CHAPTER I

INTRODUCTION

1.1 Rationale of the Study

Banana cultivation holds significant economic and nutritional value globally, with millions of people relying on the versatile fruit for sustenance and livelihoods. In the Philippines, where agriculture forms the backbone of the economy, the cultivation of Cardava banana, a local variety of banana in the country, stands out as a vital component of agricultural production with it being a priority commodity in Davao Oriental due to its high-performance growth in export, projected at 10 percent annually. Cardava banana robustness, coupled with their versatility in culinary applications has made a prominent place in Filipino cuisine, frequently appearing on dining tables and incorporated into various local dishes. It also exhibited as a staple crop and an essential source of income for smallholder farmers across the archipelago (*Invest in Davao Oriental, n.d.).*

However, the prosperity of banana farming faces a formidable obstacle in the form of plant diseases, the Bugtok disease. This bacterial ailment, widespread and endemic in various regions of the Philippines, poses a significant threat to banana cultivation, particularly affecting cooking banana cultivars like the Cardava banana. The impact of Bugtok disease on banana yields is profound, with reported reductions ranging from 26.61 to 38.53 percent, causing substantial economic losses and undermining food security efforts (*Bugtok or Tibagnol Disease of Banana*, *n.d.*).

During the inspection on May 13, 2024 at Don Martin Marundan, Mati City, Davao Oriental, a Cardava farmer expressed her sentiment about the destruction of Bugtok disease to her cultivation resulting in more than ten Cardava banana plants to be forcefully cut down. With limited external symptoms of Bugtok disease, other than cutting through the plant and fruit pulp, an expert agronomist can identify the early symptoms of the disease so it is likely to proliferate on the entirety of Cardava farm without proper inspect.

Traditional methods of disease detection, reliant on visual inspection by trained agronomists, are labor-intensive, time-consuming, and subject to human error. Moreover, the subjective nature of visual assessment may lead to inconsistencies in disease diagnosis, hindering effective disease management strategies (John et al., 2023). In light of these challenges, there is a growing imperative to explore innovative technologies that can augment existing disease detection methods and empower farmers with timely and accurate information for decision-making.

At this juncture, early detection of diseases in banana plants, coupled with pattern recognition, is crucial for timely intervention among banana farmers. Swift identification of symptoms indicative of Bugtok disease enables farmers to implement necessary measures promptly, preserving crop health, mitigating losses, and ensuring sustained productivity in banana cultivation. Advancements in image processing and GIS mapping offer promising avenues for revolutionizing disease detection in agriculture.

The proponents sought to contribute to the advancement of sustainable and resilient agricultural systems that can withstand the challenges posed by emerging threats such as Bugtok disease. They wanted to help small-scale banana farmers by developing a mobile application for early detection of Bugtok disease in Cardava and Saba banana cultivar. The app facilitates image processing to analyze the captured photo of the banana plant for effective control strategy for Bugtok disease. This would help the farmers to sustain productivity in Cardava/Saba banana cultivation.

1.2 Purpose and Project Description

The system was originally proposed in order to demonstrate a proactive approach to address the challenges of Bugtok disease in Cardava/Saba banana farm. The system served as an innovative approach to enhance timely intervention and monitoring, ultimately aiming to maximize the sustainability of Cardava production.

The Bani is designed with an intuitive interface that enables users to capture and scan photos of cross-section of the banana peduncle. Upon capturing, the system initiates an in-depth analysis of the images such as data preprocessing and augmentation, using MobileNet convolutional neural network (CNN) in color feature extraction technique to obtain relevant data to be used to perform a help for comprehensive diagnosis of the banana plant's health and potential issues. Once the analysis is complete, the system generates detailed results, which classifies the plant as healthy or diseased then integrated into the database. These results not only include the diagnosed condition of the banana plant but also capture additional metadata such as the date of the analysis. Next, is the data visualization using GIS mapping or Heat Map, to visualize the area of infected by the disease and as well as graphical data analysis interface. By centralizing this information within the database, the system facilitates efficient tracking and monitoring of banana plant health over time, enabling users to make informed decisions regarding cultivation practices, disease management, and overall crop health optimization.

The study is relevant as it empowers banana farmers with advanced technology for early disease detection and geospatial analysis, supporting sustainable agriculture, improving crop management, and enhancing food security.

1.3 Objectives of the Study

The main objective of this study is to assist the farmers in detecting early symptoms of Bugtok disease in their banana farm to prevent further escalation of the disease in the entire farm through image processing and GIS mapping. Specifically, the proponents were tasked with:

1. Designing and developing a mobile application called Bani enabling the farmers to detect Bugtok disease in Cardava/Saba cultivar which is capable of:
   1.1. Featuring an intuitive interface with an integrated camera for efficient image processing within the app.
   1.2. Capturing and scanning the cross section of banana peduncle for the disease.
   1.3. Preprocessing and augmentation of the data before further analysis of the image.
   1.4. Integrating MobileNet CNN to analyze uploaded photos of banana plant components. This algorithm extracts the color features for disease identification of banana plant.
   1.5. Integrating GIS Heat Map to visualize the Bugtok disease within Mati City.
   1.6. Incorporating an interface for data visualization analysis to generate graphical report.
   1.7. Creating a comprehensive database to store analyzed images and metadata, supporting efficient data management and long-term monitoring.
2. Examine the performance of the machine learning using the confusion matrix and accuracy metric evaluation. The confusion matrix was used to solve the accuracy metric evaluation of the model.
3. Evaluate the system using ISO/IEC 25010 standards.
4. Prepare for an implementation plan for the deployment of the system.

1.4 Significance of the Study

This study explored the potential benefits of *"Bani: Bugtok Disease Detection in Cardava and Saba Bananas Using Deep Learning and GIS Mapping"* in addressing the challenges of Bugtok disease in Cardava and Saba banana farm. After its applications was examined across different sectors, this research has reached its aim to shed light on how this system can offer diagnostic solution for individuals, organizations and communities seeking to optimize the banana farming by early prevention of disease:

Banana farmers. Farmers cultivating Cardava and Saba banana cultivars had benefitted from the findings of the study by gaining access to a more efficient and accurate method of detecting Bugtok disease. Early detection of the disease can help farmers take timely preventive measures, such as buying the necessary pesticides and bactericides thereby reducing crop losses and improving overall yield and profitability.

Researchers. Researchers in the fields of agriculture, plant pathology, and deep learning would find value in the study methodology and results. The study could contribute to the body of knowledge on disease detection in banana plants and serve as a reference for future research in similar areas. Academic institutions could incorporate the study findings into their curriculum, fostering education and innovation in agricultural sciences and technology.

Agrotechnology companies. Companies specializing in agrotechnology, agricultural biotechnology, and precision farming could leverage the study findings to develop innovative solutions and tools for disease detection and management in banana cultivation. This could lead to the creation of new products or services aimed at improving crop health and productivity for banana farmers.

Government Agricultural Agencies. Government agencies responsible for agriculture and rural development, as well as agricultural extension services, could use the study findings to inform their policies, programs, and initiatives aimed at supporting banana farmers. This could involve providing training, resources, and support for implementing disease detection and management strategies based on deep learning approaches.

1.5 Scope and Limitation

The scope of this study will be:

- The project involved design and development of Bani, a system that allowed users to manage their scan histories by saving or deleting them, and to take notes on each scan.
- The project encompassed building and training an image classification model by the use of TensorFlow to accurately detect Bugtok disease in Cardava/Saba bananas.
- The app worked on hybrid approach or it can be both online and offline.
- The app used CNN model architecture called MobileNet for image classification.
- The proponents used Google Maps API for the mapping with a geographical scope within Davao Oriental.
- Heat Map was used to collect data points of a banana farm per 1 hectare of land.

- Google Maps API was used to locate the area infected through collecting the coordinates (latitude, longitude).
- The app shaded the infected area to red while the healthy area is green.
- A minimum of one thousand (1000) images of Bugtok disease symptom found in the cross-section of banana peduncle and as well as at least one thousand (1000) images for healthy peduncle has been collected.
- Approximately 15 centimeters (cm) is being used in the distance between the camera and the surface of the freshly cross-section of peduncle in taking pictures.
- In training the model, supervised learning has been used.

The limitations are as follows:

- The performance of a CNN model like MobileNet is heavily dependent on the quality and quantity of the training data. If the data used for training is not diverse or ample, the ability of the model to generalize and accurately identify diseases in real-world scenarios may be limited.
- The system requires that banana peduncle cross-sections be captured immediately after cutting and without prolonged exposure to air. Since exposure to air and heat can cause discoloration and other changes in the peduncle, these factors could adversely affect the accuracy of disease classification by the model.
- Since mobile phones was used to gather data, image quality may vary based on the device model. To avoid capturing irrelevant data, images must be taken no further than 35 cm away. Low image quality due to factors like

insufficient resolution, poor lighting, or obstructed views may significantly impair the system's accuracy in disease identification.

- Data collection was limited to Mati city.
- The study was focused exclusively on Cardava/Saba bananas. The application is not designed for use with other trees or plants, and its accuracy may not extend to different test subjects.

1.6 Conceptual Framework

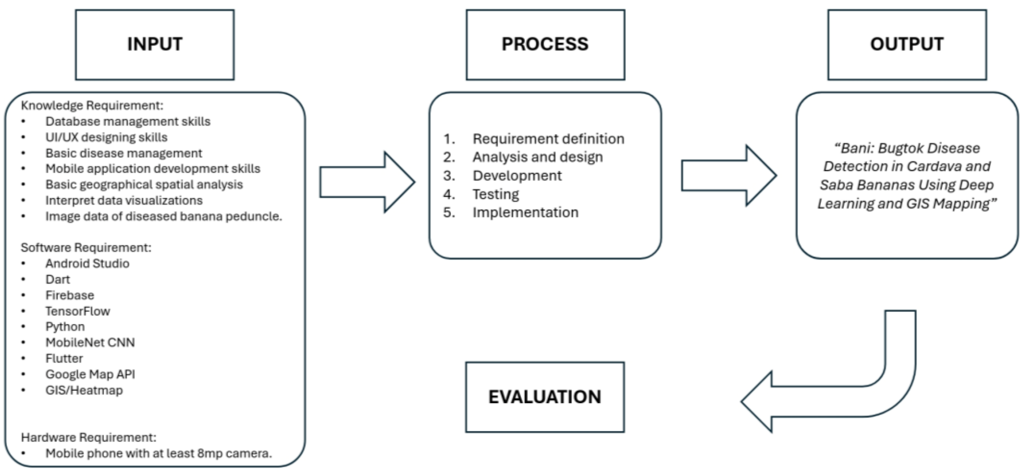

Figure 1.1 Conceptual Framework

The diagram provided illustrated the conceptual framework of IPO model guiding the development of the application, depicting the flow from user input through processing to output. Developers adhered to specific requirements: Dart and Flutter for constructing and deploying an appealing and intuitive mobile application; Python for programming algorithm; Flutter as the framework ensuring native performance on specified operating systems such as Android OS; TensorFlow for image processing; Firebase for data storage; Google Map API that allows the developers to integrate Google Maps into their applications; GIS/Heat Map to visually analyze and represent spatial data; and Android Studio for system development. To facilitate effective collaboration, developers will follow a software development life cycle, involving processes such as system examination, analysis, interface design, and code development. Implementation was proceeded once all client-required functions are integrated. Evaluation determined if the application needs further refinement or is ready for deployment.

1.7 Definition of Terms

The following terms are defined technically.

