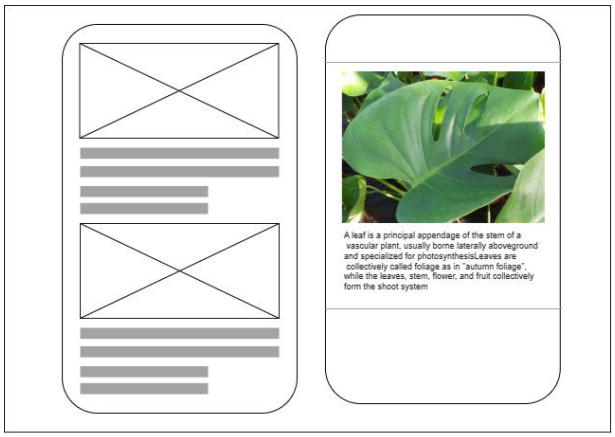

Figure 9: Common disease page

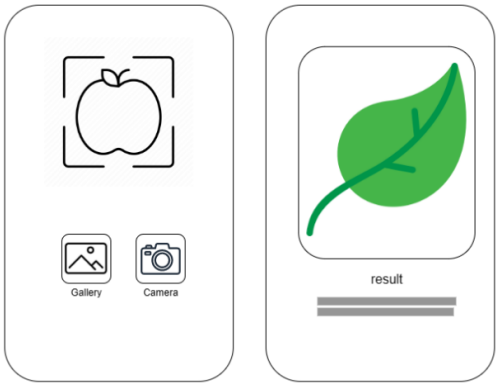

Figure 10: Disease detector

Figure 11: Weather page

The implementation of the system will be explained in the subsequent section.

4. Results and Discussion

The Implementation phase and the Testing phase of the system development process are covered in this chapter. The implementation makes use of Visual Studio Code, Android Studio, and the Dart programming language via the Flutter Framework. Additionally, the deep learning model is trained using Google Teachable Machine, while hosting and database services are provided by Firebase.

4.1 Functional Module Development.

The construction of system's functional modules is discussed in this section. The explanation is aided by a program code.

a. User login

The account login process is programmed using the dart programming language and the Flutter framework, as shown in **Figure 12**. First, the code is developed to accept email and password input through the application's user interface. These inputs are given to the Firebase authentication system to

be processed further and stored in the Firebase platform database. Using the proper authentication API offered by the Firebase platform, the code securely sends this data to the Firebase backend.

```
Future<void> checkDetails() async {
  if (_emailTextController.text.isEmpty ||
      _passwordTextController.text.isEmpty) {
    MessageShow(context, false, "Empty field!", 'All fields are required');
    // succsessMessage(    "succsess": Unknown word.
    //     context, false, 'Please fill in all field', 'Empty Feild!');    "Feild": Unknown w
  } else if (!EmailValidator.validate(_emailTextController.text)) {
    MessageShow(context, false, 'Invalid Email Adress!',    "Adress": Unknown word.
        'Please insert a valid email');
  } else if (!regex.hasMatch(_passwordTextController.text)) {
    MessageShow(context, false, 'Invalid Password!',
        'Please enter a valid password ');
  } else {
    try {
      await Authent()    "Authent": Unknown word.
          .loginWithEmailAndPassowrd(    "Passowrd": Unknown word.
              _emailTextController.text, _passwordTextController.text)
          .then((value) async => {_getdata()});    "getdata": Unknown word.
      print("This User is  $role");    Avoid `print` calls in production code.
      // Navigator.pushReplacement(
```

Figure 12: The Login Code Segment

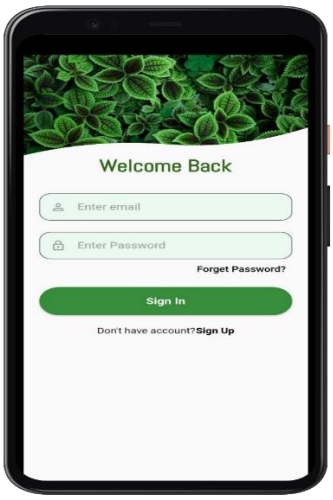

Figure 13: User Login Interface

The user interface for the application login page is next shown in **Figure 13**. In general, the user interface is made up of several essential elements, like an input box for the user's email address. The user's account is often identified by their email address. as well as a password entry field where the user can type their password. A secure masking of the password prevents unauthorized access. and it must contain both capital and small letters, numerals, and special characters. The user's identity is verified using it along with the email. and the Register button, which takes them to a page where they can register for an account.

