

Plant Nursery Management and Disease Detection Using Artificial Intelligence

¹Mrunal Rajesh Omana

Department of Computer Science and Engineering
Ashokrao Mane Group of Institutions
Vathar Tarf Vadagaon, Kolhapur, India
mrunalomanna@gmail.com

²Dr. Shrenik Suresh Sarade

Department of Computer Science and Engineering
Ashokrao Mane Group of Institutions
Vathar Tarf Vadagaon, Kolhapur, India
shreniks2k7@gmail.com

Abstract— Although buying a plant is easy to do online today, knowing how to keep that plant alive afterward is not, and diagnosing what has gone wrong with it later is harder still. This study presents an Android application that combines both problems into one tool. On the retail side, it allows a buyer to shop for plants, seeds, fertilizers, and pots without a nursery visit, with care notes attached to each listing so that the guesswork of watering, light, and feeding schedules is removed. On the health side, it adds a camera-driven diagnosis feature: when a leaf starts showing symptoms, a photo of it is run through a Support Vector Machine (SVM) classifier trained to recognize common plant diseases, and the app responds with a likely diagnosis plus a practical remedy. The system was built iteratively under an agile process, with Android Studio driving the client, PHP and MySQL handling storage and business logic, and a Python/Flask service hosting the classifier. Unit, integration, and acceptance testing across these pieces confirmed that the combined shopping plus diagnosis workflow holds together end-to-end.

Index Terms— Mobile Application, Plant Nursery, Disease Detection, Machine Learning, Support Vector Machine, Convenience, User-Friendly, E-Commerce.

I. INTRODUCTION

While online retail now covers almost every product category, purchasing plants, seeds, fertilizers, and pots still typically means a trip to a physical nursery. A face-to-face visit is no guarantee of getting the full picture either: how often a species needs watering, how much sun it tolerates, or which diseases it is prone to are details that staff at a busy nursery counter often do not have time to explain. This mismatch between the ease of shopping online and the specialized plant-care knowledge that rarely travels with it motivates the work described here.

The system built for this study, referred to here as the Plant Nursery Management and Disease Detection Application, allows customers to browse, order, and pay for plants, seeds, fertilizers, and pots through a single Android interface. The second half addresses a concern that only surfaces after the purchase: what happens when the plant becomes sick. If a leaf starts showing symptoms, the owner photographs it and submits the image using the app. A trained classifier examines the image, names the most probable disease, and returns a remedy that the owner can immediately act on. The result is a single tool that covers both the transaction and the plant's ongoing well-being.

II. LITERATURE SURVEY

Several prior studies have examined either e-commerce platforms for nurseries or automated plant-disease diagnosis, but rarely both together.

Jeger and van den Bosh discuss how planting material for many perennial crops is first propagated and multiplied within a nursery, and model how epidemics can spread through such plantation-nursery systems, underlining the importance of early disease recognition during the nursery stage [1].

Makesar, Nikam, and Dudhkawde described the design and implementation of a web-based application for plant nurseries, noting that customers often lack sufficient information about

the plants they intend to buy, and proposed an e-nursery platform that consolidates plant information in one place [2].

Krishanan, Kalia, and Tiwari examined nursery management practices more broadly, observing that well-run nurseries create employment for skilled and unskilled labor alike, and that recommended nursery practices are essential to producing healthy seedlings [3].

Bhosale et al. present a multi-plant and multi-crop leaf-disease detection and classification approach that combines deep neural networks, classical machine learning, and image processing for precision agriculture, demonstrating the feasibility of automated visual disease diagnosis across several crop types [4].

Sagar, Javed, and Doermann propose a leaf-based plant-disease detection approach supported by explainable AI, so that predictions are accompanied by an interpretable justification rather than being treated as a black box [5].

Siddiqui and Kabir evaluated existing plant disease detection mobile applications, reporting their overall quality and notable limitations, such as restricted disease coverage and inconsistent accuracy under real-world conditions [6].

Collectively, this body of work indicates that while nursery e-commerce platforms and plant disease detection systems have each been studied independently, there remains an opportunity to integrate the two into a single, unified mobile application, which is the gap this study addresses.

