FARM MANAGEMENT INFORMATION SYSTEM (CROP PLANNING AND TRACKING MODULES)

BY

LENG KAI YI

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONOURS)

Faculty of Information and Communication Technology

(Kampar Campus)

JAN 2023

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: Farm Management Information System (Crop Planning and Tracking modules)

Academic Session: Jan 2023

I __ LENG KAI YI _____

(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

_____

(Author's signature)

_____

(Supervisor's signature)

Address:
61, Lorong Juru 1,
Taman Juru
14000 Bukit Mertajam,
Pulau Pinang

Ts Tan Teik Boon

Supervisor's name

Date: 24/4/2023

Date: 27/4/2023

FACULTY/INSTITUTE* OF INFORMATION AND COMMUNICATION TECHNOLOGY

UNIVERSITI TUNKU ABDUL RAHMAN

Date: 24/4/2023

SUBMISSION OF FINAL YEAR PROJECT /DISSERTATION/THESIS

It is hereby certified that _____***Leng Kai Yi***_____ (ID No: __***1902845***) has completed this final year project/ dissertation/ thesis* entitled "*Farm Management Information System (Crop Planning and Tracking modules)*" under the supervision of __Ts Tan Teik Boon__ (Supervisor) from the Department of ___Computer Science_____, Faculty/Institute* of __Information and Communication_____ , and _____ (Co-Supervisor)* from the Department of _____, Faculty/Institute* of _____.

I understand that University will upload softcopy of my final year project / dissertation/ thesis* in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,

_____

(*Leng Kai Yi*)

DECLARATION OF ORIGINALITY

I declare that this report entitled "**FARM MANAGEMENT INFORMATION SYSTEM (CROP PLANNING AND TRACKING MODULES)**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : 

Name : Leng Kai Yi

Date : 24/4/2023

ACKNOWLEDGEMENTS

I would like to express my sincere thanks and appreciation to my supervisor, Ts Tan Teik Boon who has given me this bright opportunity to engage in the mobile-based farm management system and provide guidance throughout the project. The guidance helps me to have a better understanding of the project. Finally, I must say thanks to my parents and my family for their love, support and continuous encouragement throughout the course.

ABSTRACT

The purpose of the Farm Management System is to automate the existing manual system with the assistance of computerized equipment and full-fledged computer software. Therefore, by using farm management system, it will help users to optimize and manage farm activities and production activities. This project will develop a mobile based farm management application which allows user to plan, monitor and analyse all activities on the farm easily. It allows important data and information to be stored for a longer period of time with easy access. This project will focus on helping user to plan farm activities efficiently. It is important for farmer to have a better planning on the farmland because farm plan can maximize the resource use efficiency at the farm. The development tool that is used is Flutter with Dart language. Besides, the hardware will be used for the project is laptop. The final product of the project is the combination of using Google Maps, application programming interface (API), Firebase and an Android mobile application with the ability to plan and monitor farm activities and detect pests and diseases.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1
Introduction

Farm management is the process of making and implementing various types of strategies and methods that are aimed to maintain a farm's level of productivity and financial profitability, as well as ensure the farm remains organized. Farm management practices are essential because they can build or break a farm. The success of any farm management approach depends on the features of the specific farm since farms have varying differences [9]. Traditionally, farmers used to pay less attention to their farms and were just concerned with planting whatever seeds and crops they came across without worrying about the chances of successful farming. Daily changes in weather and climatic conditions, as well as condition changes in soil and air characteristics that might affect productivity levels of crops, seeds, soils, and farming techniques. However, farmers use their primitive knowledge and experience to adapt to environmental changes by altering their agricultural practices and growing crops that are suitable for the current environment. Nowadays, by using farm management system, farmers can be more deliberate and productive in their daily farm roles and tasks. Farm management system usually has the functionality to enable efficient planning and monitoring of all farm activities. Therefore, the proposed system is mobile-based farm management application that helps farmers perform their farm activities efficiently, such as planning and monitoring farm activities, as well as detecting pests and diseases.

1.1 Problem Statement and Motivation

Problem 1: Unable to plan farm activities efficiently.

The farm owner is not able to coordinate all the activities involved in farming, such as watering, placing fertilizer, spraying insecticide, trimming, wrapping fruits, and others. Decision-making is the major activity of management. Early in the cropping season, farmers have to decide what crops to plant, how much to plant, and fertilizer levels to use. Without efficient planning, farmers may not be able to make informed decisions about the crops. Furthermore, the farm owner is not able to determine whether the worker has done the tasks given. Supervising farm workers are not an all-the-time job. Without supervision, lazy farm workers may put a lot of effort into avoiding work, and

they may just pretend to complete the tasks given. This will result in failing to achieve long and short-term goals. Therefore, we expect to develop a farm management system that can help farmers to plan and monitor farm activities efficiently and ensure the task given to the worker must be completed.

Problem 2: No tracking of plant growth.

The farm owner cannot know how much produce of outcomes can be generated within a specific time because there is no tracking of plant growth from the beginning. In addition, the farmer is not able to know specifically the stages of harvesting for his/her supply to the customer. Without tracking of plant growth, farmers may miss out on opportunities to optimize their production process. They may not know when to apply fertilizers, pesticides, and other treatments, leading to poor plant health and reduced yields. Moreover, the lack of tracking makes it difficult to identify issues early on, such as plant diseases or pest infestations, which can cause significant crop losses if not addressed promptly. Therefore, having plant growth tracking features is essential to allow farm owners to have a comprehensive overview of their farm and optimize their production process for better yields.

Problem 3: No contingency plan for each plant.

There is no contingency plan when some harms occur to the farm or crops. For example, disease, critical pest attacks, extreme weather problems, and others may occur unexpectedly. Without a contingency plan, it is difficult to determine how a farm will handle a crisis or disaster and how a farm will recover quickly from a critical event to resume normal operations because the farm owner will solve the problems at the last minute. A contingency plan enables farmers to take proper action on time and reduce crop loss. Therefore, we expect to develop a farm management system that can detect pests and diseases that appear on the crops at different stages and helps farmers to plan out treatment for the diseases in terms of chemical control.

1.2 Research Objectives

The first objective of this project is **to develop crop plans in terms of fertilizer**. The system should let users plan and control farm activities. The planning part lets users plan the farm activities that need to be performed. For example, watering, trimming, wrapping, placing the fertilizer, and spraying pesticides. In addition, the system is able

to control and monitor the activities performed by the farm worker. Users can view the progress of the farm activities by the farmer. The second objective of this project is **to monitor the growing process of crops**. The system should be capable of tracking the supply chain of crops from the beginning until the harvesting period. Stages of the supply chain for a particular crop must be created for monitoring, such as for mango or durian. The system allows users to record data such as the date, quantity of crop, height, and take photos at different growing stages. This feature provides valuable insights to farmers, allowing them to monitor and understand the condition and quantity of crops in growing seasons based on the crop's growing stages. The third objective of this project is **to detect crop diseases and provide treatments**. The system is able to look for early detection of problems related to diseases, and pest problems. Therefore, the system can detect pests and diseases that appear on the crops at different stages and helps farmers to plan out treatment for the diseases in terms of biological control, chemical control or organic control and prevention tips for mitigating the risks identified.

1.3 Project Scope and Direction

The final deliverable at the end of the project will be a mobile-based farm management application that can help farmers plan, monitor, and analyse all activities on the farm easily. To create a new field in the system, the user would first need to plot a polygon shape on Google Maps that corresponds to the boundaries of the field. This can be done by using the drawing tools provided by Google Maps, such as the polygon tool. Once the polygon is drawn, the users would then need to enter the required field information, such as field name, crop type and plant date. Once the user has successfully created the field details and saved them, the system will retrieve and display field details and relevant data such as weather, temperature in Celsius, wind speed and humidity using an API. This information can be crucial for farmers to plan and schedule farm activities such as planting, fertilization, and harvesting. The system allows users to plan their daily farm activities such as watering, trimming, wrapping, placing the fertilizer, and spraying pesticides. Users can set specific dates and times for each task, which can assign farm jobs to his/herself or to another user. The jobs can be viewed by users in a list or in a calendar view. The system automatically sorts the upcoming jobs by date to help users prioritize their tasks and ensure that they are completed on time. Users are

able to track and monitor all activities and tasks performed by the farm worker to prevent them from slacking off at work.

Moreover, the system allows users to monitor the growing stage for the crops. Different crop has different growing stages. Users need to record data such as the date, quantity of crop, height, and take photos at different growing stages. The ability to take photos at different stages of growth can also help farmers to compare the progress of their crops over time and make informed decisions about when to apply fertilizers, pesticides, or other treatments. Additionally, users can update and delete the data that has been stored in the database. For example, if a farmer mistakenly records incorrect data, they can delete the entry and re-enter the correct information. Being able to update or delete data provides farmers with a level of flexibility in their record-keeping.

Besides that, the system provides users with the option to either choose a photo from their gallery or take a photo of their crops. Once the photo is uploaded, the system uses API to detect any diseases present on the crop and provides suitable treatment options in terms of biological control, chemical control, or organic control as well as some prevention tips. Additionally, the system is also able to detect any pests that are affecting the crop and displays information on the pest. Overall, this feature helps farmers take proactive steps to protect their crops from disease and pests.

1.4 Contributions

The project proposes a mobile-based farm management application that assists farmers to have a better farm planning and monitoring their farm activities easily and efficiently. This project focuses five crops which are banana, mango, durian, palm oil and papaya. The application allows farmers to plot their fields on Google Maps and fill in the relevant field details. he system enables users to create farm activities such as watering, trimming, wrapping, placing the fertilizer, and spraying pesticides in the fields. Users can assign these activities to themselves or assign them to other people, ensuring that the farm activities are carried out in a timely and efficient manner. Besides, the application allows users to monitor the growing process of their crops. Users can add, update, or delete growing details of the crops in different stages, such as the date, quantity of crop, height, and photos. This feature provides farmers with valuable insights into their crops' progress, allowing them to make informed decisions about the optimal time for harvest and apply treatments such as fertilizers and pesticides at the right time. Moreover, the application allows users to upload photos of their crops, and

the system uses an API to detect diseases and pests. The system can identify diseases and provide recommended treatments, as well as display information about pests that may be affecting the crops.

1.5 Report Organization

The seven chapters of the project are outlined. The first chapter provides an introduction to the project and presents the problem statement, objectives, project scope, and contribution of the system. The second chapter presents a literature review of five existing farm management systems, providing an overview and comparison between the systems. The third chapter discuss the system methodology and approach, including the architecture diagram, use case diagram and description, activity diagram, and database design. The fourth chapter explains the system design, which includes the system block design and UI diagram using wireframe. The fifth chapter covers the system implementation and includes the software setup process, including Firebase and Android Studio, as well as the system operation with module. The sixth chapter presents the system evaluation and testing results, providing explanations and screenshots of the output. Finally, the seventh chapter concludes the report and provides recommendations for future enhancements.

Chapter 2

Literature Review

In this information-intensive digital era, smartphone applications provide users with sophisticated task-specific services. Farm management system provide better access to farmer advisory services directly on-farm, facilitate large-scale information sharing and help farmers make better decision in farm operations.

2.1 System Review on Farm Management System

In this section, five existing farm management applications will be critically reviewed.

2.1.1 Agrivi

Agrivi farm management application gives a deep insight into farm activities and guides farmers to improve crop production and increase fertility [1]. New users should set up their farms using the map to gain insight into their farming. It helps users in crop planning, monitor the precise field and crop conditions and analyse all farm activities easily based on weather, soil performance, and overall crop condition. Recommendations on how to plant different crops and varieties more suitable for their location will be provided to those farmers who are not productive. To improve farm production efficiency, Agrivi provides a set of best-growing practices for each crop and gives detailed instructions on how to do each task, such as when it is best to perform specific practices, which products should be used, and how much they should use. Besides, Agrivi helped farmers make accurate agricultural decisions based on real-time field insights, including real-time satellite field insights, weather forecasts, risk alerts and crop progress monitoring. This feature helps to inform users about upcoming yield threats and take preventative measures to avoid any potential yield loss. It provides weather monitoring with a detailed seven-day weather forecast or three-year weather history for each field. Automated pest and disease detection uses advanced detection algorithms which alert users to remind and protect crops timely when pests and diseases are at risk of invading their fields. Furthermore, advice and relevant information on cultivation, planting, crop protection, fertilizing, harvesting, and all other farm activities can easily access and manage by the users. For example, users can record the

usage of silicon fertilizers. Agrivi allows users to track fertilizer rate used for every crop production, fields, and cost. Thus, it can help to increase yield and maintain crop productivity. Meanwhile, users allow storing farm records and reports such as sales and expense tracking in one place. Thus, users can access it anytime and anywhere. Built-in reports allow users to retrieve crucial information in PDF, Excel or Word formats. [2, 8].

2.1.2 Granular

Another review of the existing system is Granular. Granular is a farm business management software that helps users increase farm productivity and maximize revenue. Granular allows users to construct farm plans for each field and predict the most suitable water requirements and environment effects on certain crops. It is able to analyse massive amounts of data to calculate the estimated number of plant population in each field. Real-time analytics reports will be generated for users which can help them to make the right decisions at the right times. Granular provides knowledge and skills that help users to be better prepared for crop failure or unexpected changes in the farms mid-season. Besides, Granular offers high-resolution (3 meters), high-frequency satellite imagery to facilitate the relationship between growers and agronomists [3]. It can monitor entire operations such as crop health anytime from anywhere and easily identify potential at-risk fields. For example, identify crop problems like pests, diseases and nutrient deficiencies and take corrective action in a timely manner. Scheduling and job assignment for farm workers can be done by farm administrators and for farm operators to receive and complete the tasks. Farm operators are able to acquire turn-by-turn direction to the field after receiving the tasks and allows them to add notes and images to the work order and record the amounts of products they have used. Meanwhile, farm administrators are able to view the notes, images and updates from their teams in the field in real-time. Furthermore, Granular provides a complete crop history for the past five years, along with crop rotation information for specific field. The historical data assists users in deciding which crops to plant and the profit that they can expect to receive. Users allow to manage and view all upcoming farming tasks and schedules using smartphones. Granular can manages reports on soil and land data and always keep up to date when changes occur. Not only that, it allows users to create custom professional yield and profitability reports automatically [4]. As an enterprise

farm management software, Granular supports multi-location management. Farm managers and owners may keep control over a number of scattered farms in various places through one window [7].

2.1.3 Trimble

Trimble software allows users to plan their crops including managing equipment, materials, people and setting up farms and fields. Besides, it also provides users with tools to help them make better management decisions on most farm equipment, independent of the manufacturer, and covers all seasons, crops, terrains, and farm sizes. For example, designed to increase the efficiency of planting and seeding processes, eliminate wasted inputs, and increase farm revenues. Planting seeds in the right place and the correct depth is important for a yield succeed and allows users to build planting maps based on data and then deliver and utilize them accurately in the field. Furthermore, it can assign and track field work for farm administrators and farm workers can receive daily tasks through their smartphones. A set of instructions for completing in-field tasks will be provided to the users. Work orders include materials, tools, operators, weather constraints, target completion and due date. Users can update their task status and data to Trimble and farm administrators allow to monitor farm workers' progress and completion of each task [5]. Besides, Trimble also provides variable rate application control feature which controls amount used of different materials such as crop seed, fertilizer and nutrition. Yield data and crop analysis in real time can be view by users. Field record keeping is one of the major features. Users can enter and manage records and activities in real time. It can keep track of different seed types, tank mixes, pesticides and fertilizer usage for specific state or provincial reporting. Next, real-time fleet tracking and utilization. Users can view the location and status of each vehicle on the road maps. When navigating in remote areas, it allows users to map the locations of landmarks such as irrigation or drainage and view them for reference. Making use of turn-by-turn navigation can locate vehicles from users' current location [8].

2.1.4 FarmLogs

FarmLogs is a comprehensive software program that allows farmers to manage all aspects of their operations. It allows the farmer to record planting dates, watering schedules and crop yield on any mobile device and share it easily via the web. Farm administrators can assign work and monitor the status of assigned work of the farm worker while the farm worker can easily view their list of work orders and update their work status. FarmLogs has a crop health monitoring feature that utilizes multi-spectral satellite imagery to establish performance baselines with the last five years of crop health data of each field. Crop health is monitored against a performance baseline to detect subtle cues and stress signals. FarmLogs' crop health monitoring imagery shows actionable information and reduces the guesswork of farmers so that they can solve problems before yield is reduced. If detected crop health anomalies, FarmLogs will notify farmers by pinpointing the exact location of the field. Not only that, it will guide farmers to the location that needs to be monitored [11]. Besides, FarmLogs provides a soil composition map that allows farmers to view the soil type of the field. Color-coded maps easy to read which can assist farmers in making better seed and nutrient decisions according to the soil types. Moreover, it also provides interactive maps to assess the results of seasonal trials and get a big-picture view of yield performance. FarmLogs also provides growth state analysis, which enables farmers to keep track of the growth stage of each plant remotely. Whether crops are still in the early growth stage or nearing maturity, the system assists farmers schedule their fieldwork efficiently by continuosly updating growth stage and estimate maturity of the plant. Additionally, FarmLogs has field rainfall monitoring features. It has the ability to measure rain accumulation at each field and determines how much rain has accumulated by matching the exact location of fields with advanced data on an hourly basis which can help farmers to save a lot of time. It will send timely rainfall notifications to the farmer to follow rainfall events. Thus, farmers can make a better decision on their farm activities by remotely tracking because they know their field conditions without the need to physically check or monitor rain gauges [10].

2.1.5 Agworld

Agworld enables farmers, crop consultants, farm staff, precision specialists and operation managers to work together as one on a single platform. The purpose is to give users as much information as possible so that they can make better decisions. Agworld allows recording all the important events happening on the field including data records, observations, machinery and applications. For example, seeding records. Farmers can easily record seedling rates, dates, locations and machinery setup information for the fields. Not only that, rainfall data and harvest data also can be recorded in the system. The harvest data is used to round out the season and show the cost of production and agronomic performance as well as the data is reportable year on year to help farmers benchmark their seasons. Additionally, Agworld has a field scouting feature. Farmers are able to observe their fields throughout every state of the growing season such as tracking soil moisture and varietal characteristics among other things. When the crops are planted, the farmer can snap some pictures and record the emergence results. This allows other users to view all the data remotely which can increase the efficiency of every operation. Farmers can customize scouting templates, collect infield data and generate reports according to the farm scouting activities. Furthermore, Agworld provides a task scheduling feature. Farmers can plan their tasks themselves or insert the recommendations provided by the agronomists and field scouts, and then it will assign to the appropriate machines and applicators. When daily changes in the weather or machines break down, the farmer can reschedule jobs, change products, rates and update the latest information in the system. In order to complete the most important tasks, Agworld allows farmers to set up routes and tasks and display them based on the priority of numbers with different colours. This helps to draw farmers' attention and show specific processes. Not only that, Agworld provides real-time captured data for reporting. There are several types of reports can be generated such as production plan and budget reports, farm performance reports, farm worker reports or customized own reports. Reports help farmers to make better decisions on planning the following farm activities [12].

Features	Agrivi	Granular	Trimble	FarmLogs	Agworld
Crop planning	✓	✓	✓	✓	✓
Crop / field monitoring	✓	✓	✓	✓	✓
Weather monitoring	✓	✓	✓	✓	✓
Task scheduling	✓	✓	✓	✓	✓
Predicting crop yield	×	✓	×	×	×
Pest / disease detection	✓	✓	×	×	×
Provide growth state analysis	×	×	×	✓	×
Alerts/ Notifications	✓	✓	✓	✓	✓
Report	✓	✓	✓	✓	✓

Table 2.1: A summary of features of existing system

2.2 Limitation of existing system

In this subsection, the disadvantages and weakness of each reviewed system will be discussed. A summary of the limitation of each reviewed system is shown in table 2.2.1.

Existing system	Limitation
Agrivi	• Does not provide preparation of appropriate recommendations • Unable to predict crop yield • Unable to track the growth state of each plant
Granular	• Require a longer time to set up each field and enter the data • Unable to track the growth state of each plant
Trimble	• Unable to predict crop yield • No pest or disease detection feature • Unable to track the growth state of each plant
FarmLogs	• Unable to predict crop yield • No pest or disease detection feature
Agworld	• Majority of the crop types are not suitable to plant in Malaysia • Unable to predict crop yield • No pest or disease detection feature • Unable to track the growth state of each plant

Table 2.2: A summary of limitations of existing system

For Agrivi system, one of the limitations is it does not provide preparation of appropriate recommendations. Although the system allows farmers to plan their farm activities easily, it does not support farmers with optimal strategic decision-making in the crop planning system. Besides, the main limitation of Granular system is it requires a longer time for a user to set up each field and enter the data. For example, exploring fields as a rough draft planning process and entering different inputs and crops can be difficult. Not only that, it needs a significant amount of time needs to be spent in the Granular system in order to keep the information accurate and up to date. Therefore, new users are hard to adapt to the system. Next, for the Agworld system, the main limitation is it supports the majority of the crop types that are not suitable in Malaysia. For example, apples, oranges and grapes are not suitable to plant in Malaysia because of the weather. Based on the above five reviewed existing systems, apart from Granular system, the four other systems do not provide crop yield prediction features. Farmers may apply their old-fashioned techniques of farming which predict crop yield based on their experience. Moreover, Trimble, FarmLogs and Agworld systems do not provide pest or disease detection. The quality and quantity of a crop may decrease when the crop is attacked by different insect pests or diseases. Therefore, most of the farmers rely on traditional techniques of pest management which is the regular spray program based on schedules rather than the presence of insect pests on the fields. Furthermore, apart from FarmLogs system, the four other systems are unable to track the growth state of each plant. They cannot provide farmers with a quick visual estimate of when a crop is likely to mature and the crop growth status.

2.3 Mobile development platforms

2.3.1 Flutter

Flutter is an open-source framework developed by Google. The development language for Flutter apps is Dart. It is mainly used to develop applications for mobile platforms like Android and iOS. Both flutter web apps and Flutter desktop applications can be developed using Flutter. For building native user interface designs, it offers a collection of fully customizable widgets. For example, the Material Design library and Cupertino widgets enable users to build user interfaces and rich motion APIs. These widgets offer pre-built and customizable functionalities for creating native user interfaces for platforms. Besides, Flutter has a great “hot reload” functionality that enables users

modify the code and view the changes immediately without debugging the whole project and losing state on the emulator. Rendering engine, pre-made widgets, and development tools along with Flutter help speed up an application's development process. It has a consistent unified Object model, the Widget, which integrates views, view controllers, layouts, and other properties.

2.3.2 React Native

React Native is a JavaScript framework that creates native mobile applications. It uses the React framework and offers several built-in components and APIs. React Native integrates React JavaScript library to develop fast and responsive application interface. It has excellent rendering capabilities and applies a component-based method, making users easily build simple and complex user interface designs. React Native components and native development components have a 1:1 mapping. It combines JavaScript with the components of native user interface which results in native-like appearance of the application. React Native offers various third-party plugin options, including JavaScript-based and native modules, as developing an app from scratch might be expensive. The third-party plugins help enhance the application's functionality and performance and remove the need for specific web view features.

2.4 Maps

2.4.1 Google Earth

Google Earth is a software that includes a digital globe, maps, and geographical information. It maps the Earth by superimposing satellite photos, aerial photography and geographical information system (GIS) 3D globe. Google Earth has become the ultimate source of spatial data and information for both private and public decision-support systems. Google Earth is a powerful tool for precision agriculture. Besides, Google Earth is suitable for farmers to plan their farms, monitor crop health and helps to analyse the chemical and physical conditions of the soil in a specific location. In addition, Google Earth displays a geo-located base layer of an aerial photographic image. The base layer can use different layers of information including chemical and physical conditions of soil, mineral deposits and crop production.

2.4.2 Bing Map

Bing Map was created in 2005 and is a web mapping service offered by Microsoft's Bing. Bing Map comes with the Microsoft package solution, which comprises REST Services, Spatial Data Services, Web Control, Windows Store apps control and WPF control. They offer a variety of integration options that can be included with new or existing applications, and they provide most of the services that are comparable to the Google Map API. These services include displaying maps in two dimensions, three dimensions, and street view. Bing map has the highest resolution aerial photography and also offers oblique photography known as bird's eye. Images from a bird's-eye perspective are captured at a 45-degree angle, giving a better depth perception for geography. Furthermore, Bing map offers a rich Silverlight experience to users and developers. .NET developers can easily make map-based applications using their .NET programming skills. However, offline street maps are not supported other than Windows 10 apps and users are required to download the maps in advance.

Chapter 3

System Methodology/Approach

3.1 System Design Diagram

3.1.1 System Architecture Diagram

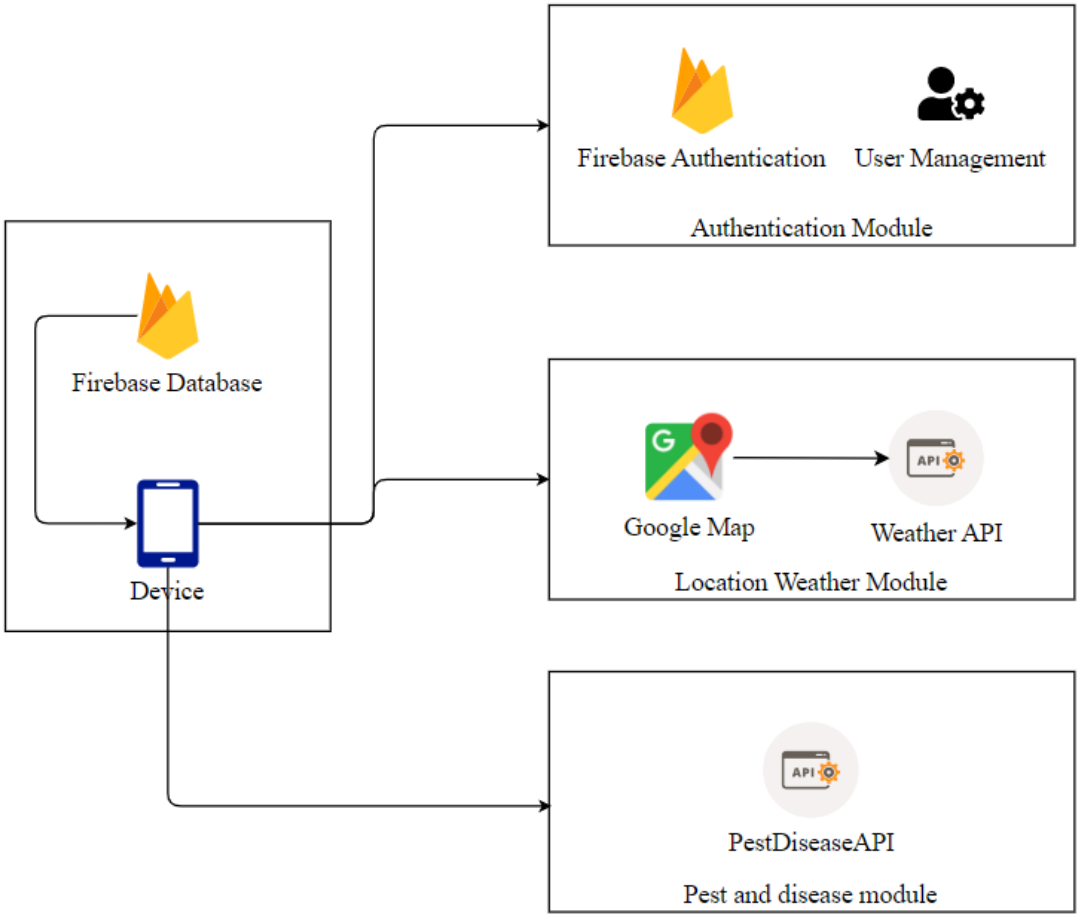


Figure 3.1: System Architecture Diagram

The figure 3.1 illustrates that the architecture diagram of the proposed application. The proposed architecture for the mobile-based farm management application includes several key components: the device, the Firebase database, an authentication module, a Location Weather module, and a pest and disease module. The device serves as the primary interface for the user to access the application. It communicates with the Firebase database to store and retrieve data related to the user's farm management

activities. The database includes information such as field locations, crop data, and job activities.