Bugtok disease is a local term in the southern Philippines used to describe the infected fruit which are discolored and hard even when ripe. It is used to describe different symptoms of the same disease caused by the bacterium *Ralstonia solanacearum* and was the main subject of this study (Soguilon, et al., 1995).

Cardava Cardava bananas, often mistaken for Saba bananas, are botanically classified within the Musa acuminate x balbisiana (ABB Group) ‘Saba’ category. They are primarily used for cooking and are prevalent in the Philippines, where they are among the most common cooking banana cultivars. Cardava banana appears to be the host of Bugtok disease (Specialty Produce, n.d.).

Convolutional Neural Network (CNN)	is an effective machine learning tool, particularly when used for computer vision applications. Convolutional neural networks, or CNNs, are a subclass of neural networks specifically engineered to handle data that resembles a grid, like pictures. This was used as an architecture for the image processing of banana bugtok disease detection (GeeksforGeeks, 2024).
MobileNet	MobileNet, TensorFlow's first mobile computer vision model, uses depthwise separable convolutions to significantly reduce the number of parameters compared to networks with regular convolutions. This results in lightweight deep neural networks. Released as open-source software by Google, MobileNet is ideal for creating compact and fast classifiers (Pujara, 2023).
Saba	Saba bananas, scientifically known as Musa acuminata x balbisiana, are a triploid hybrid of the ABB group within the Musaceae family. These short, thick, and angular bananas grow in compact clusters and originate from the Philippines, where they are widely cultivated for local consumption (Specialty Produce, n.d.).

CHAPTER II

REVIEW OF RELATED LITERATURE

2.1 Technical Background

Bani is a tool intended to provide comprehensive assistance to control strategy for Bugtok disease. It is a tool that helps farmers in detecting early symptoms of Bugtok disease, and ultimately increase the sustainability of Cardava and Saba banana production and farming. This application is available only in mobile phones to run the system. In developing the project, the following technology framework, tools and programming languages was used:

2.1.1 Dart

Dart is a programming language utilized for developing software applications, known for its simplicity in syntax and structure, making it easy to grasp. With its focus on object-oriented programming, Dart is object-oriented, meaning it focuses on creating reusable pieces of code called objects. It is used for building web, mobile, and desktop applications. Dart can be used for creating standalone programs or for developing apps using frameworks like Flutter. It is used for a variety of purposes, such as building user interfaces, handling data, and creating server-side applications (Badkar, 2023). This tool has been utilized as a programming language specifically for developing Android application systems within the Bani. Using this tool, developers built strong and efficient mobile apps specifically designed for the unique requirements of the Bani environment.

2.1.2 Flutter

Flutter is an open-source UI framework crafted by Google, empowers developers to craft high-quality mobile, web, and desktop applications using a unified codebase. It serves as a contemporary solution for producing visually appealing cross-platform applications, ensuring native performance across various devices and operating systems including Android, iOS, Linux, macOS, and Windows (Montaño, 2024). The researchers will use this framework to make the mobile application compatible to Android OS. The researchers utilized this framework to ensure the mobile application is fully compatible with Android operating systems. By doing so, it provided a seamless user experience across the two most widely used mobile platforms. This involved through testing and optimization to guarantee smooth performance and functionality on Android devices.

2.1.3 Firebase Realtime Database

The Firebase Realtime Database is a cloud-based NoSQL database that enables organizations to store and synchronize data in real time across all users' devices. This allows for the development of apps that remain consistently updated, even when users are offline (Hanna & Rosencrane, 2023). This tool has been useful to the development of the mapping system, since it allowed data to be stored and accessed offline, and then automatically syncs the data with the cloud once an internet connection is available. This ensured that the application remained functional and up-to-date regardless of connectivity status especially to real-world agricultural settings where internet connectivity can sometimes inaccessible.

2.1.4 Python

Python is a widely-used programming language employed for developing websites, software, automating tasks, and performing data analysis. Being a general-purpose language, Python is adaptable to various applications and is not tailored to any specific task. Its flexibility, coupled with its ease of use for beginners, has contributed to its widespread popularity in the programming community. It is also frequently utilized in website and software development, automating tasks, analyzing data, and visualizing information. Its accessibility has led to its adoption by individuals outside of the programming field, such as accountants and scientists, who leverage its simplicity for everyday activities like financial organization (Coursera, 2024). The researchers had utilized Python as the primary programming language for coding the image processing component within the TensorFlow framework. This decision reflects Python versatility, robustness, and extensive libraries, making it an ideal choice for implementing sophisticated image processing algorithms seamlessly integrated with TensorFlow's deep learning capabilities.

2.1.5 TensorFlow

TensorFlow, developed by Google, is an open-source platform and framework for machine learning that encompasses libraries and tools primarily based on Python and Java. Initially conceived for training machine learning models on data, TensorFlow has evolved to become a versatile tool suitable for both conventional machine learning and deep learning applications. Despite its original focus on large numerical calculations outside of deep learning, TensorFlow has emerged as a valuable resource for deep learning development as well. It supports data in the form of tensors, which are multidimensional arrays capable of managing vast amounts of data efficiently (BasuMallick, 2022). The researchers have employed TensorFlow as the framework for machine learning in the image processing of Bugtok disease detection. This decision is driven by TensorFlow's reliability, scalability, and wide range of pre-existing models, making it suitable for intricate image classification tasks. By utilizing TensorFlow's deep learning capabilities, the researchers improved the precision and effectiveness of Bugtok disease detection, ultimately advancing crop management practices and promoting sustainable agriculture.

2.1.6 Android Studio

Android Studio is the official integrated development environment (IDE) for developing Android applications. Built on IntelliJ IDEA, a Java-based IDE for software development, it includes IntelliJ’s code editing and developer tools (TechTarget Contributor, 2023). The researchers utilized Android Studio as their main tool for programming and running Dart code. This decision is based on the user-friendly interface, advanced features for coding and debugging, and smooth integration with Dart’s development environment. By using Android Studio, the researchers streamlined their coding workflow, boost productivity, and facilitate efficient development of Dart-based applications.

2.1.7 Convolutional Neural Network (CNN)

One kind of deep learning algorithm that works especially well for tasks involving picture recognition and processing is the convolutional neural network (CNN). Convolutional, pooling, and fully connected layers are some of the layers that make it up. The human brain’s visual processing served as the inspiration for

CNN architecture, which makes them ideal for identifying spatial connections and hierarchical patterns in images (GeeksforGeeks, 2024). The researchers have chosen CNN for a reason that it is more recently-used algorithm.

MobileNet

TensorFlow's first mobile computer vision model is called MobileNet. In comparison to other networks with regular convolutions and the same depth in the nets, it employs depthwise separable convolutions to drastically reduce the number of parameters. Lightweight deep neural networks are the outcome of this. Convolutional neural networks (CNNs) of the MobileNet class were made available as open-source software by Google, making them a great place to start for developing really small and incredibly quick classifiers (Pujara, 2023). Since the proponents will develop a mobile application, MobileNet is a suitable choice for this system development.

2.1.8 Google Maps API

The Google Maps API enables various software platforms to interact with one another. For instance, website owners can integrate Google Maps into their custom websites, like a real estate directory or a community services page (Zola, 2022). The researchers thought in this mapping project, Google Maps API can be highly beneficial in a mapping system project by providing robust tools for integrating real-time geographic data, enabling precise visualization of locations, and supporting features like geolocation, heatmaps, and custom markers, which enhance the analysis and presentation of spatial information in an accessible and interactive way.

2.1.9 Geographic Information System (GIS)

A Geographic Information System (GIS) is a computer system designed to analyze and visualize geographically referenced data. It worked with information that is linked to specific locations (United States Government for A Changing World, n.d.).

Heatmap

A heatmap is a graphical representation of data in two dimensions, where varying colors indicate different values. Heatmaps can be applied to various types of data, such as showcasing foreclosure rates in the real estate market, illustrating the distribution of credit default swaps (CDS), or analyzing website traffic by displaying the number of hits a site receives (Kenton, 2022). The researchers thought this tool, heatmaps are a valuable tool for visualizing spatial data density and patterns, enabling users to identify trends and make informed decisions by highlighting areas of high and low activity or intensity.

2.1.10 Adobe Photoshop CS6 (Version 13)

Adobe Photoshop CS6 is a leading software used for editing bitmap images across various fields, including game development, web and app design, fashion, and more (Adobe Labs, 2012). This tool was used during the preprocessing of the image dataset in which to remove the background of the image to reduce the noise.

2.1.11 Camera (iPhone 11 Pro Max)

This tool features a triple-lens 12-megapixel rear camera setup and a 12-megapixel front-facing selfie camera. The rear system includes an ultra-wide lens with an $f/2.4$ aperture and a 120° field of view, a wide lens with an $f/1.8$ aperture, and a 12-megapixel telephoto lens (Phonetradr, n.d.). This was used during the collection of the image dataset.

2.2 Related Literature

2.2.1 Cardava and Saba banana

Bananas are the top fruit crop in the Philippines, playing a vital role in both domestic and export markets. While Cavendish remains the leading export variety, Cardava is rapidly emerging as a key commodity, particularly in the processed food sector, such as banana chips (DA Press Office, 2022). In Davao Oriental, Cardava cultivation spans 7,652 hectares, producing 87,425 metric tons annually, with an expected growth rate of 10% per year (Province of Davao Oriental, 2024). Aside from its economic significance, Cardava also serves as a staple in local households, with its various snack forms—including banana cue, fried ripe Cardava, and turon—providing essential livelihood opportunities.

One of the main advantages of Cardava is its natural disease resistance, attributed to its *Musa balbisiana* genetic composition. Studies have highlighted its resilience against three major banana diseases. First, *Saba*, a variant of Cardava, demonstrated complete resistance to Fusarium wilt (TR4), with a 0% incidence rate in both primary and ratoon crops (Molina et al., 2014). Second, Cardava exhibits resistance to Black Sigatoka, as its *Balbisiana* genome prevents the disease from progressing beyond early infection stages (Kimunye et al., 2021). Lastly, research in Laguna, Philippines, revealed that wild *Musa balbisiana* accessions remained BBTV-free even after prolonged exposure, maintaining resistance for up to five years in field trials (Dela Cueva, 2023).

The ability of Cardava to withstand these devastating diseases reinforces its importance in sustainable banana farming. Its resilience not only benefits smallholder farmers but also offers valuable genetic resources for breeding programs focused on disease-resistant cultivars. Additionally, the increasing market demand for processed Cardava products strengthens its economic significance, making it a crucial agricultural asset in tropical regions like the Philippines.

2.2.2 Bugtok disease

The increasing prevalence of Bugtok disease threatens Saba and Cardava banana crops in the Philippines, impacting food security and livelihoods. These bananas, rich in essential nutrients, are vital to Filipino cuisine and processed products (Blomme, 2017). Bugtok, a bacterial wilt caused by *Ralstonia solanacearum* phylotype II, shares similarities with Moko disease but primarily affects cooking bananas, causing internal fruit rotting without visible wilting (Fegan & Prior, 2006). Originally linked to Moko, *R. solanacearum* race 2, biovar 1, has been introduced into the Philippines, with molecular analysis tracing its origin to Indonesia (Blomme, 2017).

Once a minor concern in the 1960s, Bugtok now poses a significant threat to bananas with a strong Musa balbisiana genetic makeup. Infected plants may appear healthy, but abnormal bracts and vascular streaking signal disease presence. Insect vectors spread the pathogen, with outbreaks reported in

Mindanao, Visayas, and Cavite, affecting 133 hectares in 17 barangays (Regional Crop Protection Center, 2023).

