Faculty of Engineering & Technology

Electrical & Computer Engineering Department

Farm Supervisor: A Mobile App Supporting Multi-Type Recommendations for Palestinian Farmers

---

Prepared by:

Alaa Sehwail - 1191741
Ahmaide Al-Awawda - 1190823
Abdullah Ayoub - 1190994

Supervised by:
Dr. Abdullatif Abuissa

A Graduation Project submitted to the Department of Electrical and Computer Engineering in partial fulfillment of the requirements for the degree of B.Sc. in Computer Engineering

Birzeit University

February 2024

المستخلص

تحتّل الزراعة مكانةً هامة في فلسطين والتي تلعب دوراً هامّاً وحيوياً في تلبية احتياجات الشعب الفلسطيني من الغذاء وتعزيز الاقتصاد المحلي. حيث تعتبر الزراعة مصدر دخل للكثير من الفلسطينيين، وتعزز بذلك الدخل الشخصي وتحسّن من مستوى المعيشة خاصة للعائلات الريفية. إضافة إلى ما سبق، فإنها تحافظ على التراث الزراعي والثقافة الزراعية المحليّة في فلسطين في ظل كافة الظروف التي تتعرض لها.

ولكن تكمن المعضلة لدى المزارعين في عدم وجود المعلومات الكافية التي تساعدهم في زيادة الإنتاجية وازدهار زراعتهم، ومن هنا تظهر الحاجة لوجود تطبيق يساعد في تقديم التوصيات والاقتراحات المناسبة للمزروعات للحفاظ عليها وزيادة إنتاجيتها.

"فارم سوبر فايزر" هو تطبيق للمزارعين الفلسطينيين يعمل على إعطاء مقترحات للمزارع باستخدام تطبيقات الذكاء الاصطناعي وتعلم الآلة بهدف توجيه وإفادة المزارع وتحسين إنتاجيتهم الزراعية، وذلك من خلال تحليل البيانات المتعلقة بالمحاصيل والأراضي الزراعية والطقس وتقنيات الزراعة وكل ما يزوده المزارع للتطبيق. يمكن أن تشمل المقترحات المقدمة من التطبيق توقيت الزراعة المثلى، تركيبة التربة المناسبة، تقنيات الري الفعالة، بالإضافة إلى بعض المعلومات التي قد تخص الآفات والأمراض التي تصيب المحاصيل، والاستخدام الصحيح للأسمدة والمبيدات الزراعية وملائمتها للنبات. هذا المشروع يهدف إلى تعزيز إنتاجية المزارعين الفلسطينيين، وتحسين جودة المحاصيل، وتقليل التكاليف الزراعية، وتعزيز استدامة القطاع الزراعي في فلسطين.

Abstract

Agriculture occupies an important position in Palestine, playing a vital role in meeting the Palestinian people's food needs and enhancing the local economy. Agriculture is considered a source of income for many Palestinians, thereby boosting personal income and improving living standards, especially for rural families. In addition to the above, it preserves agricultural heritage and local agricultural culture in Palestine under all circumstances it faces.

However, the dilemma for farmers lies in the lack of sufficient information to help them increase productivity and thrive in their farming endeavors. Hence, there is a need for an application that provides recommendations and suitable suggestions for crops to preserve them and increase their productivity.

"**Farm Supervisor**" is an application for Palestinian farmers that works by providing suggestions to farmers using artificial intelligence and machine learning applications, aiming to guide and benefit farmers and improve their agricultural productivity through analyzing data related to crops, agricultural land, weather, farming techniques, and all information provided by the farmer to the application. The suggestions provided by the application may include optimal timing for planting, suitable soil composition, efficient irrigation techniques, as well as information regarding pests and diseases affecting crops, and the correct use of fertilizers and agricultural pesticides and their suitability for plants. This project aims to enhance the productivity of Palestinian farmers, improve crop quality, reduce agricultural costs, and promote the sustainability of the agricultural sector in Palestine.

Acknowledgment

First, we are grateful to the Palestinian Ministry of Agriculture, for having a meeting with our team and supervisor we thank the minister Mr. Riad Atari for his time in helping us, as he and the ministry provided some of the most pointed out current Palestinian issues, that can be useful to be started off with the current phase of this project.

We also like to thank the two agricultural engineers Eng. Sameer Duhaidi and Eng. Mohammad Odeh for their time in helping our team and guiding us, as Eng. Duhaidi have had many meetings with the team pointing out the issues that the application can include focusing on the needs of the farmers in the Palestinian society, while Eng. Odeh provided his research about the caring methods and pests that are related to the avocado plant.

A special thanks to our supervisor Dr. Abdellatif Abuissa for his guidance and support and giving us the needed knowledge to start off and work on this project

Table of Content

List of Figures

List of Table

List of Abbreviation

Abbreviation	Original Sequence
ANN	Artificial Neural Network
API	Application Programming Interface
CF	Collaborative Filtering
CNN	Convolutional Neural Network
CSS	Cascading Style Sheet
GPS	Global Positioning System
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IDE	Integrated Development Environment
JDB	Java Database
JS	Java Script
JSON	JavaScript Object Notation
NoSQL	No Structured Query Language
RDBMS	Relational Database Management System
REST	Representational State Transfer
XML	Extensible Markup Language
TF-IDF	Term Frequency-Inverse Document Frequency

Chapter One: Introduction

This chapter explains what the objectives are from creating this application, explains the motivations that helped create the idea of this application, in addition to clarifying what is the problem statement, also it shows the questionnaire details and questions that were asked to the Palestinian farmers to get their thoughts about the application, also it includes the organization of this report.

1.1. Motivations

The Palestinian society, which is an eastern third-world nation with a great agricultural past and love for the land, depends significantly on the contributions of its farmers. Numerous Palestinians rely heavily on agriculture and animals as sources of income, and it is also essential to the region's socioeconomic stability and food security. However, the productivity and profitability of Palestinian farmers are hampered by several issues. For instance, the country's political challenges, the constraints imposed by the Israeli occupation, and the lack of resources, as well as the country's limited access to information about contemporary farming methods, inadequate understanding of crop management and selection, insufficient irrigation techniques, and a lack of awareness of weather patterns and the effects of climate change. This requires the presence of a system to aid them in solving this problem.

1.2. Problem Statement

The problem lies in attempting to increase crop productivity and maintain their prosperity among Palestinian farmers, which may be hindered by several factors beyond the control of the Palestinian farmer. Firstly, farmers lack sufficient resources and suitable conditions for their crops. For example, they may not have the means to monitor weather conditions to determine if there is an extreme heatwave or frost that could affect their crops, requiring them to take necessary measures in irrigation methods to preserve them. On the other hand, helping the farmer identify the disease based on the symptoms appearing on the plant and provides treatment and appropriate care methods for the plant.

For the aforementioned reasons, a farm advisor can be the solution to assist farmers in providing all the appropriate suggestions and recommendations for their crops, taking into account their various types and the nature of the geographical area in which they are cultivated.

1.3. Objectives

The main objective of our application is to empower Palestinian farmers by providing them with accurate and timely agricultural recommendations. By leveraging machine learning applications and agricultural expertise, our application aims to address the unique challenges

faced by Palestinian farmers, such as limited resources, water scarcity, and unpredictable weather patterns. Farmers will receive personalized recommendations on crop selection, optimal planting and harvesting times, caring method techniques, pest and disease management, and sustainable farming practices through this application. Our goal is to enhance productivity, increase crop yields, promote sustainable agriculture, and ultimately improve the livelihoods of Palestinian farmers.

The idea of "Farm Supervisor" came after noticing the problems suffered by farmers. There is a gap between the knowledge they have and the agricultural problems they face, and it is difficult for some to communicate with agrarian specialists to get the necessary consultations.

Many farmers misuse pesticides and agricultural fertilizers, either in terms of the method of use, the amount used, and the type of treatment needed for the plant. These practices affect the health of farmers first, and they also affect their production yield, as they may lead to damage to the entire crop at times or lack of plant production because of some diseases. In addition, weather factors can pose a real challenge for farmers, as strong winds or frosts at some times of the year destroy crops in winter, and significantly higher temperatures lead to drying out and sometimes death of plants in summer. What helps in this is the inexperience of farmers in dealing with these conditions, so there was a need to monitor the weather situation accompanied by recommendations to farmers on what to do based on this.

Some farmers were asked about the problems they face during their work, and what contents can help them and increase their productivity. Some agricultural experts were also contacted to find out how to improve the situation of farmers in Palestine. Scientific papers on similar topics were also researched and applications with similar ideas were considered in other areas.

A recommender system for agriculture can be a big help for farmers as it enhances their agricultural practices. With the use of a user-friendly and accessible tool, farmers will gain access to a wealth of knowledge on planting, seeding, irrigating, and other critical aspects of farming. The application will offer tailored recommendations based on predictions that come from local conditions, compost quality, current climate, crop characteristics and more. These guidance’s will lead farmers into making informed decisions, which will increase the productivity of their crops.

Figure 1-1 shows what types of recommendations the system can provide.

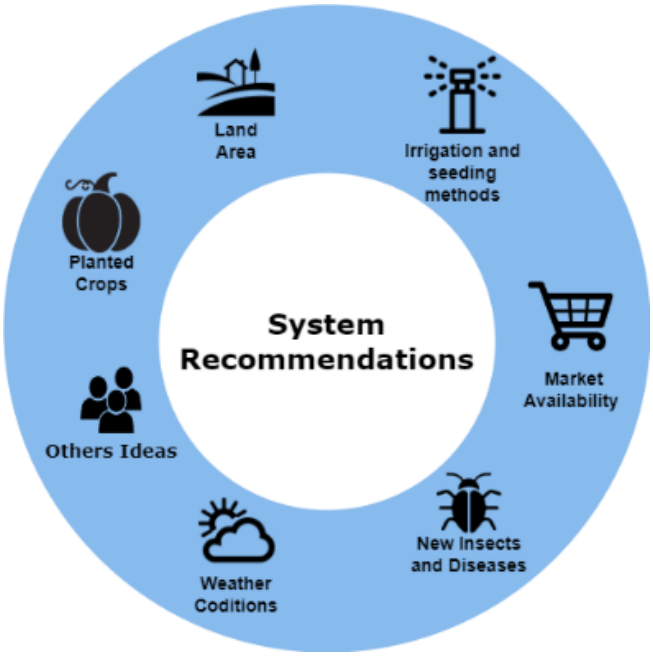


Figure 1-1: Recommender System Services in Agriculture

Furthermore, the recommender system application seeks to nurse experience sharing as it will improves collaboration among the Palestinian farmers. This will give the farmers a chance to learn from one another, share experiences, and exchange innovative ideas hoping to build a sense of community and collective learning among farmers.

The application holds great potential to transform the agricultural sector by empowering farmers, addressing challenges, fostering collaboration, and promoting sustainability. As it aims to revolutionize Palestinian farming practices, enhance productivity, and contribute to the overall development of the agricultural sector in Palestine.

1.4. Needs Assessment

To have a proof of concept, a google form was made and spread around the West-Bank to reach out as many farmers from different generations and areas, to get the needed information to build the application, this step helps in detecting the features for the application to guarantee that it will help as many Palestinian farmers as possible.

The form includes the following questions:

1. **Range of age of the farmer:** Which helps checking the compatibility of different age groups with the idea of the application.

2. **The area of cultivation**: Which helps identifying the different needs of each area.

3. **Most planted types of crops:** To know what suits each type and which area it can be found in.

4. **Thoughts about the idea of the application:** To see if farmers are into the idea of the recommender system application and getting the use of other farmers experiences.

5. **Thoughts about the features in the application:** To see if the idea of the features is useful for the farmers.

6. **The willingness to share results & experiences:** To see how willing farmers are to help collaborate with each other.

7. **Any other thoughts or ideas:** To get more ideas and get to know more about the needs of the agricultural sector in Palestine.

Figure 1-2 shown the statistical results for the second question in the form, the whole questions and responses of the form can be found in **Appendix 1**:

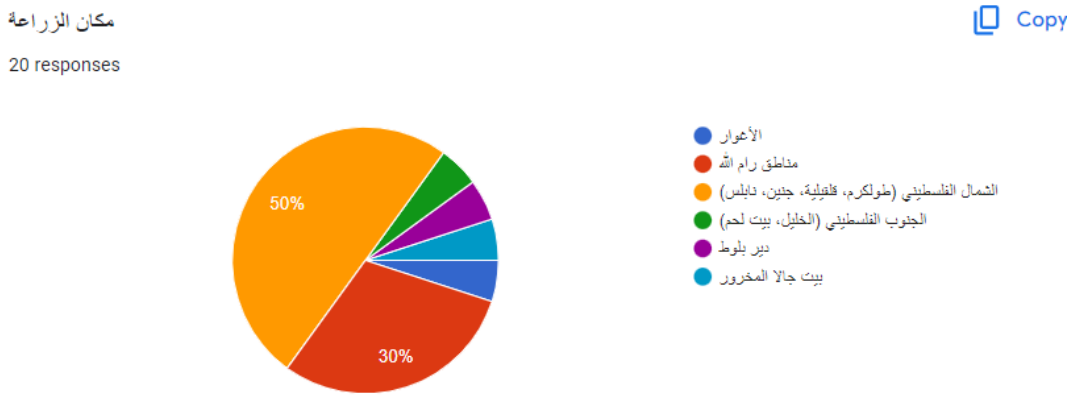


Figure 1-2: Questionnaire Response about Farmers Areas of Cultivation

In addition to the questionnaire, a brochure was designed to the application, to announce the application to the farmers, it can be found in **Appendix 2**: Farm Supervisor brochure**.**

As noticed, most of responses are from the north Palestinian, so we found that Avocado plant be a suitable choice for this graduation project.

1.5. Organization Of Report

This report provides a detailed overview of all the studies and works that have been implemented to develop the Farm Supervisor application, which has been divided into several chapters as follows: Chapter 1 includes a comprehensive explanation of the concept and clarification of the main problems that led to the development of this application. It also discusses the desired objectives of this application. Chapter 2 consists of a summary of the papers and research studies that have been reviewed, as well as a preview of a summary of searches for similar applications and extract the features. As for Chapter 3, it covers the software mobile application and technologies used. Chapter 4 holds the accomplished design and implementation. The last chapter, which is Chapter 5, provides a conclusion and the challenges we met, also it provides a description of the future work to be done.