Authentication module:

The authentication module is responsible for handling user authentication and management within the mobile-based farm management application. It uses Firebase authentication, which provides secure and easy-to-use authentication services such as email and password login. The authentication module ensures that only authorized users can access the application and its features. It also handles user registration and password reset processes, as well as user profile management.

Location Weather module:

The Location Weather module utilizes Google Maps and Open Weather APIs to provide users with real-time weather information based on their farm location. This module helps farmers make informed decisions related to their crops, such as when to water or apply fertilizers, by providing accurate weather data. The Google Maps API is used to plot fields on the map, while the Open Weather API is used to retrieve weather data based on the user's location.

Pest and disease module:

The pest and disease module utilizes the Plant.id API to detect and identify potential pest and disease issues with crops. The Plant.id API provides a comprehensive database of known pests and diseases, as well as recommended treatments based on crop type and issue severity. The module allows users to take photos of their crops and receive recommendations for treatments based on identified issues. Once an issue has been identified, the module provides farmers with recommended treatments to address the problem. The recommended treatments fall into three categories: chemical control, biological control, and prevention.

3.1.2 Use Case Diagram and Description

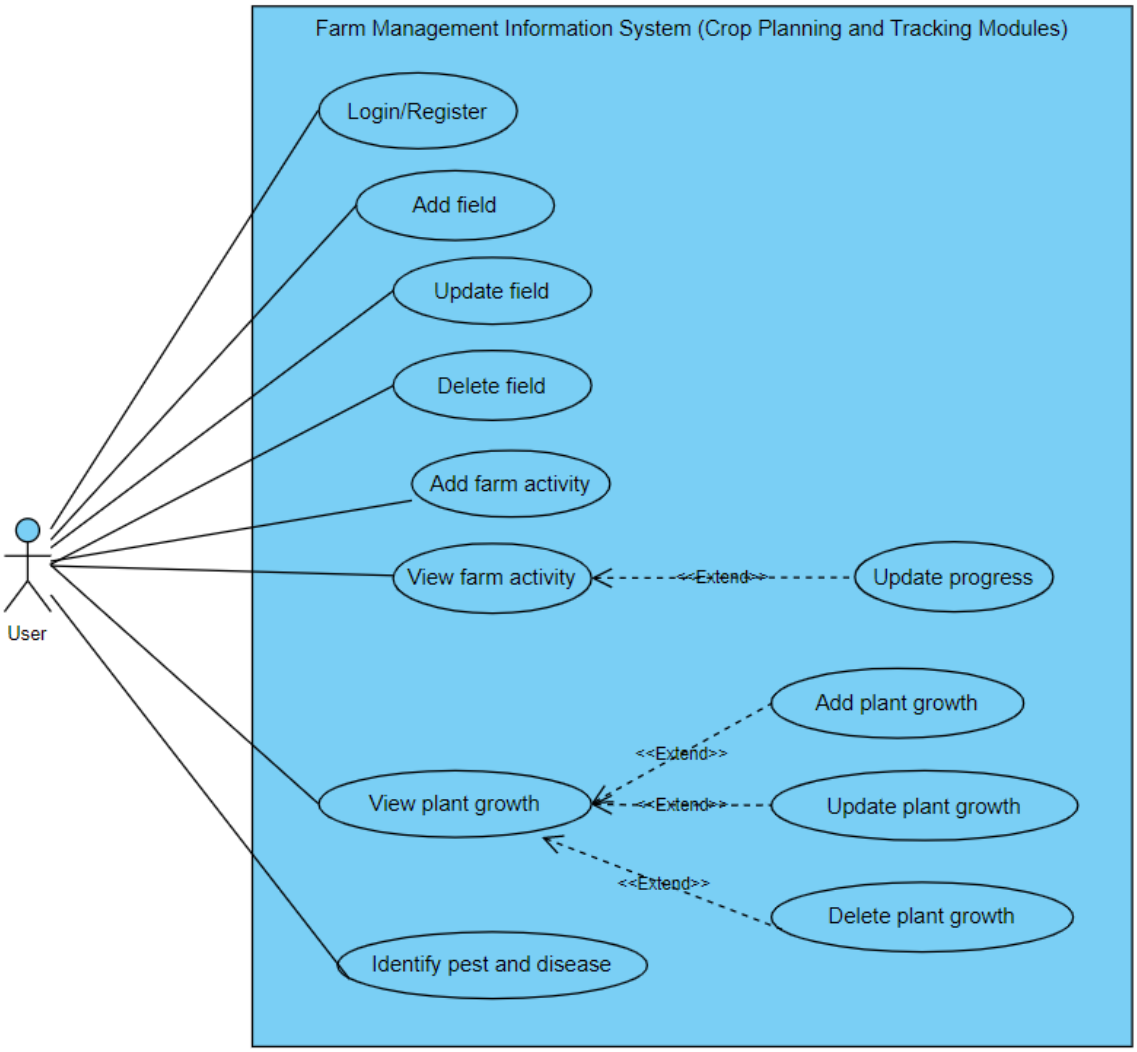


Figure 3.2: Use Case Diagram

Figure 3.2 elaborates the use case diagram for the system. **Login/Register** use case allows a user to create a new account or log in to an existing account in the mobile app. It ensures that only authorized users can access the app's features. **Add field** use case allows a user to add a new field to the app by plotting it on a Google Map. This feature is useful for farmers who want to keep track of the different areas of their farm. **Update field** use case allows a user to update the details of a field that has already been added to the app. This could include changes to the field's data, or crop type. **Delete field** use case allows a user to delete a field from the app. This feature could be used if a farmer decides to no longer cultivate a particular area of their farm. **Add farm activity** use case allows a user to add a new farm activity to a field, such as watering, fertilizing, or

spraying pesticides. This feature is useful for keeping track of which activities have been performed on each field and when. **View farm activity** use case allows a user to view a list of all the farm activities that have been performed on a particular field and user is able to update progress if they had done the farm activity. **View plant growth** use case allows a user to monitor the growth of a particular crop in a field. The user can add, update, or delete data related to the plant's growth at different stages of its development. **Identify pest and disease** use case allows a user to upload an image of a plant that is suspected to be suffering from a pest or disease. The app uses the image to identify the problem and provide treatment recommendations.

Login/Register Use Case

Use Case Name: Login/Register	ID: 1	Important Level: High
Primary Actor: User	Use Case type: Detail, essential	
Stakeholders and Interests: User – wants to login or register an account.		
Trigger: User wants to login or register an account. Type: External		
Relationships: Association: User Include: Extend: Generalization:		
Normal Flow of Events: 1. The application displays the login/register screen. 2. If the user already has an account, they enter their username/email and password and click the login button. 3. If the user does not have an account, they click on the sign-up button. 4. The application displays the registration form, which requests the user's first and last name, email, and password. 5. The user fills in the registration form and clicks the register button. 6. The application verifies the registration details and saves them in the database. 7. It the user has successfully enter email and password, the application displays the main screen.		

Sub Flows: Not applicable
Alternative/Exception Flows: 7a. If the user enters an incorrect username/email or password, the application displays an error message and prompts the user to try again.

Table 3.1: Login/Register Use Case Description

Add Field Use Case

Use Case Name: Add field	ID: 2	Important Level: High
Primary Actor: User	Use Case type: Detail, essential	
Stakeholders and Interests: User – wants to add field details.		
Trigger: User wants to add field details. Type: External		
Relationships: Association: User Include: Extend: Generalization:		
Normal Flow of Events: 1. User selects the "Add Field" button. 2. The application displays a map interface to the user. 3. The user plots a polygon shape on the map to define the boundaries of the field. 4. The user fills in the required details such as field name, crop type, and other relevant information. 5. The user saves the field information.		
Sub Flows: Not applicable		

Alternative/Exception Flows: 4a. If the required fields are not filled in, the system should display an error message to prompt the user to fill in the missing fields. The system should not save the incomplete data until all required fields are filled in.

Table 3.2: Add Field Use Case Description

Update Field Use Case

Use Case Name: Update field	ID: 3	Important Level: Medium
Primary Actor: User	Use Case type: Detail, essential	
Stakeholders and Interests: User – wants to update field details.		
Trigger: User wants to update field details. Type: External		
Relationships: Association: User Include: Extend: Generalization:		
Normal Flow of Events: 1. User selects the field that they want to update. 2. The system displays the details of the selected field. 3. User selects the edit button. 4. User modifies the details as required. 5. User clicks the "Update Field" button. 6. The system validates the updated details to ensure that all required fields are filled in. 7. The user saves the updated field information.		
Sub Flows: Not applicable		

Alternative/Exception Flows: 6a. If the required fields are not filled in, the system displays an error message prompting the user to fill in the missing fields. 7a, If the updated details are valid, the system saves the changes to the database and displays a confirmation message to the user.

Table 3.3: Update Field Use Case Description

Delete Field Use Case

Use Case Name: Delete field	ID: 4	Important Level: Low
Primary Actor: User	Use Case type: Detail, essential	
Stakeholders and Interests: User – wants to delete the field.		
Trigger: User wants to delete the field. Type: External		
Relationships: Association: User Include: Extend: Generalization:		
Normal Flow of Events: 1. User selects the field that they want to delete. 2. The system displays the details of the selected field. 3. User selects the delete button. 4. The system prompts the farmer to confirm their action. 5. User selects “YES” if they want to delete the field. 6. The system deletes the field from the user’s farm and removes all associated data.		
Sub Flows: Not applicable		

Alternative/Exception Flows: 5a. If the farmer cancels the deletion process, the system returns them to the "Fields" section without deleting the field.

Table 3.4: Delete Field Use Case Description

Add Farm Activity Use Case

Use Case Name: Add farm activity	ID: 5	Important Level: High
Primary Actor: User	Use Case type: Detail, essential	
Stakeholders and Interests: User – wants to add farm activity.		
Trigger: User wants to add farm activity. Type: External		
Relationships: Association: User Include: Extend: Generalization:		
Normal Flow of Events: 1. User selects a field to add a farm activity. 2. User selects the type of farm activity to add (e.g., watering, trimming, wrapping, placing fertilizer, spraying pesticides). 3. User selects the date and time for the farm activity. 4. User enters any additional details about the farm activity. 5. User saves the farm activity.		
Sub Flows: Not applicable		

Alternative/Exception Flows: 4a. If the required fields are not filled in, the system displays an error message prompting the user to fill in the missing fields.

Table 3.5: Add Farm Activity Use Case Description

View Farm Activity Use Case

Use Case Name: View farm activity	ID: 6	Important Level: High
Primary Actor: User	Use Case type: Detail, essential	
Stakeholders and Interests: User – wants to view the list of farm activities that have been planned for a specific field.		
Trigger: User wants to view farm activity. Type: External		
Relationships: Association: User Include: Extend: Update progress Generalization:		
Normal Flow of Events: 1. User selects a field from the list of fields. 2. The app displays a list of farm activities for the selected field. 3. User views the list of farm activities. 4. If user want to update the job progress Execute the Update progress use case		
Sub Flows: Not applicable		

Alternative/Exception Flows: 4a. If the farm activity has already been completed, the app displays a message indicating that the farm activity has already been completed.

Table 3.6: View Farm Activity Use Case Description

View Plant Growth Use Case

Use Case Name: View plant growth	ID: 7	Important Level: High
Primary Actor: User	Use Case type: Detail, essential	
Stakeholders and Interests: User – wants to monitor the growth of their crop.		
Trigger: User wants to monitor the growth of their crop. Type: External		
Relationships: Association: User Include: Extend: add plant growth, update plant growth, delete plant growth Generalization:		
Normal Flow of Events: 1. The app displays a list of user's fields. 2. The farmer selects a field from the list of fields. 3. User can view the growth details of their crops.		
Sub Flows: Not applicable		
Alternative/Exception Flows: Not applicable		

Table 3.7: View Plant Growth Use Case Description

Add Plant Growth Use Case

Use Case Name: Add plant growth	ID: 8	Important Level: High
Primary Actor: User	Use Case type: Detail, essential	
Stakeholders and Interests: User – wants to add details about the growth of a plant in different stages on a specific field.		
Trigger: User wants to add details about the growth of a plant. Type: External		
Relationships: Association: User Include: Extend: Generalization:		
Normal Flow of Events: 1. User selects the "Add more" option. 2. The system displays a form for the user to enter the plant growth details. 3. The user fills in the required fields including the crop height, crop quantity, date of observation, and any notes. 4. The farmer clicks the "Add" button. 5. The system validates the data and saves the plant growth information to the database. 6. The system displays a message confirming that the plant growth information has been successfully saved.		
Sub Flows: Not applicable		

Alternative/Exception Flows: 5a. If the required fields are not filled in, the system displays an error message prompting the user to fill in the missing fields.

Table 3.8: Add Plant Growth Use Case Description

Update Plant Growth Use Case

Use Case Name: update plant growth	ID: 9	Important Level: High
Primary Actor: User	Use Case type: Detail, essential	
Stakeholders and Interests: User – wants to update details about the growth of a plant in different stages on a specific field.		
Trigger: User wants to update details about the growth of a plant. Type: External		
Relationships: Association: User Include: Extend: Generalization:		
Normal Flow of Events: 1. The application shows a list of previously added plant growth details. 2. User selects the plant growth detail that he/she wants to update. 3. The application displays the details of the selected plant growth, including the crop height, crop quantity, date of observation, and any notes. 4. User modifies the plant growth details that he/she wants to update. 5. The application validates the updated details and ensures that all required fields are filled in. 6. User selects the "Update" button to save the changes. 7. The application saves the updated plant growth details in the database and displays a success message to the farmer.		

Sub Flows: Not applicable
Alternative/Exception Flows: 5a. If the required fields are not filled in, the system displays an error message prompting the user to fill in the missing fields. 6a. If the user decides not to update any details, he/she can select the "Cancel" button to return to the list of plant growth details without making any changes.

Table 3.9: Update Plant Growth Use Case Description

Delete Plant Growth Use Case

Use Case Name: delete plant growth	ID: 10	Important Level: High
Primary Actor: User	Use Case type: Detail, essential	
Stakeholders and Interests: User – wants to delete details about the growth of a plant in different stages on a specific field.		
Trigger: User wants to delete details about the growth of a plant. Type: External		
Relationships: Association: User Include: Extend: Generalization:		
Normal Flow of Events: 1. User selects the "Delete" button for the plant growth record they wish to delete. 2. The system removes the selected plant growth record from the database.		
Sub Flows: Not applicable		
Alternative/Exception Flows: Not applicable		

Table 3.10: Delete Plant Growth Use Case Description

Identify Pest and Disease Use Case

Use Case Name: identify pest and disease	ID: 11	Important Level: High
Primary Actor: User	Use Case type: Detail, essential	
Stakeholders and Interests: User – wants to identify pest and disease affecting crops in the fields and obtain treatment recommendations.		
Trigger: User wants to identify pest and disease. Type: External		
Relationships: Association: User Include: Extend: Generalization:		
Normal Flow of Events: 1. The user uploads an image of the crop affected by pest and disease. 2. The system processes the image using a pest and disease recognition API. 3. The system provides the user with recommendations to treat the crop disease. 4. The system provides the user with pest details that appear on their crops. 5. The farmer can choose to follow the recommendations or seek further advice.		
Sub Flows: Not applicable		

Alternative/Exception Flows: 2a. If the pest and disease recognition API is unable to identify the pest or disease, the system notifies the user that the issue could not be identified.

Table 3.11: Identify Pest and Diseases Use Case Description

3.1.3 Activity Diagram

Login and Register

Users need to enter registration details to register an account. The system will redirect users to login page if users register successfully. Users log in to their account by entering email and password. If the email and password are correct, the system will redirect users to Home Page.

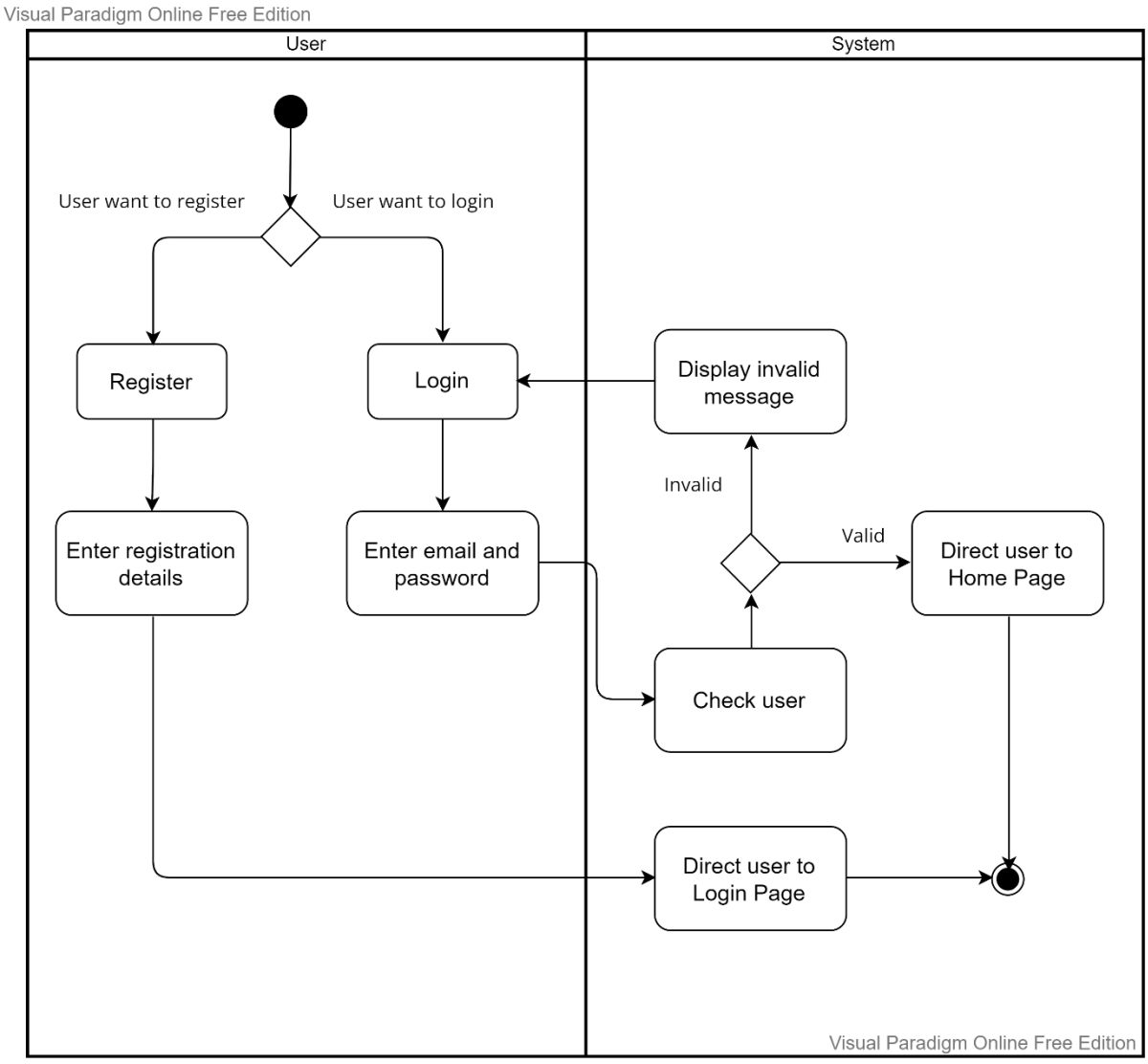


Figure 3.3: Activity diagram for login and register use case

Add field

Users need to enter field information such as field types, field name, crop categories, plant date and others. If users enter all the required fields, the system will display add successfully message.

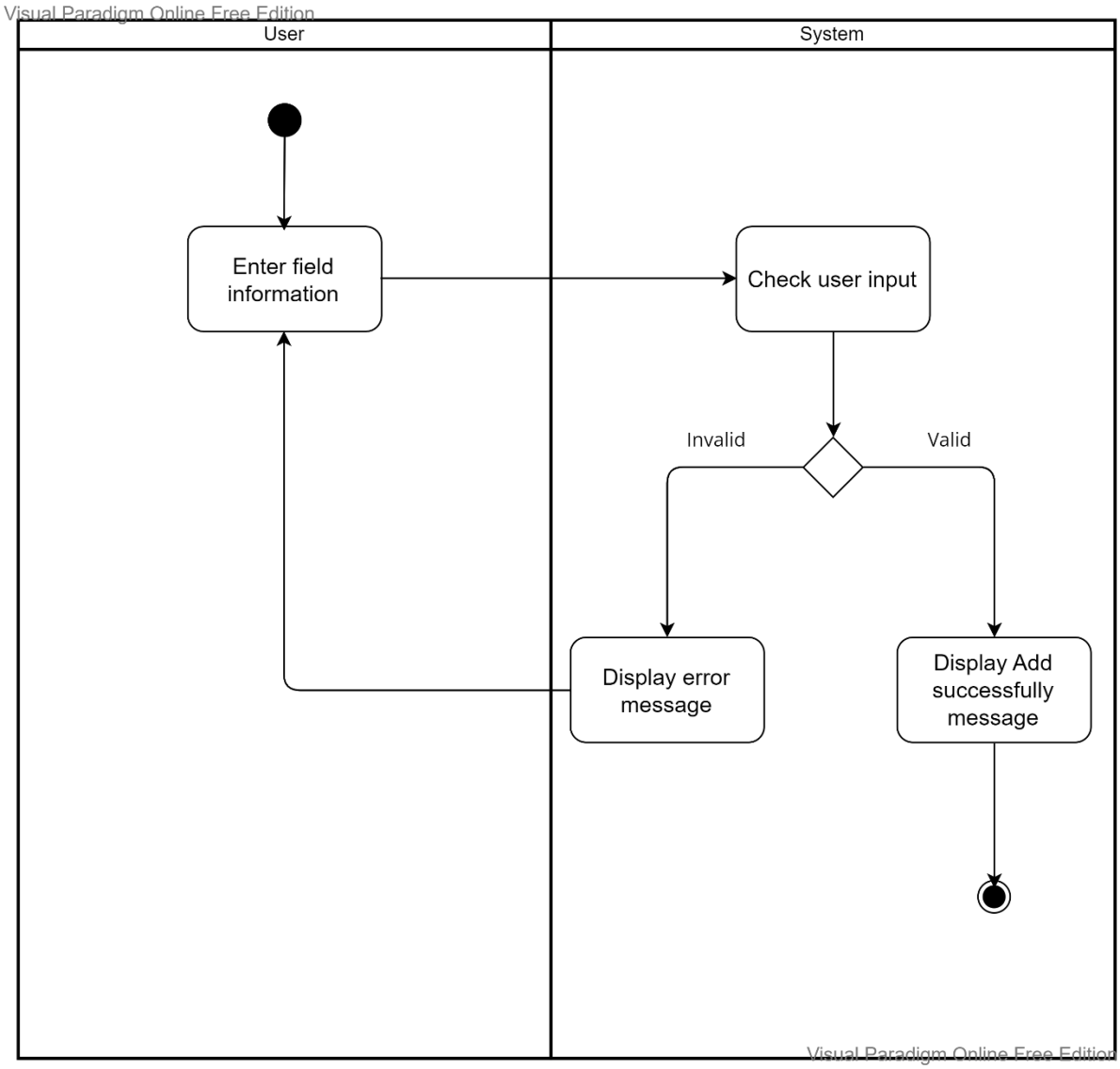


Figure 3.4: Activity diagram for add field use case

Update field

Users can select field in the list. Users allow to modify the field information such as field types, field name, crop categories, plant date and others. If users enter all the required fields, the system will display update successfully message.

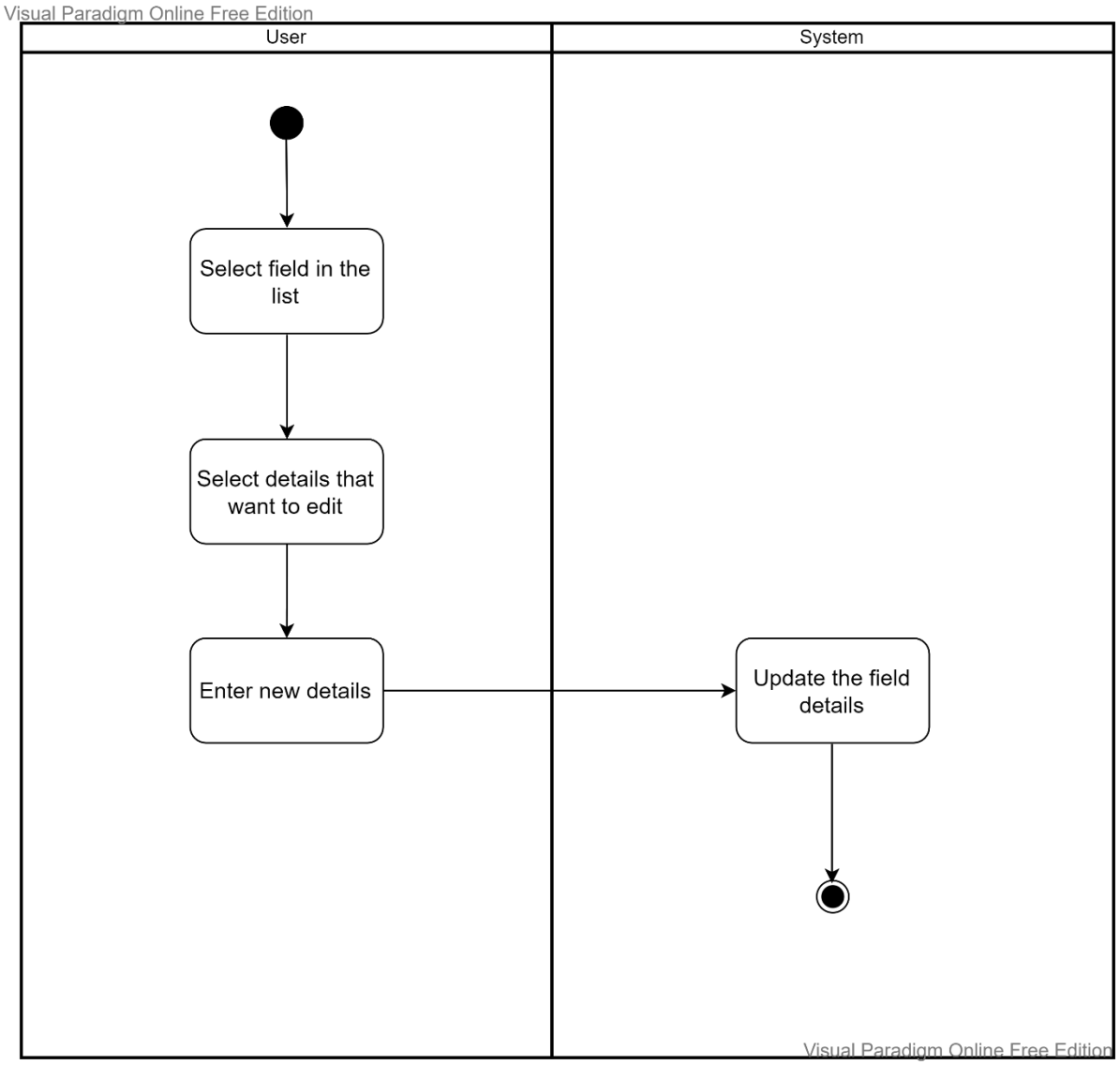


Figure 3.5: Activity diagram for update field use case

Delete field

Users can select a field in the list. Users allow to delete field by selecting delete button.