Despite Cardava's resilience, it remains vulnerable to Bugtok, necessitating monitoring and intervention. Bacterial wilt diseases, including Moko and blood disease, cause global economic losses, yet no resistant varieties exist. However, pilot studies in Bukidnon show promise in controlling Bugtok through early debudding and table salt application (Pava et al., 2003).

Bugtok's spread endangers banana production and related industries. Strengthening collaboration among researchers, farmers, and policymakers is crucial to developing disease-resistant varieties and sustainable management strategies.

2.2.3 MobileNet CNN

MobileNet, a lightweight CNN, is widely used in plant disease detection due to its efficiency and adaptability. Its depthwise separable convolutions reduce computational demands, making it ideal for mobile and embedded systems, enabling real-time disease detection and intervention. By analyzing leaf, stem, and fruit images, MobileNet helps diagnose plant diseases like blight, mildew, and rust, supporting early intervention and improved crop management.

Ashwinkumar (2022) developed OMNCNN, integrating MobileNet with machine learning to detect plant leaf diseases in India, addressing limited expert access and reducing productivity loss. Bi et al. (2020) demonstrated MobileNet's effectiveness in detecting apple diseases with precision comparable to ResNet152 and InceptionV3. Elfatimi et al. (2022) achieved over 97% accuracy in classifying bean leaf diseases, while Ghoury et al. (2019) applied MobileNet and Faster R-CNN for real-time grape disease detection, reducing economic losses.

In summary, MobileNet's efficiency and adaptability make it a valuable tool for automating plant disease detection. Its real-time capabilities and high accuracy contribute to improved crop management, mitigating agricultural losses, and enhancing global food security.

2.3 Related Systems

2.3.1 Plant Identification & Disease

The AI-powered plant identification system is a robust solution crafted to cater to the needs of gardening enthusiasts and botanical experts alike. Utilizing cutting-edge image recognition technology, it swiftly identifies thousands of plant species from a single snapshot, offering comprehensive insights into their natural habitat and growth patterns. This application was released on November 20, 2023 in Google Play Store and have not recently updated. Alongside this, users benefit from the expertise of the ChatGPT AI Assistant, providing round-the-clock assistance and resolving queries on plant care and horticultural matters. The system also offers personalized care guides, proactive disease detection, and an array of helpful insights and tips on plant care and management, making it an indispensable tool for plant enthusiasts (Kitchen Apps LLC, 2023).

Figure 2.1 Sample Image of Plant Identification and Disease (Kitchen Apps LLC, 2023)

The Plant Identification and Disease system offers AI-powered plant identification and a ChatGPT AI Assistant for expert plant care guidance. It provides personalized advice on watering, sunlight, and soil preferences, with 24/7 accessibility for continuous support. It is a mobile application similar to the proposed system, Bani, with a photo scanning interface for plant identification. However, unlike Bani, it does not detect Bugtok disease, cannot identify banana plant or fruit images, does not use MobileNet CNN, only functions online, and lacks an integrated GIS heatmap.

2.3.2 DoctorP

DoctorP is an innovative plant disease detection and treatment recommendation app, serving as a reliable advisor for users seeking to maintain the health of their crops. With a focus on a diverse array of crops such as apples, cherry, corn, grapes, and more, DoctorP utilizes state-of-the-art neural network technology trained on real-life datasets to accurately diagnose plant diseases. This application was introduced on September 27, 2021 in Google Play Store and was updated on February 8, 2024. Users can conveniently capture a photo of their plant or select an existing image from their device, prompting DoctorP to deliver precise diagnoses alongside comprehensive descriptions, examples, and treatment recommendations. It is important to note that an internet connection is necessary to access the app's diagnostic capabilities. Additionally, DoctorP encourages user feedback to enhance its database and neural networks continually (Uzhinsky, 2024).

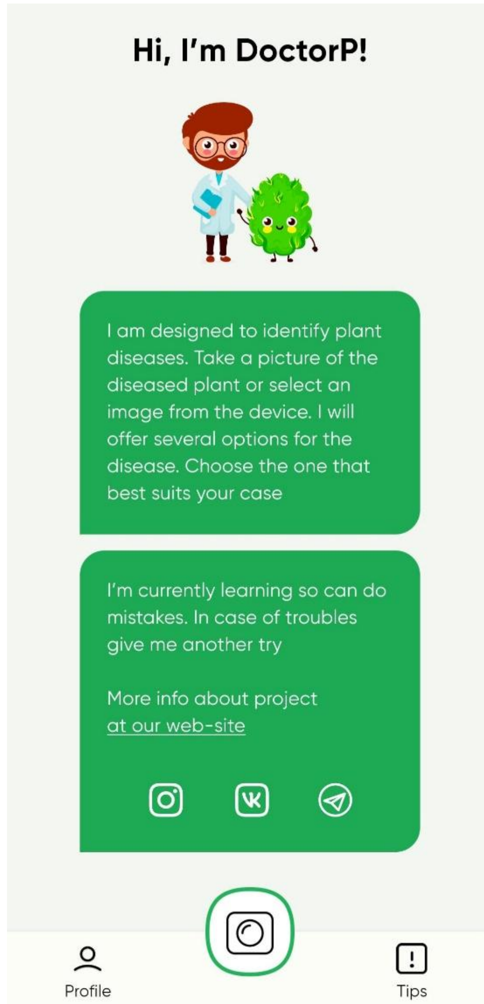

Figure 2.2 Sample Image of DoctorP (Uzhinsky, 2024)

DoctorP is a plant disease detection app that uses advanced neural networks for accurate diagnosis across a wide range of crops, including apples, cherries, corn, and grapes. It offers detailed descriptions, examples, and treatment recommendations for identified diseases. The app features a user-friendly interface for easy photo capture or image selection, but it requires an internet connection to access its diagnostic capabilities. This mobile application shares similarities with the proposed system, Bani, including a photo scanning interface for plant identification. However, it differs from Bani in that it cannot detect Bugtok disease, does not identify banana plant or fruit images, does not utilize MobileNet CNN, only operates online, and lacks an integrated GIS heatmap.

2.3.3 Plantix

Plantix, the premier agricultural app, revolutionizes crop management by offering instant diagnosis and treatment recommendations for crop pests and diseases. With over 10 million downloads worldwide, Plantix has earned its reputation as the #1 choice for farmers seeking to enhance yields and protect their crops. The app was introduced on June 4, 2015 and was updated on April 23, 2024 in Google Play Store. This comprehensive mobile crop doctor covers 30 major crops and identifies over 400 plant damages with just a simple photo. Available in 18 languages, Plantix serves as a vital tool for farmers globally, empowering them with essential insights and community support for optimal crop production (PEAT, 2024).

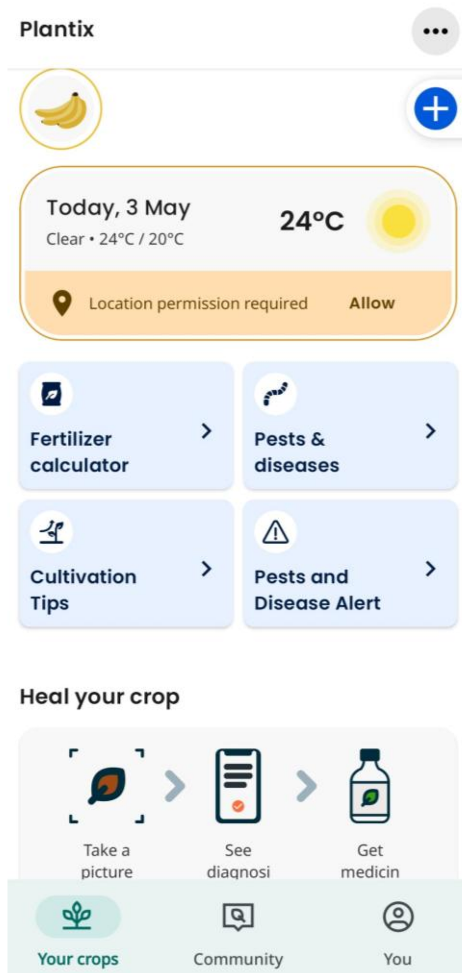

Figure 2.3 Sample Image of Plantix (PEAT, 2024)

Plantix is an agricultural app that provides instant diagnosis of pests, diseases, and nutrient deficiencies through photo analysis, offering immediate treatment suggestions. It alerts users to potential disease outbreaks in their area and provides real-time, agriculture-specific weather forecasts. The app also offers cultivation tips throughout the crop cycle and fosters a large farmer community, enabling knowledge-sharing and expert advice. Plantix is a mobile application similar to the proposed system Bani, featuring a photo scanning interface for plant identification. It can identify banana plant and fruit images and provides instant diagnosis of pests, diseases, and nutrient deficiencies, along with treatment suggestions. The app also offers alerts for potential disease outbreaks, real-time weather forecasts, and actionable cultivation tips. However, unlike Bani, Plantix does not detect Bugtok disease, does not use MobileNet CNN, operates only online, and lacks an integrated GIS heatmap.

2.3.4 OneSoil Yield

OneSoil is a free agriculture app to remotely observe your crop development, monitor the weather, and add notes. In the web version, you can also view files from machinery and calculate seed and fertilizer rates for agriculture needs (IMC, n.d.).

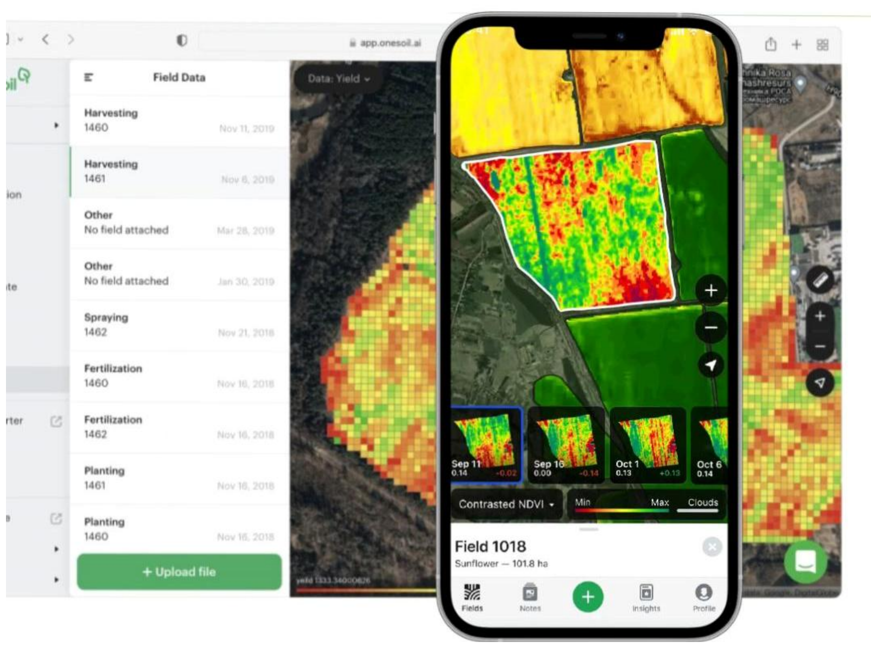

Figure 2.4 Sample Image of OneSoil Yield (IMC, n.d.)