Chapter Two: Literature Review & Related Work

This chapter shows the results of background about recommender systems in agriculture and shows results of research about similar applications and their features, it also provides comparison between "Farm Supervisor" and similar applications.

2.1. Recommendation Systems

The main factor in this application is that it will be basically giving recommendations to the Palestinian farmers judging by their previous actions or the information they give to the application.

2.1.1. The Impact of Recommender Systems

In today's modern world, the exponential growth of artificial intelligence has led to an overwhelming abundance of applications to make life easier for their users in various domains such as e-commerce, streaming platforms, and social media [1]. Users frequently ask for help to find relevant and customized information that matches their tastes since navigating this large sea of possibilities may be overwhelming. Recommender systems are crucial in this situation.

Recommender systems, are basically intelligent information filtering tools, hope to assist users in their decision-making processes, by providing them recommendations based on their interests, past behaviors, other similarities, preferences, and more. These systems use machine learning, and deep learning algorithms to analyze user's data and make precise predictions about the products or content a user will find appealing. They do this by looking at either the user's past content or the collective that results from similar users' behavior [2]. Many applications nowadays have been transformed by recommender systems.

Human life has been profoundly impacted by recommender systems by offering individualized recommendations that raise the satisfaction of its users and increase customer experiences across a range of different industries. Their technologies make decision-making simpler in many fields, adding up to boosting sales and revenue by providing customized suggestions based on the user's preferences. Plus, they provide targeted advertising, and produce insightful data-driven insights.

2.1.2. Types of Recommender Systems

The recommender systems can be built based on two filtering methods, either by content filtering which matches the features of the item to the interests of the user and the user's history, and the collaborative filtering which depends on other users' similarities towards the available item, which made recommender systems to branch into different types [3].

There are multiple approaches used for designing recommendation systems, the most two common approaches are:

A. Content-based Recommender Systems

Content-based recommenders uses content filtering which goes by checking the users previously interested-in or highest rated and searched for content, then as it analyses the attributes of the contents or items to create a user profile based on the historical interactions between the user and the item.

This type of recommender system has two models of implementation: the first one is the vector space model which makes a vector of items and gives them ratings based on ratings that has been done on similar items that the user is familiar with or has high ratings for, the second model is a classification method which comes by creating a decision tree to predict the user's thoughts on an item [4].

B. Collaborative Recommender Systems

Collaborative recommender systems are based on previous interactions to find similar users, then uses the similarity to make the recommendations, as it gives the user recommendations based on the highest rated items for by similar users then gives top-n recommendations [5]. This recommender system can be divided into two types:

1. User-based collaborative filtering: which gives recommendations based on user's who had similar item's interests' history or similar ratings [5].

2. Item-based collaborative filtering: which gives recommendations for similar items based on the preferences or interests of users as it analyzes the past interactions for similar recommended items for users where the recommended content or items are similar to previously interested items by the user [6].

Collaborative recommendation models of implementation can be done in different ways such as: clustering algorithms, machine learning algorithms such as (Random Forest, naïve Bayas), and deep learning methods [6].

2.1.3. Recommender System in Agriculture

Recommender systems can be a very valuable tool in agriculture, after researching related works such as (Smart Farming: Internet of Things IoT-Based Sustainable Agriculture [3], and more that can be found in the **2.2. Recommender System** Background section), then trying out applications with similar ideas such as (agriApp[12], FarmGo[13], and more that can be found in the **Similar Applications** section).

After checking the related papers and similar apps, it was shown that recommender systems can support farmers in plant cures, disease management, and seasonal crop recommendations and provides recommendations based on the weather conditions.

By analyzing data on climate patterns, and historical crop performance, recommender systems can give suggestions for crops that are appropriate for various seasons, giving farmers the information, they need to make decisions and improve agricultural output.

The two diagrams show the workflow of how the recommender system can work, where figure 2-1 shows how the system can generate content-based recommendations, and figure 2-2 shows how the system can generate collaborative recommendations.

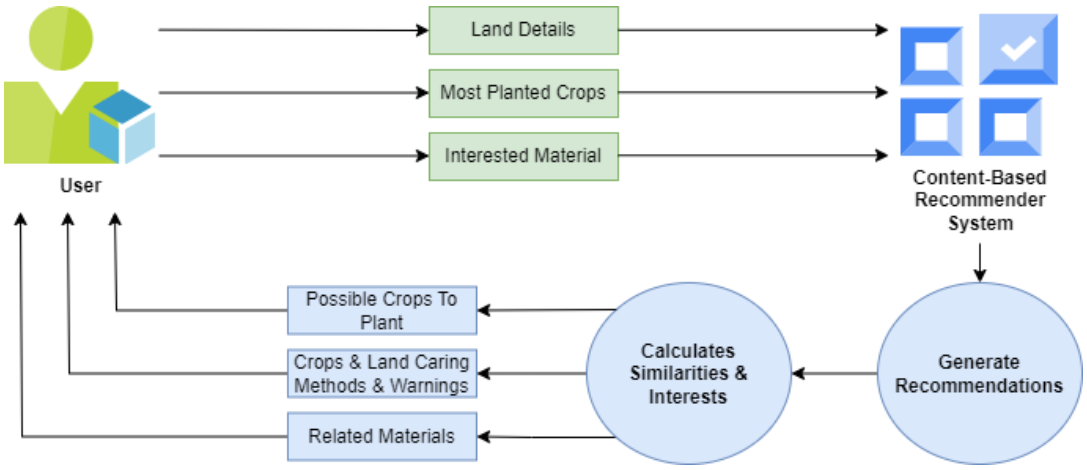


Figure 2-1: Content-Based Recommender System in Agriculture

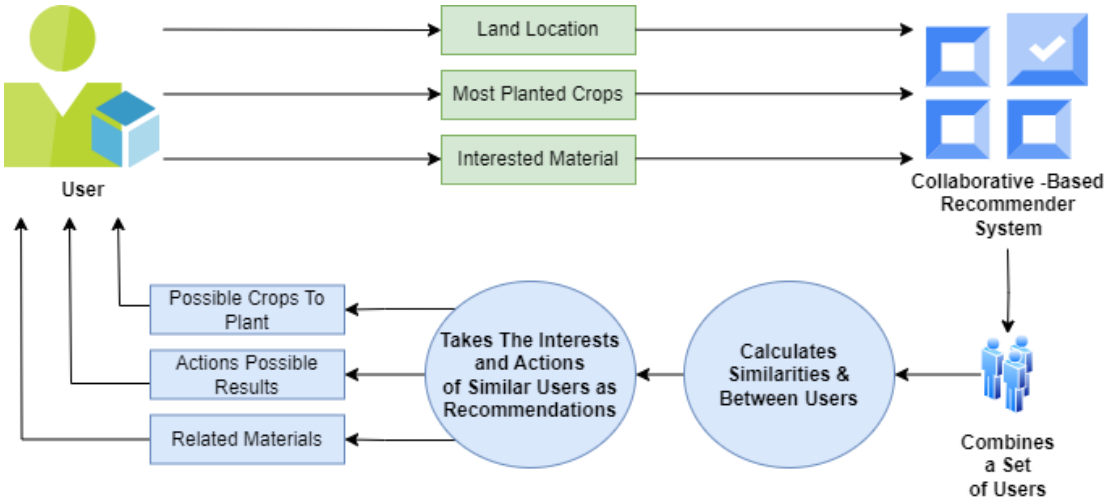


Figure 2-2: Collaborative-Based Recommender System in Agriculture

2.2. Recommender System Background

This section talks about the background method to the recommender system.

A paper discussed the application of intelligent recommendation for agriculture information, where it discusses the content-based recommendation system where is a fundamental approach in recommendation systems [7]. Unlike relying on user ratings, this algorithm actively generates recommendations based on the user's historical interactions such as evaluations, forwarding, collecting, and downloading. It creates documents aligned with the user's interests and calculates the statistical matching degree between the item to be recommended and the user's preferred content. The recommended item with the highest similarity is then suggested to the user.

The central challenge in content-based recommendation algorithms lies in calculating item similarity. Pearson Correlation Coefficient (PCC) and cosine similarity are commonly employed for this purpose. The article provides a detailed explanation of these two similarity calculation methods. Other frequently used similarity calculation methods include modified cosine similarity, Tanomi, Euclidean distance, among others.

Furthermore, the paper discusses various improvement strategies for similarity measurement. These enhancements involve proposing new similarity functions, incorporating correlation weight factors to calculate similarity, and utilizing similarity propagation to refine the similarity function. Overall, the content-based recommendation algorithm focuses on personalized recommendations derived from user behavior without relying on explicit ratings.

Also, it discussed collaborative filtering (CF) recommendation algorithms come in three main types: user-based, item-based, and model-based. The user-based CF algorithm calculates similarity between users, forming a user similarity matrix. It selects nearest neighbors for a target user, predicting their interest in unseen items based on neighbors' ratings and recommending the item with the highest predicted score. Conversely, the item-based CF algorithm analyzes similarity between items using aggregated user ratings and suggests new items to an active user based on their past preferences.

Another paper discussed the problem of the plant being exposed to diseases that lead to its death and thus not benefiting from it, in addition to the losses it causes to the farmer [8]. A recommendation system was proposed to analyze the type of soil, determine which diseases the plant is infected with, and provide fertilizers and appropriate treatment methods.

Where it collected data related to the disease of a particular plant, what fertilizers are suitable for it, and methods of caring for it, and then used the data entered by the farmer in describing his plant, and by photographing the leaves of the affected plant, and using CNN method, the image was processed and then the expected disease was suggested.

Another research paper discussed the topic of providing recommendations related to weather and climate conditions such as frost and hail, or solar radiation to protect the plant from weather

factors, where a set of weather data was used that was collected previously and using ANN technology to predict weather conditions [9].

There is another paper that discussed the creation of a simple web application using JSP, Servlet and MySQL for weather forecasting and displaying various weather parameters [10]. Also, it provides how it used API's for collecting data and displaying it in the web application for accurate conditions and to make the application reliable. Basically, weather data in this application includes the current weather, weather forecast of 5 days, it also includes seeing the weather on different location. The paper shows that a collaborative filtering approach was used to offer crop recommendations by use the combination of user data and the weather data to give predictions about the weather and by that it helps in increasing productivity and preventing plant from any damage.

2.3. Farm Supervisor Features

Based on the researched papers, some features were extracted to incorporate them into Farm Supervisor. A content-based recommendation system was implemented, relying on data entered by the farmer regarding the disease symptoms affecting their plants and the specifications of the insects infesting the plants. Consequently, a set of potential diseases affecting the plant is suggested, along with appropriate treatments and care methods. Additionally, the idea of obtaining the location where the plant is cultivated was executed. Weather conditions for the upcoming days were predicted based on this information, and recommendations were provided to the farmer for protecting their plant from potential weather-related damages. Notifications are sent through the application to alert the farmer about these recommendations.

2.4. Similar Applications

Similar applications for agriculture recommender systems for different countries can be very useful in giving some extra features for them or to measure the quality of the application. Many apps were checked to get an idea for the possible features of the app and how to manage them.

2.4.1. AgriApp [11]

An Indian mobile application that serves for Indian agriculture that serves in many things such as:

a. **Adding new crops to the truck:**
   - Add the date of sowing: when you select a crop, you must select the date of sowing.
   - Calendar: adding tasks about what to do in this period of time. E.g., after a week, you need to put agricultural fertilizer to your crop.
   - Address: Spray for crop development
   - Practices: (After 20 days of planting, spray zyme 3ml/liter of water)

- Soil Testing: you can do testing for your soil samples.
- Sat-In: you can select your land using GPS, and you can get additional services which are paid.
- Crop Advisory: you can send a chat to the app, and they will try to solve your problem.
- When a task is waiting to be done for a crop, a notification will appear on the logo of that crop.

b. Shop:

A shop to sell agricultural materials including fertilizers, pesticides, medicines, and seeds. You can buy these products directly from the app, where you have a Cart to add products, then you can checkout and pay.

At the checkout page, you can choose either to pay online or cash on delivery. If you choose to pay online, you will be directed to the payment page, where you have to insert the information of your card then you will pay.

c. Crop Practices:

In this section, you have different categories (Plantation, Oil seeds, Fruits, Vegetables, etc.). If you choose any crop from the lists, you will have information about it in general, in addition to information about the climate it needs to grow, the soil, fertilizers, and other things.

d. Crop Care (Crop Calendar):

This section gets you a page to find all your crops that you are currently planting.

e. Soil Testing:

You can do testing for your soil samples, where you choose the area, the soil type, test type, and you will buy the package that you need.

f. Community:

Here you can communicate with other users, where you can post, like, or comment on others posts. The purpose of this section is to ask about problems that face you, or if you need any small help that the other users can do for you.

g. Weather Forecast:

In this section you can get details about the weather in the next days, and any warning or recommendations to do.

h. Market Price:

In this section, you can choose the area, the market, and the crop that you want to check its price in that market. Using this feature, you can know the price of a crop, it's rising or falling, and you can decide if it's good to plant it or not.

i. Drone Service:

In this section, you can request a drone to spray your land for a disease or something. You will choose the crop, the number of acres, your address, the date, and the problem that you need to spray for.

j. Cart and Wishlist:

The app has a cart to add products to, and a Wishlist to add favorite products.

Figure 2-3 shows the main page, crop care, crop practices, and community features that are mentioned above.