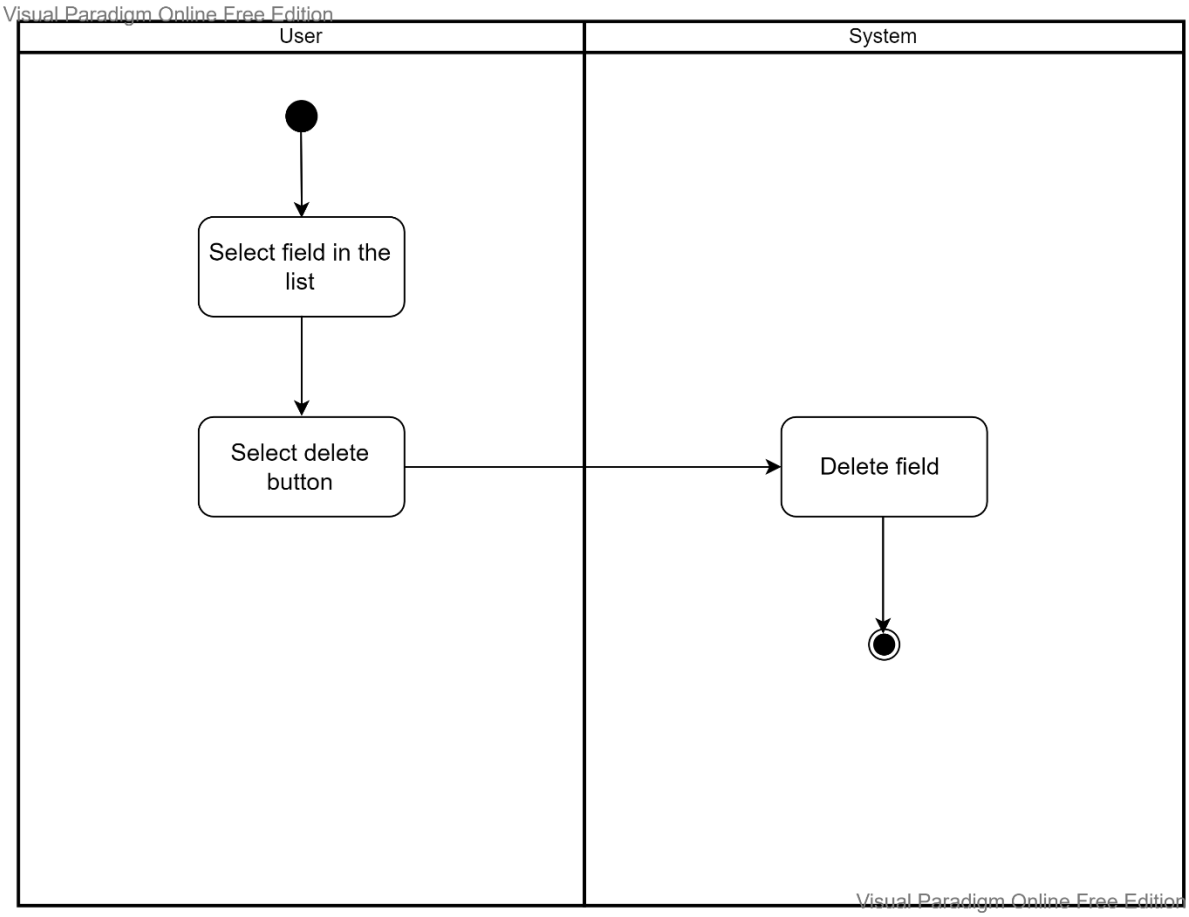


Figure 3.6: Activity diagram for delete field use case

Add farm activity

Users need to select a field in the list. There are five types of farm activities which are spraying, fertilizing, trimming, watering and wrapping. Users need to choose type of farm activity and enter required details. All the details will be saved after users select the add button.

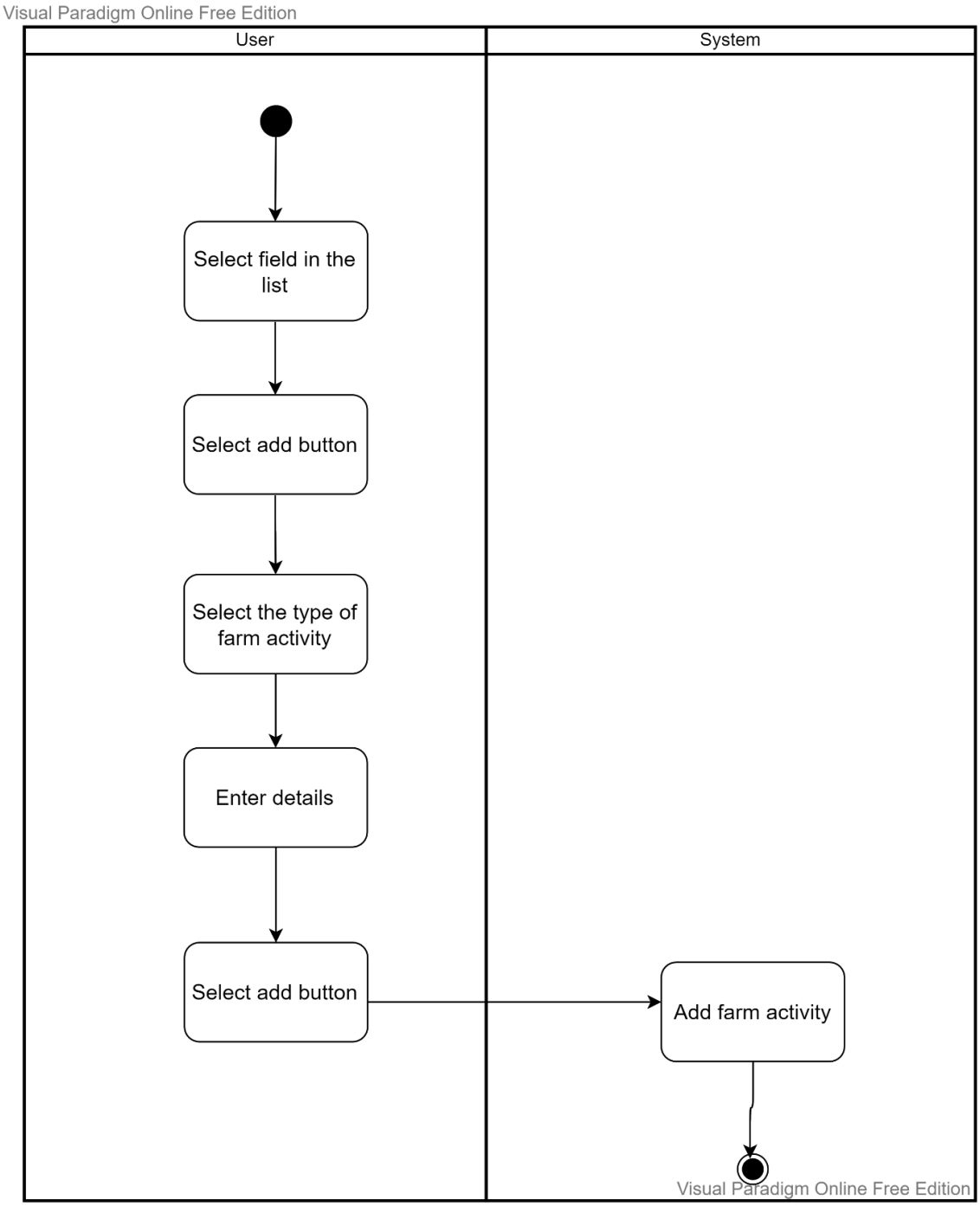


Figure 3.7: Activity diagram for add farm activity use case

View farm activity

Users need to select a field in the list. Then, select View farm activities button. The system will display all farm activities on the selected field such as type of farm activities, product name, application rate and others. If users had done their farm activity, users can update their progress to “Completed”.

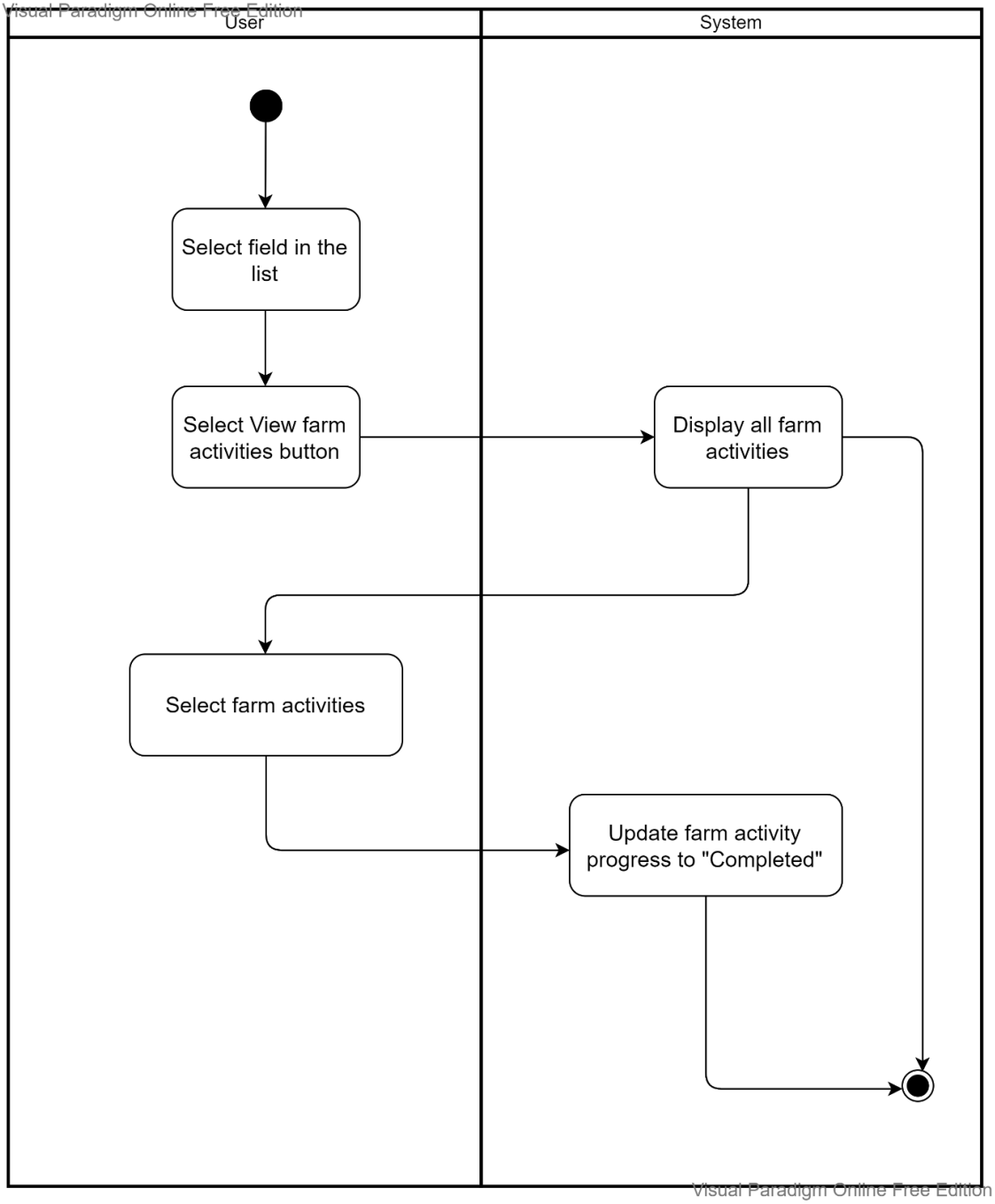


Figure 3.8: Activity diagram for view farm activity use case

View plant growth

Users need to select a field in the list. The system will display all growing stage of the crop on the selected field.

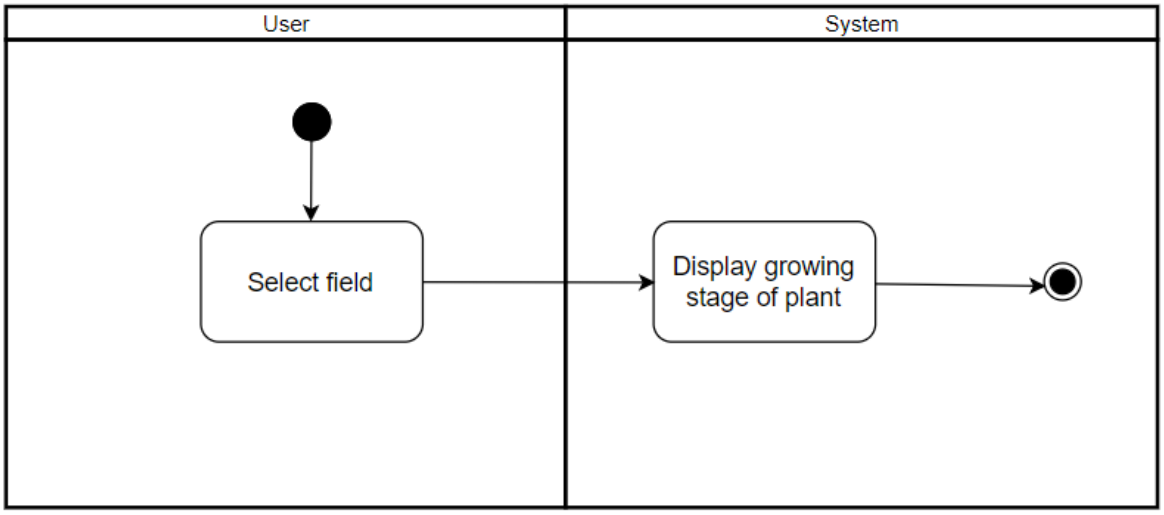


Figure 3.9: Activity diagram for view plant growth use case

Add plant growth

Users need to select a field in the list and select the growing stage of the crops that want to add. Then, users need to enter all the required details. All the details will be saved after users select the add button.

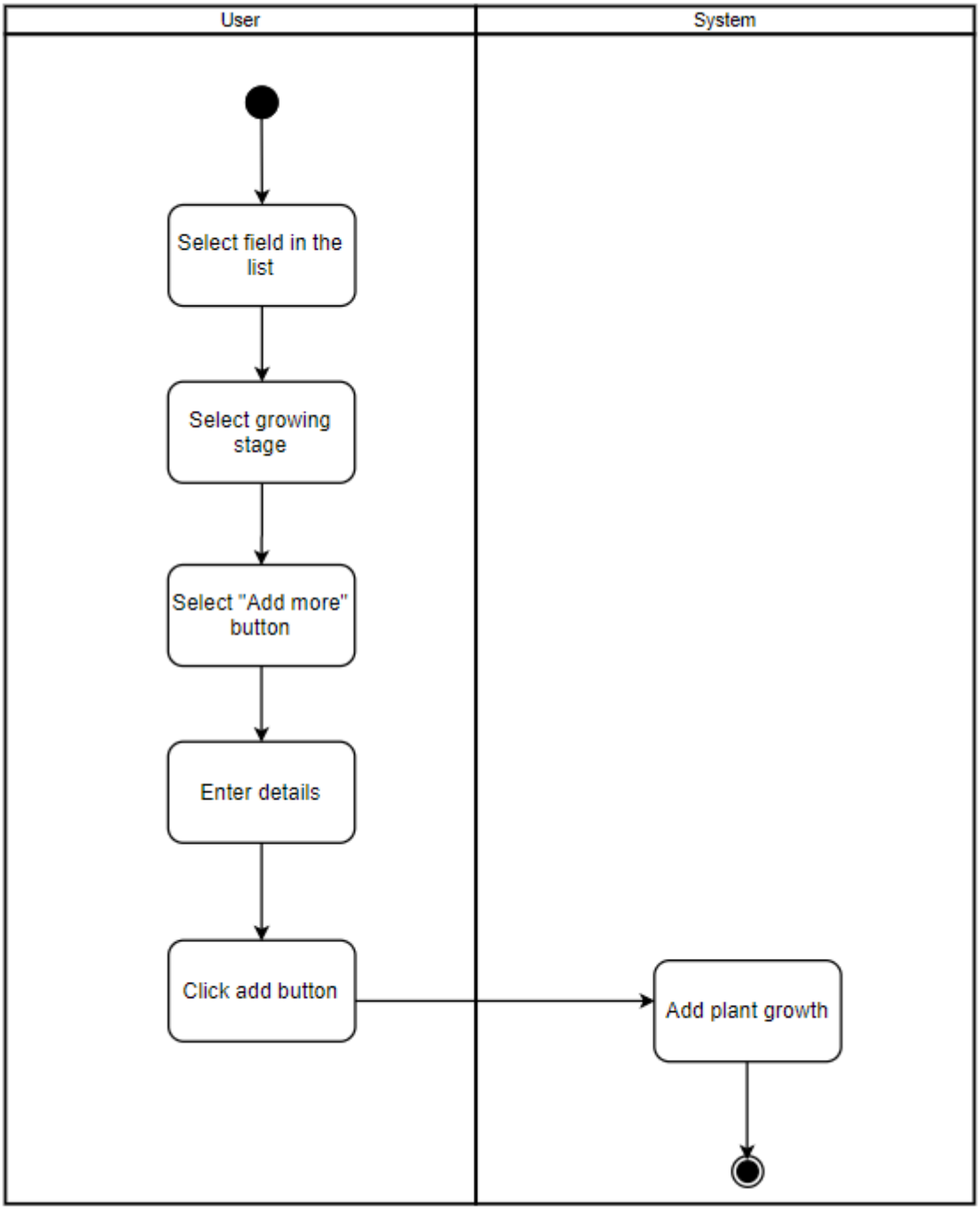


Figure 3.10: Activity diagram for add plant growth use case

Update plant growth

Users can select field in the list then select the growing stage detail that want to update. Users allow to modify the field information such as crop height, date, quantity of crops, comments and upload new image. If users enter all the required fields, the system will display update successfully message.

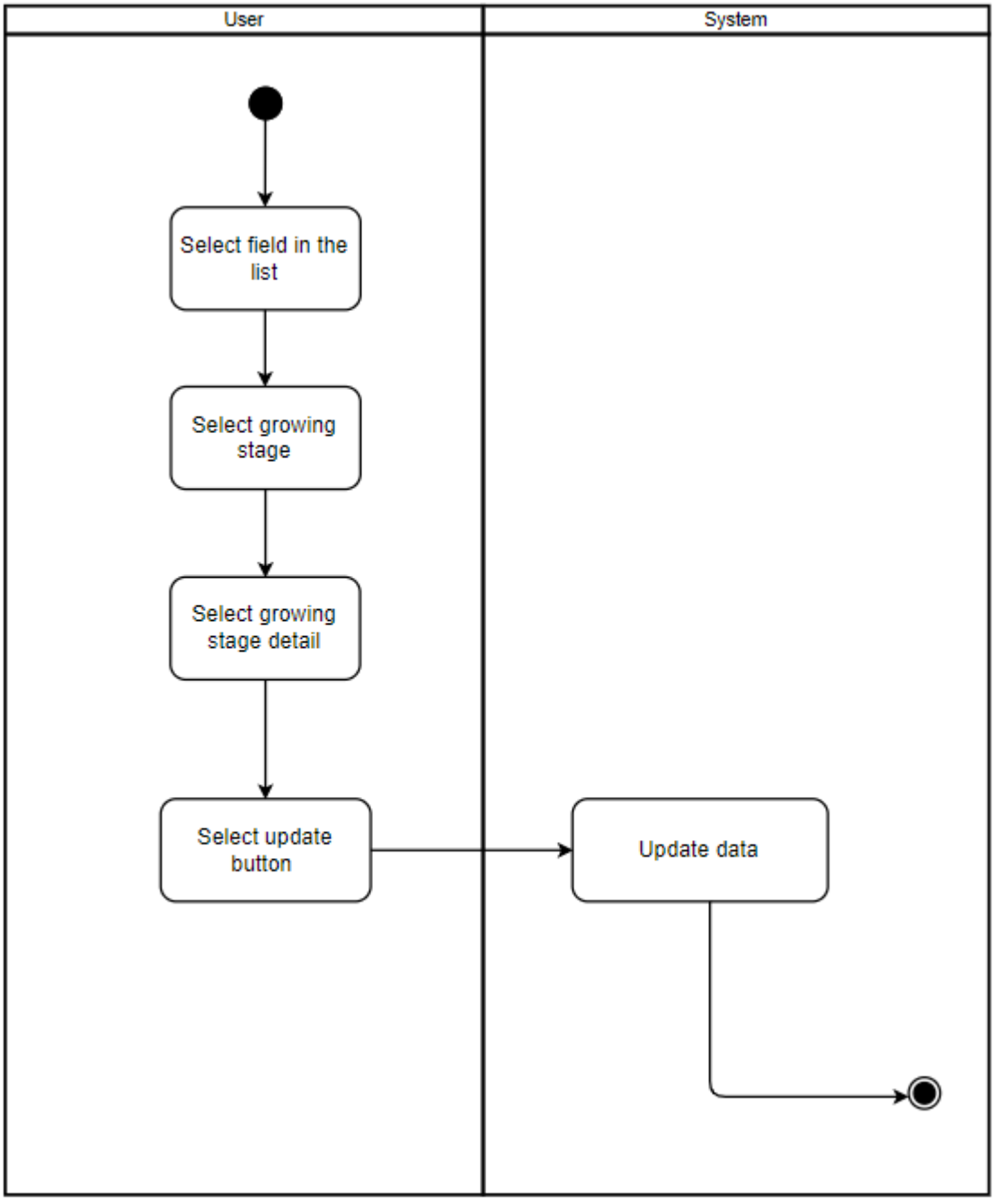


Figure 3.11: Activity diagram for update plant growth use case

Delete plant growth

Users can select a field in the list and select the growing stage that want to delete. Users allow to delete field by selecting delete button.

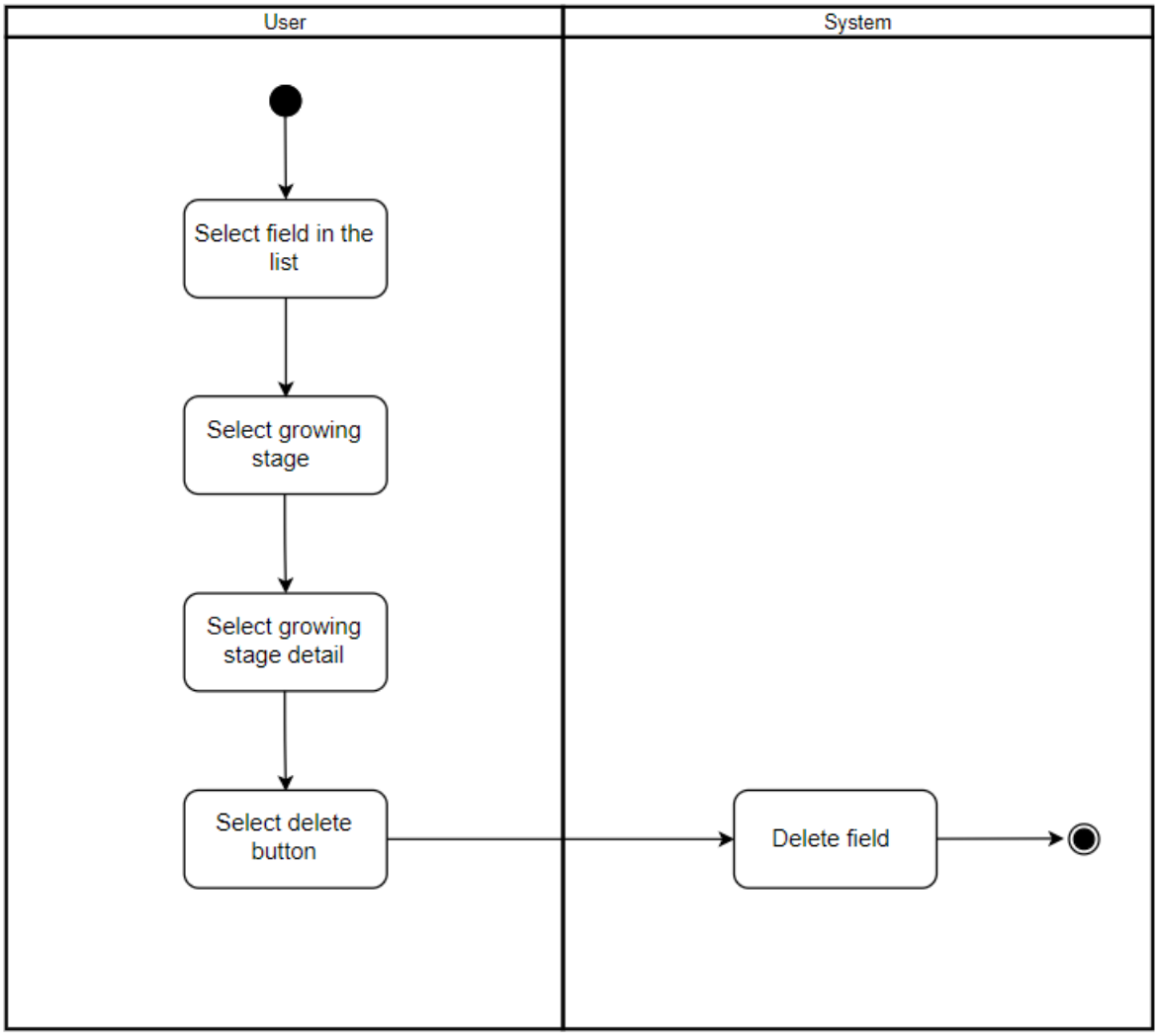


Figure 3.12: Activity diagram for delete plant growth use case

Identify pest and disease

Users upload an image of a crop disease or pest. This triggers the system to process the image and extract relevant information. If the image is of a crop disease, the system will display the crop disease and the corresponding treatment options. On the other hand, if the image is of a pest, the system will display details about the pest.