b. Registration Module

The account registration procedure is coded using the Dart programming language and the Flutter framework, as shown in **Figure 14**. The source code is created to establish a new account for new users and pulls all the necessary data from the registration page, including the profile image, username, first and last names, phone number, email, and password. These values are recorded by the programme, which then stores them in code variables. Through Firebase's API, these variables, which contain the user-provided data, are sent to the database of Firebase to add a new user record.

```
Reference ref = FirebaseStorage.instance.ref().child(_image!.path);
UploadTask uploadTask = ref.putFile(_image!);    The value of the local variable 'uploadTask

imgUrl = await ref.getDownloadURL();

print(imgUrl);    Avoid `print` calls in production code.

await Authent()    "Authent": Unknown word.
    .createUserWithEmailAndPassword(
        imgUrl!,
        _emailTextController.text,
        _passwordTextController.text,
        _userNameTextController.text,
        _firstNameTextController.text,
        _lastNameTextController.text,
        _phoneNumberTextController.text)
    .then(
      (value) => Navigator.push(
        context,
        MaterialPageRoute(
          builder: (context) => const NavigationHomeScreen(),
        ), // MaterialPageRoute
      ),
    );
```

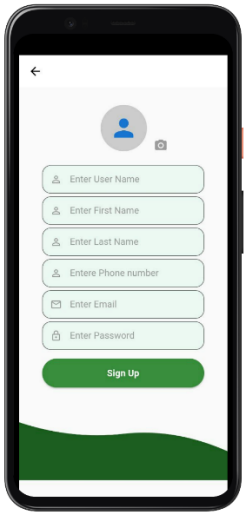

Figure 14: The Registration Code Segment

Figure 15: User Registration Interface

Figure 15 displays the application registration page's user interface. The user interface typically comprises of several essential elements, including the profile image, username, first and last names, phone number, email, and password. To properly create a new account, it is crucial that new users enter accurate information in all of the needed fields. Once they have completed this step and clicked the "sign up" button, the registration process will begin. Additionally, the system verifies the data entry to guarantee its integrity and completeness. The system securely stores the user's information in the firebase database following the data validation process.

c. User Dashboard

Based on **Figure 16**, the code is used for the user dashboard which serves as a community facilitating the sharing and comments of plant leaf photos and related information among users. The user dashboard page of the application is then depicted in **Figure 17**. Users can participate in a variety of activities on this website, such as submitting images and connecting with one another through comments. The "+" icon, which is prominently displayed, acts as a button that enables users to upload images with information about the plant, such as its name, species, and other specifics, and post them inside the app.

```
final Stream<QuerySnapshot> _usersStream =
    FirebaseFirestore.instance.collection("posts").snapshots();    "F
final Stream<QuerySnapshot> _comments =    The value of the field '_c
    FirebaseFirestore.instance.collection("comments").snapshots();
```

Figure 16: User Dashboard Code Segment

Figure 17: User dashboard-Interface

d. Weather Profile

The weather function in the code uses the JSON format for data interchange, as seen in **Figure 18** Using a weather JSON API to perform requests, receive responses via the API, and extract weather information. The weather JSON API obtains current weather conditions, forecasts, and other associated data for the area. This makes it possible for the system to incorporate real-time weather data. Weather interface is shown in **Figure 19**. This page gives users a visual representation of the current weather conditions and forecasts as well as a condensed display of those conditions. It includes features like weather conditions, data on temperature, humidity, wind speed and direction, gusts, pressure, and the time since the last update.

```
Weather.fromJason(Map<String, dynamic> json) {
  try {
    cityname = json['location']['name'];    "cityname": Unknown word.
    icon = json['current']['condition']['icon'];
    condition = json['current']['condition']['text'];
    temp = json['current']['temp_c'];
    wind = json['current']['wind_kph'];
    humidity = json['current']['humidity'];
    wind_dir = json['current']['wind_dir'];
    pressure = json['current']['pressure_mb'];
    pricipe = json['current']['precip_mm'];    "pricipe": Unknown word.
    last_update = json['current']['last_updated'];
    gust = json['current']['gust_kph'];
    uv = json['current']['uv'];

    date = json['dt'];
  } catch (e) {
    Text('asdas $e');    "asdas": Unknown word.
  }
}
```