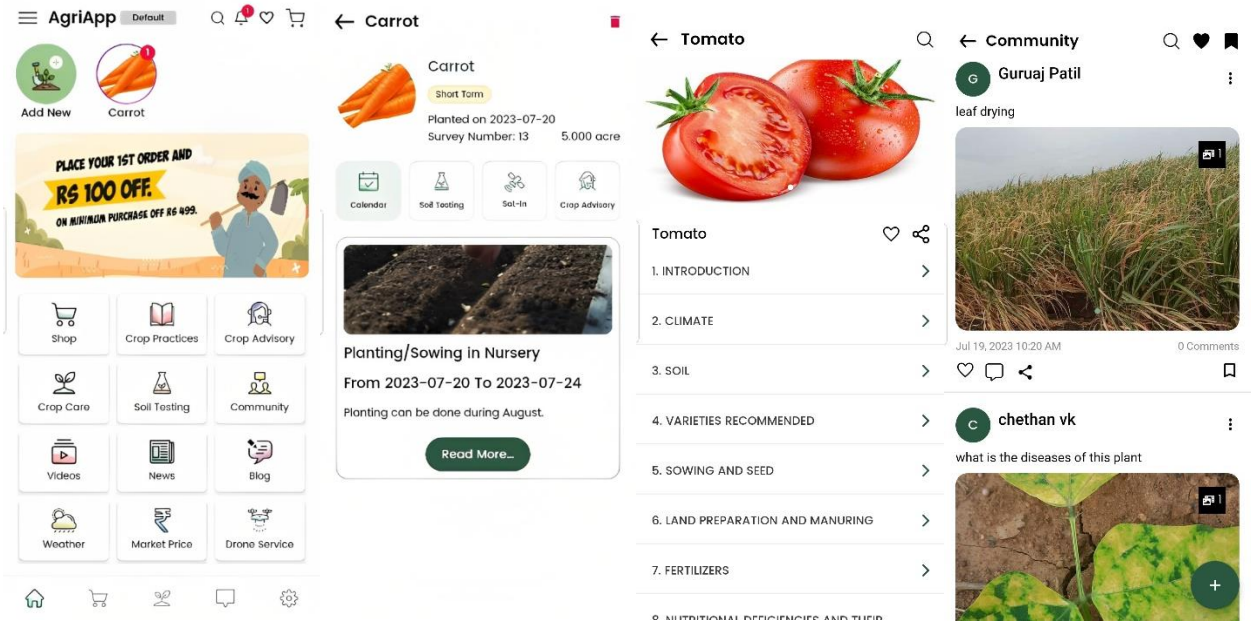


Figure 2-3: AgriApp Features [11]

2.4.2. Bushel Farm [12]

An application that simplifies farm management tasks to improve the operational efficiency and increase the productivity, as it contains the following features:

a. **Field Mapping:**
   aerial view of where your fields are and what you've planted with satellite maps, and access driving directions for your fields with the push of a button.
b. **Recent Rainfall:**
   tracking how much rain your fields receive and when.
c. **Rain and Heat History:**
   Track rainfall history and heat unit accumulation for each field and compare it to prior seasons.
d. **Scouting:**
   Log geotagged notes and pictures while you're out in the field to recall or securely share.
e. **Soil Maps:**
   see each soil type area and elevation features so you can evaluate new ground and find opportunities to improve management on existing fields.
f. **Future Prices:**
   See current futures prices, and track price fluctuation and market carry.

g. **Inputs:**
Keep organized records of how much you're spending on inputs and easily view the fieldwork associated with each product.

Figure 2-4 shows the Field Mapping feature that is mentioned above.

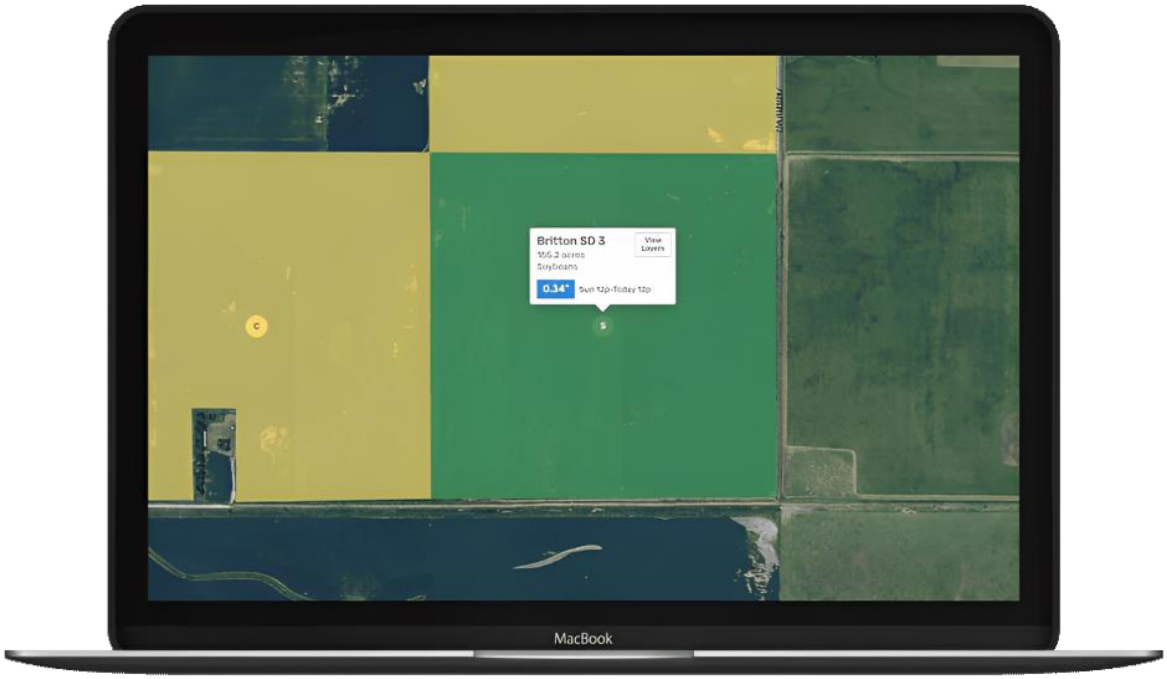


Figure 2-4: Bushel Farm Field Mapping [12]

2.4.3. FarmGo [13]

A mobile application that was made for the Philippines agriculture which provides features to enhance farmers productivity, with some of its features:

a. **Add farm location:**
The application adds the farm's location using GPS, and then you can choose what crops you are planting in your farm and the size of your farm.
b. **real-time weather forecast:**
Get real-time forecasts spanning over a 6-hour period.
c. **Monitor the weather forecasts for the next four days including the current day:**
You can see the probability of rain, humidity, temperature, and wind speed for each day.
d. **Brief weather forecasts for the next 14 days:**
Get a brief weather forecast for each day of the next 14 days.

2.4.4. FieldView [14]

A platform that was created by the climate corporation to help the farmers make more informed decisions containing the following features:

a. **Mapping the Fields:**
   After setting up your account, you will map your fields using GPS.
b. **Rainfall:**
   You can view how much rain has fallen on your field.
c. **Yield analysis:**
   It uses digital maps to compare critical field data layers and determine how agronomic practices can impact yield.
d. **Recent Activity:**
   Here you can find the latest activities done in your fields, including planting and harvest.
e. **Notifications**

2.5. Farm Supervisor Features & Comparing with Other Applications

This section explains the features implemented in "**Farm Supervisor**" and compares these features with other application's features.

2.5.1. Farm Supervisor Features

The application will have many features that will serve Palestinian farmers, by using the application the farmer can be able to get many advantages that make his agriculture life easier and more convenient here are some of the initial or planned features for the application:

a. Crop Caring Methods: This section gives helpful caring methods related to the Avocado plant to increase productivity.

b. Weather Forecast: This section will have the weather expectations for the current day, including the probability of rain, humidity, temperature, and wind speed. In addition, you will receive recommendations based on the weather if necessary.

c. Weather Related Caring Methods Notification: The application gives alerts about the upcoming weather every three days and the most suitable caring methods for it.

d. Gives Alerts About Spread Pests as Notifications: The application gives alerts about the pests that are spread in a farmer nearby location.

e. View Profile Page: The farmer can view its profile page and edit his basic information (name, email, … etc.).

f. Search for a Pest: The application enables the farmer to search for a specific pest if he knows the name of that pest and show description and the image of that pest.

g. Checking the Market: The application gives the ability to search the market for agricultural items that can be either plant seeds or used tools in agriculture.

2.5.2. Applications Comparison

Table 2-1 shows the comparison of "**Farm Supervisor**" and the other similar agriculture applications.

Table 2-1: Feature Matrix

Feature Matrix Planning ✗ Missing ✓ Existing

Feature	Farm Supervisor	AgriApp	Bushel Farm	FarmGo	Field View
Crops Data	✓	✓	✗	✗	✗
Caring Method	✓	✓	✗	✗	Recent Activities
Weather	✓	✓	✓	✓	Rainfull
Shop	✓	✓	✓	✗	✗
Price Changes	✗	✓	With Future Price	✗	✗
Location & Maps	✓	✓	Included With Scouting	✓	✓
Notification	✓	✓	✓	✓	✓
Soil Types	✗	✓	✓	✓	✗
Communication	✓	✓	✓	✓	✗
Area Comparison	Planning	✗	✗	✗	✗
Reminders	✗	✗	✗	✗	✗

2.6. Plant Related Data

To generate recommendations for plant that are related to their pests, diseases, weather conditions effects, caring or medicating methods, some research was done to understand the necessary needs for plants, since the focus in this phase is on the avocado plant, the searched and gathered information were all related to that plant.

2.6.1. Local Resources

The first source was the Palestinian Ministry of Agriculture, where they've explained the issues that Palestinian farmers face with the many plants these days as the pests and diseases for the plant aggravating with the farmers having trouble distinguishing many of them, they also mentioned the most plants that can be useful to start the application with such as Avocado and palm. [15]

Also meeting up with some agricultural engineers was very helpful pointing out the specified features in the application that farmers intend to find such as pests effects on fruits, and the needed actions to be taken in critical weather conditions, they also provided a research about the caring methods and issues related to the avocado plant in Palestine which was a big reason in choosing the plant to start off with in this phase. [16] [17]

2.6.2. Online Resources

To get more information about the plant of Avocado to be stored in the system to be used in generating initial recommendations, some online resources were taken.

First the Plant village website provided some information about some diseases and pests that can harm the avocado plant affecting the skin and the fruit itself, those issues were taken as warnings and recommendations that are included in the application. [18]

Another main resource was the University of California Agriculture and Natural Resources website, which provided many effective pests, insects and diseases giving how they can develop and the ways that plant can avoid them pointing out the effects of the different weather conditions on the situations of those plants, as helped feeding the database with the plant information that is needed. [19]

Chapter Three: Software Mobile Application & Technologies

This chapter talks about the technologies used for implementing the mobile application, also for defining the architecture and organization of the application.

3.1. What is a Mobile Application?

Mobile applications, also known as mobile apps, are software applications designed to run on mobile devices such as smartphones and tablets.

3.1.1. Mobile App Architecture

Mobile app architecture refers to the structure and design principles that govern the development of mobile applications. Farm Supervisor architecture consists of three layers, each one serves a specific purpose as figure 3-1 shows. The first is the presentation layer which is the face of the mobile app, where the user interacts with, and its user-friendly and well-designed. The second layer is the business layer, which is the heart of the mobile app, where it is responsible for coordinating the data transfer between the user interface and the data access layer. The last layer is the data layer which is accountable for managing all data-related activities, including caching and synchronization, as well as for storing and retrieving data [20].

3.1.2. Mobile App Organization

The Mobile app organization is used to deploy, develop, and manage the app that consists of three stages, the Front-end development stage that focuses on the user interface, the Back-end server stage that is required for user authentication-authorization, date services, and reusable business logic. Finally, the Database stage that is used to organize the data.

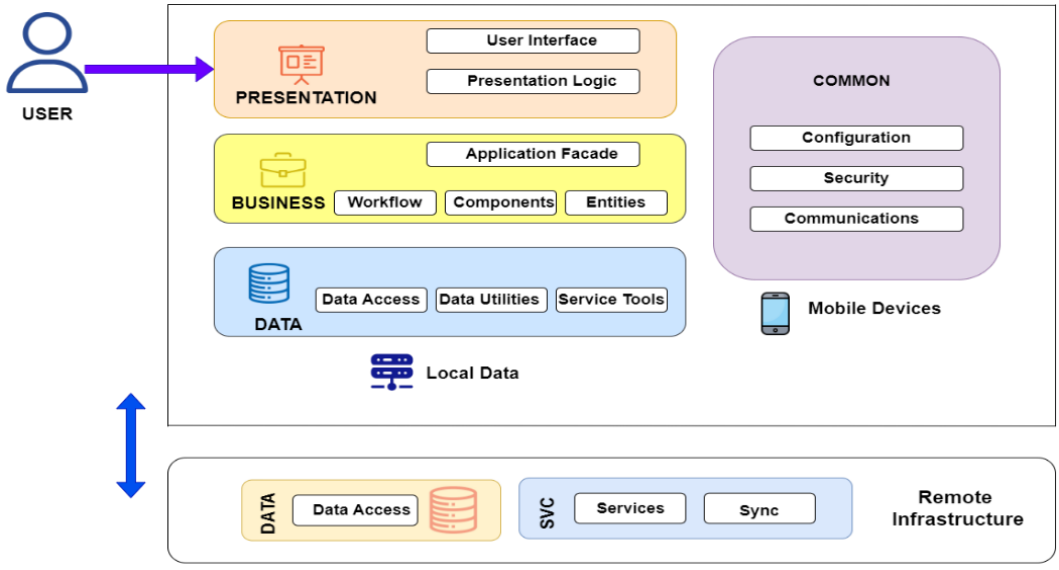


Figure 3-1: Mobile App Architecture

3.2. Hybrid Vs. Native Mobile Application

3.2.1. Hybrid Mobile Application

Hybrid apps are built using a combination of web technologies (like Hypertext Markup Language (HTML), Cascading Style Sheet (CSS), and JavaScript (JS)) and native code. They can work on any platform including browser, native, and mobile apps. They're easier to build than native applications.

3.2.2. Native Mobile Application

Native apps are written in the native programming language like objective-C for iOS and Java for Android, they have access to all platform features making them more flexible, they can be customized specifically for each platform, which can make them more user-friendly.

3.2.3. Hybrid or Native for Farm Supervisor

This graduation project needs to create high-performance app for specified platform (Android), also to provide seamless user experience, especially this app deals with Palestinian farmers which probably need to have user-friendly interfaces; therefore, a native mobile app used.

3.3. MongoDB

MongoDB is an open source No Structured Query Language (NoSQL), it can manage document-oriented information, store and retrieve it. It is quite useful for a large set of data [21].