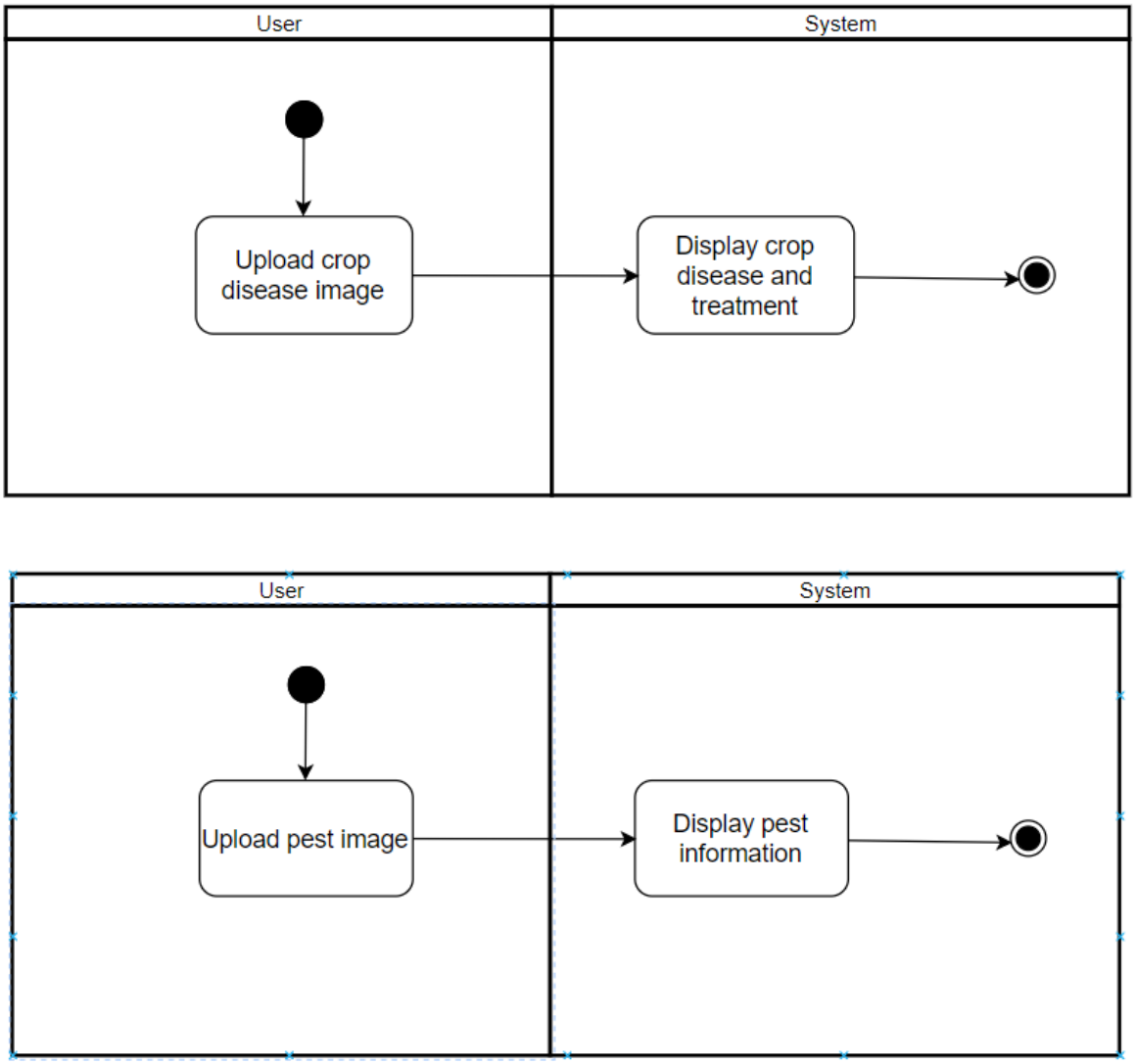


Figure 3.13: Activity diagram for identify pest and disease use case

Chapter 4

System Design

4.1 System Block Design

The project proposed a mobile based farm management application. The system can help famers plan, monitor, tracking and analyse all activities on the farm. Therefore, the proposed system will include four modules which are crop planning, job assignment, growth state tracking and pest and disease detection. An overview of the system will be shown in the block diagram 4.1 below.

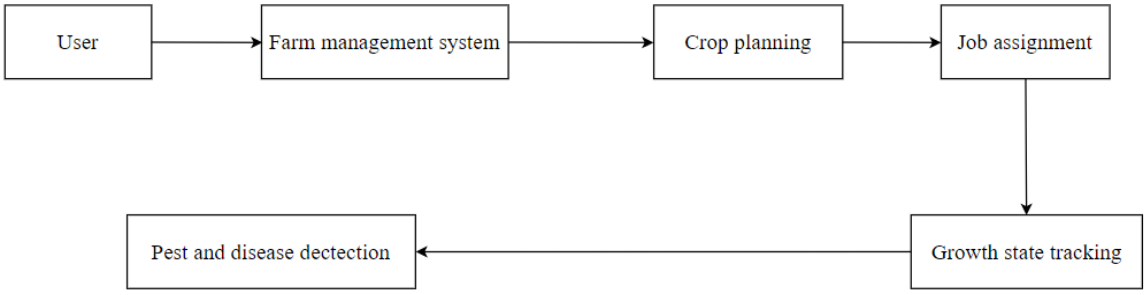


Figure 4.1: An overview of farm management system (Block Diagram)

Users need to login to the farm management system. Farm management system serves as the main interface between the user and the various modules of the app. It allows users to manage their farm, add fields, and view important information such as weather forecasts and crop growth data. It also includes features for managing user accounts and settings. In **crop planning module,** the system allows users to plan and manage their crops. The user can plot their fields on a map using Google Maps API and input crop data such as crop type, planting date, and crop details. All data entered by the user is stored in Firebase Realtime Database for easy access and manipulation. The module can also use a weather API to provide the user with information on weather patterns and conditions that may affect their crops. For example, weather temperature, wind speed and humidity. In **job assignment module**, the system allows the user to assign tasks and responsibilities to themselves or others involved in the farming process. The user can create a list of tasks, assign them to a specific person, and set a specific date

and time. This can include activities such as planting, fertilizing, watering, and wrapping.

The user can also keep track the status of assigned tasks and update them accordingly. All task-related data is stored in Firebase Realtime Database for easy access and manipulation. In **growth state tracking module**, this module enables the user to track the growth of their crops. The user can add, update, delete, and view growing stages of each crop. Each crop has different growth stages that need to be monitored and managed accordingly. All data related to crop growth stages is stored in Firebase Realtime Database for easy access and manipulation. In **pest and disease detection module**, this module is designed to help the user identify any diseases or pests that may be affecting their crops. The user can upload an image of the affected plant, and the system uses an API such as the Plant.id API to analyse the image and provide information on the type of pest or disease affecting the plant. The system can then provide the user with recommended treatment options. The treatments include biological control, chemical control, and prevention tips on the crop. This can avoid the spread of pests and diseases.

4.2 User Interface Design

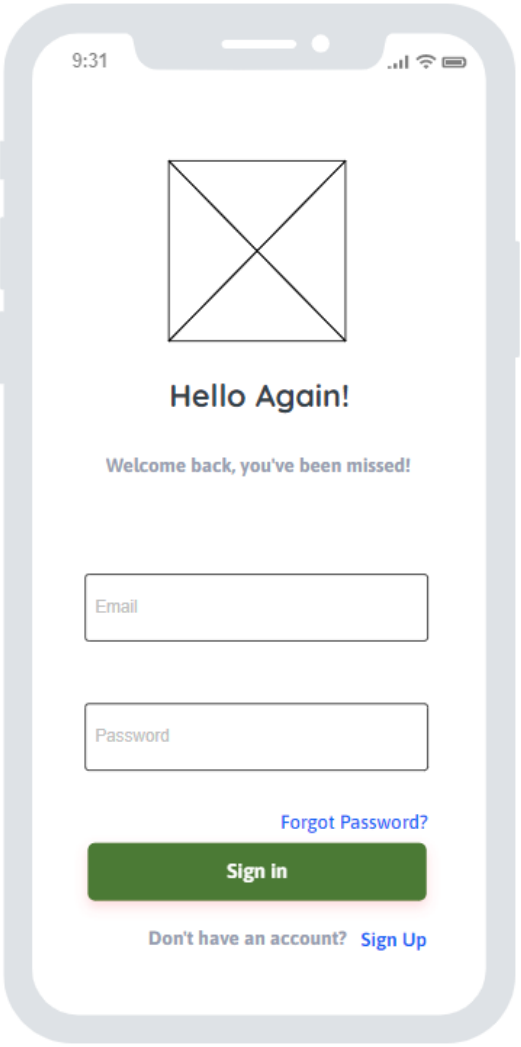


Figure 4.2: Login Page

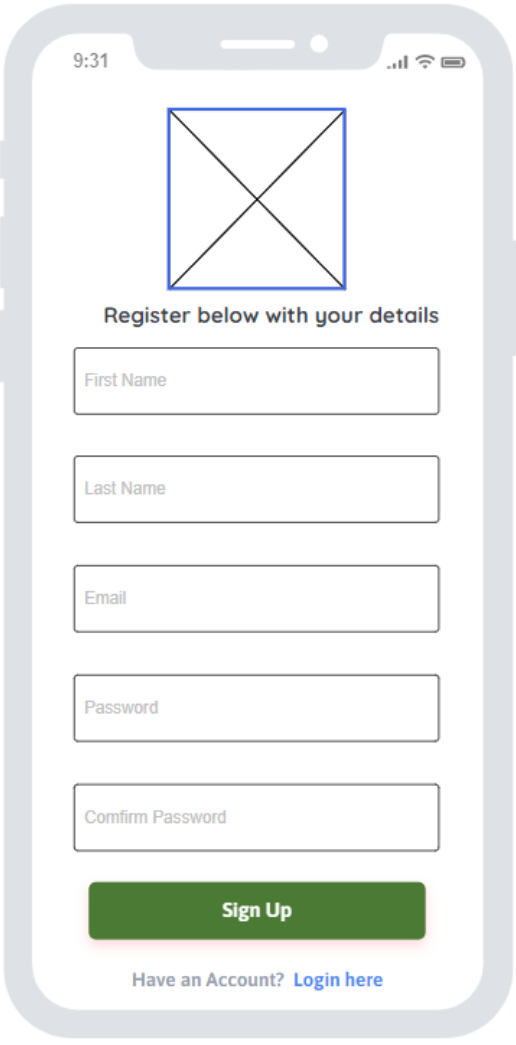


Figure 4.3: Register Page

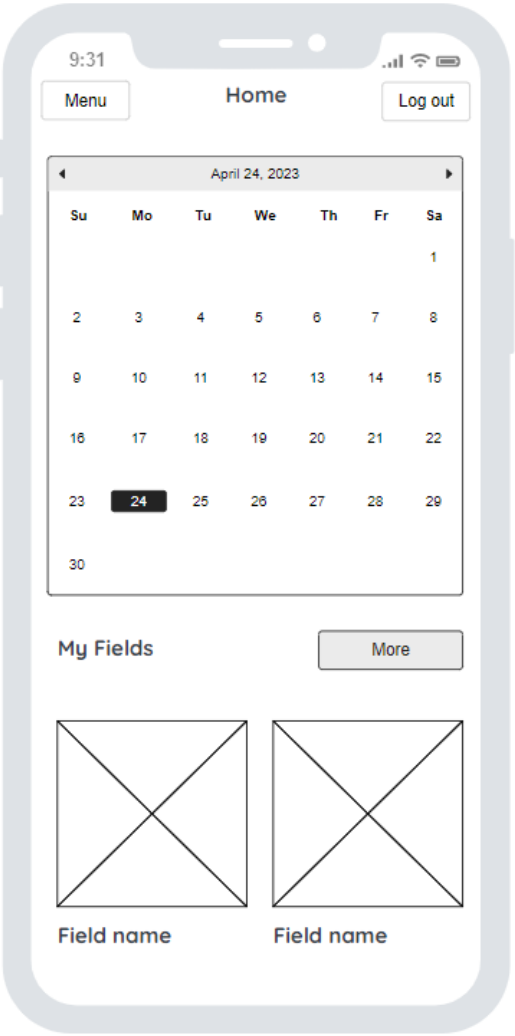


Figure 4.4: Main Page

Figure 4.5: Add Field Page

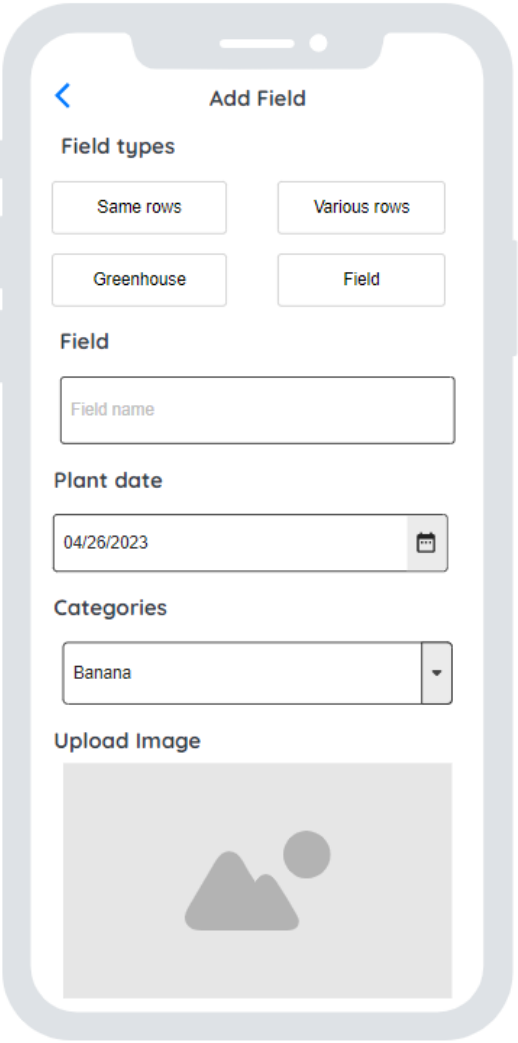


Figure 4.6: Add Field Page (Cont.)