Figure 18: Weather Code Segment

Figure 19: Weather Information-Interface

e. Disease detector

The Disease Detector Function Code Segment and the Disease Detector-Interface are shown in **Figures 20** and **21**, respectively. The model.flite formatting file, which is in charge of processing the photos, uses integrated deep learning, as demonstrated in the source code. The loadModel function loads the TFLite model. Additionally, the classifyImage function is employed to perform model inference on the downloaded or recorded image and save the results in the _outputs variable. Following selection, an image is shown in a container with the TFLite model's anticipated label. GetImageCamera and getImageGallery each offer two ways to take and post pictures. The /assets/ directory contains the files model.flite and plant_labels.txt. TFLite is also a dependency to the app/build.gradle file.

The plant disease detector's user interface can be seen in **Figure 21**. It offers two options for all users: the camera option, which enables real-time photo capture, and the gallery option, which enables selecting pre-existing photos from the device's gallery. Using either method, the image that is selected is sent to the deep learning model for processing based on the image condition, and the model responds as a result of the condition to which the interface is responding.

```
loadModel() async {
  await Tflite.loadModel(    "Tflite": Unknown word.
    model: "assets/model_unquant.tflite",    "unquant": Unknown word.
    labels: "assets/labels.txt",
    numThreads: 1,
  );
}

classifyImage(File image) async {
  var output = await Tflite.runModelOnImage(    "Tflite": Unknown word.
      path: image.path,
      imageMean: 0.0,
      imageStd: 255.0,
      numResults: 2,
      threshold: 0.2,
      asynch: true);    "asynch": Unknown word.
  setState(() {
    _loading = false;
    _outputs = output;
  });
}
```

Figure 20: Disease Detector Code Segment

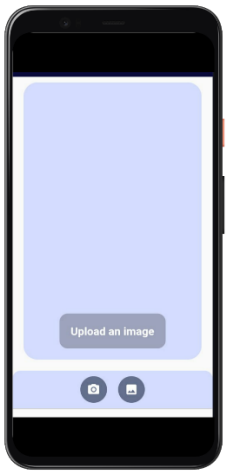

Figure 21: Disease Detector Interface

f. Common disease information

The common disease information code segment and the common disease information interface are seen in **Figures 22** and **23**, respectively. The source code depicts the Common disease information function

in accordance with **Figure 22**. The administrator is in charge of controlling and curating the content that is made available to users, and this code is used to force the administrator to develop content such as images and information by uploading photos and providing pertinent information about the plants. The data is taken and sent to the Firebase database when an administrator creates a new post. After the material has been stored in the database, the system retrieves it through the Firebase server and displays it to the user.

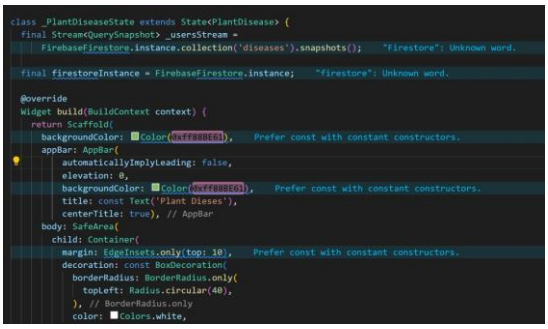

Figure 22: Common Disease Information Code Segment

Figure 23: Common Disease Information Interface

The common disease information interface is shown in **Figure 23**. This interface acts as a platform for the administrator to submit and distribute useful data regarding various plant diseases. Informational facts, such as descriptions, symptoms, precautions, and remedies for plant diseases, are included in the administrator's content that is disseminated. While all users of the application have access to this interface and can use it to gain knowledge, utilise common resources, and better understand and manage plant health.

g. Update profile

The update profile code segment is depicted in **Figure 24** and is implemented in the Flutter framework using the Dart programming language. With the use of the code, users will be able to change their username, last name, phone number, email address, and profile image. The code retrieves the updated data from the corresponding input fields when the user modifies the data via the application interface. The updated data is subsequently safely transferred through API to the Firebase database. The database contains the updated profile information.