III. PROBLEM STATEMENT AND OBJECTIVES

A. Problem Statement

Two gaps stand out in the way people currently buy and care for plants. First, sourcing plants, seeds, fertilizers, and pots still leans heavily on visiting a nursery in person, and even that visit rarely comes with reliable care instructions. Second, once a plant is in the buyer's home, there is no straightforward way to determine what is wrong with it if it later falls ill, or what to do about it, a gap that matters because a disease left unchecked can quickly damage the health and yield of the plant.

B. Objectives

- Shop on your own schedule: A buyer can browse and order plants, seeds, fertilizers, and pots whenever and wherever suits them, rather than during nursery opening hours.
- Put care instructions in reach: attach watering, feeding, and light requirements, along with general growing tips, directly to each listing so that buyers are not left to guess.
- Catch problems early: let a photo of a leaf be checked automatically against a trained model so a disease can be named and treated before it spreads.
- Get the order right: Back every purchase with dependable ordering, packaging, and dispatch flow so that plants reach the buyer in good condition.

IV. PROPOSED SYSTEM AND METHODOLOGY

Development followed an agile process rather than a single fixed build cycle: work was split into short iterations, each ending in a version of the app that actually ran, and each round of testing fed directly into what was built next. Working this way kept the requirements open to change and meant that a demonstrable product existed at every stage, not only at hand-off.

A. System Architecture

A new user starts by creating an account, and every subsequent session begins with a login check against the stored profile in the central database. From there, the app opens onto the product menu, plants, seeds, fertilizers, and pots, where items can be added to a cart and paid for either by cash on delivery or through an online gateway. The second path through the system runs in parallel: at any point after a purchase, the buyer can hand the app a photo of a troubled leaf, at which point it is routed to the disease-detection module and comes back paired with a diagnosis and a suggested course of action.

B. Modules

Module 1 handles the buying side of the app, which includes creating an account, signing in, browsing what the nursery has listed under plants, seeds, fertilizers, and pots, building up a cart, and pushing an order through to checkout, all sitting on top of a relational store that the Android client talks to.

Module 2 is where the health check occurs. A leaf photo submitted by the user is handed to a Support Vector Machine model on the server side, which scores it against known disease patterns and sends back whichever label fits best, together with a short note on what to do about it.

V. EXPERIMENTAL SETUP AND IMPLEMENTATION

A. Technologies and Tools

Java inside Android Studio drove the mobile client. On the server, business logic was implemented in PHP, records were stored in MySQL, and XAMPP hosted the local development stack. The classifier was written in Python and wrapped in a small Flask service, so the mobile app could send it an image over HTTP and receive a prediction back without embedding the model directly on the device.

B. Hardware and Software Requirements

Development was carried out on machines with at least 4 GB of RAM, approximately 500 GB of free storage, and a processor clocked at 2 GHz or above, running either a 32-bit or 64-bit

build. On the software side, the toolchain pulled together Android Studio, Notepad++, XAMPP, Flask, and MySQL, plus a standard browser for testing the web-facing components.

C. Application Interface

Fig. 1 is the screen a user lands on first, offering a choice between signing up as a new customer or signing in with an existing account; Fig. 2 shows the sign-in form itself. Past that screen, the catalogue, cart, and order flow are described above.

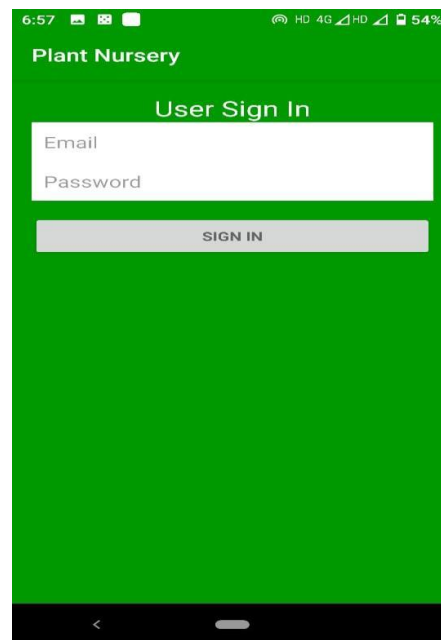


Fig. 1. Application landing screen with Sign Up and Sign In options.



Fig. 2. User sign-in screen.