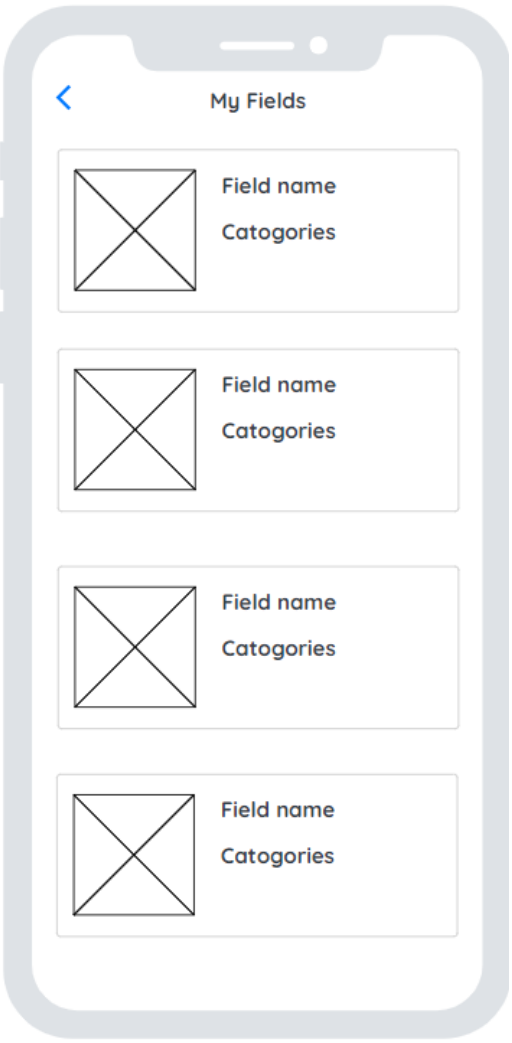


Figure 4.7: View Field Page

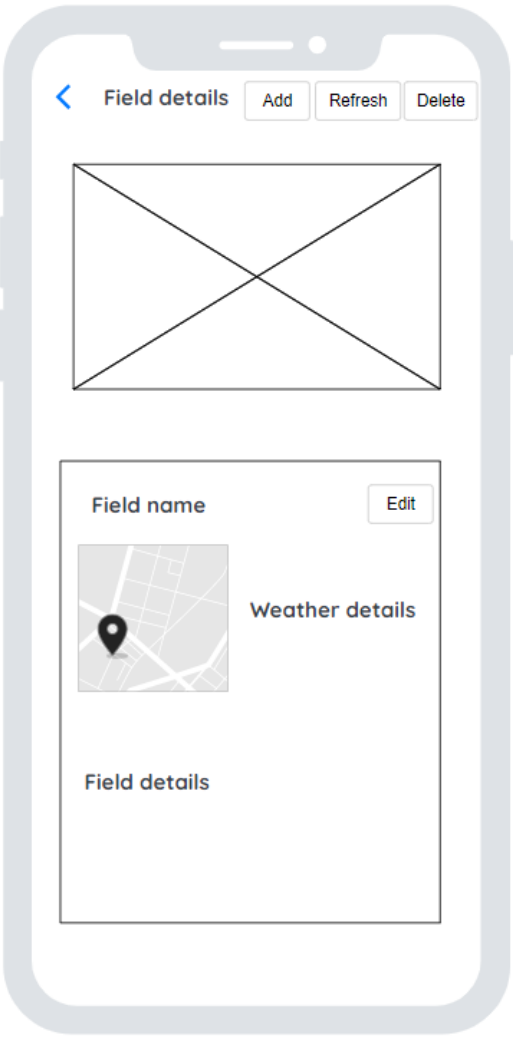


Figure 4.8: View Field Details Page

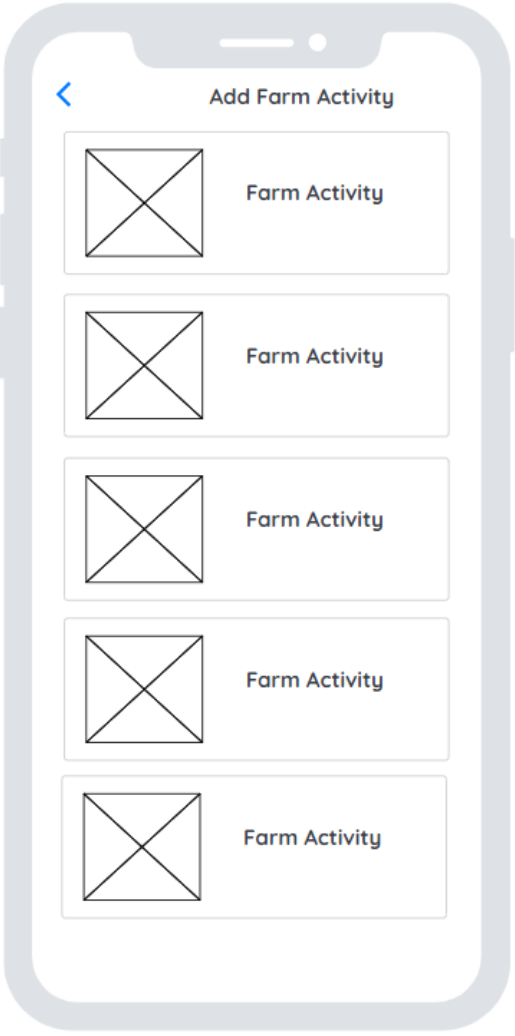


Figure 4.9: Farm Activity Page

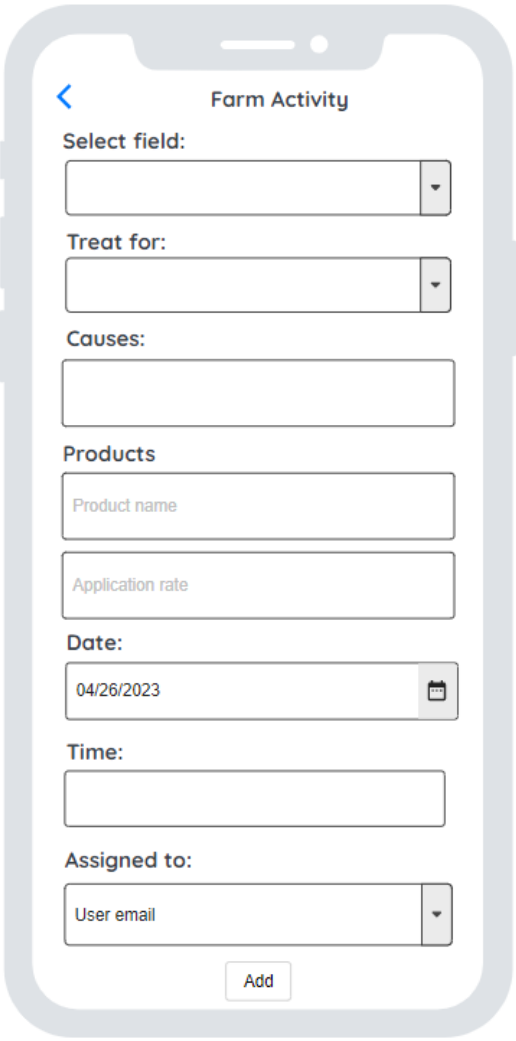


Figure 4.10: Add Farm Activity Page

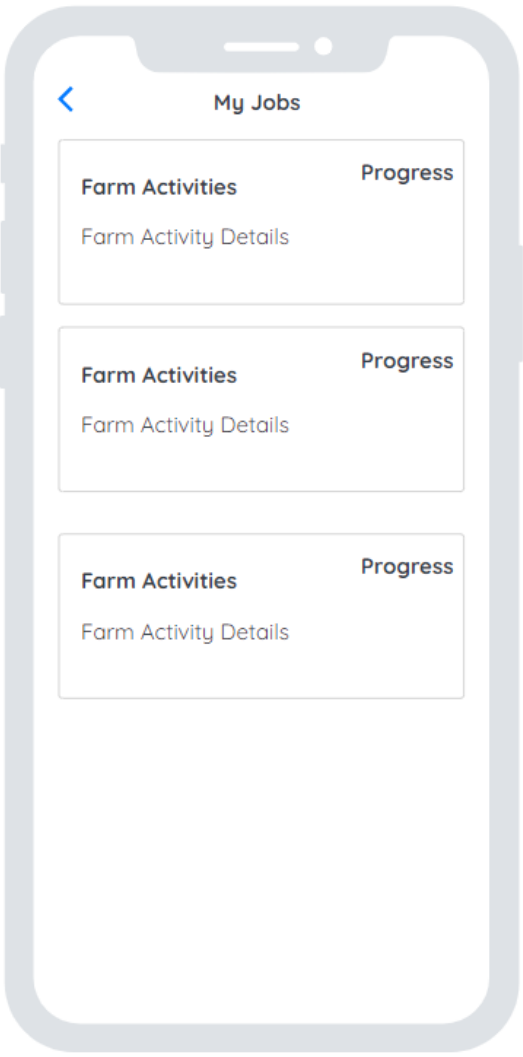


Figure 4.11: View Farm Activity Page

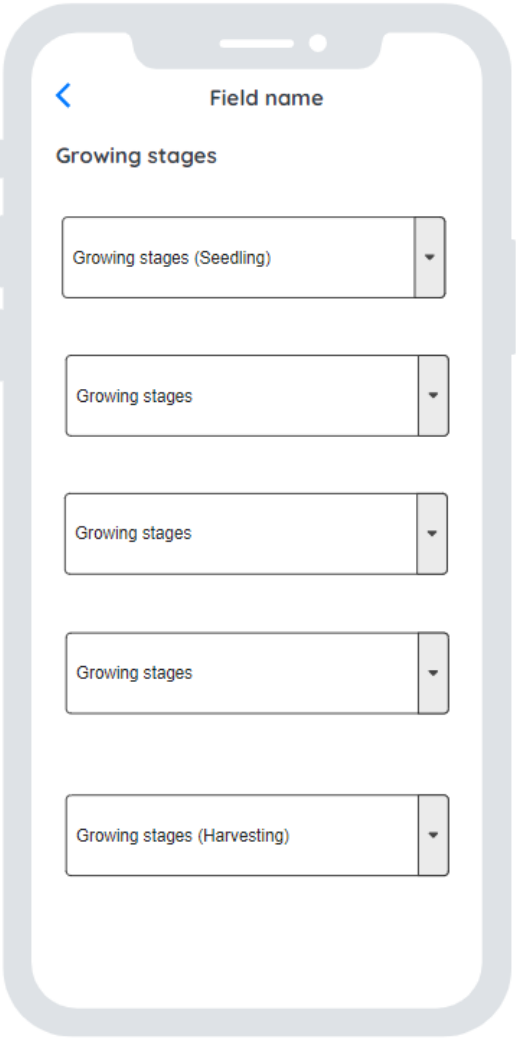


Figure 4.12: Growing stage Page

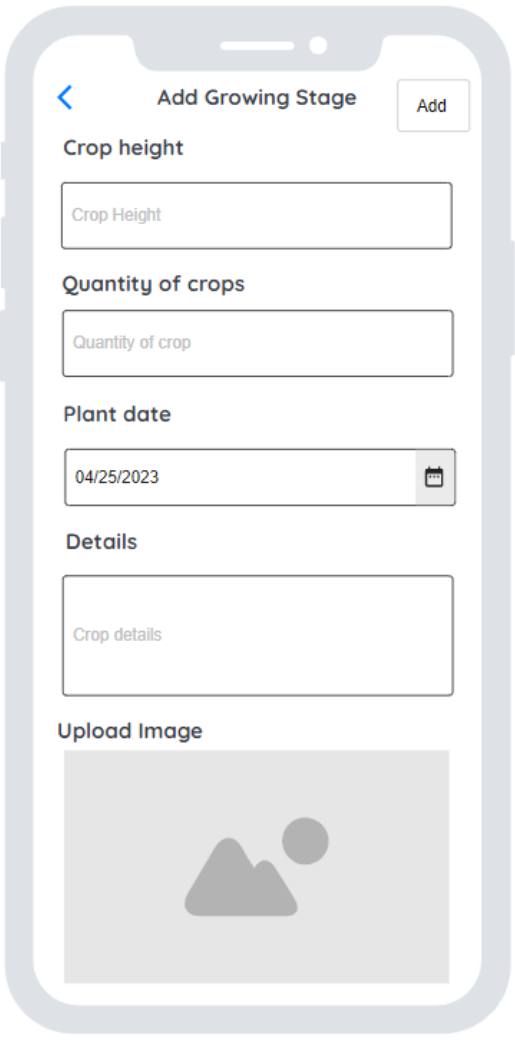


Figure 4.13: Add growing stage Page

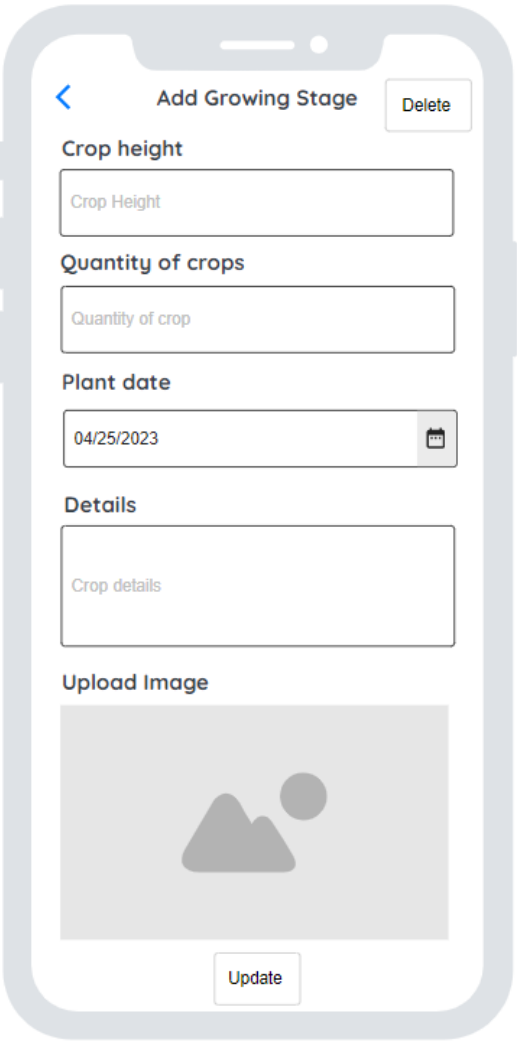


Figure 4.14: Update growing stage Page

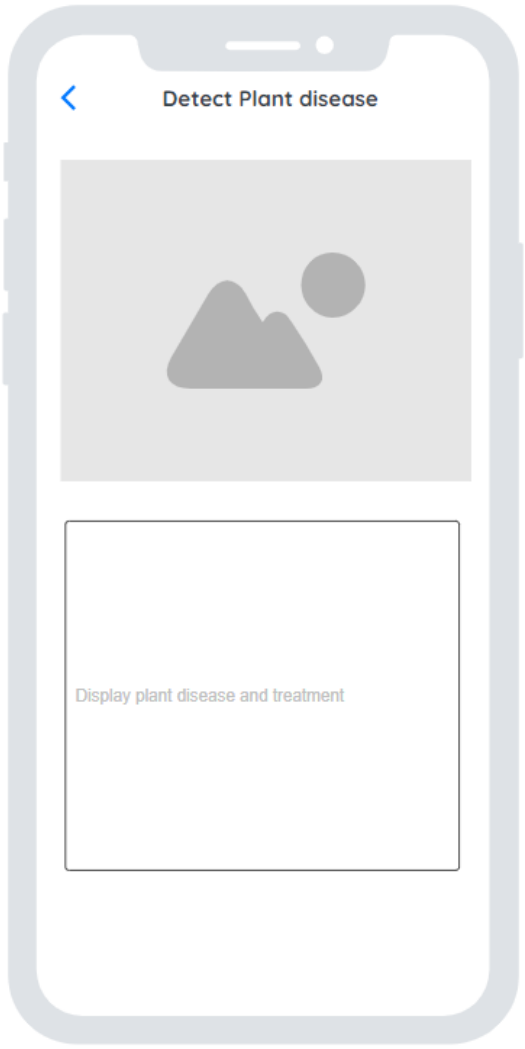


Figure 4.15: Detect Plant Disease Page

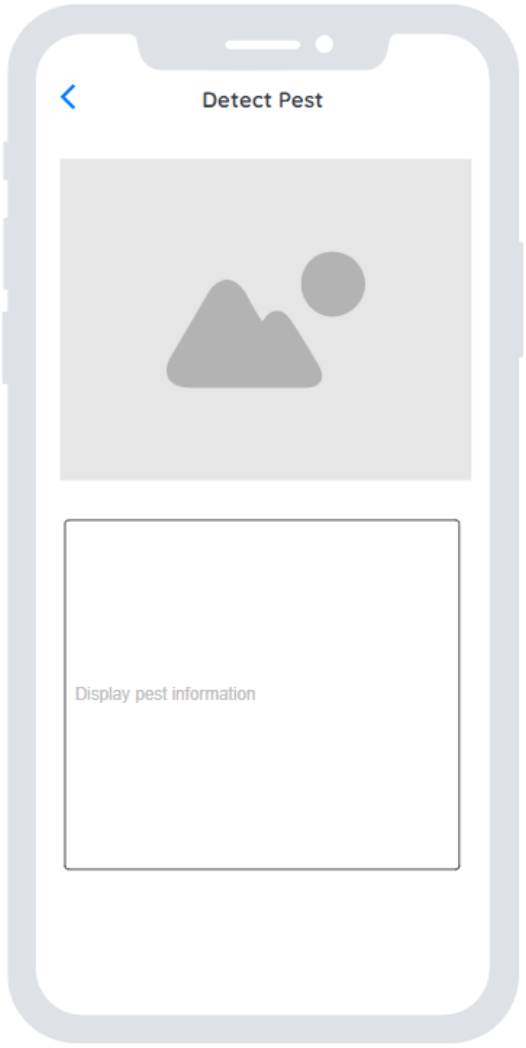


Figure 4.16: Detect Pest Page

4.3 Database Design

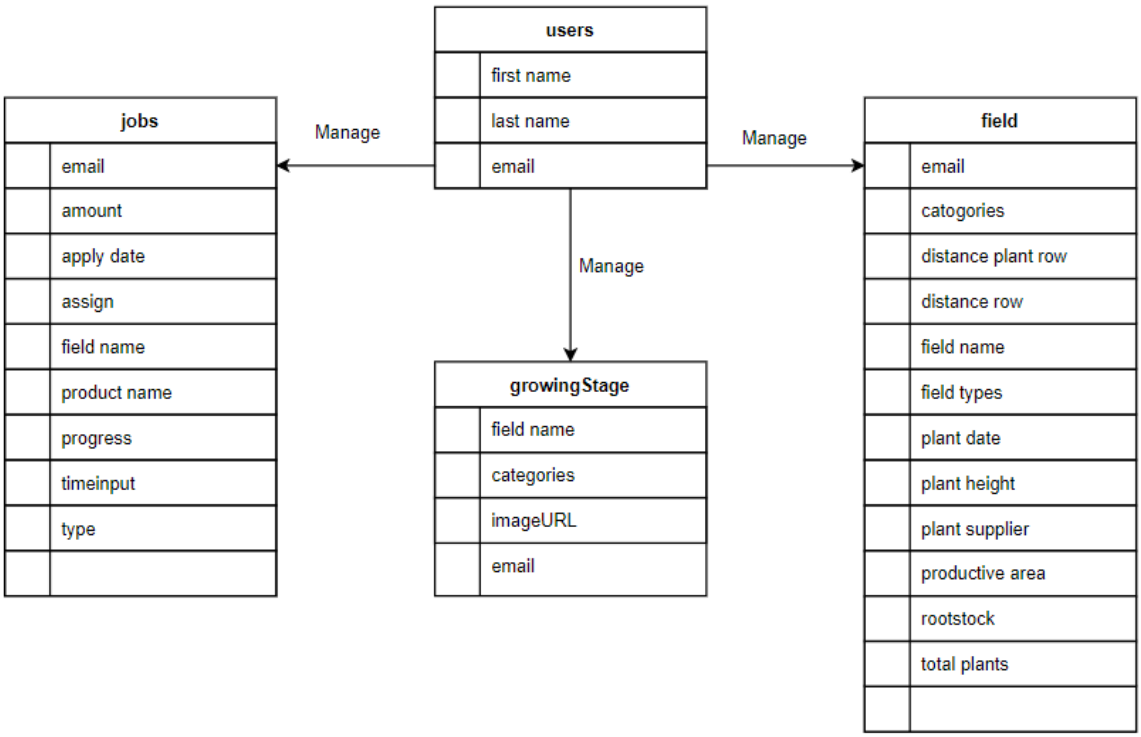


Figure 4.17: Database design

In this project, the system uses Firestore database to store data. From the figure 4.17, there are total four tables and the relationships between each table are shown in the following entity relationship diagram. All the user details will be stored in the users table. Each user can manage one or many fields by adding, modifying and deleting field record. Each user can add, view farm activities and update the progress of farm activities status. Each user can view, add, update and delete growing stage of the plant.

Chapter 5

System Implementation

5.1 Software Setup

5.1.1 Installation of Android Studio

First, download the Android Studio installer from the official Android Studio website.

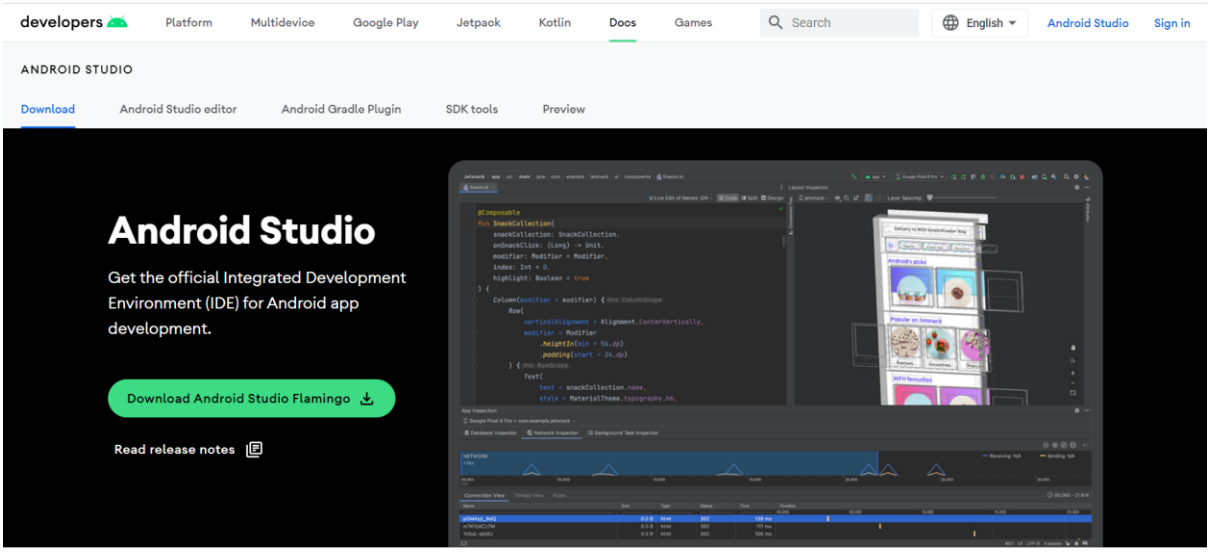


Figure 5.1: Installation of Android Studio

Follow the installation instructions provided by the installer. Users will be asked to choose the installation location, select the components you want to install, and accept the terms and conditions. Android Studio will also prompt to download and install any required SDKs, build tools, and other dependencies. Once have completed the setup process, you can start developing Android applications using Android Studio.

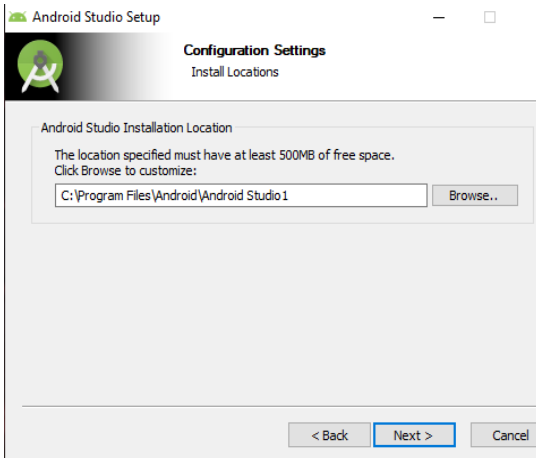


Figure 5.2: Installation of Android Studio (cont.)

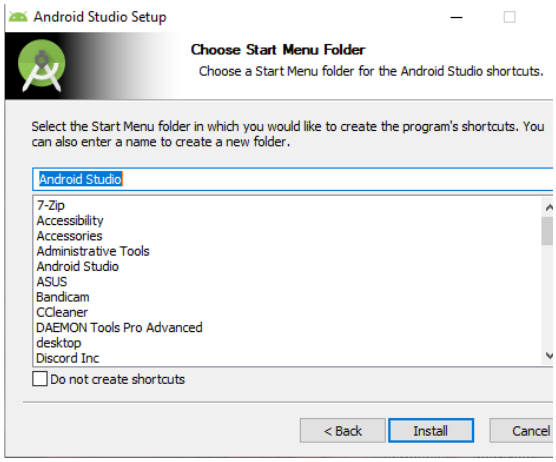


Figure 5.3: Installation of Android Studio (cont.)

5.1.2 Installation of Flutter

Select the operation system.

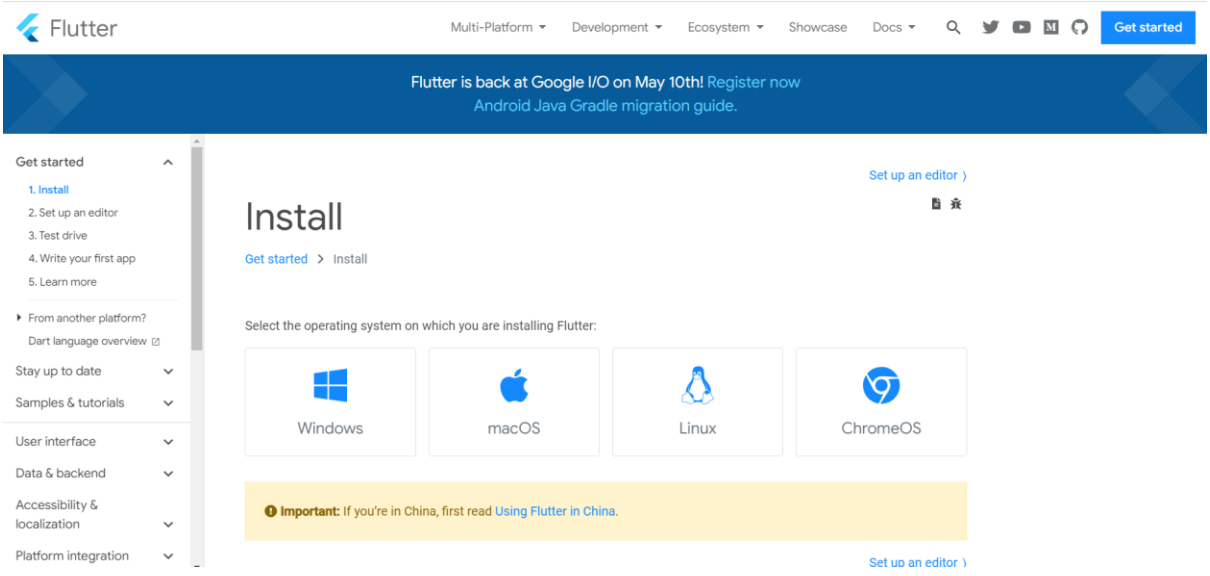


Figure 5.4: Installation of Flutter

Download the latest stable version of Flutter for operating system from the official Flutter website. After successfully download, extract the contents of the downloaded file into a directory.

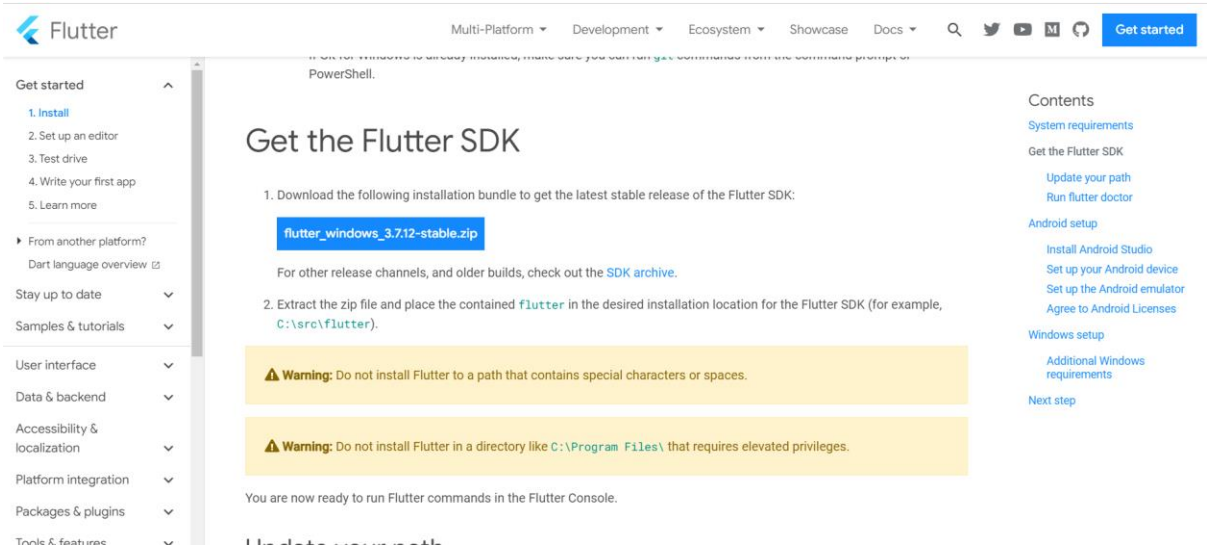


Figure 5.5: Installation of Flutter (cont.)

In the settings, search for “Edit the system environment variables”. Add the full path of the ‘flutter/bin’ directory to the PATH environment variable. Then, click “OK”.

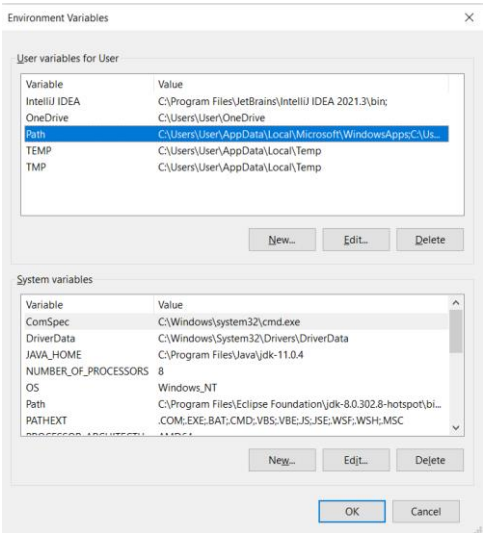


Figure 5.6: Installation of Flutter (cont.)

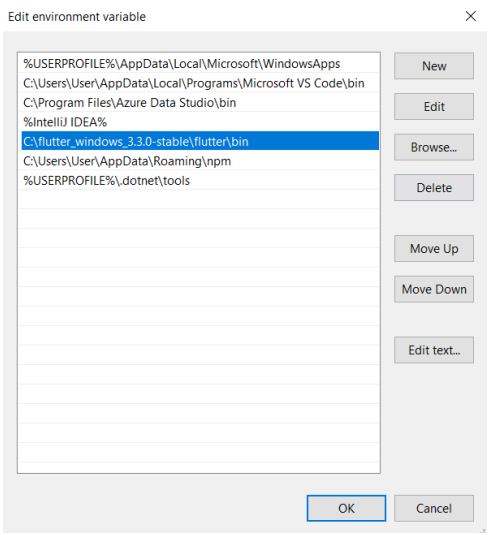


Figure 5.7: Installation of Flutter (cont.)

Open a terminal or command prompt and run the command ‘flutter doctor’. This command checks your environment and displays a report of the status of your Flutter installation, including whether any additional components are needed.

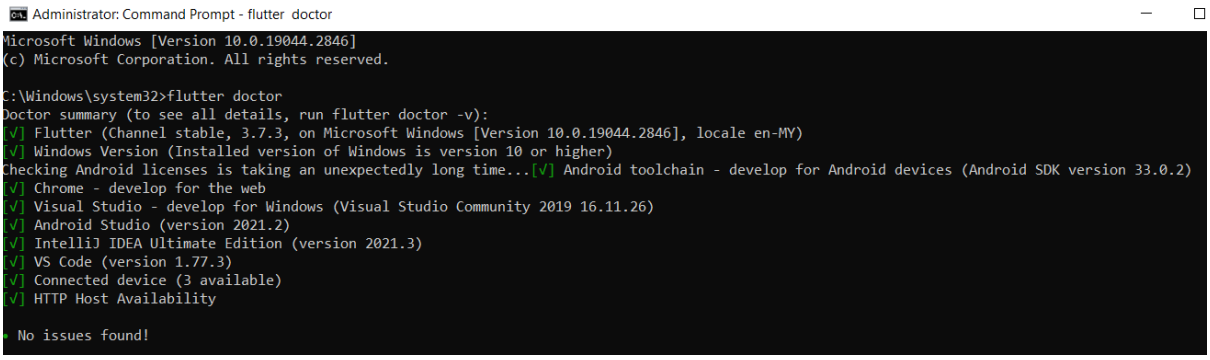


Figure 5.8: Installation of Flutter (cont.)

5.1.3 Creation of Google Firebase

Go to the Firebase website and sign in using Google account. Click on the "Get Started" button and click on the "Add project" button. Then, enter a project name and select country/region. Once the project is created, user will be redirected to the project dashboard.

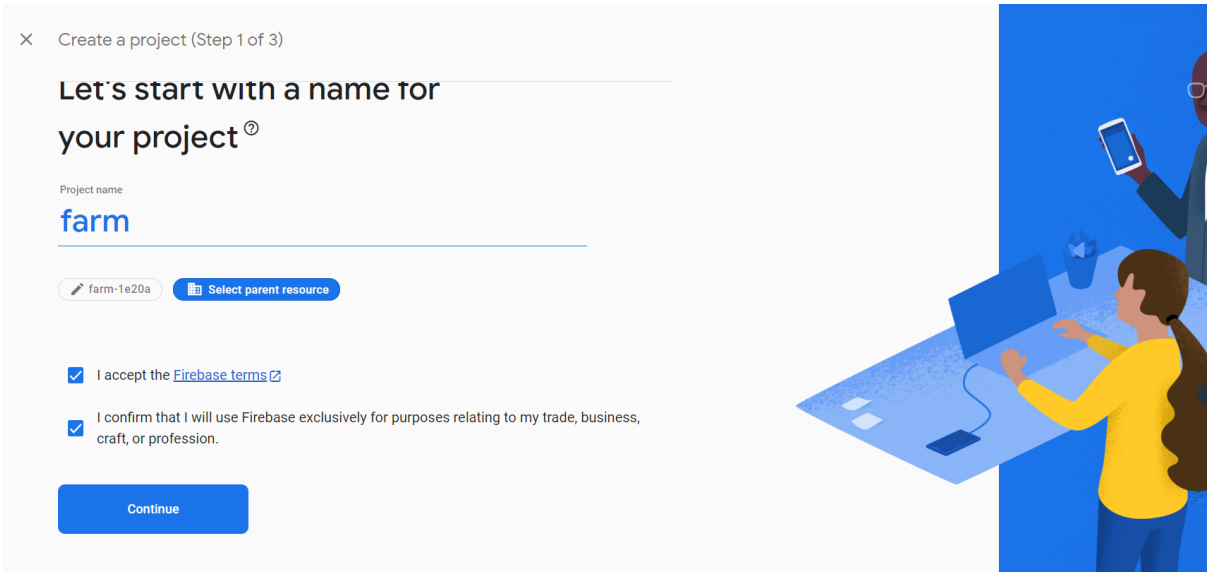


Figure 5.9: Creation of Google Firebase

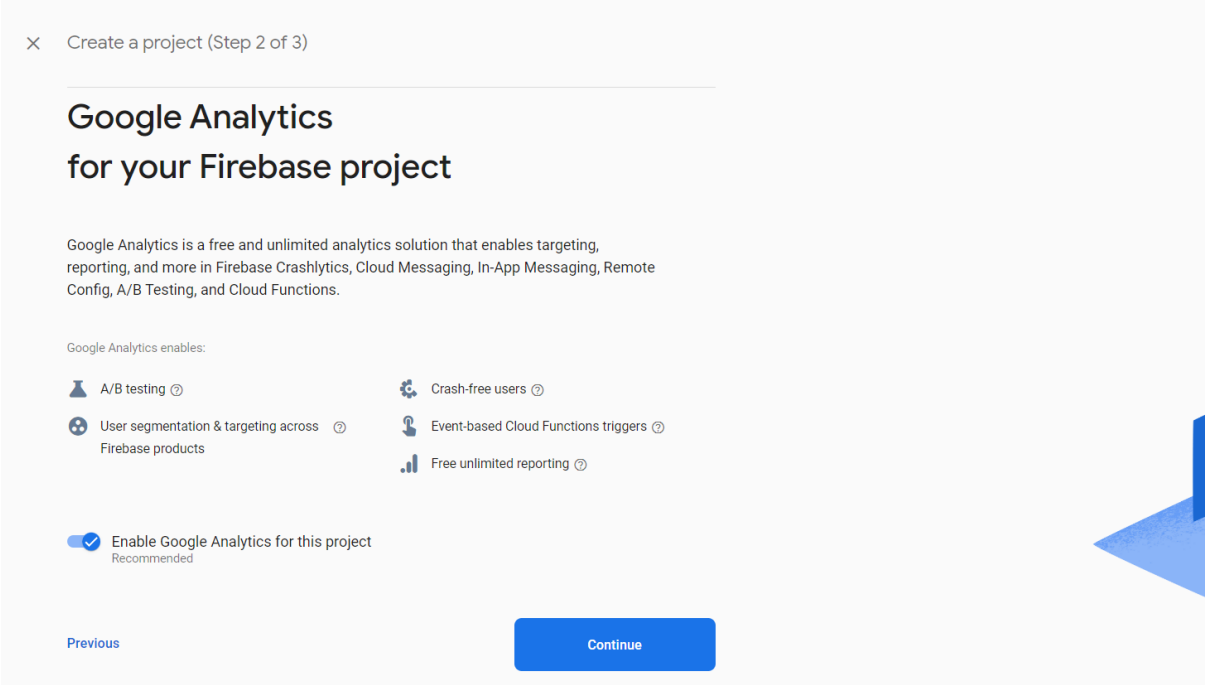


Figure 5.10: Creation of Google Firebase (cont.)

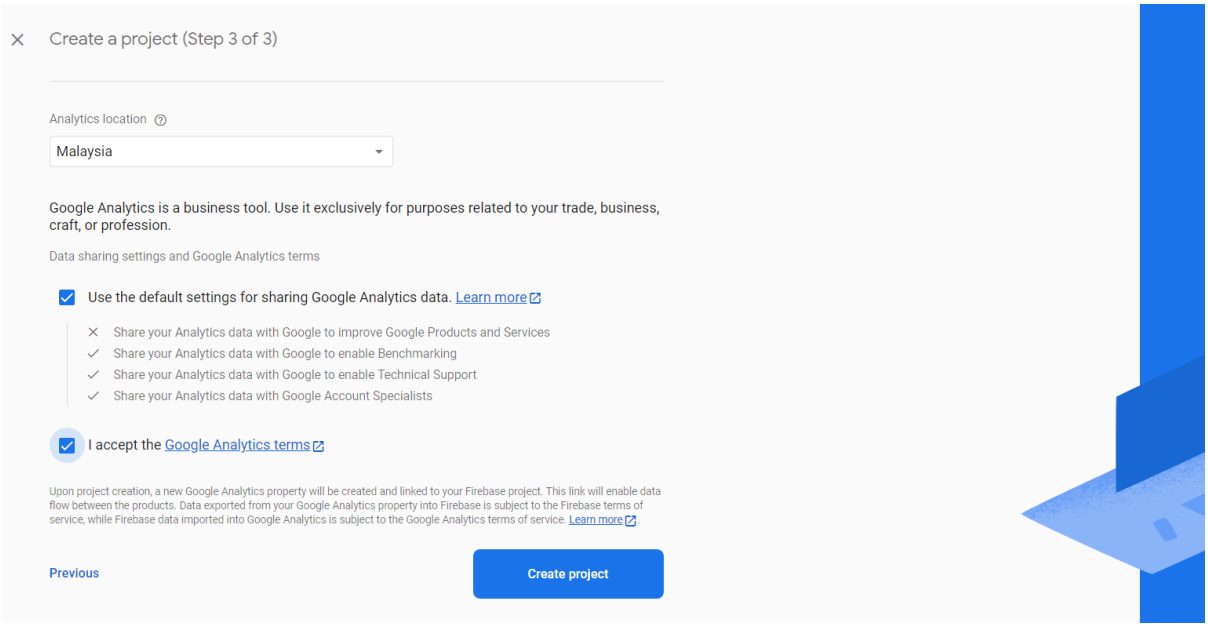


Figure 5.11: Creation of Google Firebase (cont.)