The OneSoil app provides key tools for precision agriculture, including satellite monitoring with NDVI for tracking field vegetation, growing degree-days, and precipitation. It allows for field scouting through smartphones to detect issues, make notes, and categorize fields. Users can access weather forecasts to optimize timing for activities like spraying. Additionally, the app offers web-based tools for variable-rate technology and data analysis from on-board computers. The OneSoil app is similar to the proposed system Bani as it uses a GIS heatmap, is a mobile application, and is utilized in agricultural settings. However, it does not incorporate image processing, cannot identify banana plants or fruit, does not detect Bugtok disease, does not use MobileNet CNN, and operates exclusively online without offline functionality.

2.4 Synthesis

Cardava bananas, with their *Musa balbisiana* genetic makeup, are essential to the agricultural and economic landscape of regions like Davao Oriental in the Philippines. Their natural resistance to diseases such as Fusarium wilt, Black Sigatoka, and BBTV supports sustainable farming and robust livelihoods, while also offering promising prospects for breeding stronger varieties. However, the rise of Bugtok disease—particularly affecting Cardava and Saba bananas—necessitates early and expert detection. MobileNet-based disease detection has emerged as a successful tool across various contexts, from plant leaf disease identification in India and apple diseases in China to bean and grape disease classification. Utilizing MobileNet's efficiency and real-time capabilities is imperative to mitigate the impact of banana diseases, safeguard crop productivity, and contribute to global food security.

The table below compares the proposed system with existing image processing systems, highlighting how it addresses the gaps in current solutions.

Table 2.1 Tabulated Comparison

Features	Proposed System	Existing Image Processing Systems			Existing Mapping System
	Bani	Plant Identification and Disease	DoctorP	Plantix	OneSoil Scouting: Farming Tool
A mobile application system	YES	YES	YES	YES	YES
Can identify Bugtok disease in Banana cultivar	YES	NO	NO	NO	NO
Can identify banana plant and fruit	YES	NO	NO	YES	NO
Uses MobileNet CNN	YES	NO	NO	NO	NO
Has photo scanning interface for scanning	YES	YES	YES	YES	NO
Performs the system in both hybrid mode	YES	NO	NO	NO	NO
Has integrated GIS heatmap	YES	NO	NO	NO	YES

CHAPTER III

MATERIALS AND METHODS

3.1 Software Methodology

The development of the Bani application follows a structured Agile methodology to ensure a systematic approach to addressing the needs of banana farmers and effectively managing Bugtok disease. The methodology is divided into distinct phases, each involving iterative sprints of 1-2 weeks.

Figure 3.1 Agile Methodology (Okeke, 2021)

Requirement Definition – The team collaborates with banana farmers and agriculturists to identify and prioritize key features like image processing for Bugtok disease and GIS heatmap visualization.

Analysis and Design – Requirements are translated into a structured design using Dart, Flutter, and Python. Wireframes, mockups, and user stories guide the integration of MobileNet CNN for image classification and Google Maps API for GIS mapping.

Development – The application is built in iterative sprints, incorporating image processing, GIS heatmap functionality, and a user-friendly interface. Flutter, Firebase Realtime, TensorFlow, and MobileNet CNN are used for development and model training.

Implementation – Features are integrated, the UI is finalized, and Firebase is set up for data storage. GIS mapping and image processing functionalities are tested for full operational efficiency.

Testing – Unit, integration, and user acceptance testing are conducted using automated tools. MobileNet CNN and GIS heatmap features are evaluated through a confusion matrix, accuracy metrics, and ISO/IEC 25010 standards.

Deployment – The application undergoes final refinements based on testing and user feedback before release. The goal is to ensure seamless adoption by Cardava/Saba banana farmers for effective disease monitoring.

3.1.1 Requirement Analysis

The Department of Agriculture in Davao Oriental currently relies on a traditional system for managing banana diseases. Inspections are conducted through on-site visits, and control strategies typically involve distributing informational flyers, mitigation guidelines, and medication for infected banana plants. However, due to budget constraints, these efforts are often inconsistent, and cases of infection are sometimes overlooked or dismissed.

3.1.2 Requirement Documentation

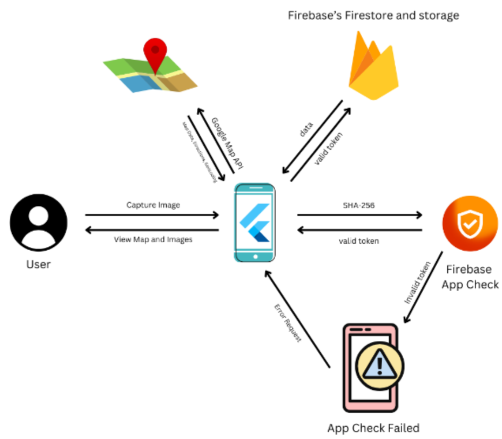

Figure 3.2 System Architecture of Bani App

The application retrieves heatmap point data directly from Firebase Firestore/Storage. This data is then used to generate heatmap overlays on a Google Map, which is rendered using the Google Maps API. The Google Maps API is used independently of the data retrieval from firebase.

3.1.3 Requirement Specification

3.1.3.1 Product Perspective

The proponents developed a mobile application to help Saba/Cardava banana farmers and other agriculturists in mitigating Bugtok disease strategically. The system operates as a general-use application with a single interface, allowing anyone to access its features as long as they have the app installed.

3.1.3.2 Product Features

The system can accurately classify healthy and non-healthy, and a non-banana peduncle with the help of careful collection of datasets performed by the proponents which is one of the important features of Bani app. Afterwards, the core feature, map visualization, takes initiative to display the infected area location. The system empowers farmers and agriculturists to strategically manage their Saba/Cardava banana farms by providing a digital tracking tool for efficient monitoring and decision-making.

3.1.3.3 User Class and Characteristics

The system only accommodates one (1) type of user. This type of user was assumed to have basic digital literacy and use offline mode and Android device frequently.

3.1.3.4 Operating Environments

During the project's development, the proponents utilized various software components while ensuring compliance with the necessary hardware requirements.

3.1.3.4.1 Software Components

The proponents utilized various software components as per requirement in developing the system. For mobile application development, they used Dart and Flutter to build and deploy an intuitive and visually appealing interface, ensuring native performance on Android OS. For algorithm programming, Python was utilized to implement core functionalities. In terms of image processing, TensorFlow was integrated to analyze and process captured images. For data storage, Firebase was employed to manage and store relevant system data efficiently. Additionally, Google Maps API was incorporated to enable location tracking and visualization, while GIS/Heat Map was used for spatial data representation. Lastly, Android Studio served as the primary development environment for system implementation and testing.

3.1.3.4.2 Hardware Requirements

In developing the system, the proponents ensured that the hardware requirements were met, specifically that the laptop or PC had a minimum of 2 GB of RAM and a stable internet connection.

3.1.3.5 Design and Implementation Constraints

A number of constraints in design and implementation was considered during the development of the system:

Hardware Constraints

- Requires a smartphone with a functional camera for image capture.
- Minimum device specifications (e.g., at least 2GB RAM) for smooth app performance.
- GPS-enabled devices are necessary for location tracking and mapping.

Software Constraints

- Requires an Android OS device.

- Internet connection is necessary for online mapping and data synchronization.
- Limited offline functionality since GIS and Firebase require real-time access.

Environmental Constraints

- Accuracy of disease detection may be affected by poor lighting or blurry images.
- Weather conditions (e.g., rain or extreme sunlight) may impact image quality.
- Farmers in remote areas with weak internet signals may experience delays in data upload.

Algorithmic Constraints

- TensorFlow's model performance depends on training data quality.
- The detection algorithm may struggle with ambiguous or unclear symptoms of Bugtok.
- False positives or false negatives could affect decision-making for farmers.

User Constraints

- Requires basic knowledge of mobile apps and digital tools for effective use.
- Farmers must correctly position the phone for accurate image capture.

- Some users may have difficulty interpreting GIS heatmaps without proper guidance.

3.1.3.6 User Documentation

The system includes an embedded user manual accessible through the “Help” page, where end-users can find answers to common questions. It also provides step-by-step instructions on how to use the system, with responses directly from the developers.

3.1.3.7 Other Non-functional Requirements

Listed below are various non-functional requirements addressing aspects such as safety, security, and software quality attributes.

3.1.3.7.1 Safety Requirements

To safeguard users, the proponents ensured that the app is free from malicious or harmful content, such as clickbait, malware, viruses, adware, and similar threats.

3.1.3.7.2 Security Requirements

Since the Bani app operates as a unified interface without user authentication, security measures focus on data integrity, safe access, and protection from external threats rather than strict user-based access control. Since the app does not include user login, these security measures focus on preventing data leaks, securing external connections, and ensuring system integrity rather than individual user access control.

- The system ensured that all uploaded images and location data remain unaltered during processing and storage.
- Firebase security rules restrict unauthorized modifications to the database.
- The app does not include external links or clickbait content that could expose users to malicious websites.
- The system had undergone security scans to detect vulnerabilities before deployment.
- The app request only essential permissions (e.g., camera, GPS) and avoid unnecessary access to sensitive device data.
- Users were informed of how their data is collected and used to ensure transparency.

3.1.3.7.3 Software Quality Attributes

3.1.3.7.3.1 Usability

The system's user interface was designed to be intuitive and user-friendly, allowing users to quickly become familiar with its operations within minutes. Additionally, the developers included a “Help” page to address common questions and provide guidance on using the system effectively.

3.1.3.7.3.2 Reliability

The system underwent multiple rounds of testing and debugging to minimize bugs and prevent system crashes.

3.1.3.7.3.3 Functionality

The proponents carried out an evaluation to ensure the system operates at its full capacity.

3.1.3.7.3.4 Portability

The proponents ensured that the system can run in hybrid approach or both online and offline mode.

3.1.3.7.3.5 Maintainability

The proponents ensured that the software is designed for easy updates and modifications whenever system changes are needed.

3.1.3.7.3.6 Efficiency

The proponents utilized available resources to develop the system in alignment with the project proposal.

3.1.4 Design

3.1.4.1 Use Case Diagram

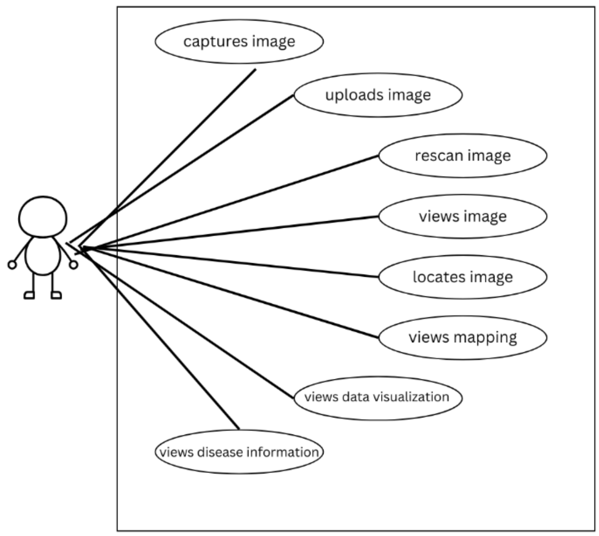

Figure 3.3 Bani Use Case Diagram

3.1.4.1.1 Use Case Description

The diagram summarized the system and its users, illustrating their interactions through various use cases. It showed that a single type of user, referred to as a general user, was able to perform multiple actions within the system. These actions included capturing and uploading images, rescanning images, viewing and locating images, accessing mapping features, exploring data visualization, and reviewing disease-related information.