The profile update page of the application's user interface is depicted in **Figure 25**. Users can change their profile information using the interface. The user interface often comprises of a few essential elements, such as the profile image, username, first and last names, phone number, and email. To guarantee a successful update, users should enter accurate and legitimate information in all essential fields. After the data has been entered and has been verified by the system as accurate and complete, it is securely transmitted and saved in the database.

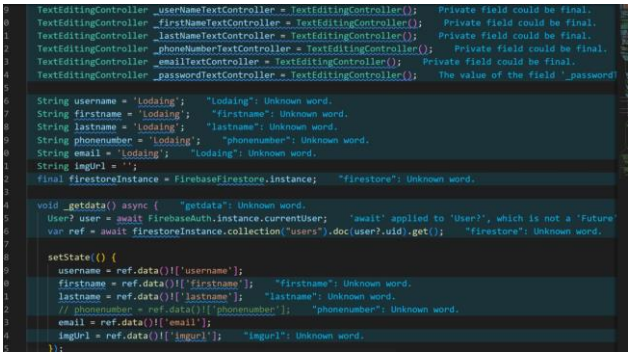

Figure 24: Update Profile Code Segment

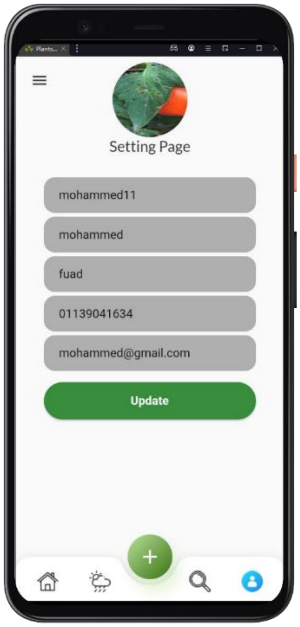

Figure 25: Update profile-interface

4.2 System Testing

Testing is a critical element in the software development life cycle. By finding unwanted errors and unforeseen difficulties, testing assists in avoiding these errors once the application has been published. In this section, a test will be carried out to assess the functionality of each module. A User Acceptance Test (UAT) method is utilized to perform testing.

a. Account Registration and Login

Table 6 show the test case for Account Registration and Login module. The purpose of this test is to verify whether the administrator and users are allowed to log in and Register into the system and whether the system will restrict login and Registration if an incorrect credential is entered.

Table 6: Test Case for Account Registration and Login Module

Module: Account Registration and Login				
Test Case ID	Description	Expected Result	Actual	Result
M1-1	already administrator has an account	The user should be able to create for an account	The user has successfully created for an account	Pass
M1-2	To check whether a administrator can login into the system	The user should be able to login into the system	The user has successfully logged into the system	Pass
M1-3	To check whether the system will restrict login whenever a wrong credential is entered	The system should restrict login when an incorrect credentials has been entered	The system restricted the login when an incorrect or no credentials has been entered	Pass

Table 7: Test Case for User Dashboard Module

Module: user dashboard module				
Test Case ID	Description	Expected Result	Actual	Result
M1-1	Post photo	The user should be able to take photos or upload from the device gallery	The photo is successfully uploaded and displayed in the user's dashboard	Pass
M1-2	View posted photos	The user should be able to see the photos in the user dashboard	The posted photos are displayed in the user's dashboard	Pass
M1-3	Comment on a photo	The user ought to be able to respond by leaving a comment on any image displayed on the dashboard.	The comment is successfully posted. The user's name and comment content are visible to other users.	Pass

b. User dashboard module

The test case for the user dashboard module is shown in **Table 7**. This test's objective is to determine whether users are permitted to post photos and respond with comments.