5.2 Setting and Configuration

5.2.1 Android Studio configuration

Open Android Studio. Select “New Flutter Project” button.

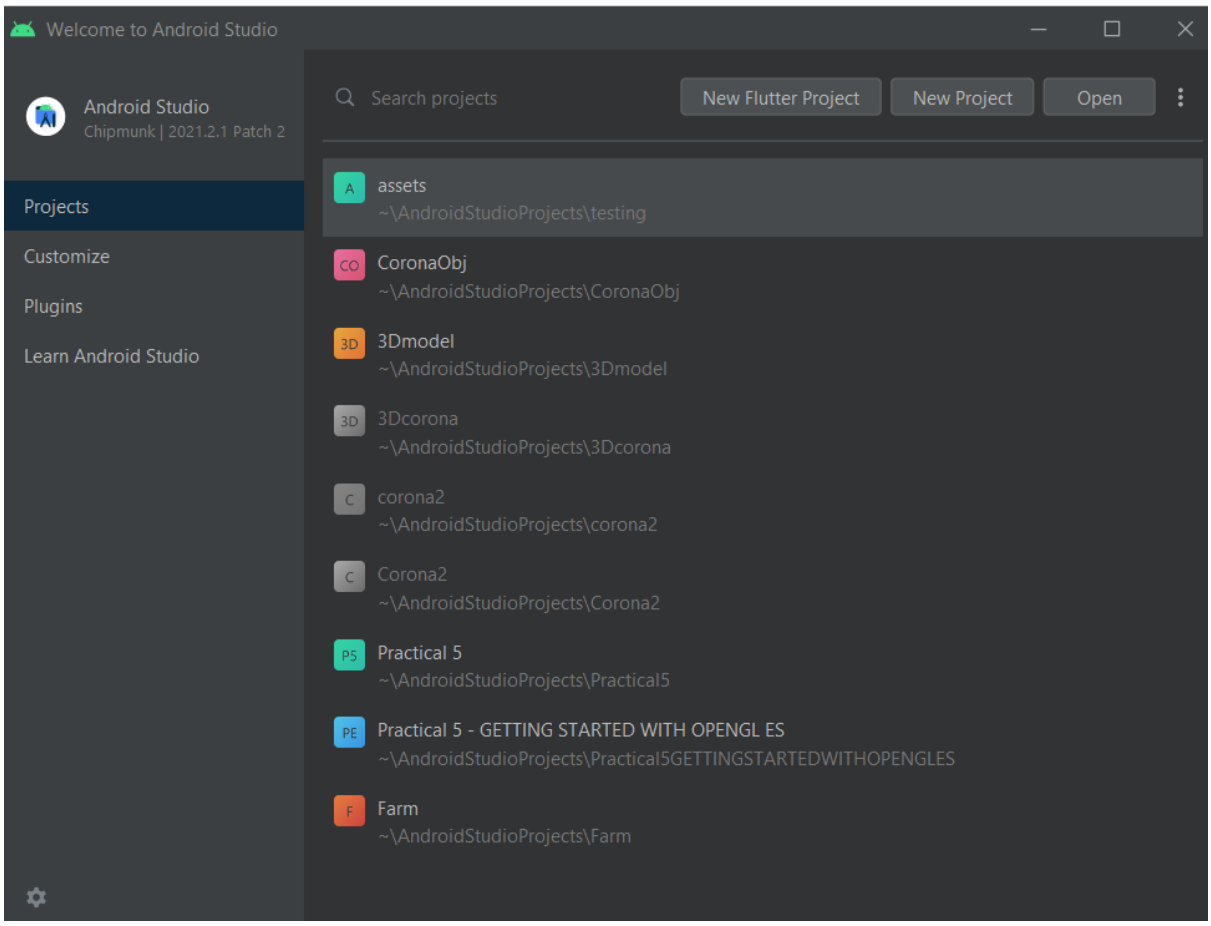


Figure 5.12: Android studio configuration

Enter project name. Click “Finish” button and the project is successfully created.

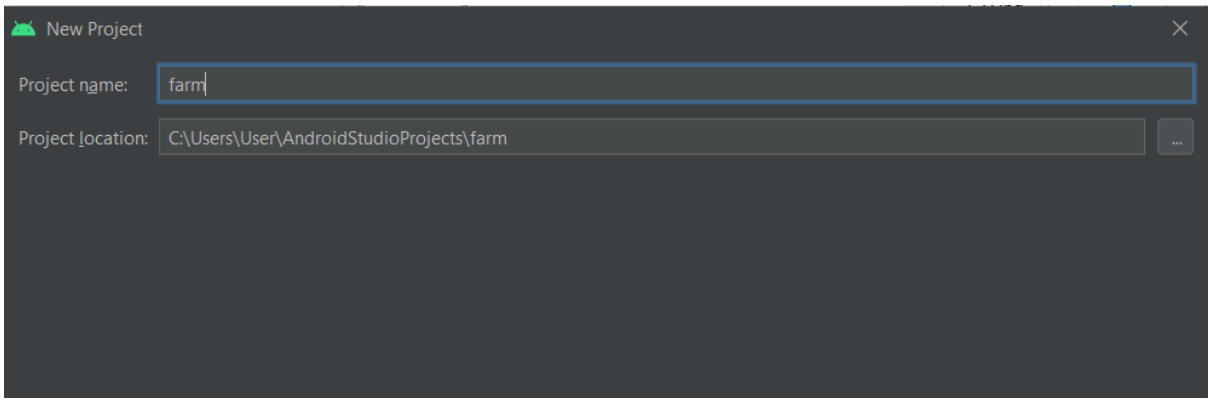


Figure 5.13: Android studio configuration (cont.)

5.2.2 Google Firebase configuration

Click on the "Add Firebase to your Android app" button.

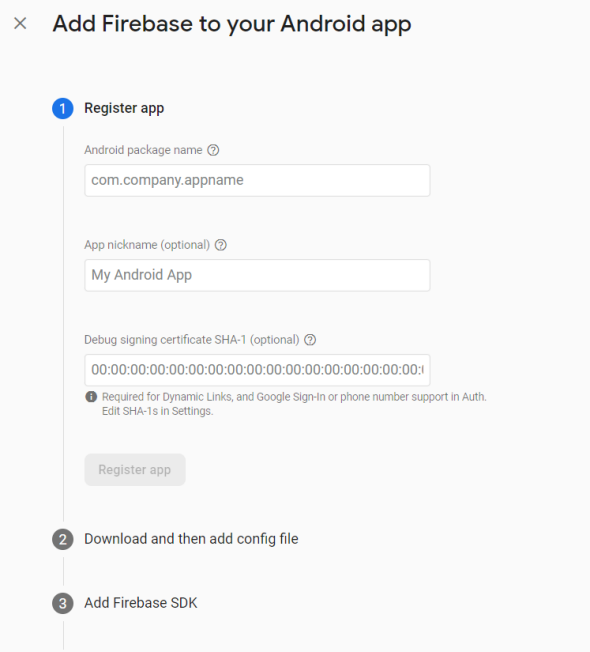


Figure 5.14: Google Firebase configuration

Download the ‘google-services.json’ configuration file.

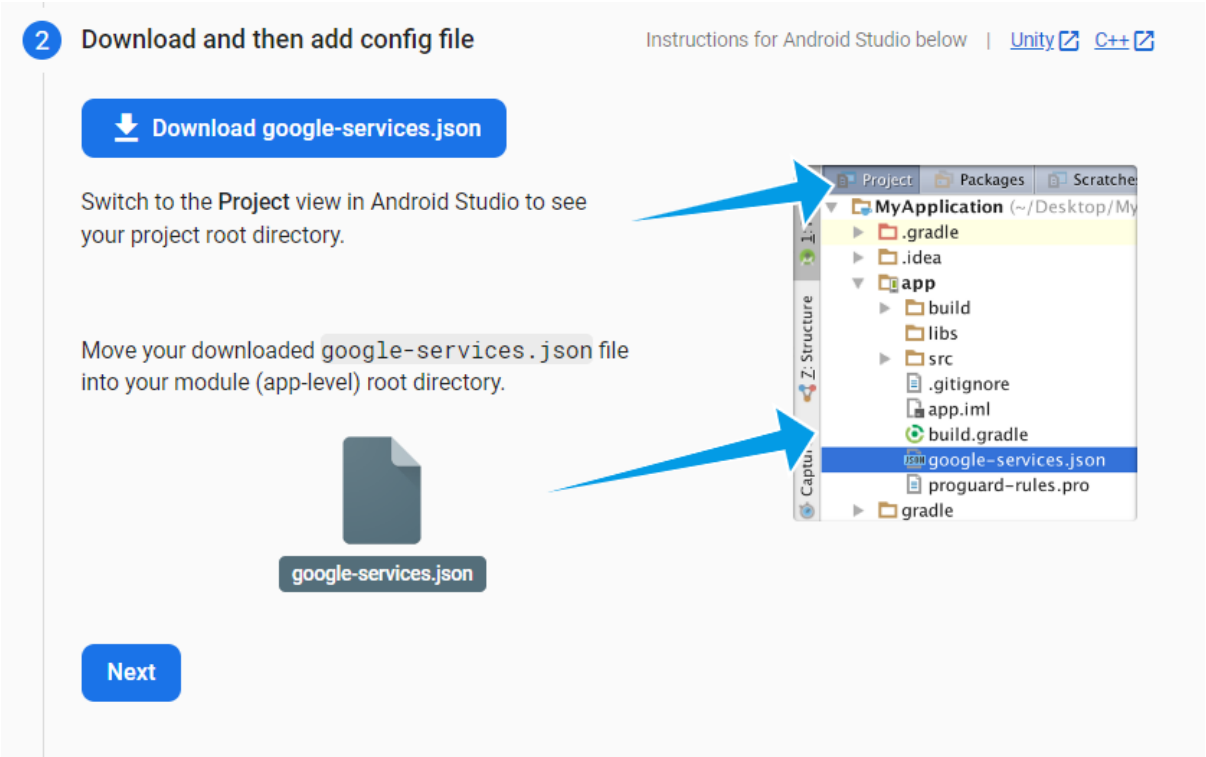


Figure 5.15: Google Firebase configuration (cont.)

Copy the ‘google-services.json’ file into the app folder of Android Studio project. Add the Firebase SDK to the app by adding the following dependencies to build.gradle file.

```
buildscript {
    ext.kotlin_version = '1.6.10'
    repositories {
        google()
        mavenCentral()
    }

    dependencies {
        classpath 'com.android.tools.build:gradle:7.1.2'
        classpath "org.jetbrains.kotlin:kotlin-gradle-plugin:$kotlin_version"
        classpath 'com.google.gms:google-services:4.3.13'
    }
}

allprojects {
    repositories {
        google()
        mavenCentral()
    }
}
```

Figure 5.16: Google Firebase configuration (cont.)

Initialize Firebase in app by adding the following code to application class or main activity.

```
// Import the Firebase SDK
import com.google.firebase.FirebaseApp;

// Initialize Firebase in your app
FirebaseApp.initializeApp(this);
```

Figure 5.17: Google Firebase configuration (cont.)

5.3 System Operation

5.3.1 Login/Register module

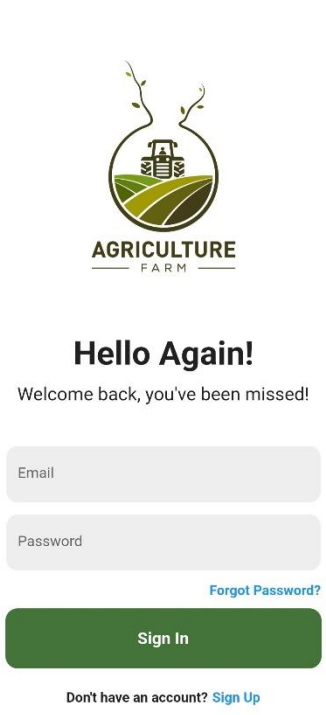


Figure 5.18: Login screen

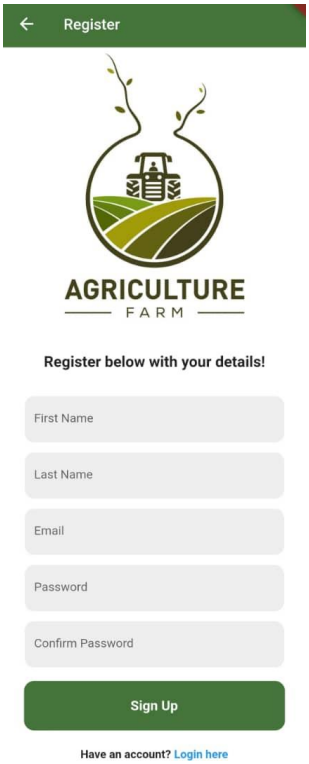


Figure 5.19: Register screen

The login/register module is an essential part of any application that requires users to create an account to access its features. In this module, the user can either log in using their existing credentials or create a new account by registering with the application. The registration process involves the user providing their basic details, such as their name, email address, and password, which the application will use to authenticate the user's identity. The user's information is then stored securely in the application's database using encryption methods to ensure that it is protected against unauthorized access. On the other hand, the login process allows the user to access the application's features by entering their registered email address and password. The application then verifies the user's identity and grants access to the features if the user's details match the information stored in the database.

5.3.2 Crop planning module

Figure 5.20: Google Map screen

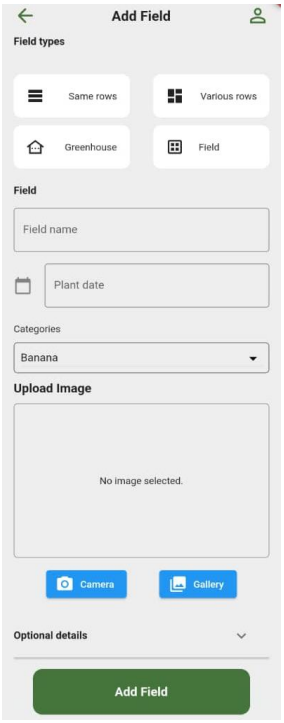


Figure 5.21: Add Field screen

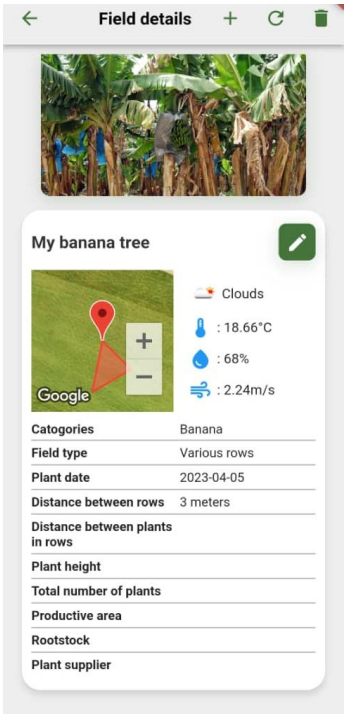


Figure 5.22: View Field Screen

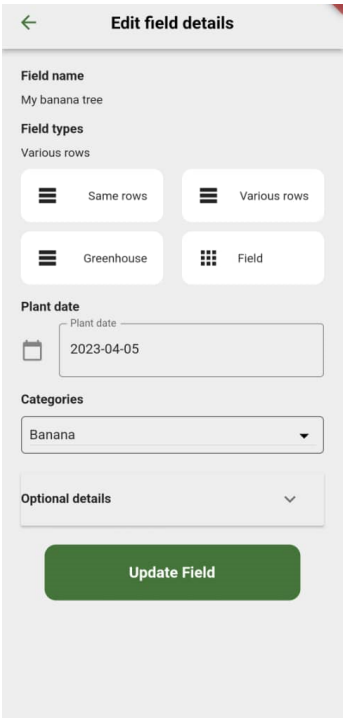


Figure 5.23: Update Field Screen

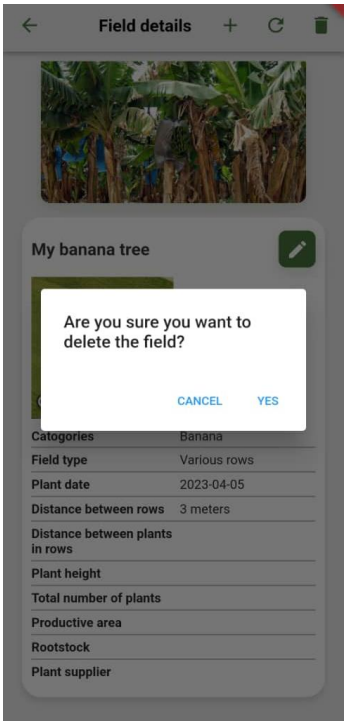


Figure 5.24: Delete Field Screen

Crop planning module includes several screens to help farmers add, view, update, and delete fields. The Google Map screen is the first screen of the Crop Planning module, where farmers can view their farm on a map and plot the boundaries of their fields. This screen is essential for ensuring accurate planning and management of crops, as farmers need to know the exact location and size of their fields. The Add Field screen is where farmers can add a new field to their farm. This screen requires farmers to input information such as the field's name, field's details, and type of crop to be grown. The View Field screen allows farmers to view their fields and their details such as the size, location, and crop type and crop details as well as weather forecast data. The Update Field screen allows farmers to make changes to their existing fields. Farmers can update information such as the field name, crop type and crop details. These changes are saved to the system's database and can be accessed later. The Delete Field screen allows farmers to delete fields they no longer need. This screen is particularly useful when farmers need to make changes to their farm's layout, and fields are no longer needed.

5.3.3 Job assignment module

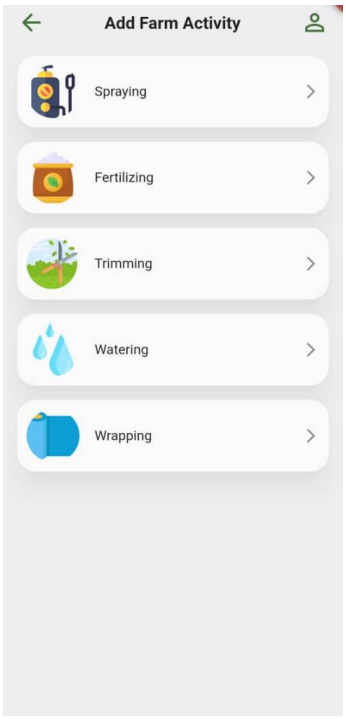


Figure 5.25: Farm Activity Screen

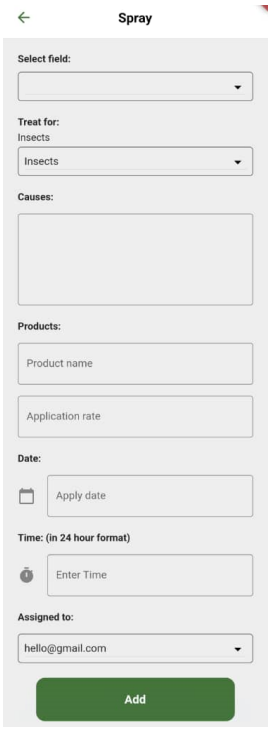


Figure 5.26: Add Farm Activity Screen

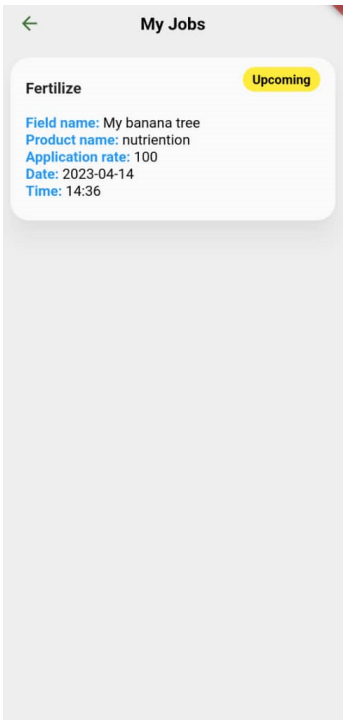


Figure 5.27: View Farm Activity Screen

The job assignment module in the farm management system allows users to assign various types of farm activities to specific fields. Users can access the Farm Activity

screen to view a list of five different types of farm jobs, such as fertilizing, watering, spraying, wrapping and trimming. From there, user can select a specific field and add a job activity, which will take them to the Add Farm Activity screen. On the Add Farm Activity screen, users can input details about the job activity, including the type of activity, the reason or cause for performing the job, the products needed to complete the job, and the amount of product required. Users can also specify the start and end dates for the job activity. Once a farm job has been added, users can access the View Farm Activity screen to see all the upcoming farm jobs that have been assigned to specific fields. This screen can also be sorted by date to help users plan and prioritize their farm activities. Users can select a job from the list and update their job progress by clicking on it. When they do so, a dialog box pops out, allowing them to update their progress.

5.3.4 Growth state tracking module

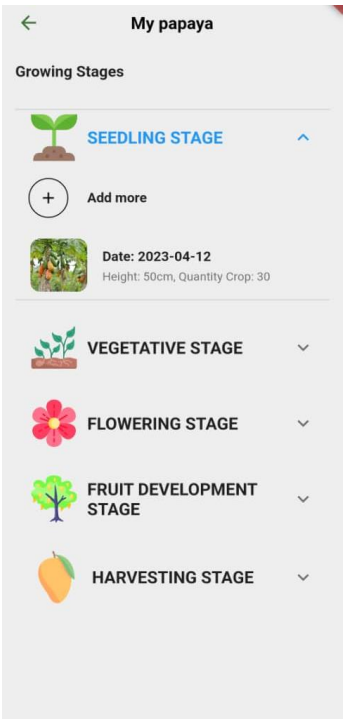


Figure 5.28: View Plant Growth screen

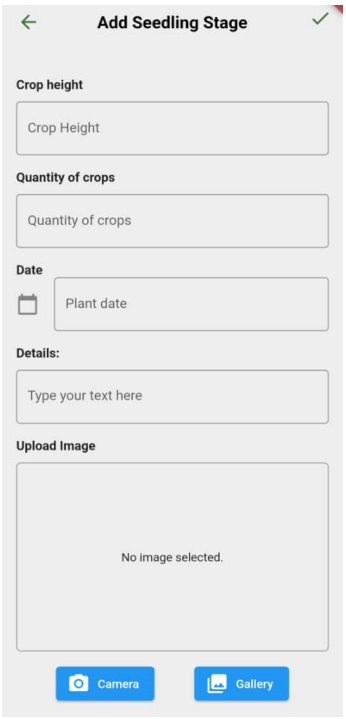


Figure 5.29: Add Plant Growth screen

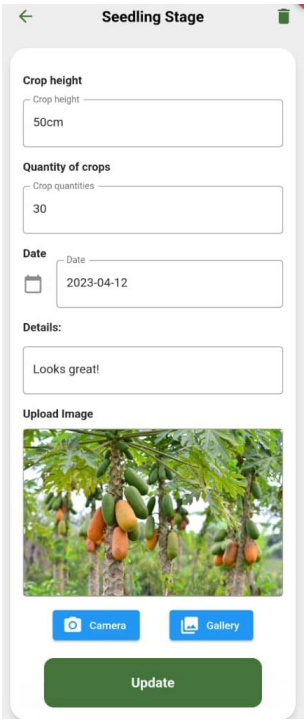


Figure 5.30: Update Plant Growth screen

The Growth State Tracking module is an important part of the farm management system as it allows the user to track the growth of their crops. The View Plant Growth screen shows the different growing stages of each crop and allows the user to expand the tiles to see all the data under each growing stage category. This makes it easy for the user to keep track of the growth of their crops and identify any issues that may arise. The Add Plant Growth screen allows the user to add details about the growth of their crops, such as the crop height, quantity of crops, date, details, and photo. This information is important for tracking the growth of the crops and identifying any issues that may need to be addressed. The user can add this information as frequently as they want, such as daily or weekly, to keep a continuous record of their crop's growth. The Update Plant Growth screen allows the user to update the details of their crops as they grow. This could include changing the crop height, adding more detail, or uploading a new photo. The user can update this information as frequently as they want to ensure that their records are accurate and up to date. Besides, user can delete the plant growing state by selecting the delete button.

5.3.5 Pest and disease detection module

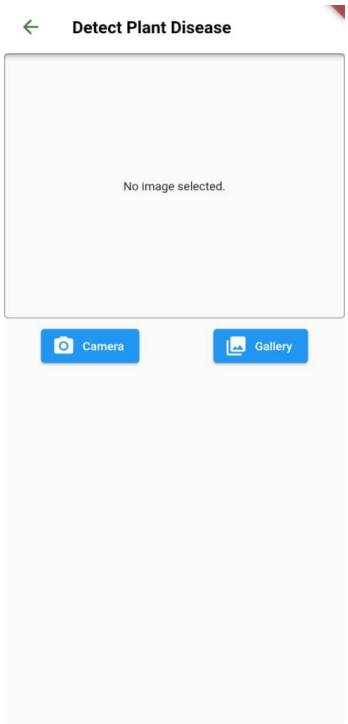


Figure 5.31: Detect disease screen

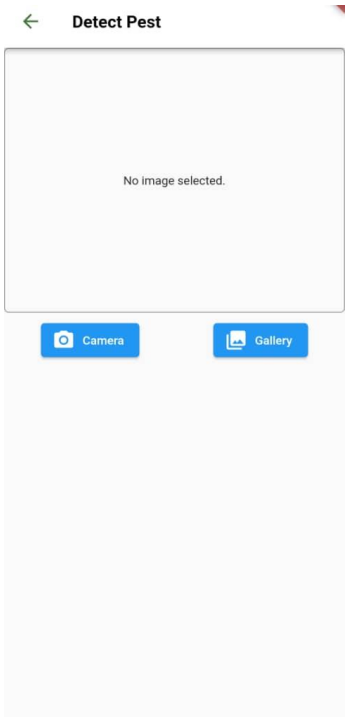


Figure 5.32: Detect pest screen

The Pest and Disease Detection module utilizes an API to detect and identify pests and diseases in crops. The module assists farmers in diagnosing crop issues early on, preventing yield losses and reducing the need for harsh chemicals. Using the API, the system can detect the presence of diseases in crops and provide users with detailed information, such as the name of the disease, a description of the disease, and a treatment plan that includes biological and chemical control measures and prevention tips. This information can be extremely useful in preventing the spread of the disease and managing its effects on crops. The system can also detect pests in crops and provide users with information such as the scientific name of the pest, common name, and URL of the pest. This information can help farmers identify the specific pest affecting their crops and determine the best course of action for control. By providing early detection and identification of pests, the system helps farmers reduce the potential for crop damage and control pests before they become a severe problem.