3.3.1. NoSQL

NoSQL databases are non-tabular databases, where it does not store the data as relational databases in terms of rows and columns of table but in terms of an object/document in a collection, so there is no need to map an object to a table in the database.

3.3.2. Document Oriented

In MongoDB data is stored in document/object which translates to rows in relational databases, multiple documents form a collection which translates to tables in relational databases. Every document in a collection possesses a unique identifier by which the document can be accessed, this identifier is generated automatically in the creation of a document.

3.3.3. Schema Less

MongoDB does not require predefined schemas, which means it stores any type of data and this gives flexibility to the user compared with the relational database.

3.3.4. Query Language

The query language in MongoDB is based on JavaScript Object Notation (JSON), where documents are added/deleted from collection by specifying the operation in a JSON object. Furthermore, the data retrieved from the database is encapsulated in a JSON object.

3.4. PostgreSQL

PostgreSQL is an open-source relational database that supports both SQL (relational) and JSON (non-relational) querying. It supports a lot of programming languages like (Java, Python, C++, etc....).

3.4.1. Relational Database Management System (RDBMS)

PostgreSQL follows the relational database models, where it stores the data in tables with rows and columns.

3.4.2. Data Types

PostgreSQL supports a set of data types, including the traditional data types like string, integers as well as more advanced types like array, JSON, XML.

3.4.3. PgAdmin4

PgAdmin4 is an open-source management tool for PostgreSQL database, it offers a user-friendly graphical interface, making it easier for the user to manage PostgreSQL database without relying on command-line tool.

3.5. Android Studio

Android Studio is the official Integrated Development Environment (IDE) for android application development, it has a flexible Gradle-based build system, also it provides fast and feature-rich emulator for app testing, it can be developed for all Android devices [22]. In addition, it provides debugging tools that help in fixing issues in the code, also it includes a visual designer for creating and previewing user interface for Android apps.

3.6. Recommendation System Libraries

The implementation of the recommendation system requires the use of built-in libraries which can build effective models and give decisions, for this application as the recommender system is built in the python language, several built it libraries were used.

3.6.1. FuzzyWuzzy Library

FuzzyWuzzy is a powerfull Python library that comes in measuring how similar two sequences are with the use of Levenshtein Distance. This technique provides a precise metric for comparing strings by counting the number of edits needed to change one string into another. FuzzyWuzzy's true power is revealed by its process module, which is designed with great care to make finding strings that closely resemble a given pattern as easy as possible. [23]

The process module enables a variety of functions designed to locate and extract strings that closely resemble a given target. Using complex algorithms, it sorts through a set of strings and compares them all to a target string to determine which matches, based on similarity scores, are the best matches.

3.6.2. Scikit-Learn Library

The scikit-learn library, commonly referred to as sklearn, stands as a cornerstone in the Python data science ecosystem, providing a vast range of tools for statistical modeling and machine learning. the TfidfVectorizer stands out. This component converts textual data into a matrix of features known as TF-IDF (Term Frequency-Inverse Document Frequency) features, which represent the relative importance of words in documents.

Sklearn not only provides preprocessing tools but also streamlines the entire machine learning process, from data splitting to model training. A simple yet effective tool for splitting datasets into training and testing sets is the train test split function, which makes sure models are tested on untested data to determine how well they generalize. Moreover, the Random Forest classifier is one of the many algorithms for model building that are included in the sklearn suite. [24]

3.7. The Weather API

For a wide range of applications, including historical weather, air quality, real-time and forecast weather data, and more, WeatherAPI.com provides a full suite of geolocation and weather APIs. It is optimized for developers and offers flexibility in data access via city names, coordinates, and other identifiers, as well as high uptime and quick response times. It is a platform that can be integrated into coding projects of various sizes because it prioritizes security, scalability, and ease of use with both HTTP and HTTPS access.[25]

Chapter Four: System Implementation & Design

This chapter explains about the system implementation with all the databases, micro services alongside the design of the application, including the front-end user interface and back-end, and the notification service, also explains how the recommender system applied in this application.

4.1. Databases

This section explains the types of databases used in this project and their implementations and use.

4.1.1. PostgreSQL Database

The first used database, which stores all the needed information for the users (farmers) and their authentication requirements. This database is connected to a spring boot service that handles all its needed APIs for the application.

This database includes the required information for each farmer including the login and sign-up requirements, the available locations for the application, the available lands, and available crops as they are considered as belongs to the farmer, which is the user of the application, and for each crop the disease's history is also included (here the diseases can be between either a pest or disease), as it won't focus on any details of the diseases.

Another important schema included in this database is the user's log in token which includes the signed in user's (farmer's) claims to get the log in info and send separate notifications to different users. Figure 4-1 shows the class diagram of the database.

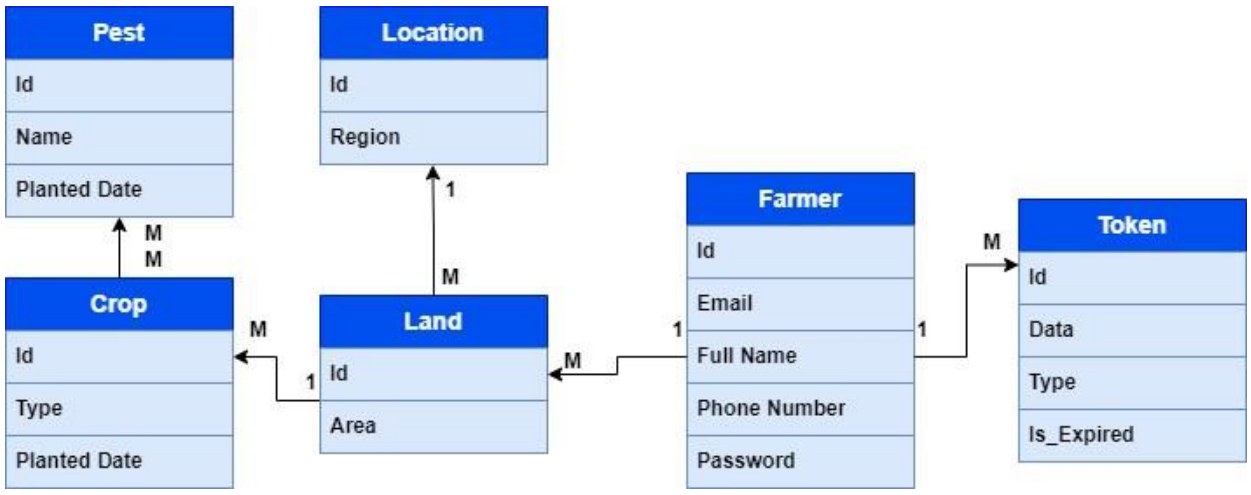


Figure 4-1: PostgreSQL Database Class Diagram

4.1.2. MongoDB

The second used data base in the project is the well-known documented database MongoDB. This database stores the following documents, focusing mainly on the plant's details as it can be used for giving recommendations, alerts, warnings, and related weather conditions.

At this point in the application (focusing for now on the avocado plant), the database contains the required information for all the related pests and diseases for the included plant including their symptoms prevention caring methods, medication methods, and needed tools, alongside the weather conditions with their suitable caring methods.

This database includes documents related to the user farmer's without focusing on their identity or login information, including their locations, yet here they are only included to be used in the recommender system later where including them in the database is necessary for generating all the needed recommendations and warnings based on similar users. The database stores the location's relationship with a pest or a disease which comes from the farmers at those locations.

The found data from the resources that are related to the plant was arranged in a JSON file to be inserted as a document to the database which contained (pests, diseases, symptoms, caring methods, cures, tools, and weather conditions)

Figure 4-2 shows the class diagram of the database.

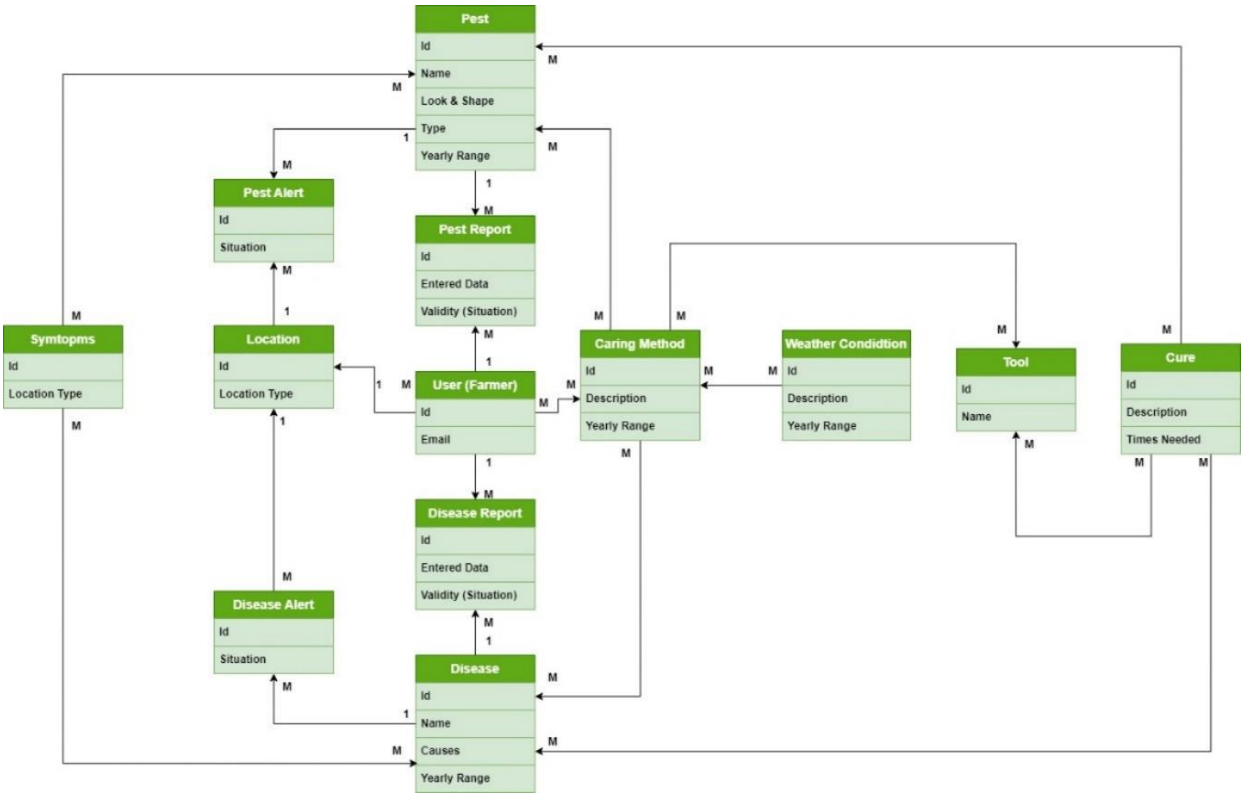


Figure 4-2: Mongo Database Class Diagram

4.2. Micro Services

This mobile application is built with the use of the micro services architecture, as it includes two main services other than the mobile application where each one has its own database from the previous two databases as each micro service comes it to focus on a job that is different from the other service. Figure 4-3 shows a diagram of the used architecture.

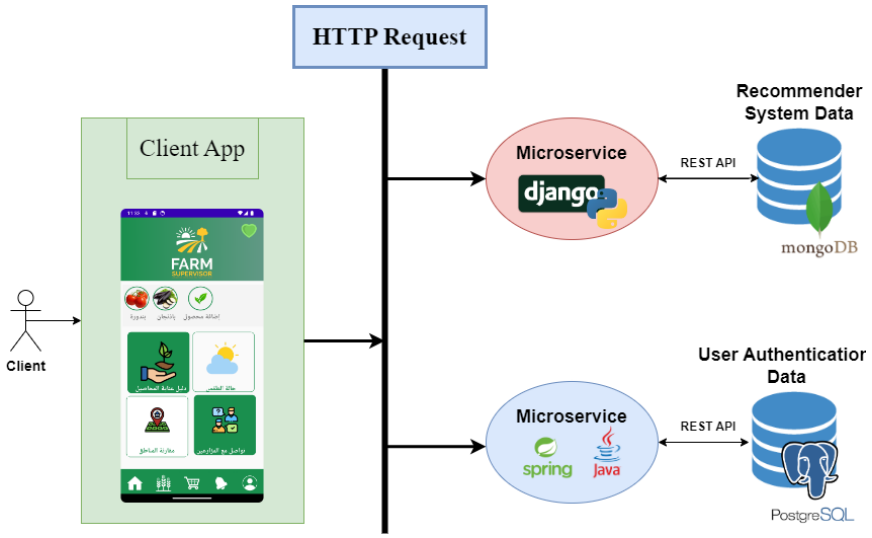


Figure 4-3 System Architecture Diagram

4.2.1. Spring Boot Authentication Service

This service comes in as a Java application created in the spring boot framework which provides many useful tools such as: REST APIs, database JDB connections, security, authentication and more.

This service plays its role as a mediator between the PostgreSQL and the mobile application as it handles RESTful APIs that read and write to the database, including handling the user's authentication to the system.

This service handles all the information related to the farmer. Including building different schemas in the database for the Farm Supervisor. After verifying all the information, a **POST** Application Programming Interface (API) request was sent to the PostgreSQL database (farm-system) to save all the information. This process enables farmers to log in to the application after email verification, with their registration information stored in the database through a **GET** API (based on the entered email) and a correct password associated with the registered email.

The service also includes spring security which gives the ability to authenticate users, and make the system more secure, first it encrypts the user's password using the Bcrypt API, an second it generated a valid token for each time a user logs in to the application so that the system stores the user's (farmer's) log in information and claims making it easier to communicate with the user and send them notification, the user can have more than one token in a time as if they log in from many different places at once.

In the mobile application on the profile page, the farmer's information is displayed using a GET API specific to the registered farmer. For the edit information page, when the farmer edits their information, a **PUT** API is applied to the registered email to update their information. Table 4-1 describe some of the APIs used for this database.