3.1.4.2 Entity Relationship Diagram

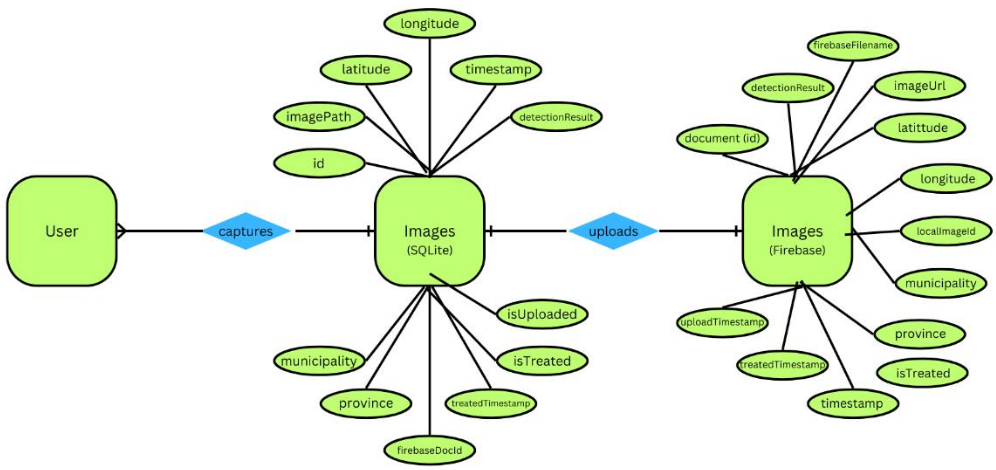

Figure 3.4 Bani Entity Relationship Diagram

3.1.4.3 Data Dictionary

TABLE NAME	ATTRIBUTE NAME	CONTENTS	TYPE	FORMAT	PK	FK
Images	id	Images' unique ID	TEXT	xxxxxx	✓	
	imagePath	Images' image path	TEXT	xxxxxx		
	latitude	Images' latitude	REAL	8 bytes (64-bit)		
	longitude	Images' longitude	REAL	8 bytes (64-bit)		
	timestamp	Images' timestamp when detection	TEXT	xxxxxx		
	detectionResult	Images' detection result	TEXT	xxxxxx		
	isUploaded	Images' upload status	int	0		

	isTreated	Images' treat status	int	0		
	treatedTimestamp	Images' timestamp when treated	TEXT	xxxxxx		
	firebaseDocId	Firebase's document id	TEXT	xxxxxx		✓
	province	Images' location province	TEXT	xxxxxx		
	municipality	Images' location municipality	TEXT	xxxxxx		
bugtok_images	firebaseDocId	Firebase's collection unique id	STRING	xxxxxx	✓	
	detectionResult	Images' detection result firebase	STRING	xxxxxx		
	firebaseFilename	Images' filename firebase	STRING	xxxxxx		
	imageURL	Images' URL	STRING	https://xxxxxxx		
	isTreated	Images' status	boolean	True/False		
	latitude	Images' location latitude	numbers	0000.00		
	longitude	Images' location longitude	numbers	0000.00		
	localImageId	Images' sqlite primary key id	STRING	xxxxxx		✓
	municipality	Images' located municipality	STRING	xxxxxx		
	province	Images' located province	STRING	xxxxxx		
	timestamp	Images' timestamp when detection	STRING	xxxxxx		

	treatedTimestamp	Images' timestamp when treated	STRING	xxxxxx		
	uploadTimestamp	Images' timestamp when uploaded	STRING	xxxxxx		

3.1.4.4 Graphical User Interface Designs

3.1.4.4.1 Design 1

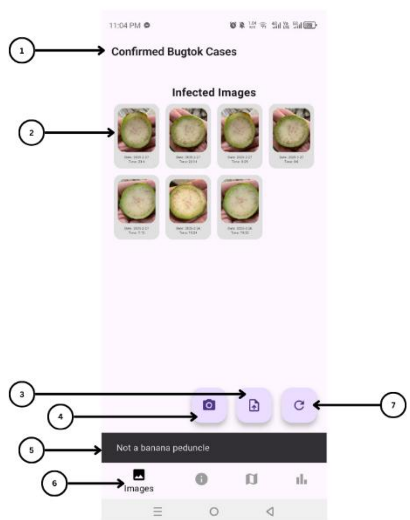

Figure 3.5 Confirmed Bugtok Cases

This interface displays the title, a collection of successfully scanned images, buttons for capture, upload and rescan image, and also shows the captured images along with the detected disease. Additionally, a snackbar for displaying the image processing result.

Table 3.2 Confirmed Bugtok Cases

No.	UI Components	Name	Description
1	Text	Title	Display the interface's title
2	Image	Images	Displaying the successfully scanned pictures
3	Button	Upload Button	Button for uploading photos
4	Button	Capture Button	Button for capturing images
5	Snackbar	Result Display	Display the result of the image processing
6	Bottom Navigation Bar	Images Button	Selects the Confirmed Bugtok Cases Interface
7	Button	Rescan	Button for rescan image

3.1.4.4.2 Design 2

Figure 3.6 Capture Image

This interface displays the camera interface and the capture button.

Table 3.3 Capture Image

No.	UI Components	Name	Description
1	Button	Capture	Capture the image

3.1.4.4.3 Design 3

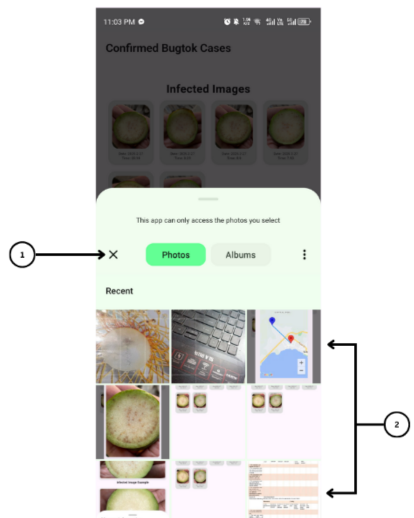

Figure 3.7 Upload Image

This interface displays the modal sheet with selection of images.

Table 3.4 Upload Image

No.	UI Components	Name	Description
1	Button	Close Button	Close the model sheet
2	Images	Images	Displaying the selection of image for image processing

3.1.4.4.4 Design 4

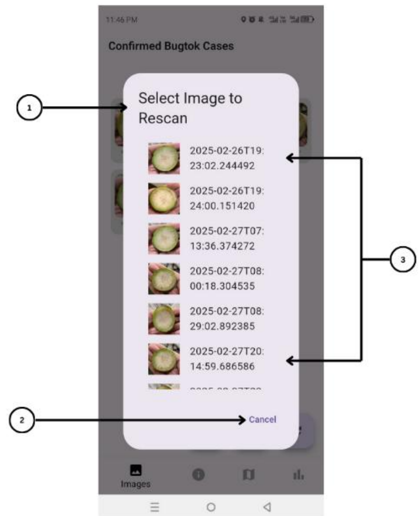

Figure 3.8 Rescan Image

This interface displays the title, and the selection of images for rescanning.

Table 3.5 Rescan Image

No.	UI Components	Name	Description
1	Text	Title	Display the interface's title
2	Image	Images	Displaying the selection of images for rescanning
3	Button	Cancel Button	Button for cancelling

3.1.4.4.5 Design 5

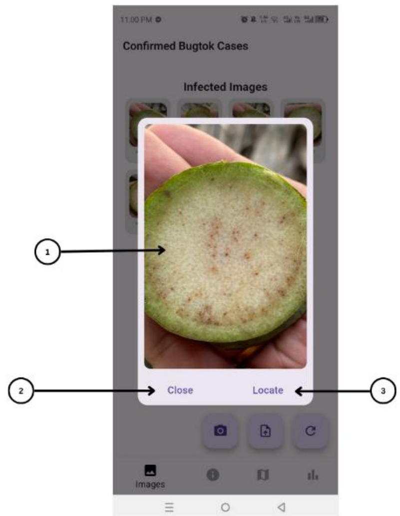

Figure 3.9 Image Preview

This interface displays the selected image and includes a "Locate" button for determining its location.

Table 3.6 Image Preview

No.	UI Components	Name	Description
1	Image	Selected Image	Display the selected image
2	Button	Close Button	Close the preview image
3	Button	Locate Button	Button for determining image location

3.1.4.4.6 Design 6

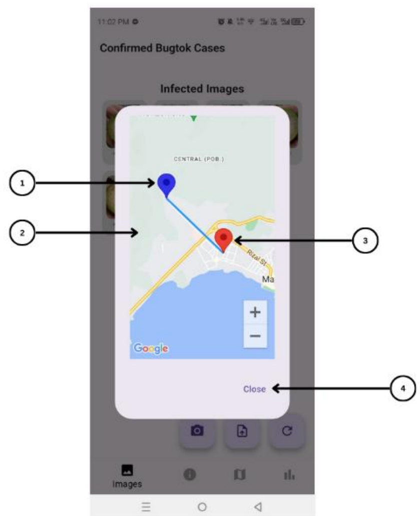

Figure 3.10 Locate

This interface displays map interface with markers for determining locations of the image and device.

Table 3.1 Confirmed Bugtok Cases

No.	UI Components	Name	Description
1	Marker	Device Marker	Marks the user's current location
2	Google Map	Map	Display the map for visualizing the location
3	Marker	Image Location Marker	Marks the image location
4	Button	Close Button	Button for closing the interface

3.1.4.4.7 Design 7

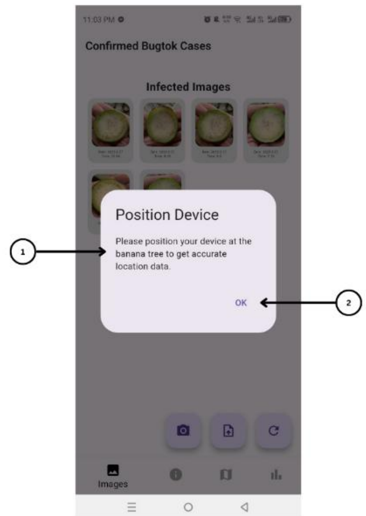

Figure 3.11 Image Location Prompt

This interface displays a prompt that will instruct the user to put the device near the banana tree or MAT (clump of a banana plant's rhizome).

Table 3.8 Image Location Prompt

No.	UI Components	Name	Description
1	Text	Prompt	Display the instruction for user
2	Button	Confirmation	Confirming if the device is near the plant

3.1.4.4.8 Design 8

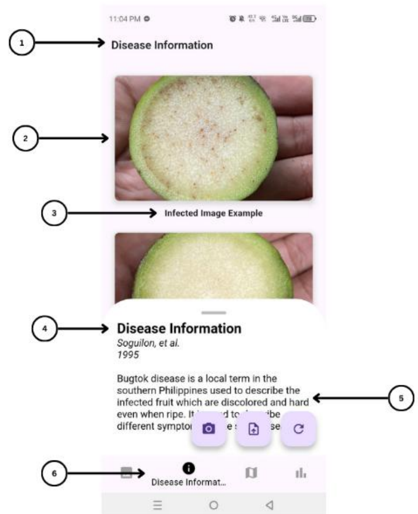

Figure 3.12 Disease Information

This interface displays the title, example images of healthy and infected with its title and show modal sheet for disease information.