5.4 Implementation Issue and Challenges

There are few challenges faced during implementation phase in this proposed project. The first challenge is the failure of operating system. A system failure might be caused by a hardware failure or software problem. For example, code and run the program for many hours using laptop without letting it have a break. Therefore, this will cause laptop overheating and it can slow down the entire system and causes frequent crashes. Besides, I used firebase to manage database. Firebase is a web-based software tool which runs only on the browser; hence it completely depends on browser. There might be a problem where the database server is down or error when establishing a database connection. Thus, it may cause the data unable to upload correctly into the database system. As a result, inaccurate data will result in inaccurate field records. In addition to the challenges mentioned, another implementation issue that can be encountered is difficulty in finding a suitable API for the pest and disease detection. While there are many APIs available, some of them can be costly and may not fit within the project budget. Moreover, even if an API is available for free, it may not provide all the necessary functionalities required for the project.

Chapter 6

System Evaluation and Testing

6.1 Testing Result

6.1.1 Login/Register module

Login/Register

Figure 6.1: Register an account

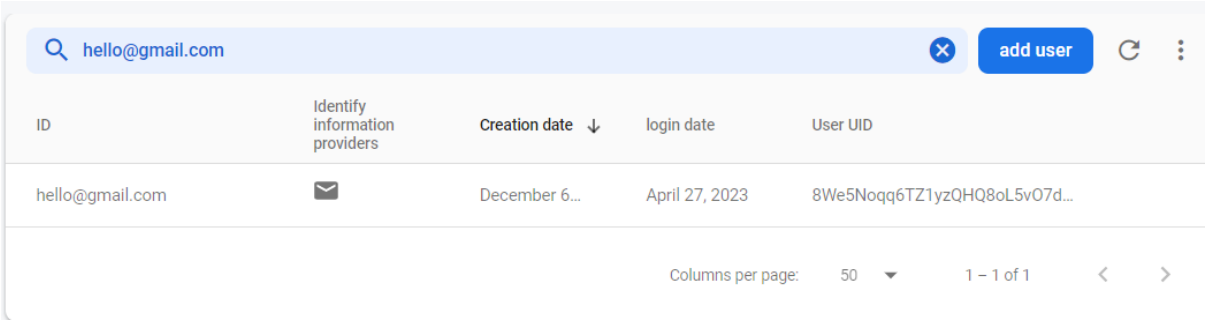


Figure 6.2: Successfully register an account

The login/register module is the first point of contact for the user to access the system. To ensure that this module works as expected, testing is a crucial step in the development process. In the testing of the registration feature, the first step is to fill out the required user details. This includes the user's first and last name, email address,

password, and confirmation of the password. Once all the necessary details are filled out, the next step is the system will check if all the data has been entered correctly. This ensures that the user is not missing any vital information and that the system can save the user's data correctly. After the user submits their details, the data should be stored securely in the Firebase Authentication system.

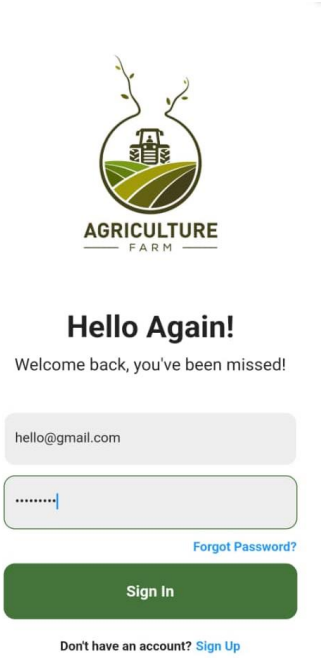


Figure 6.3: Login an account

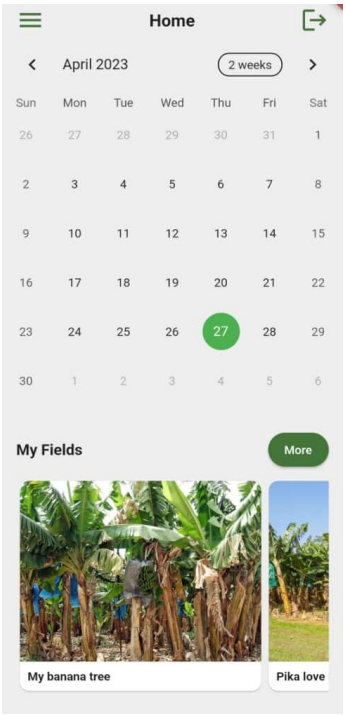


Figure 6.4: Main Page

In the testing of the login feature, the user needs to enter their registered email address and password. The system then authenticates the user's details, checks if the entered email address is valid and matches with the registered email address in the system, and also checks if the entered password matches with the registered password. Once the system confirms that the entered credentials are valid, the user is redirected to the application's home page.

6.1.2 Crop planning module

Add field

Figure 6.5: Plot a polygon on Google Map

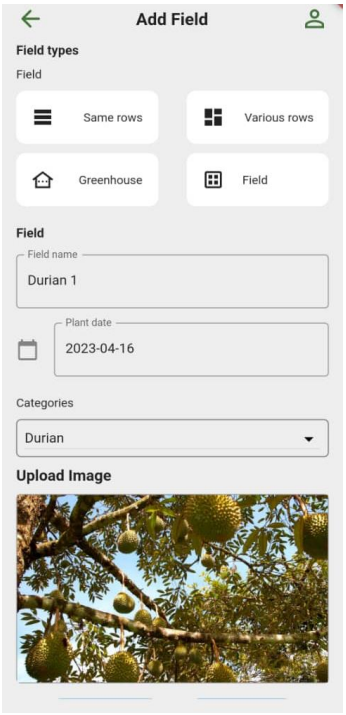


Figure 6.6: Add Field details

During the testing of the crop planning module, the add field feature was thoroughly tested. The user interface allowed users to search location and zoom in and out the Google Maps to plot their fields on a map and fill in the necessary field details. The module was found to be functional, and the data was saved correctly in Firebase. Additionally, the undo and clear buttons worked correctly, which allowed the user to make any necessary corrections while plotting the field. Once the user had finished adding the field, a “Polygon saved successfully” message was displayed, and the user was redirected to the main page. This feature allowed users to quickly and efficiently add their fields without any unnecessary delays or complications. In figure 6.7, the figure shows the data was successfully stored in the database.

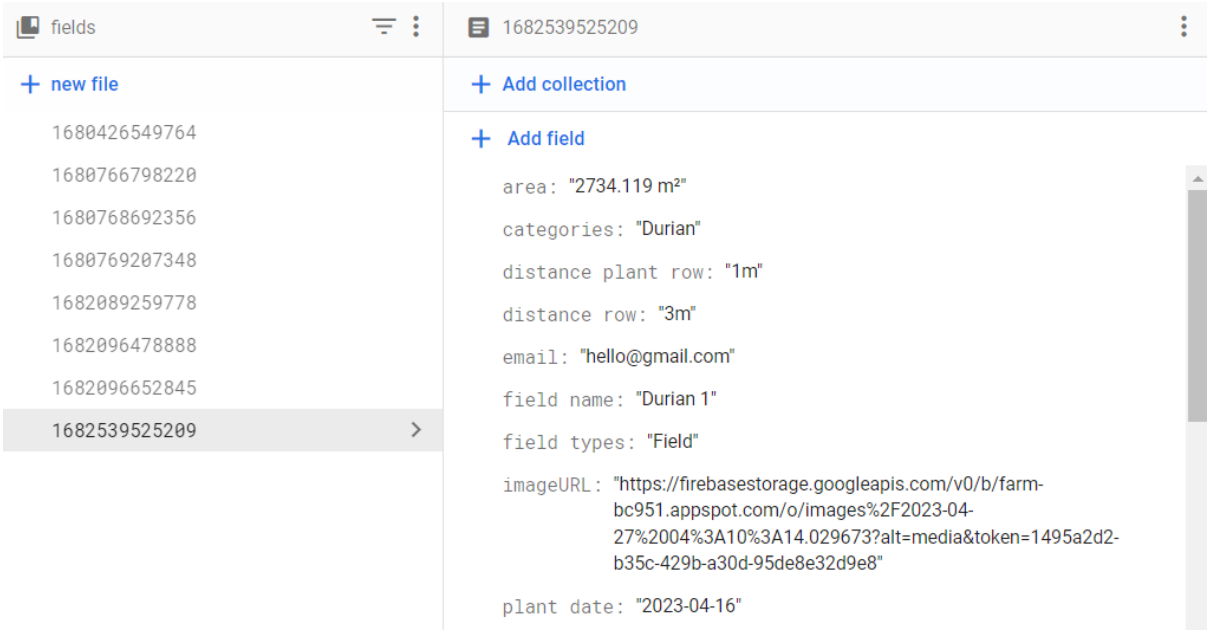


Figure 6.7: Successfully add Field details

Update field

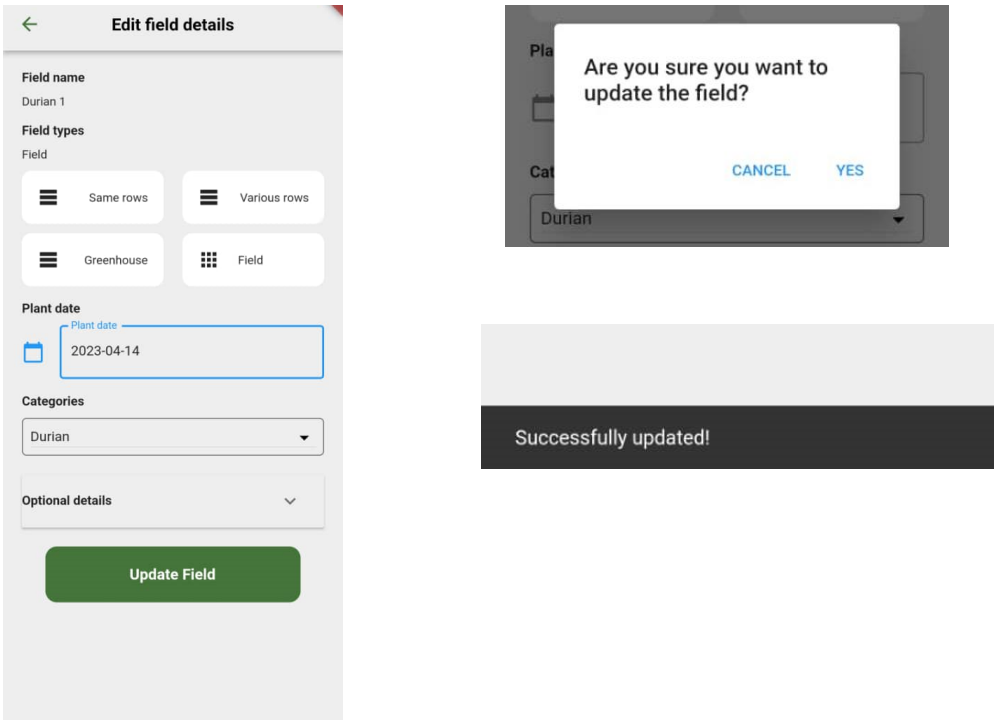


Figure 6.8: Update field

In the crop planning module, the update field functionality is tested. When the user clicks on the update field button, a dialog message appears to confirm whether they want to update the field. The user can choose to either update or cancel the action. Both options are tested and ensured that the data was correctly filled in before proceeding. If

the user selects "yes" to update, the system will update the field and display a message indicating that the update was successful. The user is then redirected back to the fields list page, where they can view the updated information. During testing, error messages are checked in case the user does not fill in the required data or if there is any other error while updating the field. Overall, the update field functionality was successfully tested, and did not encounter any major issues or errors. The system was able to update the field details correctly, and the user was able to view the updated information in the fields list page.

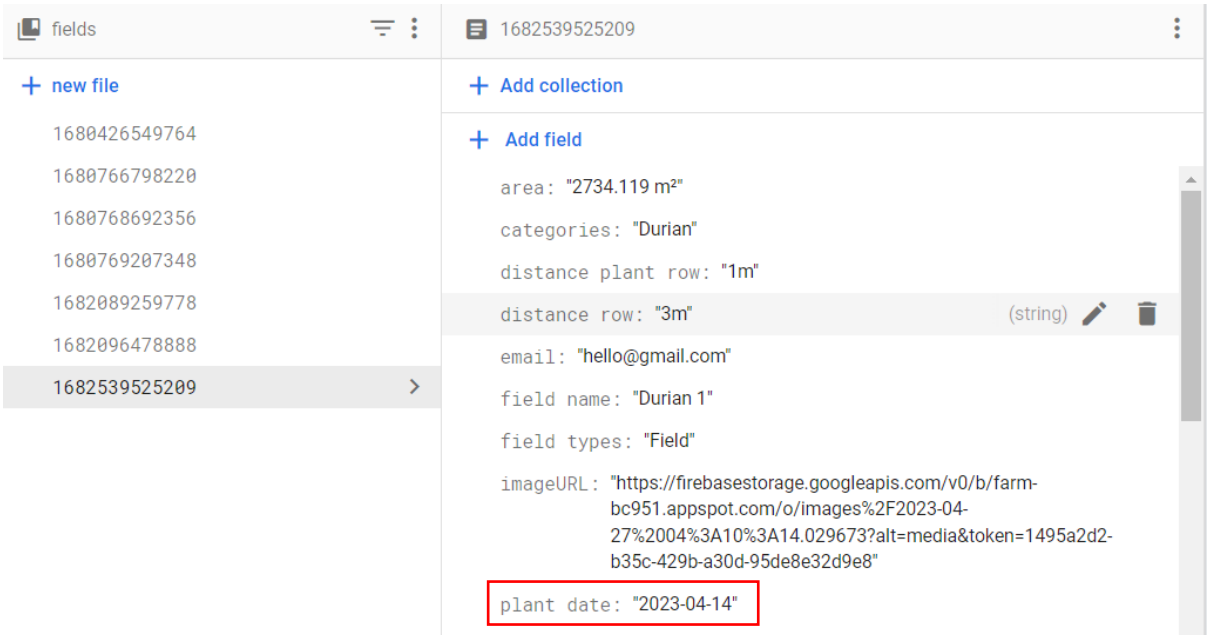


Figure 6.9: Successfully update field

Delete field

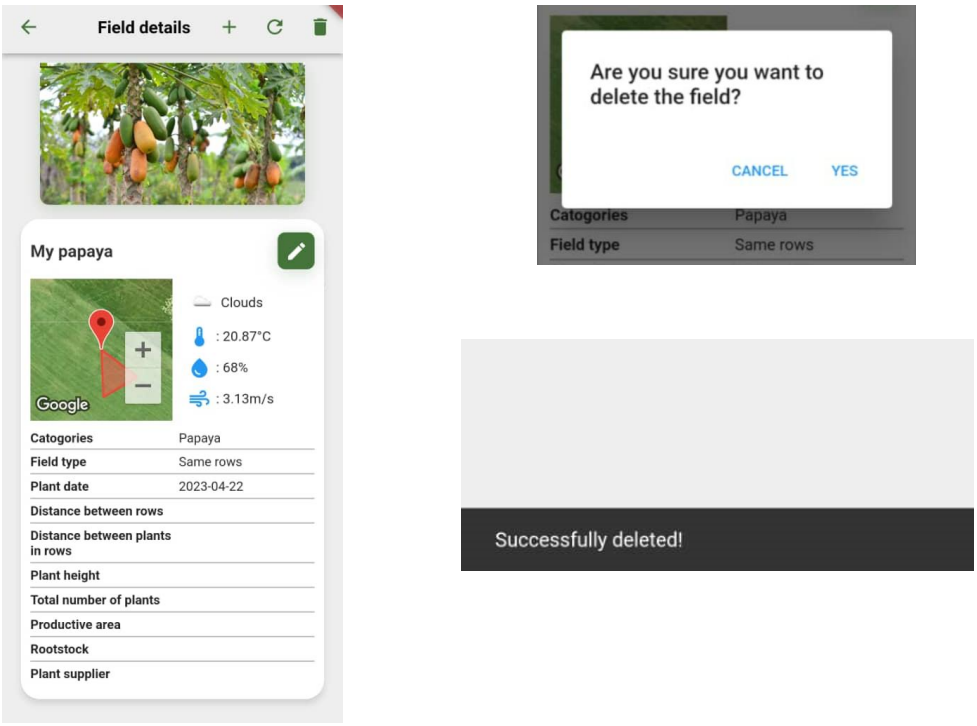


Figure 6.10: Delete field

The delete field feature was tested. The user is able to select a specific field that they want to delete, and a pop-up dialog box will appear to confirm the deletion. If the user presses the "delete" button, the system will prompt them once again to confirm if they want to delete the field or not. If the user confirms the deletion, the field will be removed from the database and a “Successfully deleted” message will be displayed. Testing this feature ensures that the user is able to manage their field list effectively, and users are able to remove any unwanted or unused fields from the list. The pop-up confirmation box is a useful feature as it helps prevent accidental deletion of fields, which could result in data loss. Additionally, the success message after deleting a field provides immediate feedback to the user, indicating that the field has been successfully removed from the system.

6.1.3 Job assignment module

Add farm activity

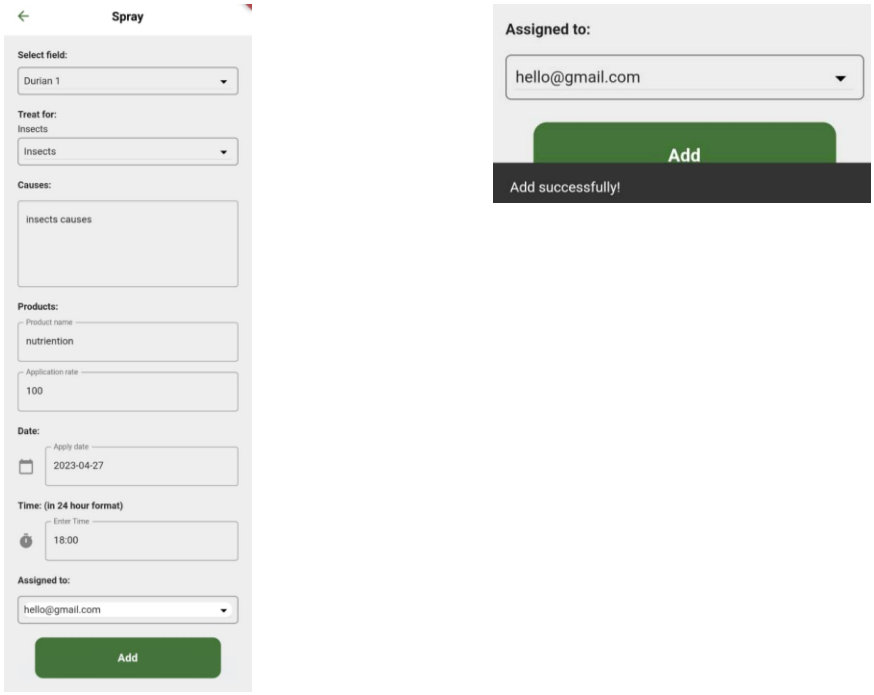


Figure 6.11: Add farm activity

In spraying page, users can choose an option to treat for the crop which are insects, diseases, weeds, nutrition and others. Besides, users can fill in the product name, application rate, date, time and the person they want to assign to. If users fill in all the required field in the spraying page, all the information will be stored in the database. The application will display “Add successfully!” message using Scaffold Messenger. In figure 6.12, the data are successfully store in database.

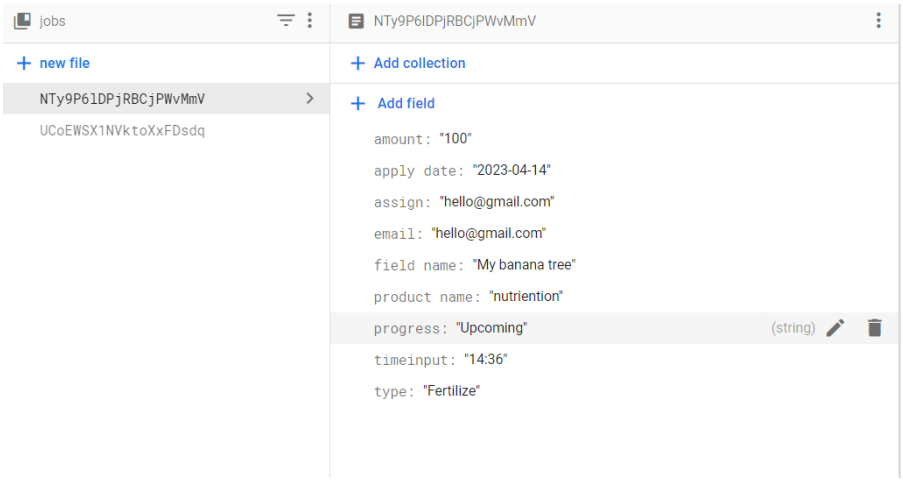


Figure 6.12: Successfully add farm activity

View farm activity

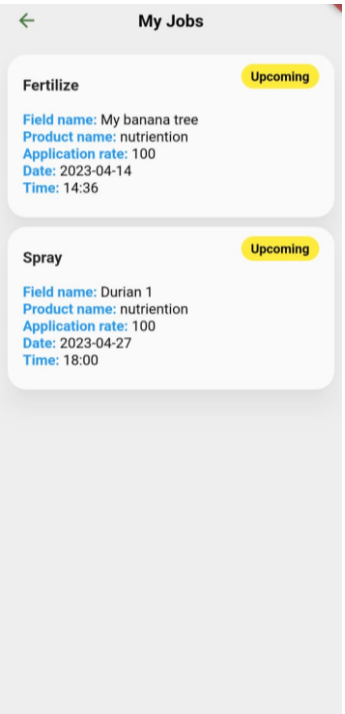


Figure 6.13: View farm activities

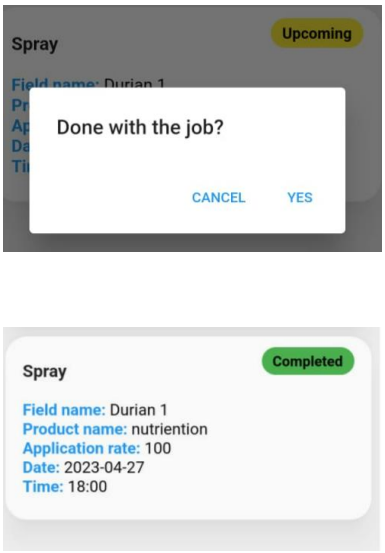


Figure 6.14: Update progress

The testing result for the job assignment module shows that the View Farm Activities screen is displaying the relevant information regarding different types of jobs, their progress, associated field name, product name, application rate, date and time. The user can easily view all upcoming jobs, which are shown in yellow colour. Moreover, the Update Progress feature works as expected, where the user is prompted with an alert box asking "Done with job?". If the user selects "Yes", the progress of the job is updated from "Upcoming" to "completed". On the other hand, if the user selects "No", the progress remains "Upcoming". The completed jobs are displayed in green color, which makes it easier for the user to track the progress of the jobs. The ability to update the job progress allows the user to keep track of the work done on the farm.

6.1.4 Growth state tracking module

View plant growth

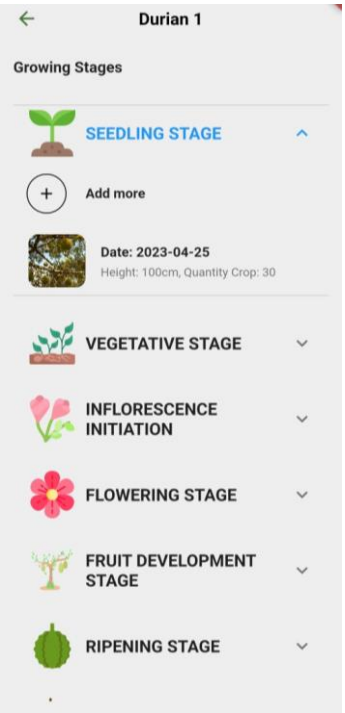


Figure 6.15: View plant growth stages

View Plant Growth screen allows users to track the progress of their crops through different growing stages, which are displayed based on the crop type. Users can select the desired stage by clicking on the corresponding expansion tile, which will display information such as the date, height, quantity, and a photo of the crop at that stage. This feature provides farmers with a visual representation of their crop's development and helps them make informed decisions about their farming practices.

Add plant growth

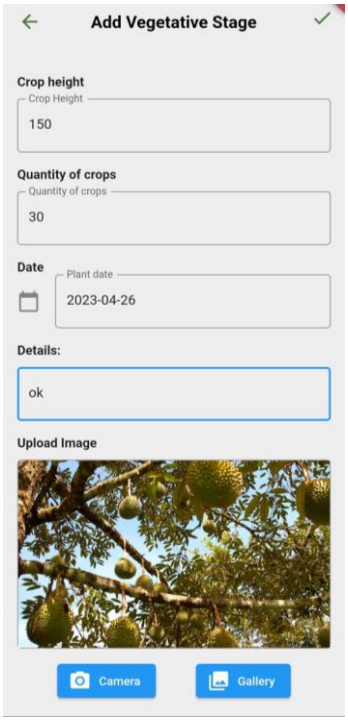


Figure 6.16: Add plant growth stages

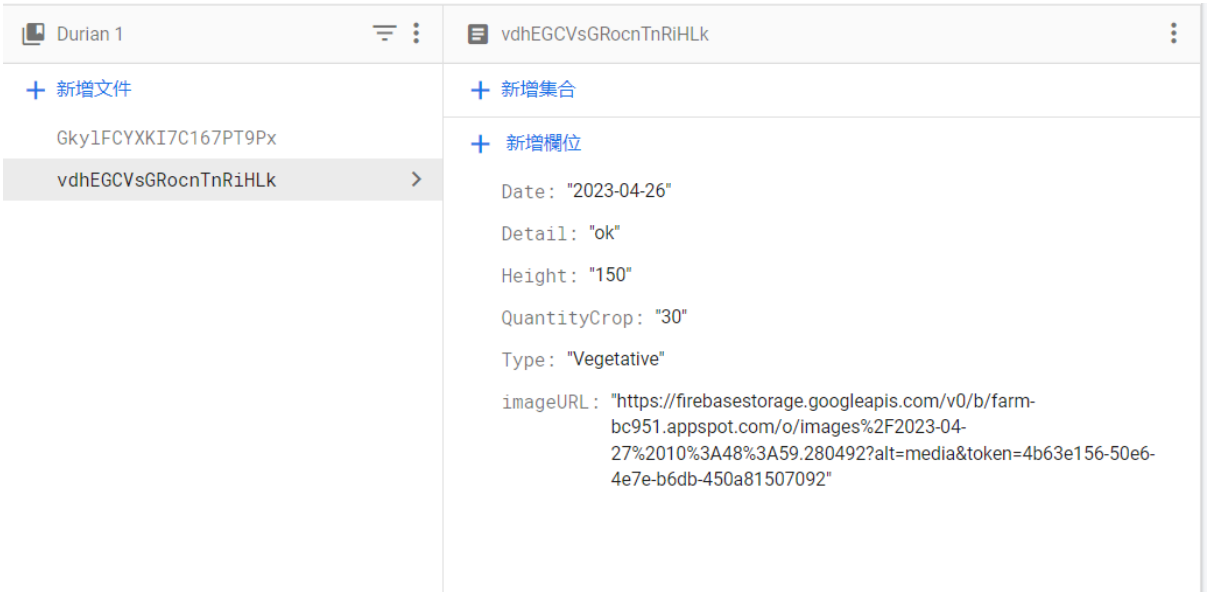


Figure 6.17: Successfully add plant growth stages

During testing of the Growth state tracking module, it is able to successfully add plant growth data using the "Add plant growth" screen. The details are entered such as the crop's height, quantity, date, and additional notes, as well as uploaded a photo to provide visual documentation of the crop's growth. After entering this information, we were

able to see the output from Firebase in Figure 6.17, confirming that the data had been successfully stored in the database. When we returned to the "View plant growth" screen, we saw that the new data was displayed within the appropriate expansion tile, which represented the corresponding growth stage for the crop.