Table 8: Test Case for Weather Module

Module: Weather				
Test Case ID	Description	Expected Result	Actual	Result
M1-1	Get Current Weather	The user should be able to see the weather condition	The app displays the current weather conditions for the specified location	Pass
M1-2	Display Weather Icons for Different Types	The user should be able to see the appropriate weather icons	The app shows appropriate weather icons (e.g., sun, cloud, rain) based on the weather conditions for the specified location.	Pass
M1-3	Request location permissions	The user is prompted by the app to grant location access.	The app displays an appropriate error message indicating that the location is invalid or displays the weather of the user current location	Pass
M1-4	Network Connectivity	The app should be connecting with the internet network	The app displays an appropriate error message indicating the lack of network connectivity and prompts the user to check their internet connection.	pass

c. Weather module

Table 8 show the test case for the Weather module. The purpose of this test is to verify whether the users are allowed to see current weather conditions and temperature, humidity, wind speed and direction, gust, pressure, and the last update time.

d. Disease detector module

The test case for the disease detection module is shown in **Table 9**. The goal of this test is to confirm the deep learning model's ability to recognise and classify plant diseases using uploaded or captured leaf photos.

Table 9: Test Case for Disease Detector Module

Module: Disease Detector				
Test Case ID	Description	Expected Result	Actual	Result
M1-1	Detect Disease from Captured Photo	The user is able to take photos from the camera	The app processes the captured photo using deep learning algorithms and identifies the type of disease affecting the plant leaf,	Pass
M1-2	Detect Disease from Uploaded Photo	The user is able to upload photos from the device gallery	The app processes the uploaded photo using deep learning algorithms and identifies the type of disease affecting the plant leaf,	Pass
M1-3	Display Disease Detection Results	the user can see the plant Disease Results	The app presents the disease detection results, including the identified disease type.	Pass
M1-4	Provide Additional Resources or Information	the user can see the Additional Resources or Information	The app offers resources or information such as articles and recommended actions	pass

e. Common disease information module

The Common Disease Information module's test case is shown in **Table 10**. The goal of this test is to make that the module works properly and that the administrator can upload images with information while allowing users to access and view the posts.

Table 10: Test Case for Common Disease Information Module

Module: Common disease information				
Test Case ID	Description	Expected Result	Actual	Result
M1-1	Add Common Disease Photo and Information	The administrator able to add the photos and information	The admin successfully adds the photo and information about the common plant disease,	Pass
M1-2	View Common Disease Photos and Information	the users are able to see the uploaded photo that uploaded by the administrator	The user can see the uploaded photo of the plant leaf affected by the common disease and read the details provided about the disease	Pass

f. Update profile module

The update profile module's test case is shown in **Table 11**. This test's objective is to confirm that it is functional for users to update their profile information. It is the goal of the tests to make sure that every user may correctly update and save changes to their profile information.

Table 11: Test Case for Update profile mod.

Module: Update Profile				
Test Case ID	Description	Expected Result	Actual	Result
M1-1	Update Profile Information	The users can update their profile information	The system successfully updates the user's profile information and saves the changes in the database.	Pass
M1-2	Display Error Message	The user sees error message for duplicate Data	The system displays an error message indicating that the entered data is the same as the existing data	Pass
M1-3	Validate Input	The users should input valid data	The system alerts the user with an error message indicating that the input provided is invalid if the user input invalid Data	Pass
M1-4	Cancel Action	The users able to cancel the update profile action	The system cancels the update action and does not save the changes made to the user's profile, reverting to the previously saved information.	pass
M1-5	Update Profile Picture	The users able to change the profile picture	The system updates the user's profile picture with the newly selected image and saves the changes, displaying the updated picture in the profile.	pass

5. Conclusion

Plantscare is an Android app that uses deep learning technology to help farmers identify plant leaf diseases. The app is easy to use and can be accessed from anywhere, making it a valuable tool for farmers of all sizes. Plantscare can identify a variety of plant diseases, including those that affect vegetables and fruits. The app also provides information on the symptoms, causes, and treatments for each disease. This information can help farmers to take steps to prevent and control diseases, which can lead to improved crop health and yield. In addition to identifying diseases, Plantscare also provides farmers with access to expert knowledge and information. The app includes a library of articles, videos, and other resources that can help farmers to learn more about plant health and production. This information can help farmers to make better decisions about the management of their crops. Plantscare is a valuable tool for improving crop health and yield. The app is free to use and has no associated costs. This makes it an accessible and affordable option for farmers of all sizes.

Acknowledgment

The authors would like to thank the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia for its support.