Table 4-1: APIs Method for PostgreSQL Database

APIs Method

PostgreSQL Database

API Name	HTTP Method	URL	Requirment & Output	Description
Add New Farmer	POST	api/v1/farmer/add	Requirment: All farmer info. in JSON object Output: if failure: Farmer already exist!	Add new farmer with information and registered in unique id
Update Farmer Info.	PUT	api/v1/farmer/update	Requirment: All new farmer info. + farmer email Output: if failure: Farmer does not exist!	Update farmer information based on the new information supplied
Return Farmer Info.	GET	api/v1/farmer/fullName api/v1/farmer/password api/v1/farmer/mobilenumber	Requirment: Email for farmer Output: if failure: Farmer does not exist!	Return all farmer information
Check Email	GET	api/v1/farmer/emailcheck	Requirment: Email for farmer Output: if failure: Email is not registered	Check if the entered email registed in the database or not
Check Login	GET	api/v1/farmer/logincheck	Requirment: Email & Password for farmer Output: if failure: Email is not registered or password incorrect	Check the registered email and password if entered correctly or not

4.2.2. Django Recommender System Service

This micro service comes in as a python application made in the Django framework, as the python language provides many APIs that can help in the artificial intelligence sector and in machine learning. The python Django framework provides REST API's and makes connections to databases easier which comes to its use in this application.

This micro service plays a vital role in this project as first comes in as mediator between the MongoDB and the mobile application, and second it contains the needed recommender systems alongside the warnings generators since it is connected to the MongoDB which contains all the needed information about the plants, area situations, and weather conditions. Adding on the data preprocessing where it preprocesses texts and objects.

The Django application provides many different services yet some of them are not implemented yet in the mobile application.

A. Implementing of Search Bar Service

This is a small service that is implemented to search for both pests and diseases, as it takes the user's entered text on the application and provides the most similar pests or diseases.

The searched text is preprocessed to be compared with the available data in the system, which is also goes through some preprocessing, so that the system can return the highest matches (in case there was available) after returning the processed data into its original shapes as objects filled with data.

B. Implementing of Content-Based Recommendations Service

As the aim of the application goes to be to give recommendations about plants, this service comes in to give the farmer some options that highly match the symptoms that they'll be facing.

This service simply takes the user's entered symptoms or the needed information about a pest preprocess it, then generates the available pests or diseases and compares them to the entered data with the use of the FuzzyWuzzy library, then ranks the available pests and diseases to choose the closest at max three available ones.

Now when it comes to pests, the system allows the user (farmer) to enter the type of the pest and how it looks like in case they know how it looks like but can't determine what it is. Figure 4-4 summarizes the workflow of the service.

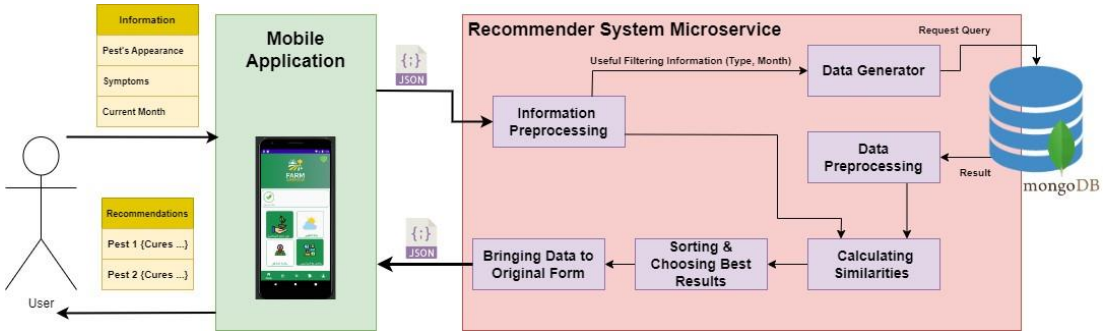


Figure 4-4: Content Based Recommender Diagram

C. Pest Alert Service for a Specific Area

This service comes in giving warnings when a pest is speeded in an area as when a threshold number of users (currently set to three) report having a specific pest in a specific time, so that the other users in the same area can be aware and start taking actions.

This service is implemented in the Django application and its needed models that connect pests and diseases to users and locations are built in the database, yet it is still not implemented in the mobile application and neither the spring boot service (the tokens are initial to send notifications to specific users), as the application still doesn't give the ability for the farmer to save his pests in the application. Figure 4-5 shows a diagram of the operation.

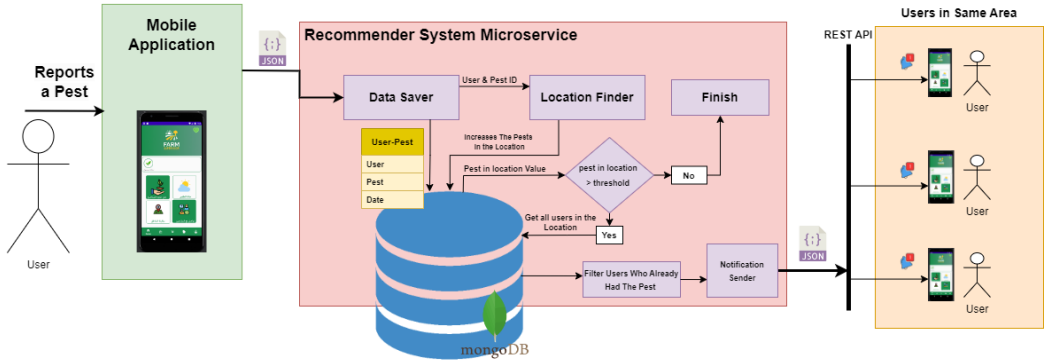


Figure 4-5: Pest Warning Diagram

D. Implementing of Weather Warnings & Recommendations

This service comes with the help of the weather API as it gives caring methods recommendations based on the upcoming three days.

The weather cases are split into four categories ordered from less extreme to most extreme (clear or sunny, cloudy, hot wave, rainy or more extreme), the service chooses the most critical or extreme weather case then checks the suitable related caring methods for it and chooses random three of them, Figure 4-6 shows the API's workflow.

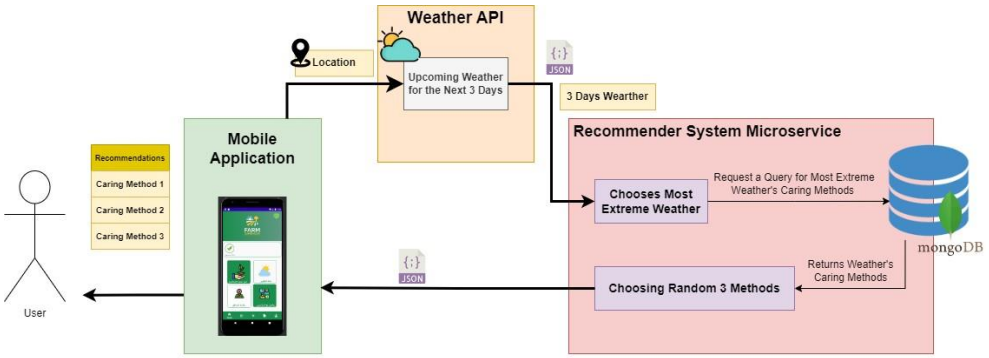


Figure 4-6: Weather Recommendations API

E. Collaborative Based Recommender

The collaborative recommender that intends to generate a pest or a disease recommendation for the farmer based on their entered symptoms and information, requires storing each farmer's problem report including the given symptoms and chosen pest or disease as the classification, as the recommendations come from other farmers behaviors (collaboration).

This service requires a large dataset of chosen users' (farmers') pests and diseases with their entered symptoms and extra information, as the content-based recommender is implemented with the addition of saving the farmer's entered pest's and diseases from the application's side on the database to be used as the dataset, the collaborative based recommender can be released.

For now, the Django service contains an alpha version of the collaborative-based recommender that can be evaluated when the application has the required amount of data for it, as it uses the Scikit-Library where it implements a random forest classifier which takes the entered farmer's data, preprocesses it then extracts the features from the preprocessed symptoms text before having the final classification for the pest or the disease.

Again, all that can be done after implementing the application with the content-based recommender with the addition of having the farmer choosing the right disease from the recommender ones to be the classification, before having the data split into training and testing to build the collaborative model, choosing the exact needed features, evaluating it, and finally releasing it in the mobile application.

4.3. Front-end User Interface

The Farm Supervisor application provides interfaces for farmers seeking recommendations about their crops, offering login and registration pages. The platform features a homepage for users, including functions that allow farmers to view and edit their personal profiles. It also incorporates a notification icon that displays all received notifications, including recommendations for crop management.

Furthermore, the application allows users to view information about their lands and the crops planted on them, providing the capability to edit crop details when a new crop is planted. Additionally, the application offers a dedicated section for crop care, where farmers can utilize a search bar to identify specific pests affecting their Avocado plants. Clicking on the pest's name displays detailed information and a picture, ensuring the farmer accurately identifies the pest affecting their plant.

The application also includes two icons enabling farmers to input information about their Avocado plants. One icon allows users to add a description of the pest type, along with details about the pest's symptoms. By providing this information, the application generates predictions for the most likely pests to affect the plant. The second icon is for inputting information about symptoms observed on the plant, with the application providing recommendations for predicted diseases that may impact the plant.

4.3.1. Registration & Login Pages

Registration in the Farm Supervisor application, as depicted in figure 4-7, involves opening an interface to enter the required user information such as email, name, password, etc. All fields are validated before being saved to the database. Once the registration process is complete, a dialog box appears, allowing the farmer to sign the application.

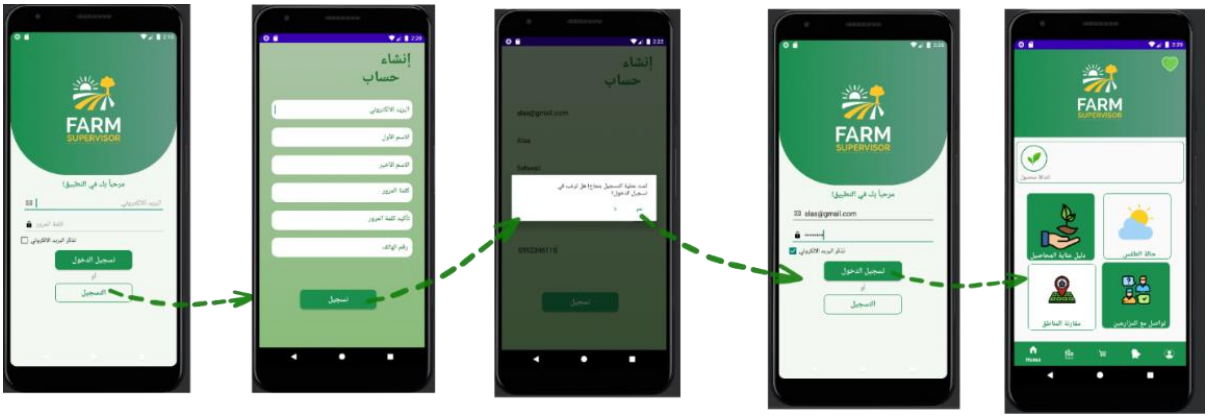


Figure 4-7: Farm Supervisor Registration & Login Page

The sign-in page displays fields for entering the email address and password. After the entered information is verified, the farmer gains access to the home page of the application.

4.4. Farmer View

This section provides an overview of the interfaces available to farmers after a successful registration and login process.

4.4.1. Crop Care Guide

This section allows farmers to search for specific types of pests that may affect the Avocado plant. It is designed for farmers who already know the names of the pests. When a farmer clicks on the name of a pest, an interface opens, displaying information about the pest, including a picture. This feature assists farmers in understanding the appropriate remedies and care methods to apply to the Avocado plant, preventing damage caused by the identified pest as figure 4-8 shows.

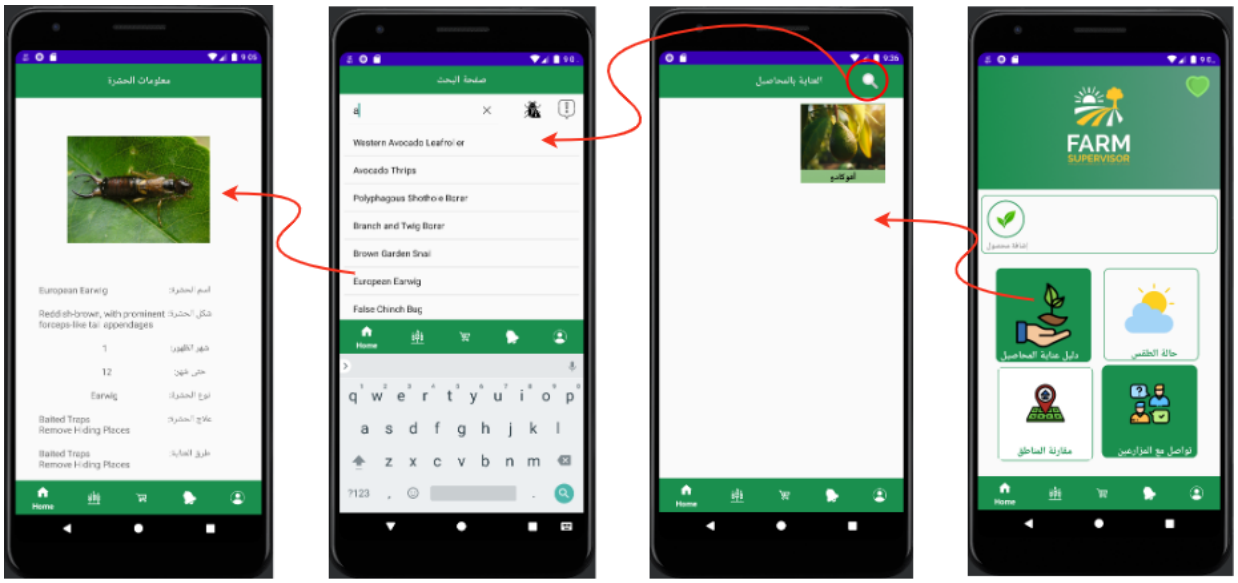


Figure 4-8: Search Pest

Additionally, there are two icons, one specifically designated for pests. Clicking on this icon opens a new interface displaying groups of checkboxes. Each group is designed for collecting information related to the pest.

The first group allows the user to specify the type of pest (Fly, Mite, Snail, etc.), the second group is intended for visual information about the pest, enabling users to describe its features (color, legs, etc.), and the last group is dedicated to documenting the symptoms observed on the plant (nests, loss of leaves, sunburn, sugary material, etc.). After clicking the send button, a new dialog displays information about the most recommended pests, as figure 4-9 shows.