The disease-check flow lives under "Leaf Disease Predict": the user picks a photo of the affected leaf and taps through to have it scanned, as shown in Fig. 3. What comes back, as shown in Fig. 4, is a named disease, a short explanation of what typically causes it, and a remedy the user can try.

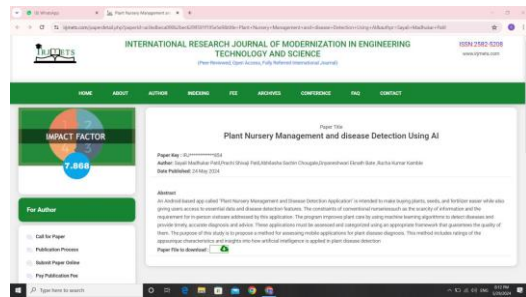


Fig. 3. Leaf image upload screen for disease prediction.

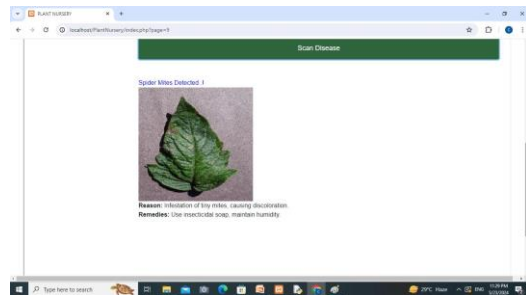


Fig. 4. The disease was predicted with reasoning, and a suggested remedy was provided.

VI. TESTING AND RESULTS

Three layers of testing were used to check the build quality. At the unit level, each piece, sign-up, sign-in, catalogue browsing, and the disease classifier, was exercised on its own to confirm that it behaved as expected before anything was wired together. Integration testing then checked the seams; for instance, placing an order actually left a matching entry in the order history view. Finally, a round of acceptance testing walked through realistic buyer scenarios in front of end-users to confirm that the app matched what had been requested. Specific checks that passed included the login form correctly rejecting a bad email/password pair, a warning appearing when a required field was left blank, and the system catching an attempt to re-enter data that already existed in the database.

VII. CONCLUSION

This work delivers a single Android app that lets someone shop for plants, seeds, fertilizers, and pots without leaving the couch, and then continues to help after the sale by reading photos of sick leaves and suggesting a fix. Building it in Agile sprints, with an SVM classifier performing the diagnostic work, kept the project adaptable while still landing a system that passed its unit, integration, and acceptance tests. Taken together, these results suggest that the app is ready to move from a college project to something closer to a real deployment.

VIII. FUTURE SCOPE

There is room for improvement in a few directions: widening the set of diseases the classifier recognizes and feeding it a larger training set, swapping the SVM for a deep-learning model if that improves accuracy, and giving buyers a forum to swap tips with each other. On the hardware side, IoT sensors reporting soil moisture and nutrient levels in real time, paired with demand-forecasting analytics on the retail side, would push the system from a shopping-plus-diagnosis tool toward something closer to a full nursery management platform.

REFERENCES

- [1] M. J. Jeger and F. van den Bosh, "Modeling plant epidemics in plantation-nursery systems," 2021.
- [2] M. Makesar, Y. Nikam, and P. Dudhkawde, "Design & implementation of web based application for plant nursery," Aug. 2022.
- [3] R. Krishanan, R. K. Kalia, and J. C. Tiwari, "Plant nursery management," unpubl.
- [4] Y. H. Bhosale, S. R. Zanwar, S. S. Ali, N. S. Vaidya, R. A. Auti, and D. H. Patil, "Multi-plant and multi-crop leaf disease detection and classification using deep neural networks, machine learning, image processing with precision agriculture," in Proc. Int. Conf. Computer Communication and Informatics (ICCCI) 2023.
- [5] S. Sagar, M. Javed, and D. S. Doermann, "Leaf based plant disease detection and explainable AI," in Proc. Int. Conf. Computer Communication and Informatics (ICCCI) 2022.
- [6] I. Siddiqui and M. A. Kabir, "Evaluating plant disease detection mobile applications: quality and limitations," Aug. 2022.
- [7] S. Ahmad and I. Ashraf, "Fruit and vegetable nursery: establishment and management," Apr. 2020.
- [8] H. Mittal, P. Nautiyal, K. Chauhan and R. Bhaskar, "Nursery management of horticultural crop and its act," Dec. 2021.