Table 3.1 Confirmed Bugtok Cases

No.	UI Components	Name	Description
1	Text	Title	Display the interface's title
2	Image	Images	Displaying the successfully scanned pictures
3	Text	Image Title	Displays the image title (Infected or Healthy)
4	Modal Sheet	Disease Information Sheet	Holds the disease information
5	Text	Disease Information	Information about the disease
6	Bottom Navigation Bar	Images Button	Selects the Disease Information Interface

3.1.4.4.9 Design 9

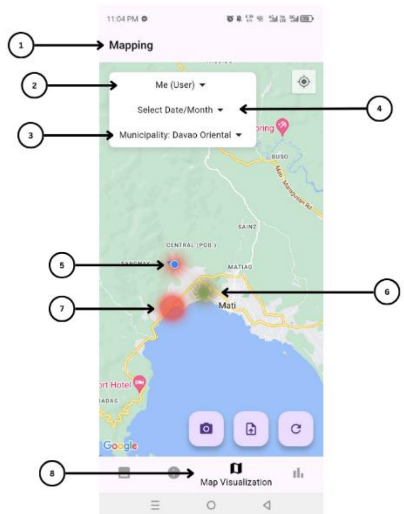

Figure 3.13 Map

This interface displays the map and visualizes heatmap points based on the results, with green indicating treated areas and red representing infected areas.

Table 3.10 Map

No.	UI Components	Name	Description
1	Text	Title	Display the interface's title
2	Drop Down Button	Users	Selection of database to fetch the users or all users results for heatmap visualization
3	Drop Down Button	Municipality	Selection of Municipalities of Davao Oriental
4	Drop Down Button	Date or Month	Selection of specific date or month
5	Location Indicator	Current Location	Display the current location of the device
6	Heatmap	Treated Point	Displays the treated heatmap point/s
7	Heatmap	Infected Point	Displays the infected heatmap point/s

8	Bottom Navigation Bar	Map Visualization Button	Selects the Mapping Interface

3.1.4.4.10 Design 10

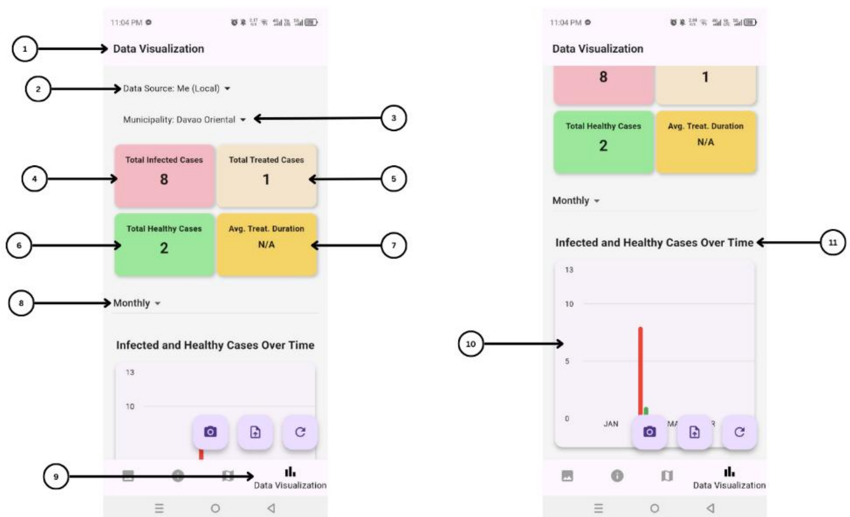

Figure 3.14 Data Visualization

This interface displays the total of infected, healthy and treated cases, and counts the average treat duration. It has a bar chart to view total infected and healthy cases over time.

Table 3.11 Data Visualization

No.	UI Components	Name	Description
1	Text	Title	Display the interface's title
2	Drop Down Button	Data Source	Selecting data sources
3	Drop Down Button	Municipality	Selection of municipalities in Davao Oriental
4	Container	Infected Card	Displays the total infected cases
5	Container	Treated Card	Displays the total treated cases
6	Container	Healthy Card	Displays the total healthy cases

7	Container	Average Treat Duration Card	Displays the average treat duration
8	Drop Down Button	Yearly or Monthly	Selection between Yearly and Monthly
9	Bottom Navigation Bar	Data Visualization Button	Selects the Data Visualization Interface
10	Bar Chart	Bar Chart	Displays the total infected and healthy cases over time
11	Text	Bar Chart Title	Displays the bar chart's title

3.1.5 Development and Testing

3.1.5.1 Development

The development and testing of the Bani App focused on integrating mapping as the core feature while utilizing image processing as a verification tool for Bugtok disease detection in Cardava/Saba banana farms. The system allows farmers to capture images of a banana peduncle's cross-section, which undergoes analysis using MobileNet CNN to verify whether the plant is diseased. If confirmed, the geolocation data is extracted and plotted on a GIS-based heat map, enabling real-time monitoring of infected areas. The app was rigorously tested for image classification accuracy, GPS tracking reliability, and system performance in real farming conditions. The mapping feature provides actionable insights for disease control, ensuring that only validated infections are added to the database. Through field testing and optimization, Bani was refined to deliver an intuitive and efficient solution, empowering banana farmers with advanced technology for sustainable crop management.

3.1.5.2 Data Analysis Plan

The design, development, and testing of the Bani app followed the ISO 25010 Software Quality Model, an international standard used to evaluate software quality by identifying key characteristics for assessment. The system underwent testing and evaluation by respondents, facilitated by a structured data analysis plan. Once all necessary data were collected, the proponents applied the Weighted Arithmetic Mean as the statistical tool for analysis. Respondents assessed each software characteristic using a Likert Scale, where 1 represented the lowest rating and 5 the highest.

The points that were used:

Strongly Agree	5
Agree	4
Neutral	3
Disagree	2
Strongly Disagree	1

In interpreting the weighted arithmetic mean, the following scale was used.

RANGE	INTERPRETATION
4.51 – 5.00	Strongly Agree
3.51 – 4.50	Agree
2.51 – 3.50	Neutral
1.51 – 2.50	Disagree
1.50 and below	Strongly Disagree

CHAPTER IV

RESULTS AND DISCUSSIONS

4.1 Design and Development of Bani App

4.1.1 Development of Bani Intuitive Interface with Integrated Camera

Figure 4.1 Interface of Bani App

The figure above displays one of the interfaces of the Bani app, allowing users to explore the functions of each button designed for navigation.

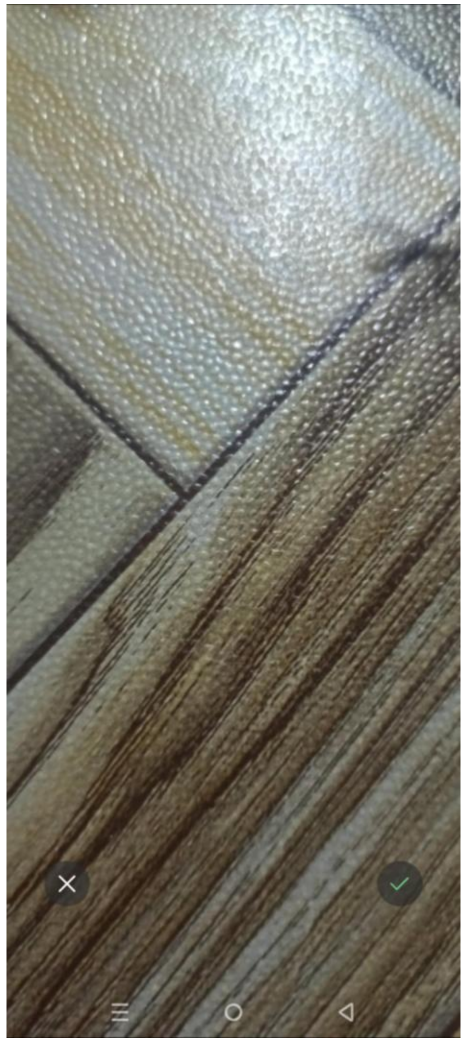

Figure 4.2 Bani Integrated Camera

The figure above enables the user to utilize the integrated camera within the application to capture and scan the banana peduncle. The scanning to verify the captured image is then processed.

4.1.2 Capture and Scan Interface

Figure 4.3 Scanning Process

After the capturing, it undergoes the scanning process where users are prompted to position the device at the banana tree to get the accurate location data. After that, it proceeds to add it under the label “Infected Images” if it is confirmed by the system as a diseased banana.

4.1.3 Preprocessing and Augmentation of the Dataset

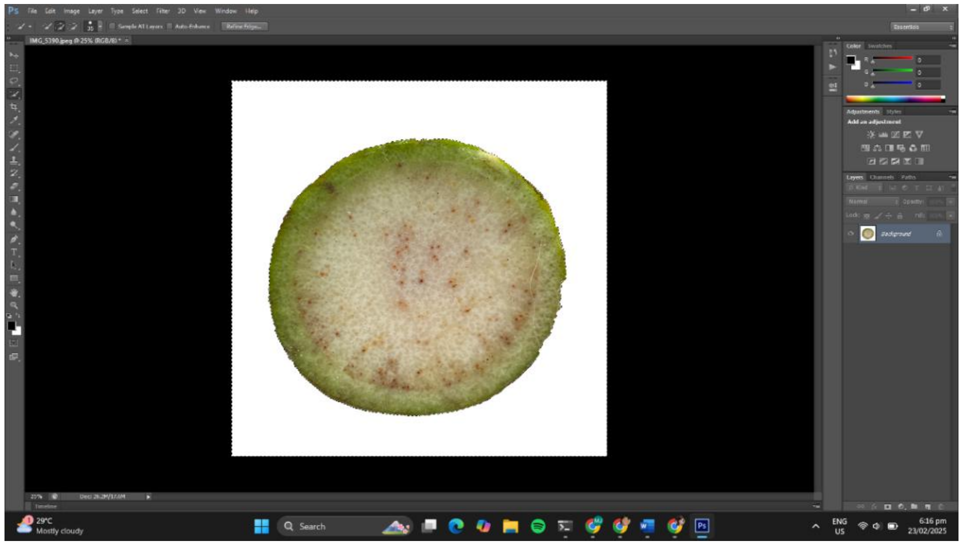

Figure 4.4 Preprocessing

In this figure, the researchers preprocessed the dataset which is an essential process for the image processing feature of the Bani App. In this process, the researchers took a total of 2400 images for both infected and healthy images of banana peduncle into photoshop to remove its background and replace it with white background.