Figure 4-9: Pest Recommendations First Icon

The other icon, specifically designed for symptoms, opens a new interface when clicked. This interface contains a list of checkboxes that describe the symptoms observed on the Avocado plant. When the farmer selects the checkboxes corresponding to the symptoms present on the plant and sends the information, the application opens a new dialog displaying the most recommended pests. This enables the farmer to address the pest issue based on the suggested remedy and care for the plant according to the recommended methods, as figure 4-10 shows.

Figure 4-10: Pest Recommender Second Icon

4.4.2. Weather Condition

This page shows the weather expectations for the current day. The user will have to give permission to detect his location, so the application asks the user for the location permission. Figure 4-11 shows the request to allow the application to access the phone's location.

Figure 4-11: Location Permission

Then, the user location will be stored depending on his location that was detected by the app. This operation demands some permissions and dependencies that were added to the application. If the user denied permission, he could use the app normally, but he will not be able to get the weather forecast information for his location. After the user gives permission, the app will detect his location and then he continues to use the app.

The page shows the city name, a photo describing the weather state, current day and date, current temperature, and a written weather status. Then the page has three important elements, which are the humidity, the rain percentage, and the wind speed. In the lower part of the page, you get the weather forecast for the next hours (the hour, photo for the weather state, and the temperature in this hour). Figure 4-12 shows the weather page.

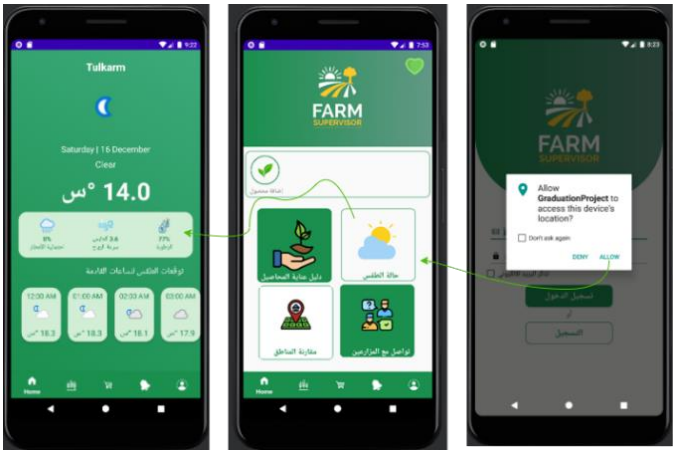


Figure 4-12: Weather Transactions

4.4.3. Communicate with Farmers Page

This page lists several farmers who offer various types of services, including plowing, pruning, and spraying. Users can view the farmers providing each service along with their names and phone numbers. Additionally, farmers can add any new service they offer that will be displayed to other users of the application. Figure 4-13 displays the communication page and other related pages.

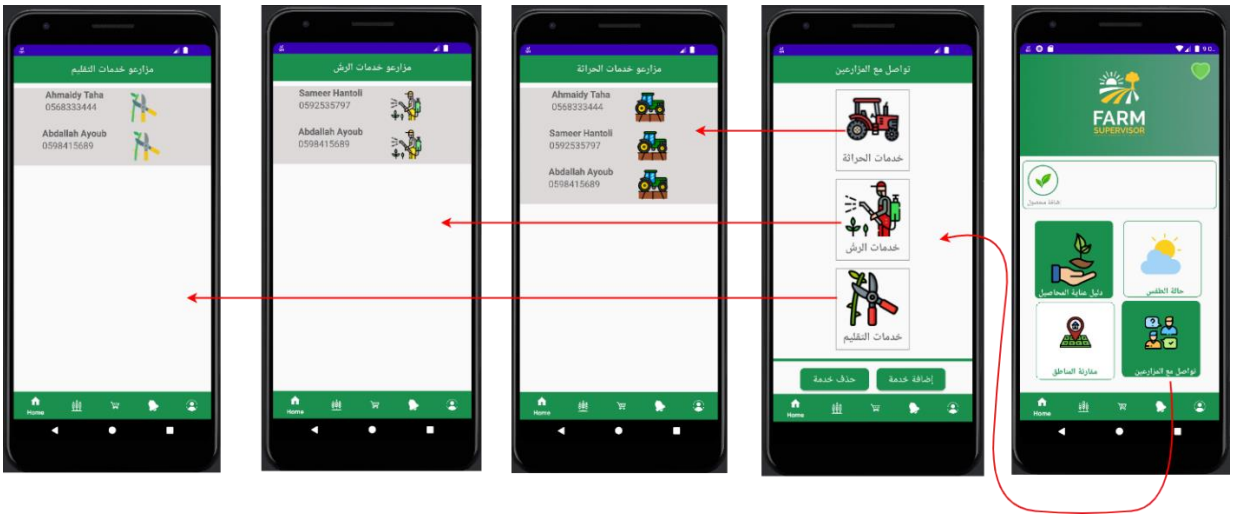


Figure 4-13: Communicate with Farmers Page

Figure 4-14 displays the pages for adding or deleting services. Users can view their current services and select the type of service they want to add from a drop-down list.

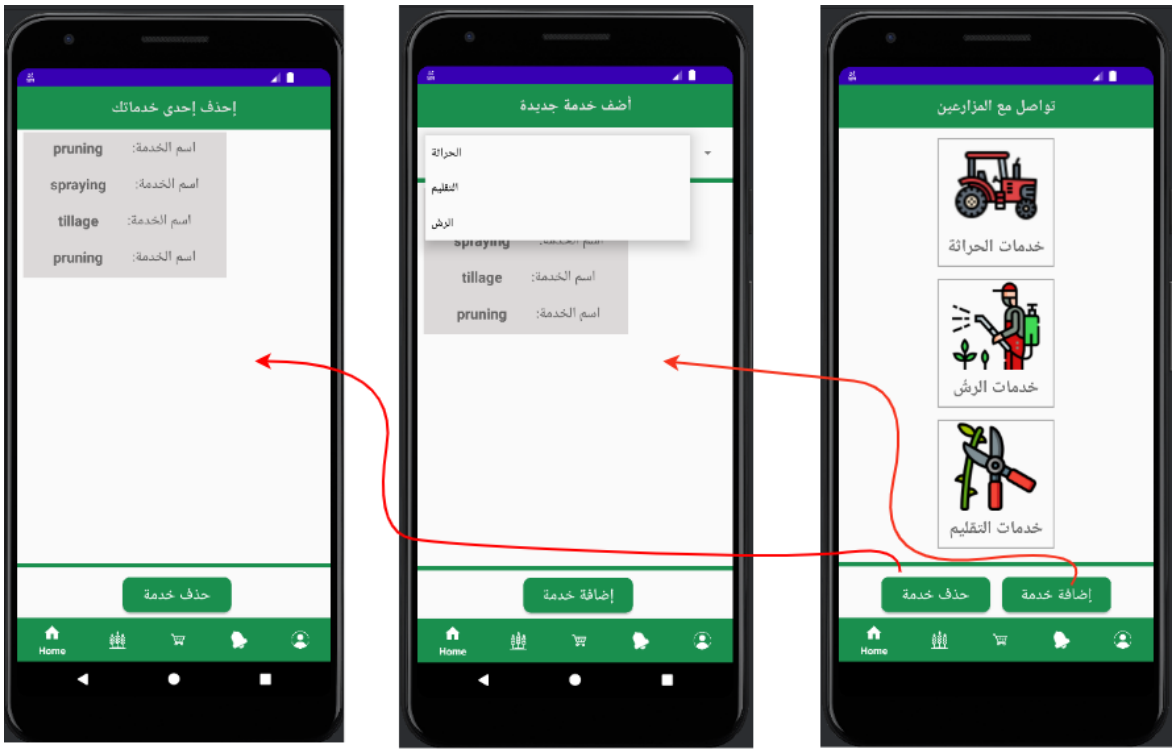


Figure 4-14: add/delete service

4.4.4. Application's Notifications

The mobile application can send notifications to the farmers having the application, these notifications can be either warnings for pests or weekly reminders, each notification provides a page having the data related to the notification, there are two notifications providing the following:

A. Weekly Recommendations Based on the Weather

This notification appears in two cases when first logging in to the application, and then once every three days with calling a broadcast receiver that sends the notification every three days.

The notification comes with the weather recommender service in the Django application, as it takes the three caring methods and their tools with the upcoming weather title and icon. Figure 4-15 shows the notification with the weather recommendations page.

Figure 4-15: Weather Recommendations Notification & Page

Another way to get to the weather recommendations page is from the weather page itself, where the user can get those recommendations at any time as shown in figure 4-16.

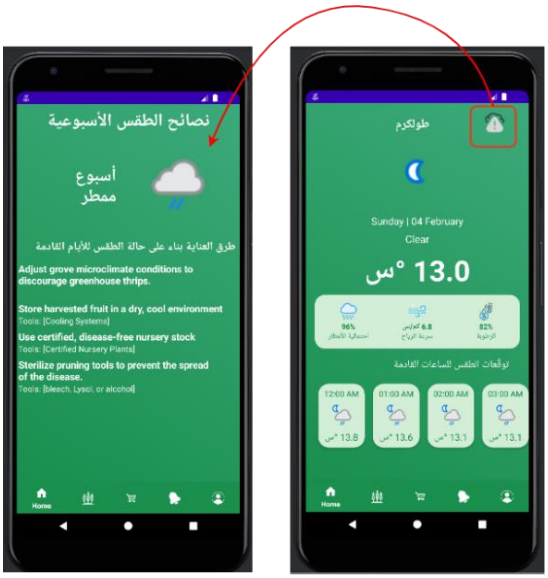


Figure 4-16: Weather Recommendations Page

B. Pest Separation in Farmer Area

This notification also appears in two cases first when first logging in to the application (on the land's page) and then when a pest is spread (which needs the pest alert service to be ready as the farmers stores their crop's pests on the Django application).

To check the page's workflow, the location that is set on the Android Studio's emulator "Mountain View" is set to have the spread pests in the MongoDB, so when the notification appears the application receives all the pests that are spread in the farmer's location with the current number of cases (minimum is three), also including an image of the pest. Figure 4-17 shows the notification appearance and the pest alert page on the application.

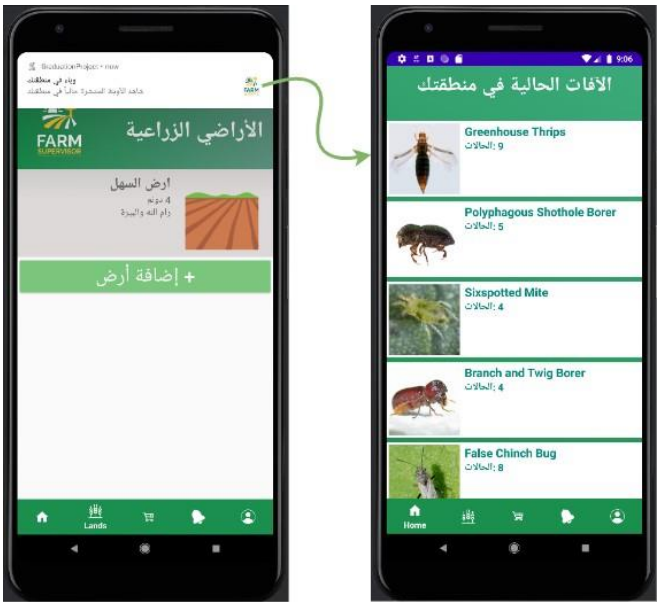


Figure 4-17: Spread Pests in Farmer's Location Notification & Page

4.4.5. Profile Page

Farm Supervisor allows the farmer to view and edit their profile, which includes basic information such as name, email address, phone number, and password. The farmer can make edits and save the updated information to the database. Additionally, the application provides an option for the farmer to sign out of their account, as figure 4-18 shows.

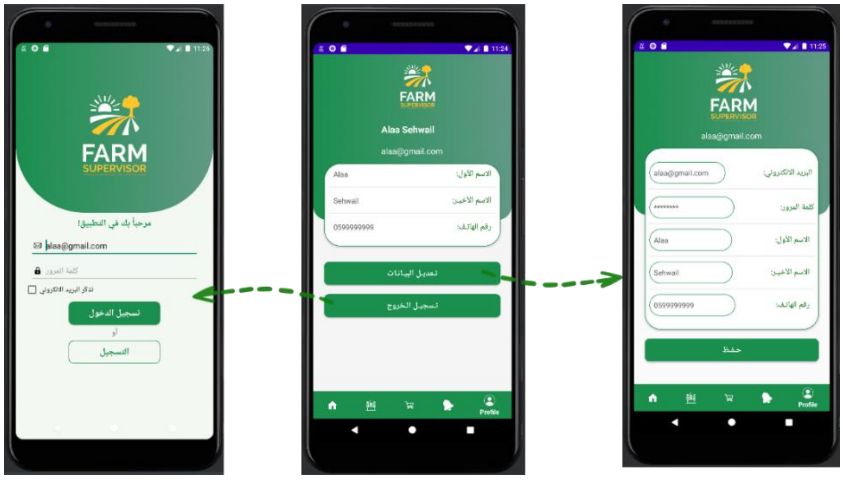


Figure 4-18: Farm Supervisor Profile Page

4.4.6. Markets Page

This page enables users to browse the available markets provided by the app and view the items available in each market. This feature helps users to identify where they can find specific items and the corresponding prices. The section includes markets for agricultural products and equipment. The user can get to this section from the bottom bar, by clicking on the shopping cart icon as shown in figure 4-19.