References

[1] Fernando, W. D. (2012). Plants: An international scientific open access journal to publish all facets of plants, their functions and interactions with the environment and other living organisms. Plants, 1(1), 1.

[2] Almusaed, A. (2011). Plants, Oxygen and Human Life Benefits. In Biophilic and Bioclimatic Architecture (pp. 159-165). Springer, London.

[3] Zhang, J., Huang, Y., Pu, R., Gonzalez-Moreno, P., Yuan, L., Wu, K., & Huang, W. (2019). Monitoring plant diseases and pests through remote sensing technology: A review. Computers and Electronics in Agriculture, 165, 104943.

[4] Sligo, F. X., & Massey, C. (2007). Risk, trust and knowledge networks in farmers' learning. Journal of Rural Studies, 23(2), 170-182.

[5] Šūmane, S., Kunda, I., Knickel, K., Strauss, A., Tisenkopfs, T., des Ios Rios, I., ... & Ashkenazy, A. (2018). Local and farmers' knowledge matters! How integrating informal and formal knowledge enhances sustainable and resilient agriculture. Journal of Rural Studies, 59, 232-241.

[6] Abu-Naser, S. S., Kashkash, K. A., & Fayyad, M. (2010). Developing an expert system for plant disease diagnosis. Journal of Artificial Intelligence, 3(4), 269-276.

[7] Yelapure, S. J., & Kulkarni, R. V. (2012). Literature review on expert system in agriculture. International Journal of Computer Science and Information Technologies, 3(5), 5086-5089. Zhang, J., Huang, Y., Pu, R., Gonzalez-Moreno

[8] Jabbar, H. K., & Khan, R. Z. (2016, March). Survey on Development of Expert System from 2010 to 2015. In Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies (pp. 1-7).

[9] Rupanagudi, S. R., Ranjani, B. S., Nagaraj, P., Bhat, V. G., & Thippeswamy, G. (2015, January). A novel cloud computing based smart farming system for early detection of borer insects in tomatoes. In 2015 international conference on communication, information & computing technology (ICCICT) (pp. 1-6). IEEE.

[10] Mekala, M. S., & Viswanathan, P. (2017, August). A Survey: Smart agriculture IoT with cloud computing. In 2017 international conference on microelectronic devices, circuits and systems (ICMDCS) (pp. 1-7). IEEE.

[11] Eli-Chukwu, N. C. (2019). Applications of artificial intelligence in agriculture: A review. Engineering, Technology & Applied Science Research, 9(4), 4377-4383.

[12] Fan, M., Shen, J., Yuan, L., Jiang, R., Chen, X., Davies, W. J., & Zhang, F. (2012). Improving crop productivity and resource use efficiency to ensure food security and environmental quality in China. Journal of experimental botany, 63(1), 13-24.

[13] Omodero, C. O. (2021). Sustainable agriculture, food production and poverty lessening in nigeria. transport, 3(6).

[14] Sabo, B. B., Isah, S. D., Chamo, A. M., & Rabiu, M. A. (2017). Role of smallholder farmers in Nigeria's food security. Scholarly Journal of Agricultural Science, 7(1), 1-5.

[15] Harvey, C. A., Rakotobe, Z. L., Rao, N. S., Dave, R., Razafimahatratra, H., Rabarijohn, R. H., ... & MacKinnon, J. L. (2014). Extreme vulnerability of smallholder farmers to agricultural risks and climate change in Madagascar. Philosophical Transactions of the Royal Society B: Biological Sciences, 369(1639), 20130089.

[16] Mustafa, M. S., Husin, Z., Tan, W. K., Mavi, M. F., & Farook, R. S. M. (2020). Development of automated hybrid intelligent system for herbs plant classification and early herbs plant disease detection. Neural Computing and Applications, 32(15), 11419-11441.

[17] Isleib, J., & Michigan State University. (2018, October 2). Signs and symptoms of plant disease: Is if fungal, viral or bacterial? Retrieved from https://www.canr.msu.edu/news/signs_and_symptoms_of_plant_disease_is_it_fungal_viral_or_bacterial.

[18] Aggarwal, C. C. (2018). Neural networks and deep learning. Springer, 10, 978-3.

[19] Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017, August). Understanding of a convolutional neural network. In 2017 international conference on engineering and technology (ICET) (pp. 1-6). IEEE.