Update plant growth, Delete plant growth

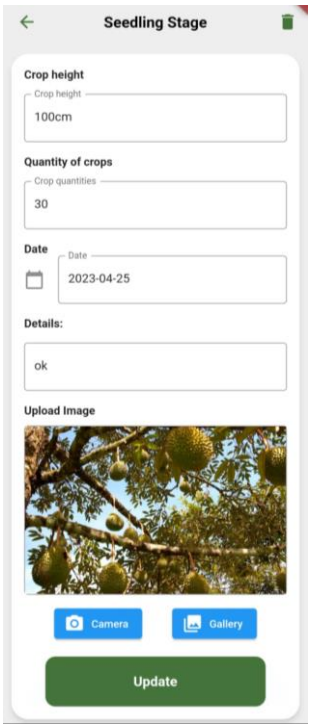


Figure 6.18: Update, delete plant growth details

In the growth state tracking module, the update functionality for plant growth information is tested. Users are able to update various details such as photos, dates, heights, and crop quantities for a specific growth stage. When the user clicks on the update button, a pop-up window appears asking if they are sure they want to update the data. If the user clicks "yes", the data is updated in the Firebase. After the update is successful, the updated data is displayed in the expansion tile for that growth stage. This allows users to easily see the progress of their crops over time and track any changes that have been made. The update functionality is important for keeping accurate records and making informed decisions about crop management. Overall, the testing of the update functionality in the growth state tracking module was successful and demonstrated the usefulness of this feature for farmers.

6.1.5 Pest and disease detection module

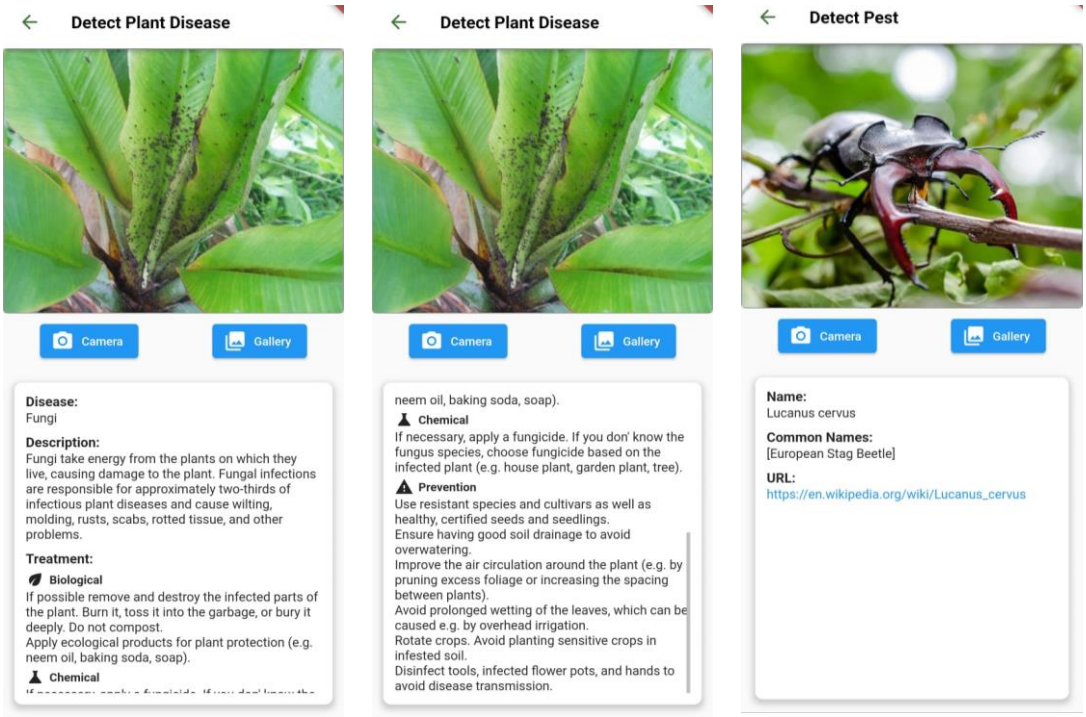


Figure 6.19: Detect disease and Pest

In the disease detection and pest module, the user can upload a photo from the camera or gallery to identify any plant diseases or pests. This is done by using the plant.id API, which retrieves the disease name, description, and treatment options. The treatment options include biological and chemical controls, as well as prevention methods. This information is valuable for farmers to manage their crops effectively and prevent further damage. Similarly, the module also uses an API to retrieve information on pests. The API provides the name of the pest, its common names, and a URL to access further information on the pest. The user can click on the URL to copy it and access more information on the pest.

Chapter 7

Conclusion and Recommendation

7.1 Conclusion

In conclusion, the Android-based farm management system presented in this project is a solution to the possible limitations and problems of the existing farm management systems. The project aims to provide a convenient, user-friendly, and efficient platform for farmers to manage their crops and increase their productivity. The system focuses on five types of crops, namely banana, durian, mango, palm oil, and papaya. These crops were selected based on their importance in the agricultural industry and their high demand in the market. The system allows farmers to plan their crops, assign jobs to their workers, track the growth of their plants, and detect pests and diseases that could harm their crops. The crop planning feature of the system enables farmers to add and update information about their fields, including the crop type, crop details, and planting date. The system also allows farmers to view the progress of their jobs, update the progress of their jobs. The job assignment feature of the system allows farmers to assign jobs to their workers and track their progress. The system displays a list of job types, progress, field name, product name, application rate, date, and time. Farmers can update the progress of their jobs and mark them as completed when they are done. The plant growth tracking feature of the system enables farmers to track the growth of their plants by adding and updating information about their growth stages, including photos, dates, heights, and crop quantity. The system also provides a list of the growing stages of each crop type, and farmers can select the stages by selecting the expansion tile. The pest and disease detection feature of the system uses an API from plant.id to retrieve information about the pest or disease, including its name, description, and treatment options. This feature allows farmers to detect and treat pests and diseases before they can cause significant damage to their crops. Overall, the Android-based farm management system presented in this project provides a comprehensive and efficient solution for farmers to manage their crops effectively. It enables them to plan their crops, assign jobs to their workers, track the growth of their plants, and detect pests and diseases that could harm their crops. The system is user-friendly and accessible, making it an ideal tool for farmers who want to increase their productivity and profitability.

7.2 Future enhancement

There are several potential areas for future enhancement of the Android-based farm management system. One key area for improvement is the user interface, which can be made even more user-friendly and accessible to farmers of all skill levels. This could involve simplifying navigation and adding more intuitive features, such as voice commands or touchless controls. Another important area for future development is automation. By integrating the system with sensors that monitor temperature, soil moisture, and humidity levels, the farm management system can provide real-time data on crop conditions and help farmers make informed decisions about irrigation, fertilization, and other key factors. This would enable more precise and efficient crop management, potentially reducing waste and increasing yields. In addition to automation, the system could also be enhanced with notifications and warning messages. For example, the system could alert farmers to changes in weather patterns, potential pest or disease outbreaks, or other risks to their crops. This would enable farmers to take proactive measures to protect their crops and reduce the risk of losses. Finally, the system could be expanded to include additional crops beyond the five currently supported. By adding support for a wider range of crops, the system could be made more widely applicable and useful to farmers.

REFERENCES

[1] L. Brezinščak and M. Mesić, "Review of software applications for agricultural production in Croatia," Agron. glas. (Online), vol. 80, no. 2, pp. 129–142, 2019.

[2] "AGRIVI: Farm management software for digital agriculture," AGRIVI, 06-Jan-2015. [Online]. Available: https://www.agrivi.com/.

[3] Corteva.ca. [Online]. Available: https://www.corteva.ca/content/dam/dpagco/corteva/na/ca/en/files/temp/DF-Corteva-Field-Guide-West.pdf.

[4] "Farming for the future," Granular, 10-Jan-2018. [Online]. Available: https://granular.ag/.

[5] A. Hughes, "Work orders," Trimble Agriculture, 25-Feb-2019. [Online]. Available: https://agriculture.trimble.com/software/work-orders/.

[6] "Agrivi," PAT RESEARCH: B2B Reviews, Buying Guides & Best Practices, 27-May-2022. [Online]. Available: https://www.predictiveanalyticstoday.com/agrivi/.

[7] "Granular," PAT RESEARCH: B2B Reviews, Buying Guides & Best Practices, 08-Jun-2022. [Online]. Available: https://www.predictiveanalyticstoday.com/granular/.

[8] "Trimble," PAT RESEARCH: B2B Reviews, Buying Guides & Best Practices, 28-May-2022. [Online]. Available: https://www.predictiveanalyticstoday.com/trimble/.

[9] "What is farm management?," Smart Capital Mind, 13-Aug-2022. [Online]. Available: https://www.smartcapitalmind.com/what-is-farm-management.htm.

[10] "FarmLogs: Farm management software and apps," Farmlogs.com. [Online]. Available: https://farmlogs.com/.

[11] A. Chowdhry, "FarmLogs is now able to alert farmers about crop threats," Forbes, 18-May-2015. [Online]. Available: https://www.forbes.com/sites/amitchowdhry/2015/05/18/farmlogs-is-now-able-to-alert-farmers-about-crop-threats/?sh=7037eff257ad.

[12] Data driven Farm Management Software for all Farms," Agworld Farm Management Software. [Online]. Available: https://www.agworld.com/au/.

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: 3**
Student Name & ID: Leng Kai Yi 1902845	
Supervisor: Ts Tan Teik Boon	
Project Title: Farm Management Information System (Crop Planning and Tracking modules)	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Finish developing crop planning module.

2. WORK TO BE DONE

- Change user interface and modify job assignment module.

3. PROBLEMS ENCOUNTERED

- No problem at this stage.

4. SELF EVALUATION OF THE PROGRESS

- Progress is on track.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: 5**
Student Name & ID: Leng Kai Yi 1902845	
Supervisor: Ts Tan Teik Boon	
Project Title: Farm Management Information System (Crop Planning and Tracking modules)	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Finish developing job assignment module.

2. WORK TO BE DONE

- Identify the different growing stage of the crop.

3. PROBLEMS ENCOUNTERED

- No problem at this stage.

4. SELF EVALUATION OF THE PROGRESS

- Progress is on track.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: 7**
Student Name & ID: Leng Kai Yi 1902845	
Supervisor: Ts Tan Teik Boon	
Project Title: Farm Management Information System (Crop Planning and Tracking modules)	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Finish identifying the different growing stage of the crop.

2. WORK TO BE DONE

- Start developing plant growth tracking module.

3. PROBLEMS ENCOUNTERED

- Difficult to find bugs and errors.

4. SELF EVALUATION OF THE PROGRESS

- The problem encountered are manageable.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: 9**
Student Name & ID: Leng Kai Yi 1902845	
Supervisor: Ts Tan Teik Boon	
Project Title: Farm Management Information System (Crop Planning and Tracking modules)	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Finish developing plant growth tracking module.

2. WORK TO BE DONE

- Find suitable API for the pest and disease module.
- Start developing pest and disease module.

3. PROBLEMS ENCOUNTERED

- API requires fee or subscription to access.

4. SELF EVALUATION OF THE PROGRESS

- The problem encountered are manageable.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: 11**
Student Name & ID: Leng Kai Yi 1902845	
Supervisor: Ts Tan Teik Boon	
Project Title: Farm Management Information System (Crop Planning and Tracking modules)	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Finish developing pest and disease module.

2. WORK TO BE DONE

- Carry out the system testing.
- Writing report

3. PROBLEMS ENCOUNTERED

- No problem at this stage.

4. SELF EVALUATION OF THE PROGRESS

- Progress is on track.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: 13**
Student Name & ID: Leng Kai Yi 1902845	
Supervisor: Ts Tan Teik Boon	
Project Title: Farm Management Information System (Crop Planning and Tracking modules)	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Completed the report.

2. WORK TO BE DONE

- Submit to Turnitin and ready for submission.

3. PROBLEMS ENCOUNTERED

- No problem at this stage.

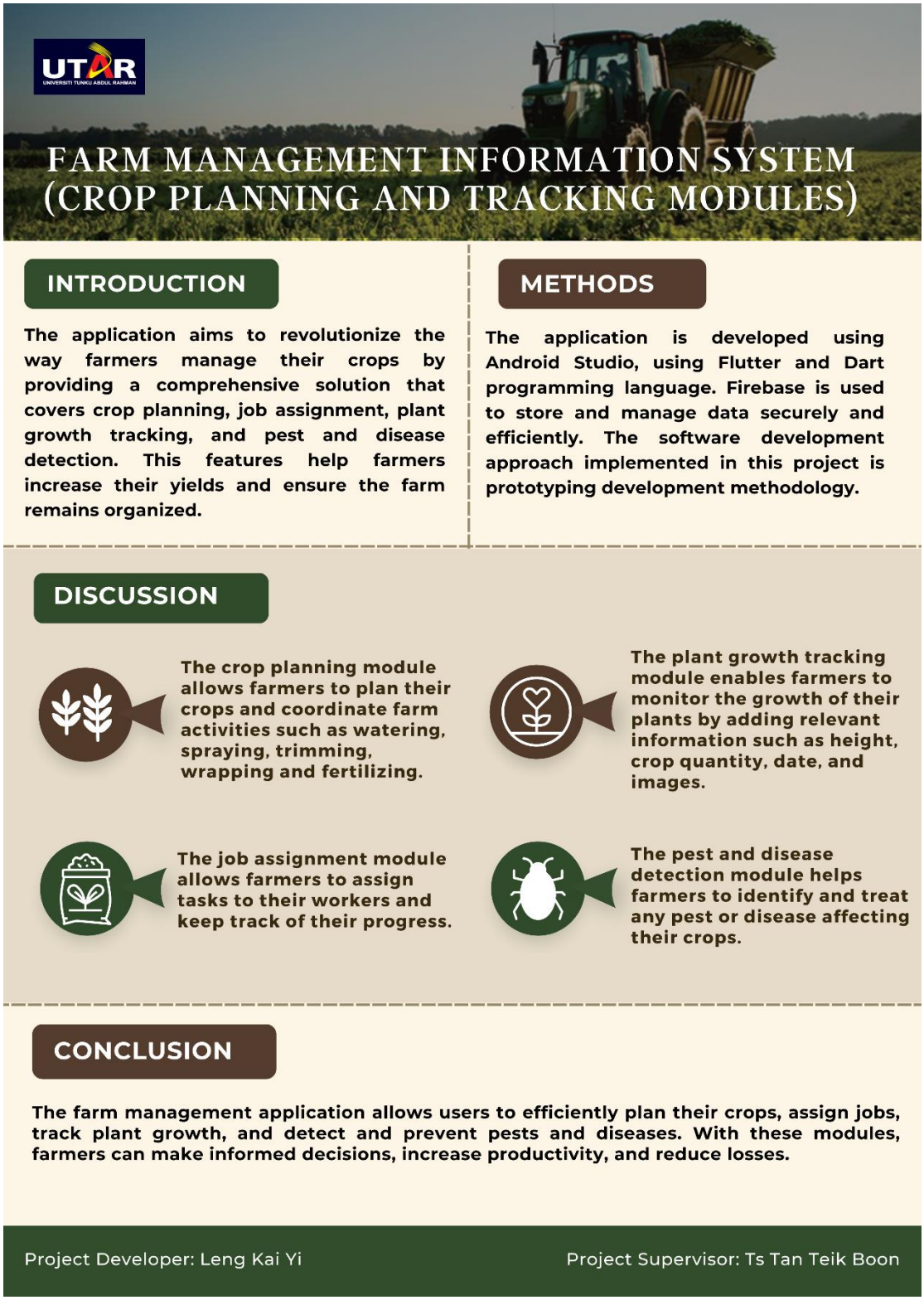
4. SELF EVALUATION OF THE PROGRESS

- Progress is on track.

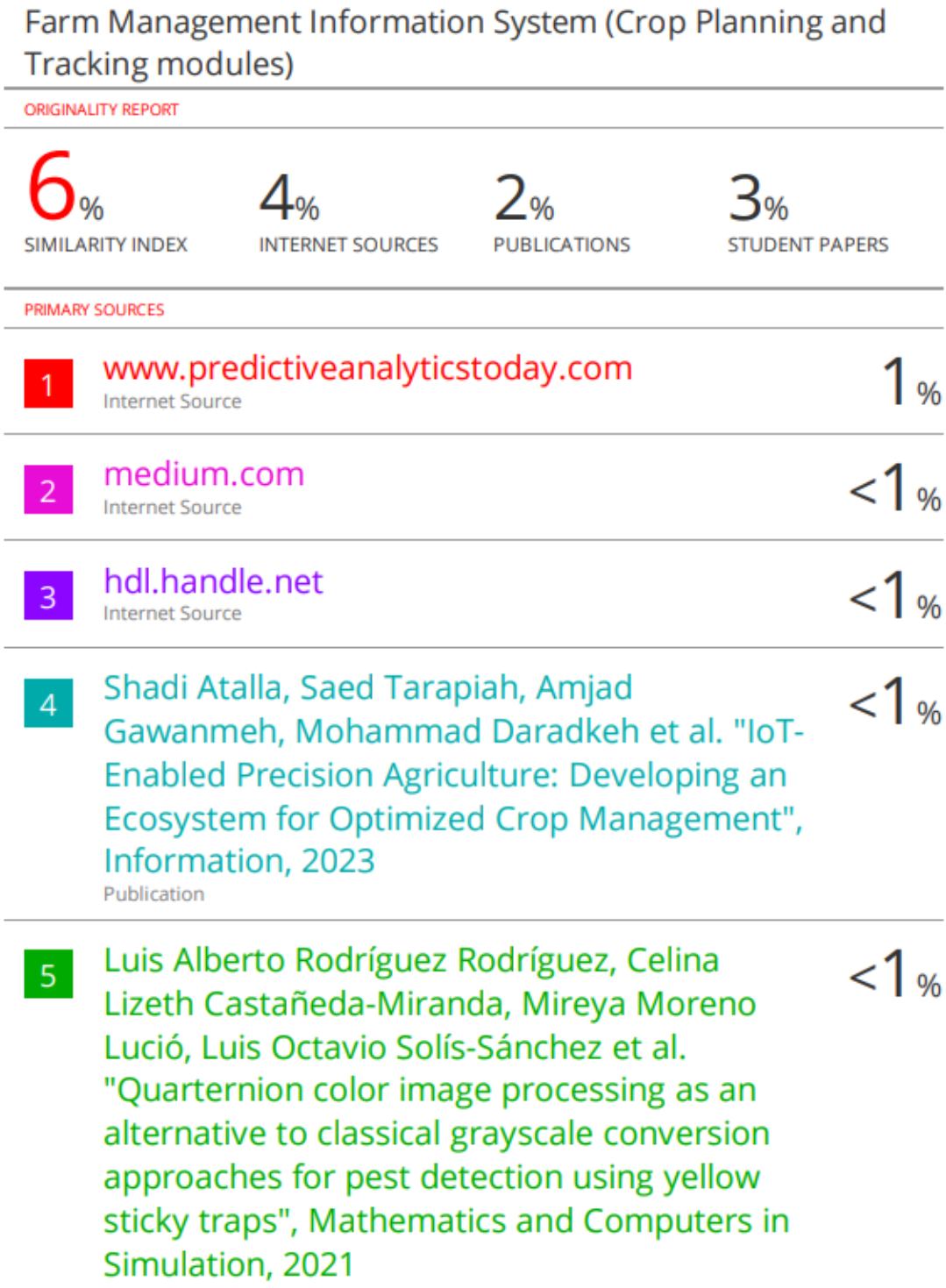
Supervisor's signature

Student's signature

POSTER


UTAR
UNIVERSITI TUNKU ABDUL RAHMAN
FARM MANAGEMENT INFORMATION SYSTEM
(CROP PLANNING AND TRACKING MODULES)
INTRODUCTION
The application aims to revolutionize the way farmers manage their crops by providing a comprehensive solution that covers crop planning, job assignment, plant growth tracking, and pest and disease detection. This features help farmers increase their yields and ensure the farm remains organized.
METHODS
The application is developed using Android Studio, using Flutter and Dart programming language. Firebase is used to store and manage data securely and efficiently. The software development approach implemented in this project is prototyping development methodology.
DISCUSSION
The crop planning module allows farmers to plan their crops and coordinate farm activities such as watering, spraying, trimming, wrapping and fertilizing.
The plant growth tracking module enables farmers to monitor the growth of their plants by adding relevant information such as height, crop quantity, date, and images.
The job assignment module allows farmers to assign tasks to their workers and keep track of their progress.
The pest and disease detection module helps farmers to identify and treat any pest or disease affecting their crops.
CONCLUSION
The farm management application allows users to efficiently plan their crops, assign jobs, track plant growth, and detect and prevent pests and diseases. With these modules, farmers can make informed decisions, increase productivity, and reduce losses.
Project Developer: Leng Kai Yi
Project Supervisor: Ts Tan Teik Boon

PLAGIARISM CHECK RESULT



Farm Management Information System (Crop Planning and Tracking modules)

ORIGINALITY REPORT

6%	4%	2%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.predictiveanalyticstoday.com Internet Source	1%
2	medium.com Internet Source	<1%
3	hdl.handle.net Internet Source	<1%
4	Shadi Atalla, Saed Tarapiah, Amjad Gawanmeh, Mohammad Daradkeh et al. "IoT-Enabled Precision Agriculture: Developing an Ecosystem for Optimized Crop Management", Information, 2023 Publication	<1%
5	Luis Alberto Rodríguez Rodríguez, Celina Lizeth Castañeda-Miranda, Mireya Moreno Lució, Luis Octavio Solís-Sánchez et al. "Quarternion color image processing as an alternative to classical grayscale conversion approaches for pest detection using yellow sticky traps", Mathematics and Computers in Simulation, 2021	<1%

	Publication	
6	Marco L. Napoli. "Beginning Flutter®", Wiley, 2019 Publication	<1 %
7	eprints.utar.edu.my Internet Source	<1 %
8	Submitted to Creighton University Student Paper	<1 %
9	brain-mentors.com Internet Source	<1 %
10	Submitted to Sheffield Hallam University Student Paper	<1 %
11	codesinsider.com Internet Source	<1 %
12	extension.missouri.edu Internet Source	<1 %
13	Submitted to Universiti Tunku Abdul Rahman Student Paper	<1 %
14	Submitted to Uva Wellassa University Student Paper	<1 %
15	Uzair Ahmad, Lakesh Sharma. "A REVIEW OF BEST MANAGEMENT PRACTICES FOR POTATO CROP USING PRECISION AGRICULTURAL TECHNOLOGIES", Smart Agricultural Technology, 2023	<1 %

	Publication	
16	www.javatpoint.com Internet Source	<1 %
17	www.mindbowser.com Internet Source	<1 %
18	Faisal Nabi, Xiaohui Tao, Jianming Yong. "Security aspects in modern service component-oriented application logic for social e-commerce systems", Social Network Analysis and Mining, 2021 Publication	<1 %
19	www.infotech.com Internet Source	<1 %
20	www.researchgate.net Internet Source	<1 %
21	Submitted to Higher Education Commission Pakistan Student Paper	<1 %
22	Mahmoud Abbasi, Mohammad Hossein Yaghmaee, Fereshteh Rahnama. "Internet of Things in agriculture: A survey", 2019 3rd International Conference on Internet of Things and Applications (IoT), 2019 Publication	<1 %
23	itac.fhi360.org Internet Source	<1 %

24	Submitted to Asia Pacific University College of Technology and Innovation (UCTI) Student Paper	<1 %
25	Submitted to Genesis Christian College Student Paper	<1 %
26	Submitted to HELP UNIVERSITY Student Paper	<1 %
27	Soundharya Sivakumar, Gowryparvathy Bijoshkumar, Athulya Rajasekharan, Vaishnavi Panicker et al. "Evaluating the Expediency of Smartphone Applications for Indian Farmers and Other Stakeholders", AgriEngineering, 2022 Publication	<1 %
28	Submitted to Southern Cross University Student Paper	<1 %
29	Submitted to University of Ghana Student Paper	<1 %
30	Submitted to Middlesex University Student Paper	<1 %
31	Submitted to Universiti Malaysia Perlis Student Paper	<1 %
32	Wayne Piekarski, Bruce H. Thomas. "Interactive augmented reality techniques for construction at a distance of 3D geometry",	<1 %


Proceedings of the workshop on Virtual environments 2003, 2003
Publication
33
www.stacktips.com
Internet Source
<1%
34
A. Martin, D. Sathish, C. Balachander, T. Hariprasath, G. Krishnamoorthi. "Identification and counting of pests using extended region grow algorithm", 2015 2nd International Conference on Electronics and Communication Systems (ICECS), 2015
Publication
<1%
35
farmlogs.com
Internet Source
<1%
Exclude quotes On
Exclude bibliography On
Exclude matches Off

Universiti Tunku Abdul Rahman			
Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1of 1

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	Leng Kai Yi
ID Number(s)	1902845
Programme / Course	CS
Title of Final Year Project	Farm Management Information System (Crop Planning and Tracking modules)

Similarity	**Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)**
Overall similarity index: 6 % **Similarity by source** Internet Sources: 4 % Publications: 2 % Student Papers: 3 %	
Number of individual sources listed of more than 3% similarity: 0	

Parameters of originality required and limits approved by UTAR are as Follows:

(i) Overall similarity index is 20% and below, and

(ii) Matching of individual sources listed must be less than 3% each, and

(iii) Matching texts in continuous block must not exceed 8 words

Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.

Note Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

Signature of Supervisor	Signature of Co-Supervisor
Name: Ts Tan Teik Boon	Name: _____
Date: 27/4/2023	Date: _____

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	1902845
Student Name	Leng Kai Yi
Supervisor Name	Ts Tan Teik Boon

TICK (√)	DOCUMENT ITEMS Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
	Front Plastic Cover (for hardcopy)
√	Title Page
√	Signed Report Status Declaration Form
√	Signed FYP Thesis Submission Form
√	Signed form of the Declaration of Originality
√	Acknowledgement
√	Abstract
√	Table of Contents
√	List of Figures (if applicable)
√	List of Tables (if applicable)
	List of Symbols (if applicable)
	List of Abbreviations (if applicable)
√	Chapters / Content
	Bibliography (or References)
√	All references in bibliography are cited in the thesis, especially in the chapter of literature review
	Appendices (if applicable)
√	Weekly Log
√	Poster
√	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)
√	I agree 5 marks will be deducted due to incorrect format, declare wrongly the ticked of these items, and/or any dispute happening for these items in this report.

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report.

_____

(Signature of Student)

Date: 27/4/2023