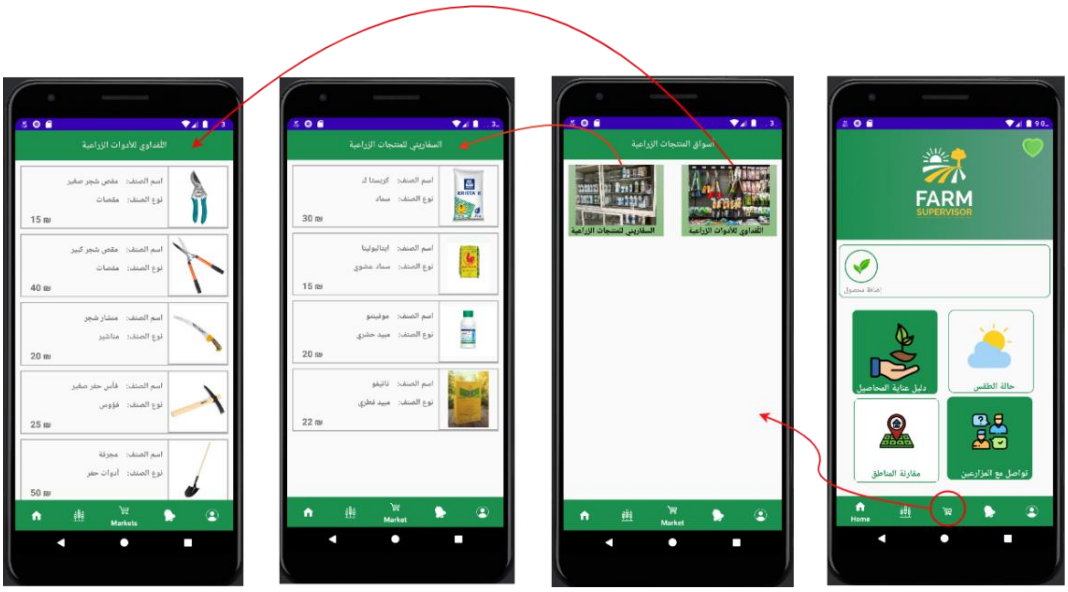


Figure 4-19: Farm Supervisor Market Page

4.4.7. Lands & Crops Tracking Page

This page shows all the farmer's lands, where the farmer can check each land's information, and planted crops. The page also includes the addition of the farmer's land. Figure 4-20 shows the land's page implementation and the addition of a new land.

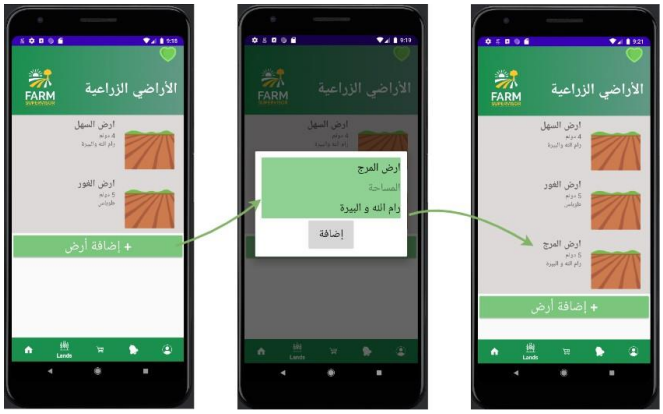


Figure 4-20: Lands Page

While clicking on each land, its crops are displayed, where the page shows the crop's information alongside checking each crop's record, Figure 4-21 shows the crop's page and the implementation of adding a new crop, and figure 4-22 shows crop record page and the implementation of a new record.

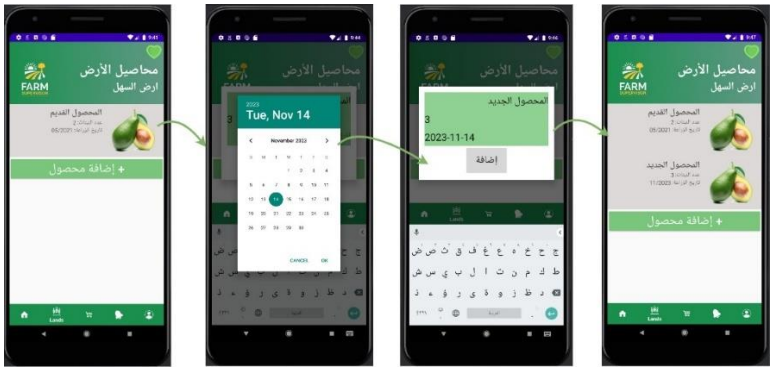


Figure 4-21: Land Crops Page

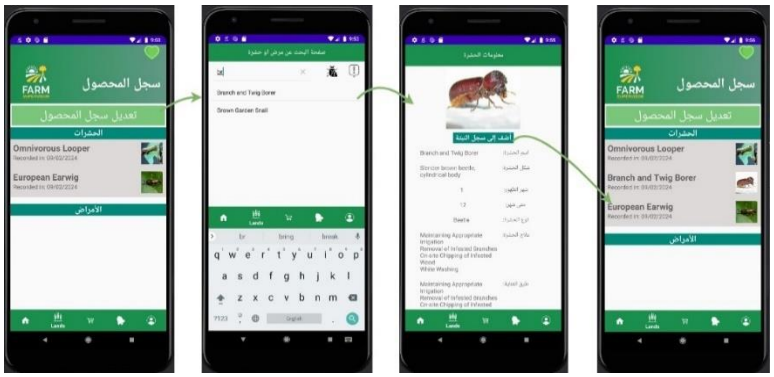


Figure 4-22: Crop Record Page

Chapter Five: Conclusion, Challenges & Future Work

This chapter explains the conclusion at the current point, the challenges faced, and the future plans for the application.

5.1. Conclusion

Farm Supervisor is providing an easy-to-use environment for Palestinian farmers seeking assistance and suggestions to help them thrive and make identifying issues easier. Additionally, Farm Supervisor offers various features, such as the ability to gain information related to their crops, access weather updates for their agricultural region, and get recommendations for any problem with his crops. Farmers are receiving essential notifications regarding the spread of pests and weekly weather issues. This is what Farm Supervisor offers during this current phase, and further developments will include additional features and the implementation of the mentioned functionalities above as the current features give high potential for the next step.

5.2. Challenges

During the implementation phase of Farm Supervisor, we encountered several challenges. The most important was accessing actual and suitable data sets, especially during the conflict on the Palestinian land that limited the resources for the application alongside arranging meetings with many agricultural sectors. Another challenge is in the Palestinian society as lack of collaboration and the poor knowledge when comes to technical side for the farmers, yet it can be said that the biggest issue is the occupation which destroys the agricultural sector in Palestine by limiting resources, taking the lands, and forcing the farmers to leave the jobs and go work in his settlements.

5.3. Future Work

Since the current work is a part of the university's graduation project for the current semester, many features and elements for the application are still up to me added, these planned features are:

1. **Provide Plant Data in Arabic:** As seen for the application when it comes to the plant related data it is provided in English, as how it was found on the resources, so the first thing to be done is to include it translated to the Arabic where the pest's diseases with their caring methods, cures and other related data to be in Arabic for the farmer to understand it.

2. **Improving Image's Storage:** Since the application requires images for crops, pests, market merchandise, these elements required images can be improved to be stored on an online server with their link being stored to be imported from the database.

3. **Improving the Addition of Crop's Records:** Improving the crop records so that the users can add a crop based on the recommendations they receive.

4. **Implementing the Collaborative Recommender System:** After having enough number of chosen pests and diseases for entered symptoms the dataset needed for the collaborative recommender can implemented to build the model that can be used to give farmers recommendations related to other farmers, as the architecture for the model is ready but needs the dataset to implement it on.

5. **Adding Plant's Age Stage:** Implementing the age stage of the plant in the application with its needed caring methods and related infects from both pests and diseases, as it can be an initial feature that determines the recommendations.

6. **Adding More Plants:** As the application is currently limited to the avocado plant, more plants will be added to the application with their related ages stages, pests, diseases, and weather conditions, alongside having the farmers to track their crops.

7. **Improving the Market:** The market that is accessible from the application will be improved as more features will be added involving customers checking agriculture-related merchandise and where to buy them from with the use of the application.

8. **Existing Crops Recommendations:** Farmers will start getting notifications that are related to their existing plants, for example if they need inspection or irrigation as reminders.

9. **Sharing Lands and Crops Between Farmers:** The application will have the ability to make land or crops belong to more than one farmer, where a farmer's actions on a plant becomes available to the other farmer.

10. **Allowing the Application to Receive Requests:** The mobile application will be set to receive requests to get notifications that are related to other user's actions such as messaging and notifications, this can be implemented with the use of Firebase Cloud Messaging.

11. **System Deployment:** When the application is ready to be launched, each service alongside PostgreSQL will be deployed with a cloud computing platform such as Amazon Web Services or Google Cloud.

12. **Mobile Application & Micro Services Improvement:** The mobile application will be improved to have a better performance when it comes to its data delivery speed. In addition to adding more plant, pest, or disease related images which can be very useful for the farmers in helping them understand their situation, and plant's needs.

The planned future work is summarized in the diagram in figure 5-1.

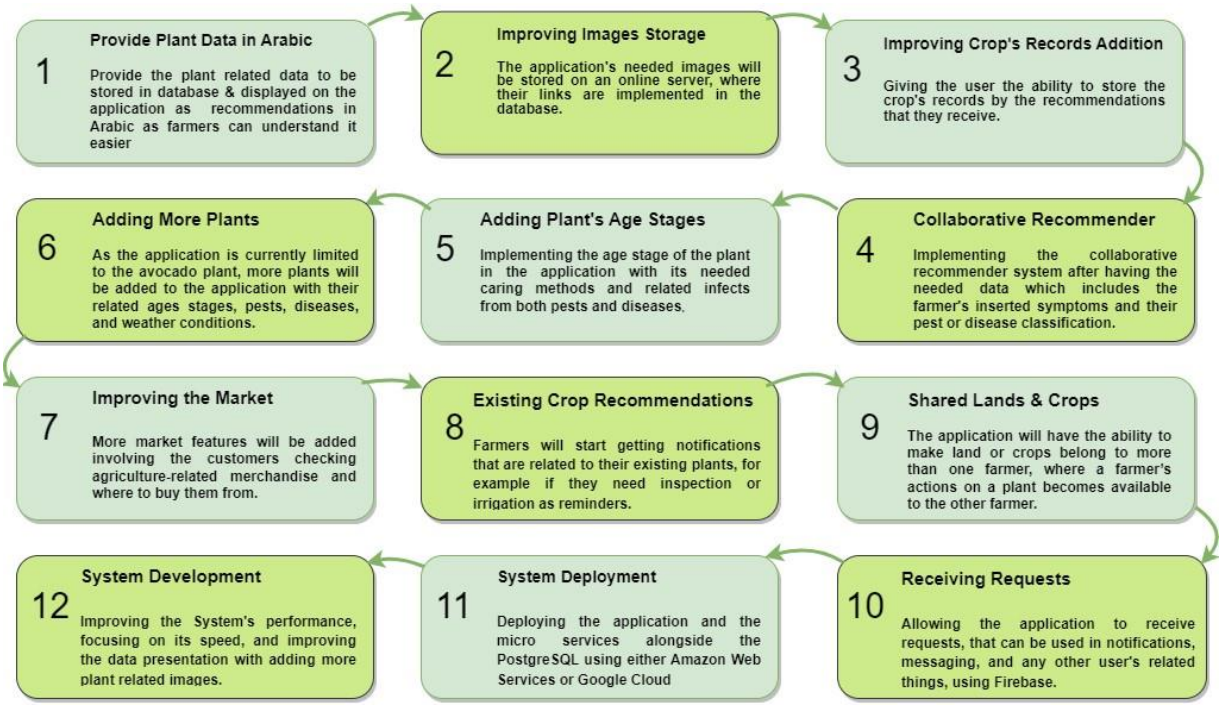


Figure 5-1: Future Work Diagram

References

[1] Cowan, D. (n.d.). “Exponential Growth. The Science of Machine Learning” https://www.ml-science.com/exponential-growth?fbclid=IwAR2AmAxePWrS6kzncXveQF4IFl_HyUTJ-SzdjRjkO76y13b5P-VD3pKnJmY Accessed on May 6th, 2023.

[2] K. Shah, A. Salunke, S. Dongare and K. Antala, (2017) “Recommender systems: An overview of different approaches to recommendations, - IEEE Xplore. https://ieeexplore.ieee.org/abstract/document/8276172/ Accessed on May 7th, 2023.

[3] Neptune.ai. (n.d.). “Recommender Systems: Lessons from Building and Deployment”. https://neptune.ai/blog/recommender-systems-lessons-from-building-and-deployment Accessed on May 8th, 2023.

[4] Turing. (2022, July 29). *How collaborative filtering works in Recommender Systems*. How Collaborative Filtering Works in Recommender Systems. https://www.turing.com/kb/collaborative-filtering-in-recommender-system?fbclid=IwAR0us0FmN0G4vBPwZ0zGfWyhcs7EubwPyxtnp5Ag9eB0DPOcclJHffL9zU8 Accessed on May 8th, 2023.

[5] Google. (n.d.). *Collaborative filtering | machine learning | google for developers*. Google. https://developers.google.com/machine-learning/recommendation/collaborative/basics?fbclid=IwAR0UPGlWx7zUojLbAGzrX35mu5DRAy-UKjBHuzqxp8Oznz5rWbFCinRHxB0 Accessed on May 9th, 2023.

[6] Google. (n.d.-b). *Content-based filtering | machine learning | google for developers*. Google. https://developers.google.com/machine-learning/recommendation/content-based/basics Accessed on May 9th, 2023.

[7] C. Song and H. Dong, (2021) "Application of Intelligent Recommendation for Agricultural Information: A Systematic Literature Review," https://ieeexplore.ieee.org/document/9611248 Accessed on Jan 14th, 2023.

[8] V. Suma, R. A. Shetty, R. F. Tated, S. Rohan and T. S. Pujar, (2019), “CNN based leaf disease identification and remedy recommendation” https://ieeexplore.ieee.org/document/8821872 Accessed on Jan 15th, 2024.

[9] panelS. Bangaru Kamatchi, & AbstractEstablishing linkages between Meteorological and climatic data. (2020, February 27). “Improvement of crop production using recommender system by weather forecasts”. Procedia Computer Science. https://www.sciencedirect.com/science/article/pii/S1877050920300314?ref=pdf_download&fr=RR-2&rr=84fa648f29210dab Accessed on Jan 16th, 2024.

[10] Hulsure, A., Kale, Y., Kalekar, A., Banswani, N., & Ganesh, V. (2021). "Weather Forecasting & Crop Recommendation. International Journal of Engineering Research & Technology (IJERT), 10(5)". https://www.ijert.org/weather-forecasting-crop-recommendation Accessed on Jan 17th, 2024.