4.1.4 Integration of MobileNet CNN

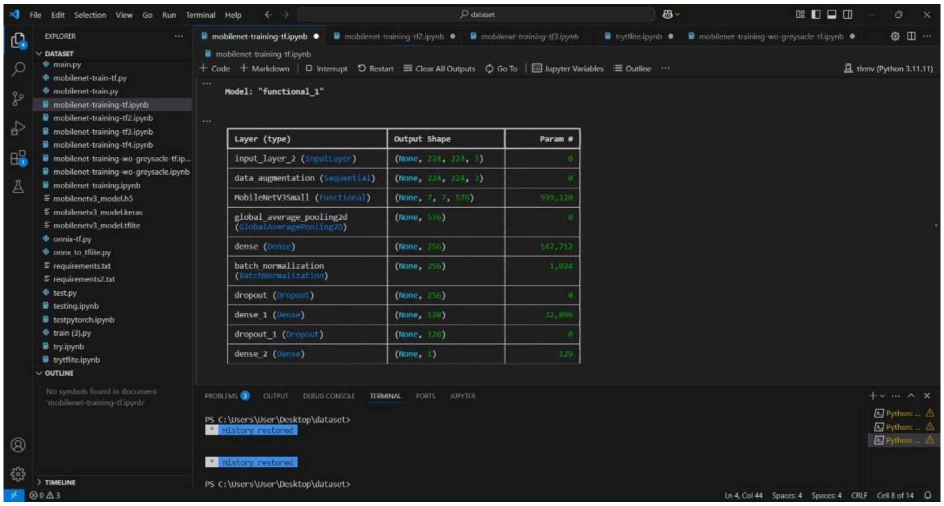

Figure 4.5 Integration of MobileNet CNN on Tensorflow

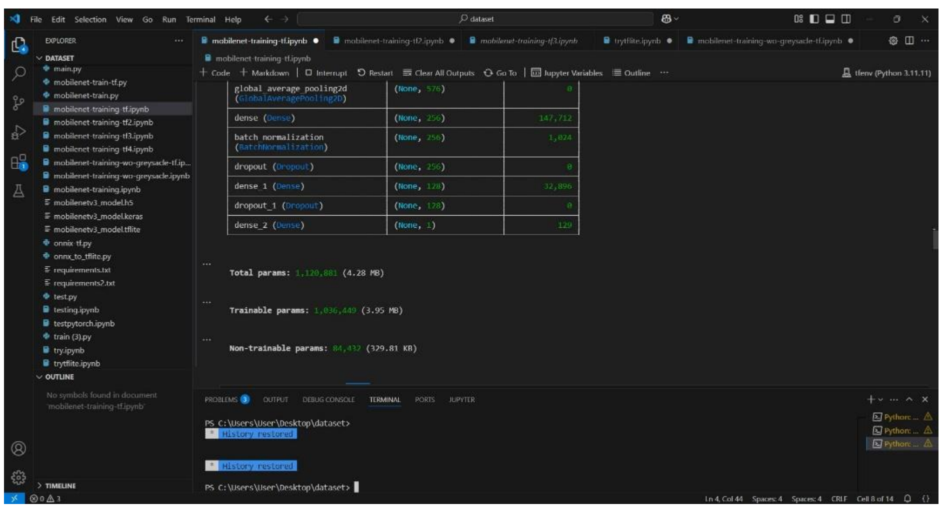

Figure 4.6 Machine Learning

The figures above display the integration of MobileNet CNN on Tensorflow—these are the tools essentially used for the development of the image processing feature for the Bani app. These are used to train the machine adding up the preprocessed dataset.

4.1.5 Integration of the GIS Heat Map

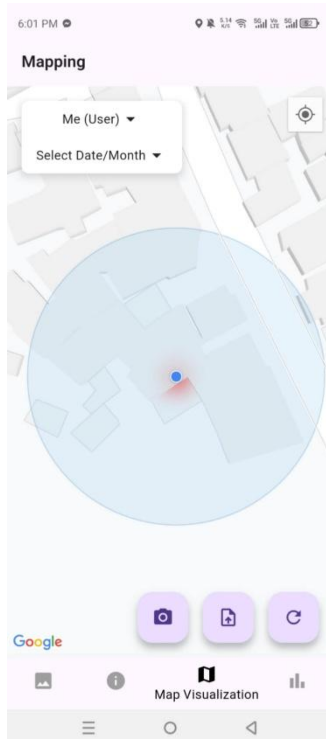

Figure 4.7 Heatmap with Radius

Figure 4.7 shows the map feature of the Bani app. After the scanning and analyzing process of the image captured, the location of the banana tree will then be recorded here as a red mark if it is confirmed a diseased banana.

4.1.6 Data Visualization Interface

Figure 4.8 Data Visualization

The figure data visualization shows the statistical tally of the recorded cases of Bugtok disease frequency, as you can see in the example data above.

4.1.7 Comprehensive Database

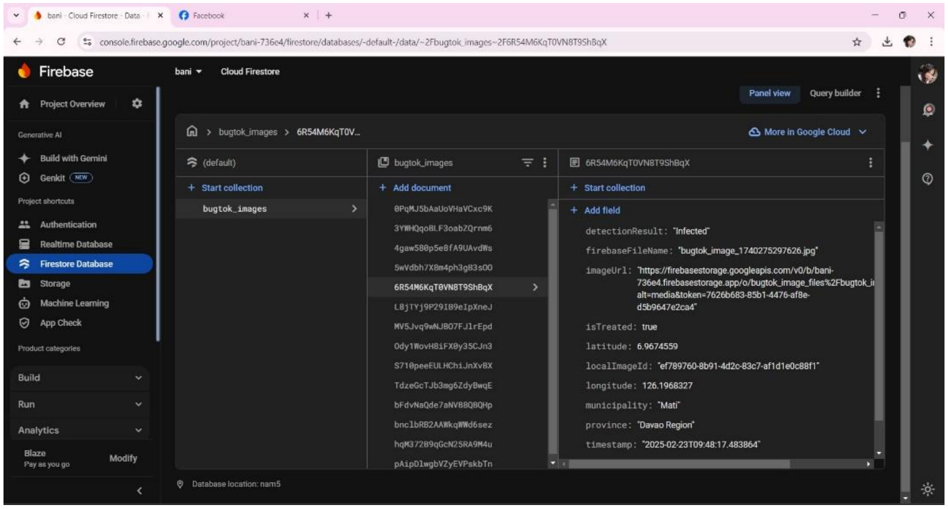

Figure 4.9 Firebase NoSQL Database

The figure above displays the database for online cloud server. This is essential for storing data of all users and getting data for data visualization.

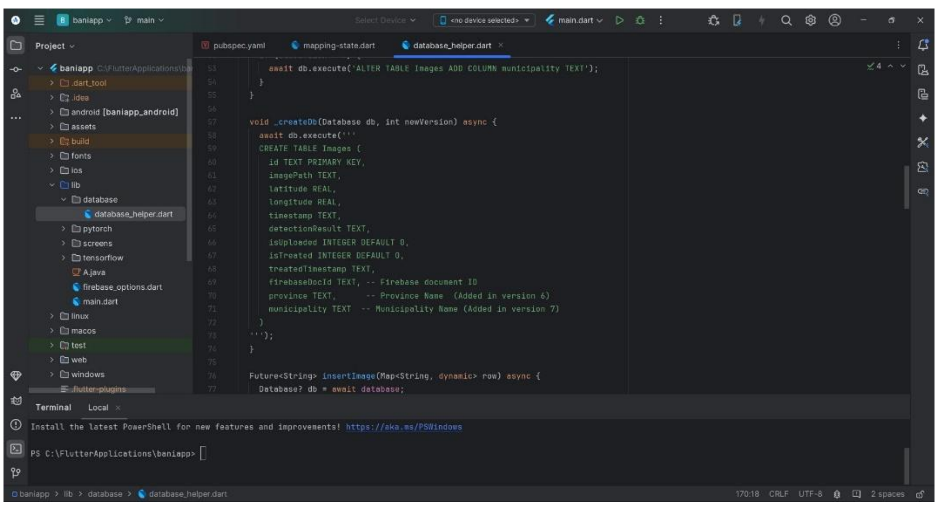

Figure 4.10 SQLite Database

The figure above displays the database for local server. This is essential for storing data even offline mode. Since the system is a hybrid application.

4.2 Metric Evaluation for MobileNet CNN

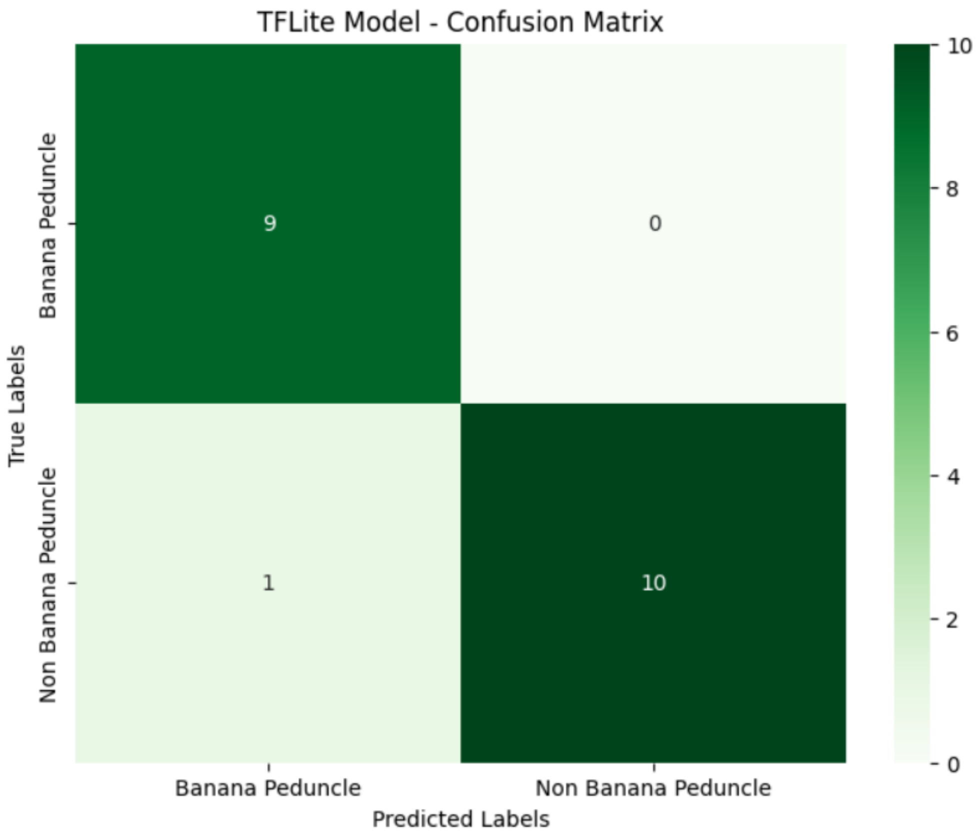

Figure 4.11 Confusion Matrix for Banana and Non-banana Peduncle Image Classification

```
...
    TFLite Model - Classification Report:
                         precision    recall  f1-score   support

        Banana Peduncle       0.90      1.00      0.95         9
    Non Banana Peduncle       1.00      0.91      0.95        11

               accuracy                           0.95        20
              macro avg       0.95      0.95      0.95        20
           weighted avg       0.96      0.95      0.95        20
```

Figure 4.12 Metric Evaluation for Banana and Non-banana Peduncle Image Classification

This report evaluates the performance of the model that is designed to classify the test samples as either “Banana peduncle” and “non-Banana peduncle”. The model perfectly predicts all the images from test samples with 9 images for banana peduncle and 11 for non-banana peduncle. The model also achieved perfect scores across all other metrics. Specifically, it shows 1.00 (See Figure 4.11 and Figure 4.12) precision and recall for both classes, meaning that all samples that are “healthy” were indeed healthy, and all samples that are “infected” were indeed infected. This resulted in a perfect precision of 1.00 for the non-banana peduncle class, and a 0.90 precision for banana peduncle class. On the other hand, the banana peduncle class gained a perfect 1.00 for the recall while the non-banana peduncle class got 0.91 and finally, both classes obtained 0.95 f1-score and overall accuracy. Thus, indicating that the model correctly classified all 20 image samples in the test set (9 for banana peduncle and 11 for non-banana peduncle). These results suggest excellent performance of the model—at least for this application system.

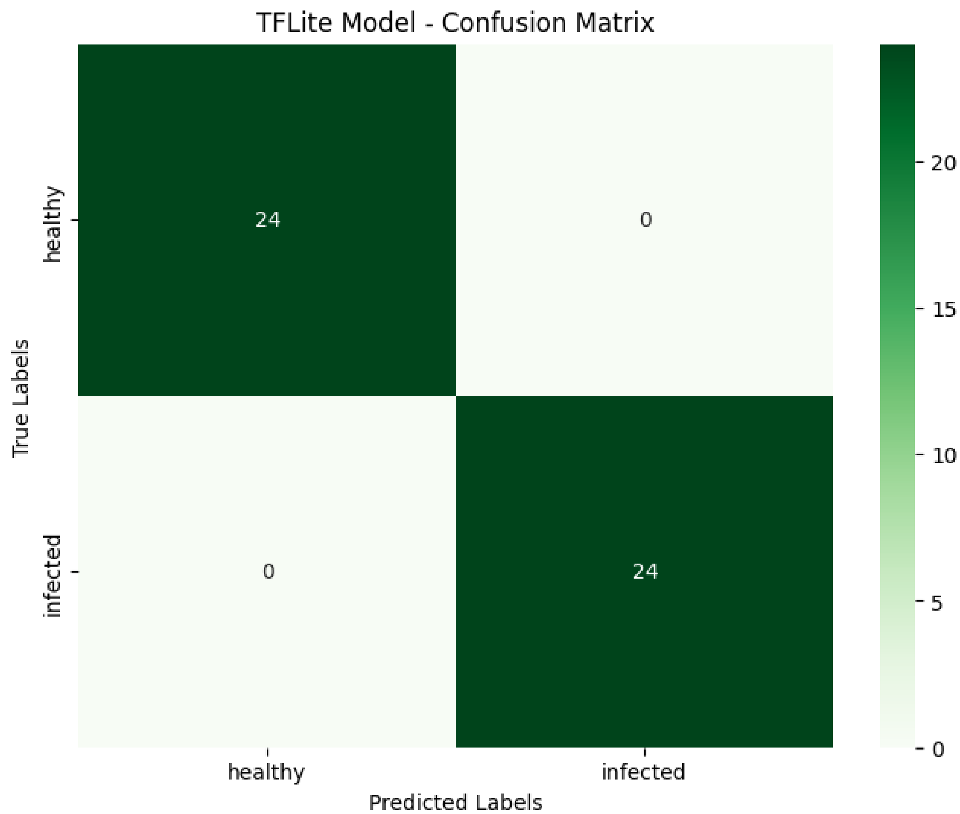

Figure 4.13 Confusion Matrix for Healthy and Infected Peduncle Image Classification

```
...
    TFLite Model - Classification Report:
                  precision    recall  f1-score   support

         healthy       1.00      1.00      1.00        24
        infected       1.00      1.00      1.00        24

        accuracy                           1.00        48
       macro avg       1.00      1.00      1.00        48
    weighted avg       1.00      1.00      1.00        48
```

Figure 4.14 Metric Evaluation for Healthy and Infected Peduncle Image Classification

This report evaluates the performance of the model that is designed to classify the test samples as either “Healthy” and “Infected”. The model perfectly predicts all the images from test samples with 24 images each for healthy and infected. The model also achieved perfect scores across all other metrics. Specifically, it shows 1.00 (See Figure 4.13 and Figure 4.14) precision and recall for both classes, meaning that all samples that are “healthy” were indeed healthy, and all samples that are “infected” were indeed infected. This resulted in a perfect F1-score of 1.00 for both classes and an overall accuracy of 1.00, indicating that the model correctly classified all 48 image samples in the test set (24 healthy and 24 infected). While these results suggest excellent performance of the model, it is important to consider the size and diversity of the test dataset to ensure the model generalizes well to real-world data.

Method for Dividing the Images through Training, Testing, and Validation

However, due to the limited dataset size of 1,204 samples per class, the researchers adjusted the split to allocate 80% for training and 20% for validation. This modification ensured that the model can learn effectively from a larger training set while still maintaining meaningful evaluation with enough validation and separate test sets. Such flexibility in the data splitting process allowed the researchers to efficiently use the limited data samples, balancing training performance with evaluation reliability which has been deemed practical by the researchers under the given constraints.

4.3 Evaluation of the System

A total of 30 respondents consisting 29 farmers and a banana expert from Department of Agriculture were chosen as participant for the user testing phase of the

Bani app. After testing, they evaluated the app using the approved survey through a Five-Point Likert Scale. Table below displays the summary of the conducted survey.

Table 4.1 Bani Evaluation Result

Indicators	Respondents	Weighted Mean	Interpretation
Functionality	30	3.95	Agree
Efficiency	30	3.63	Agree
Compatibility	30	3.71	Agree
Usability	30	3.90	Agree
Reliability	30	3.71	Agree
Security	30	3.52	Agree
Maintainability	30	4.33	Agree
Portability	30	3.98	Agree
Overall	30	**3.75**	**Agree**

These indicators listed in the first column determine whether Bani App complied with 25010 standards. Starting off with the first indicator; functionality**,** it has obtained an average of 3.95 which is then rounded off to the nearest whole number (4) and falls under the interpretation of "Agree". This is interpreted as positive feedback of the respondents and deemed the Bani app as **functional**. Next, the indicator Efficiency has obtained an average of 3.63, which, when rounded off to the nearest whole number (4), also falls under the interpretation of "Agree." This suggests that respondents find the Bani App **efficient** in terms of its performance, speed, and resource utilization. The Compatibility indicator received a weighted mean of 3.71, which rounds to 4 and is interpreted as "Agree." This implies that the app is **compatible** on most of the farmers since they use Android phones, compatibility in GPS location across different devices and with varying internet speeds. For Usability, the app scored an average of 3.90, which rounds to 4, again interpreted as "Agree." This means that respondents find the application **user-friendly**, intuitive, and easy to navigate. The Reliability indicator also received a 3.71 rating, which rounds to 4 and is interpreted as "Agree." This indicates that the app performs **consistently** and can be depended on for stable operation without frequent errors or failures. However, in terms of Security, the app obtained an average of 3.52, which rounds to 4 and is interpreted as “Agree.” This suggests that respondents agree regarding the app’s security measures. On the other hand, Maintainability received the highest rating, with a weighted mean of 4.33, rounding to 4 and interpreted as “Agree.” This indicates that the app is well-structured, **easy to update**, and can be efficiently modified for future improvements or fixes. Lastly, Portability received a 3.98 rating, rounding to 4 and interpreted as “Agree.” This suggests that the app can be easily transferred or adapted to different environments, making it flexible for various user needs. Overall, the Bani App achieved a total weighted mean of 3.75, which rounds to 4 and is interpreted as “Agree.” This indicates that the application generally meets the ISO/IEC 25010 standards based on respondent feedback. However, the security aspect needs further evaluation and enhancement to ensure that user data and system integrity are well-protected.

4.4 Implementation Plan

Table 4.2 Implementation Plan

Strategy	Activities	Persons Involved	Duration
Deployment Preparation	Verify system preparedness, review deployment steps	Proponents and DA staff	15 minutes
Access and Testing	Test for minor or major revisions for the system, test user access, functionality and features	Proponents and DA staff	5 minutes
System Preparation	Check system bugs, deploy on a server	Proponents and DA staff	15 minutes
Conduct Training Sessions	Conduct initial orientation and demonstration activities to persons involved	Proponents and DA staff	10-15 minutes

The deployment process was planned to begin with the team verifying system preparedness and reviewing deployment steps to ensure a smooth rollout. Next, the team would conduct access and functionality testing to identify any necessary revisions. Once the system was set up, they would proceed with checking for system bugs and deploying the application on a server. Finally, they would conduct initial orientation and demonstration activities to familiarize the persons involved with the system's features and usage.

CHAPTER V

SUMMARY, CONCLUSION, AND RECOMMENDATIONS

5.1 Summary

Recapping the results and discussion from the previous chapter, the researchers have successfully met the objectives for the Bani app system development. First was the development stage of the Bani interfaces; this included the integrated camera, mapping visualization, data visualization with graphical representation, and gallery for the confirmed case of Bugtok disease all paired with functional buttons.

Second was the functionality of the image processing feature of the application. It was successfully integrated since it was really essential for the Bani app to scan and confirm the Bugtok disease accurately before it marks the location in the map visualization. Big chunk of effort was poured in developing this feature.

The third objective was the highlight of the Bani app—the map visualization. If a case of Bugtok disease was recorded, a prompt is popped on screen, asking the user to turn on their GPS share location and stick the device to the banana tree. Ideally when using the app, it is recommended while doing a site inspection in the entire Cardava/Saba banana plantation/farm. Without the integration of this map feature, the system would be a typical image processing product with a half-baked purpose.

The fourth objective was the data visualization. This is where the numerical data were collected and analyzed for interpretation. In analytical approach, this is essential because the regular usage of Bani app would feed more data into the graphical representation or the bar graph. It displays the data from Bugtok cases recorded over time. This is great for descriptive analysis.

The fifth objective was the database for storing data. This is also an essential backbone of the application. It ensures efficient data management, allowing users to store and analyze information seamlessly. A well-structured database enhances system reliability, enabling smooth functionality even as data volume increases.

Sixth objective, Bani App was evaluated using ISO/IEC 25010 standards based on various indicators. It received an overall weighted mean of 3.75, rounded to 4, which was interpreted as "Agree", indicating general compliance with the standards. The app scored well in Functionality (3.95), Efficiency (3.63), Compatibility (3.71), Usability (3.90), Reliability (3.71), Maintainability (4.33), and Portability (3.98), all rounding to 4, suggesting positive feedback in these areas. However, Security scored the lowest out of 8 (3.52 rounded to 4, interpreted as "Agree"), indicating a need for improvement in security measures. Overall, while the app is functional, efficient, and user-friendly, enhancements in security are recommended to strengthen user data protection and system integrity.

Lastly, the implementation plan was an on-going process. The team continuously assessed the system performance to identify areas for improvement. Regular evaluations and user feedback were incorporated to ensure a smooth and effective deployment.

5.2 Conclusion

The Bani App successfully met its objectives, proving to be an effective tool for Bugtok disease detection and mapping in Cardava and Saba banana farms. Its image processing, mapping, data visualization, and database worked well together, helping farmers detect and track infected plants accurately. The app performed well in functionality, efficiency, usability, and reliability, based on ISO/IEC 25010 standards. However, its security needs improvement to better protect user data. Overall, the Bani App serves its purpose of helping farmers detect and monitor Bugtok disease early. While it is already useful, further security upgrades will make it even better and more reliable for long-term use.

5.3 Recommendations

Bani App was able to meet the objectives and performed well enough to pass the ISO/IEC 25010 standards. However, there are some areas that can be improved in future development.

- One key enhancement is the **addition of a predictive analytics feature**. This would allow the system to analyze past data and predict potential Bugtok disease outbreaks, helping farmers take preventive measures before infections spread. By integrating machine learning, the app could provide early warnings and improve disease management strategies.
- Another improvement is the **implementation of role-based user access**. Currently, the system only supports a single type of user with no separate roles, such as admin and farmer accounts. Adding role-based access would allow better data management, monitoring, and system control, ensuring a more organized and structured system.
- With multiple user roles, **security improvements become necessary**. If the system expands to support different users, additional security features such as

data encryption and access control should be implemented to protect sensitive information and prevent unauthorized access.

- Lastly, **user login must be added if different user roles are introduced**. This would ensure that only authorized users can access specific features, strengthening overall data security and system integrity.