[11] Agriapp. (n.d.). Agriapp. https://agriapp.com/ Accessed on May 19th, 2023.

[12] *Lite*. (n.d.). Bushel Farm. https://bushelfarm.com/lite Accessed on May 19th, 2023.

[13] *FarmGo*. (n.d.). FarmGo - Yara Smallholder Digital Solutions. https://yaradigital.com/en/farmgo#Features Accessed on May 19th, 2023.

[14] *Digital tools to analyze crop performance and maximize return*. (n.d.). Digital Tools to Analyze Crop Performance and Maximize Return. https://climate.com/features Accessed on May 19th, 2023.

[15]. Atari, Riad. (September 2023) Personal interview. Year. Palestinian Ministry of Agriculture.

[16]. Duhaidi, Sameer. (May 2023). "Personal Communication." *pointed out the necessary needs for the Palestinian farmers which a mobile application can help in*.

[17]. Odeh, Mohammad. "Avocado Farming in the Qalqilya Governorate". (August 2022)

[18]. PlantVillage. (n.d) "Avocado | Diseases and Pests, Description, Uses, Propagation." Pennsylvania State University, https://plantvillage.psu.edu/topics/avocado/infos Accessed on October 21st, 2023.

[19]. *Avocado*. (n.d.). Avocado / Agriculture: Pest Management Guidelines / UC Statewide IPM Program (UC IPM). https://ipm.ucanr.edu/agriculture/avocado/ Accessed on October 22nd, 2023.

[20] *Mobile App Architecture Basics. principles, layers, examples*. Forbytes. (2023, April 28). https://forbytes.com/blog/mobile-app-architecture-basics/?fbclid=IwAR1Il4PGU_QNcYoy86TUayEoWowMJUJqY_57p6eCXmN-FshFRxyujLsommI Accessed on January 20th, 2024.

[21] Wikimedia Foundation. *MongoDB*. Wikipedia. https://en.wikipedia.org/wiki/MongoDB Accessed on October 13th, 2023.

[22] Wikimedia Foundation. *Android Studio*. Wikipedia. https://en.wikipedia.org/wiki/Android_Studio Accessed on January 27th, 2024.

[23] Coheam, A. (2020, February 13). *Fuzzywuzzy*. PyPI. https://pypi.org/project/fuzzywuzzy/ Accessed on January 3rd, 2024.

[24] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825-2830. Accessed on January 28th, 2024.

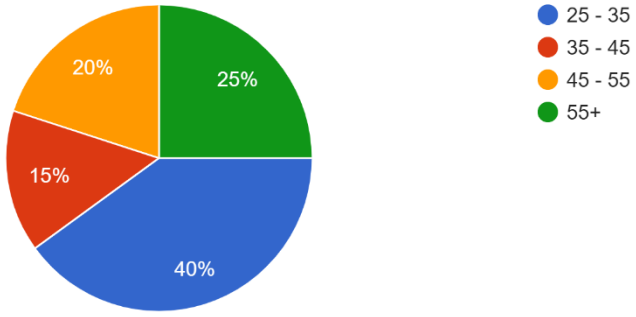
[25] "WeatherAPI.com. (n.d.). World Weather API - Online Weather API. Retrieved from https://www.weatherapi.com/api.aspx" Accessed on November 18th, 2023.

Appendix

Appendix 1: Questionnaire responses

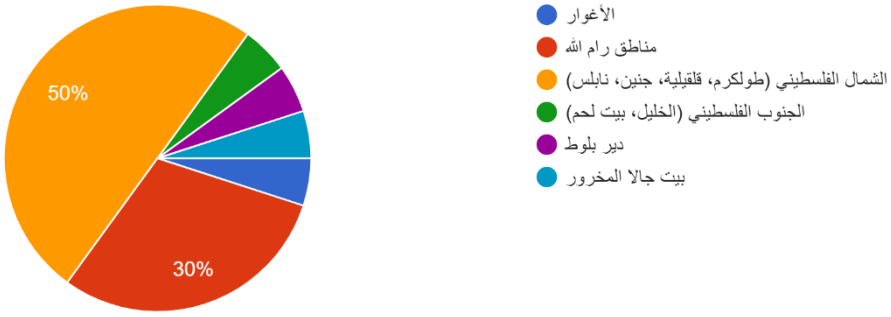
الفئة العمرية

20 responses



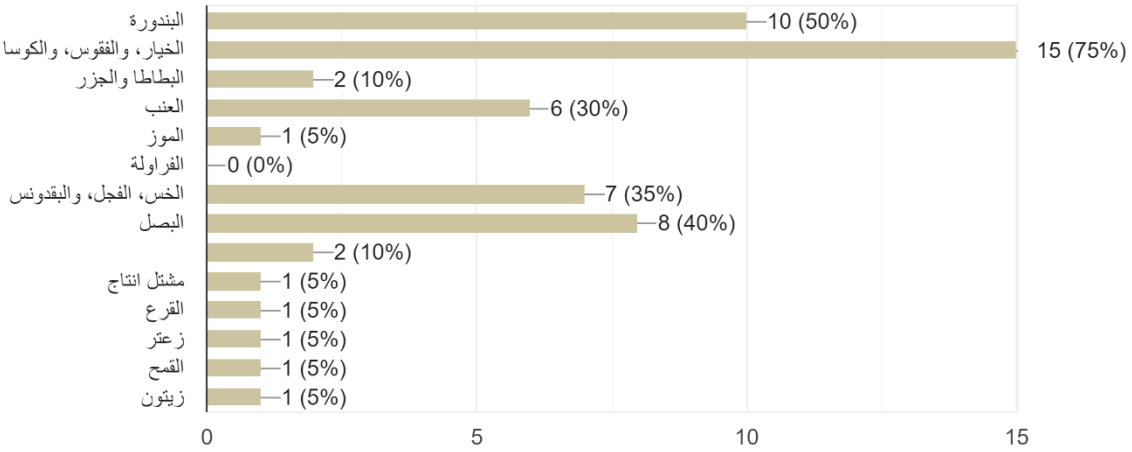
مكان الزراعة

20 responses



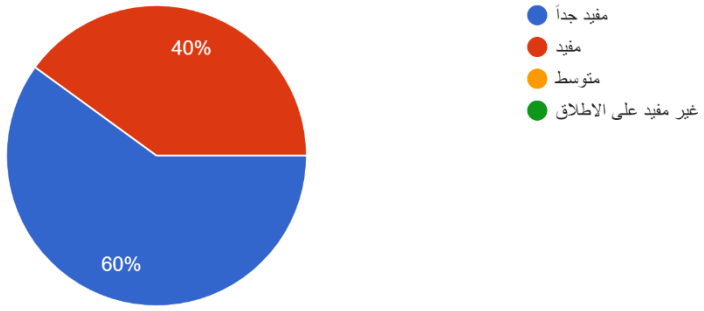
ما هي أكثر أنواع المحاصيل الزراعية التي تقوم بزراعتها او الإهتمام بها؟

20 responses

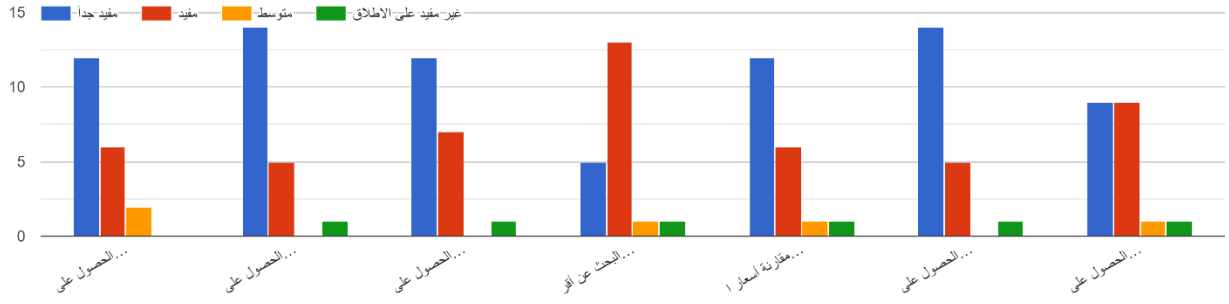


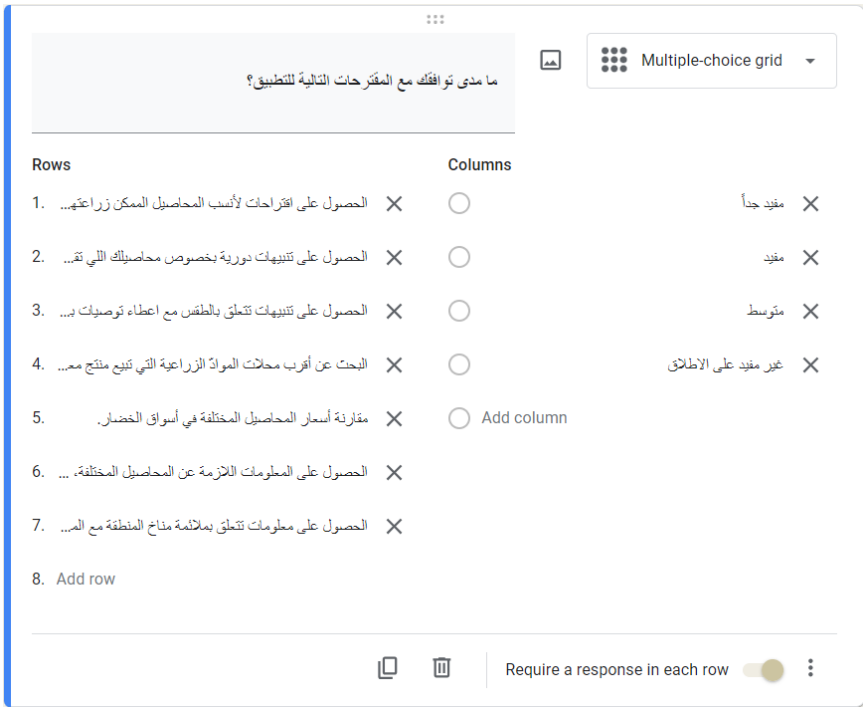
هل تعتقد أن وجود تطبيق هاتف يختص بتقديم توصيات تخص الزراعة سيكون مفيدًا لك في عملك الزراعي؟

20 responses



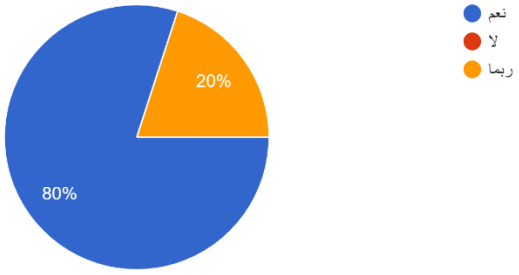
ما مدى توافقك مع المقترحات التالية للتطبيق؟





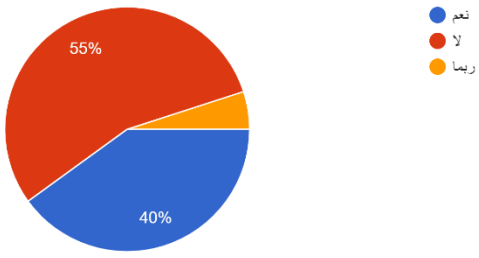
هل تظن أن هذا النوع من التطبيقات قد يساعدك في الاستفادة من خبرات المزارعين الآخرين أو من نشاطاتهم الحالية؟

20 responses



في حال استخدامك للتطبيق هل تمانع في نشر نتائج محاصيلك الزراعية، بغرض إفادة المزارعين الاخرين؟

20 responses



Appendix 2: Farm Supervisor brochure

Final Report

ORIGINALITY REPORT

11%	7%	3%	6%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Birzeit University Main Library Student Paper	3%
2	farmlogs.com Internet Source	1%
3	forbytes.com Internet Source	1%
4	digital.lib.usu.edu Internet Source	1%
5	Caixia Song, Haoyu Dong. "Application of Intelligent Recommendation for Agricultural Information: A Systematic Literature Review", IEEE Access, 2021 Publication	1%
6	www.ijert.org Internet Source	1%
7	Submitted to universititeknologimara Student Paper	<1%
8	docshare.tips Internet Source	<1%

9	iarjset.com Internet Source	<1 %
10	Recommender Systems Handbook, 2011. Publication	<1 %
11	Submitted to Federal University of Technology Student Paper	<1 %
12	Submitted to Associatie K.U.Leuven Student Paper	<1 %
13	climate.com Internet Source	<1 %
14	Submitted to DeVry, Inc. Student Paper	<1 %
15	Submitted to HTM (Haridus- ja Teadusministeerium) Student Paper	<1 %
16	Submitted to Informatics Education Limited Student Paper	<1 %
17	pergamos.lib.uoa.gr Internet Source	<1 %
18	Kulkarni, Shruti. "Machine-Learning-Assisted Recommendation System for Financial Organizations", Westcliff University, 2023 Publication	<1 %
19	Submitted to University of Sheffield Student Paper	<1 %

20	www.appypie.com Internet Source	<1 %
21	backend.orbit.dtu.dk Internet Source	<1 %
22	dora.dmu.ac.uk Internet Source	<1 %
23	link.springer.com Internet Source	<1 %
24	mic.ucmo.edu Internet Source	<1 %
25	elibrary.stipram.ac.id Internet Source	<1 %
26	projekter.aau.dk Internet Source	<1 %
27	Submitted to PSG Institute of Management Coimbatore Student Paper	<1 %
28	Recommender Systems, 2016. Publication	<1 %
29	dev.pymvpa.org Internet Source	<1 %
30	devitjobs.uk Internet Source	<1 %
31	digiresearch.vut.ac.za Internet Source	

		<1%
32	elibrary.tucl.edu.np Internet Source	<1%
33	fastercapital.com Internet Source	<1%
34	ipfs.io Internet Source	<1%
35	repository.president.ac.id Internet Source	<1%
36	www.gelbukh.com Internet Source	<1%
37	www.repositorio.unicamp.br Internet Source	<1%
38	www.researchgate.net Internet Source	<1%

Exclude quotes	On	Exclude matches	< 8 words
Exclude bibliography	On		