MAY 2024

GreenEye

Instantly identify plants with our camera-based recognition app.

OUR TEAMS

Muhammad Rihap Muhammad Syamil

Muhammad Raihan Pancadrya Y.P.

About Group Members

Muhammad Raihan Firdaus

Writer/AI Eengineer

Interest in writing a document and application content.

"*dum vita est, spes est* -- While there is a life, there is a hope"

Muhammad Rihap Firdaus

UI UX Designer/Front-End Dev/AI Engineer

Designing and creating user interfaces that are intuitive, user friendly, attractive and provide a good user experience.

"When you fail to plan, you plan to fail"

About Group Members

Muhammad Syamil Hamami

Backend Developer/AI Engineer

Implementing business logic and building backend app

“Endeavor ceaselessly, yet in the pursuit of greatness, remember the power of rest - Moon Tzu 🗿”

Pancadrya Yashoda Pasha

Designer/AI Researcher/AI Engineer

Interested and often explores things related to technology, AI, Cloud, etc.

“One of the most important things in life is to believe”

Topic and Motivation

TOPIC

Recognition of nearby plants using image recognition technology

MOTIVATION

The ability of modern humans to recognize and understand their surroundings is diminishing. This has resulted in a lack of knowledge to independently recognize plants and other related information. Therefore, there is a desire to build a plant knowledge base that is easy to access and use.

AL-QUR'AN REFERENCE

One of the Qur'anic verses that discusses plants is Surah Al-An'am verse 99. Here's how it reads:
"And He is the One Who sends down rain from the sky—causing all kinds of plants to grow—producing green stalks from which We bring forth clustered grain. And from palm trees come clusters of dates hanging within reach. ˹There are˺ also gardens of grapevines, olives, and pomegranates, **similar** ˹**in shape**˺ but **dissimilar** ˹**in taste**˺. Look at their fruit as it yields and ripens! Indeed, in these are signs for people who believe." (Al-An'am: 99)

Target User

All types of human beings experience the following in their lives:

50%

Easy to forget #1

Can't remember many things, especially if they are of different types

30%

Living in a concrete and steel environment #2

Don't have much opportunity to play in nature

10%

Just want to keep things simple #3

Utilize whatever technology is available, the easier the better

---

The target user group for this application comprises individuals who often find themselves forgetting amidst the hustle and bustle of urban life, where concrete and steel dominate the landscape. These users seek simplicity in managing their daily routines and interactions with nature.

They may be city dwellers longing for a connection to the natural world but are deterred by the complexities of identifying plants nearby. For them, the application serves as a convenient tool to bridge the gap between their urban existence and the tranquility of green spaces.

Proposed Solution

GreenEye, a plant recognition application based on Android mobile. Utilizing the "eyes" of every modern smartphone,

GreenEye recognizes captured photos using existing machine learning technology. As a result, users can enjoy the convenience of recognizing the plants around them.

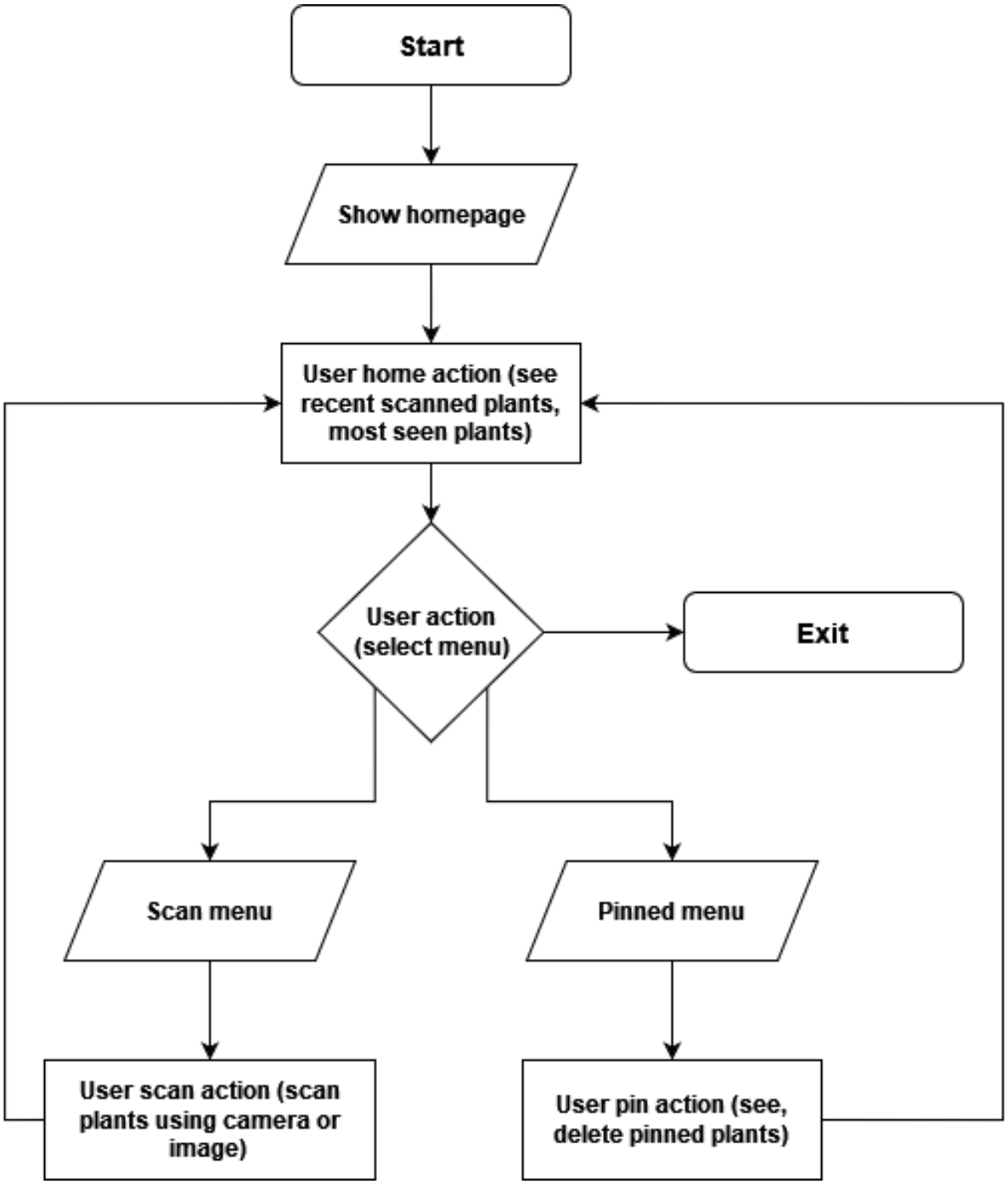
In relation to "Good health and well-being", GreenEye indirectly encourages users to explore open spaces such as parks, gardens, and nature reserves, providing opportunities for activities such as walking, hiking, and bird watching. On the other hand, GreenEye also provides information on plants that have average health benefits.

GreenEye contributes to quality education by offering users access to valuable information about the flora around them. Through features such as plant identification, users can learn about the different plant species they see. This fosters greater understanding and appreciation of biodiversity and environmental conservation.

GreenEye

GreenEye

Application Flowchart



Application Design

home page
contains a search bar that makes it easier for users to search by words.

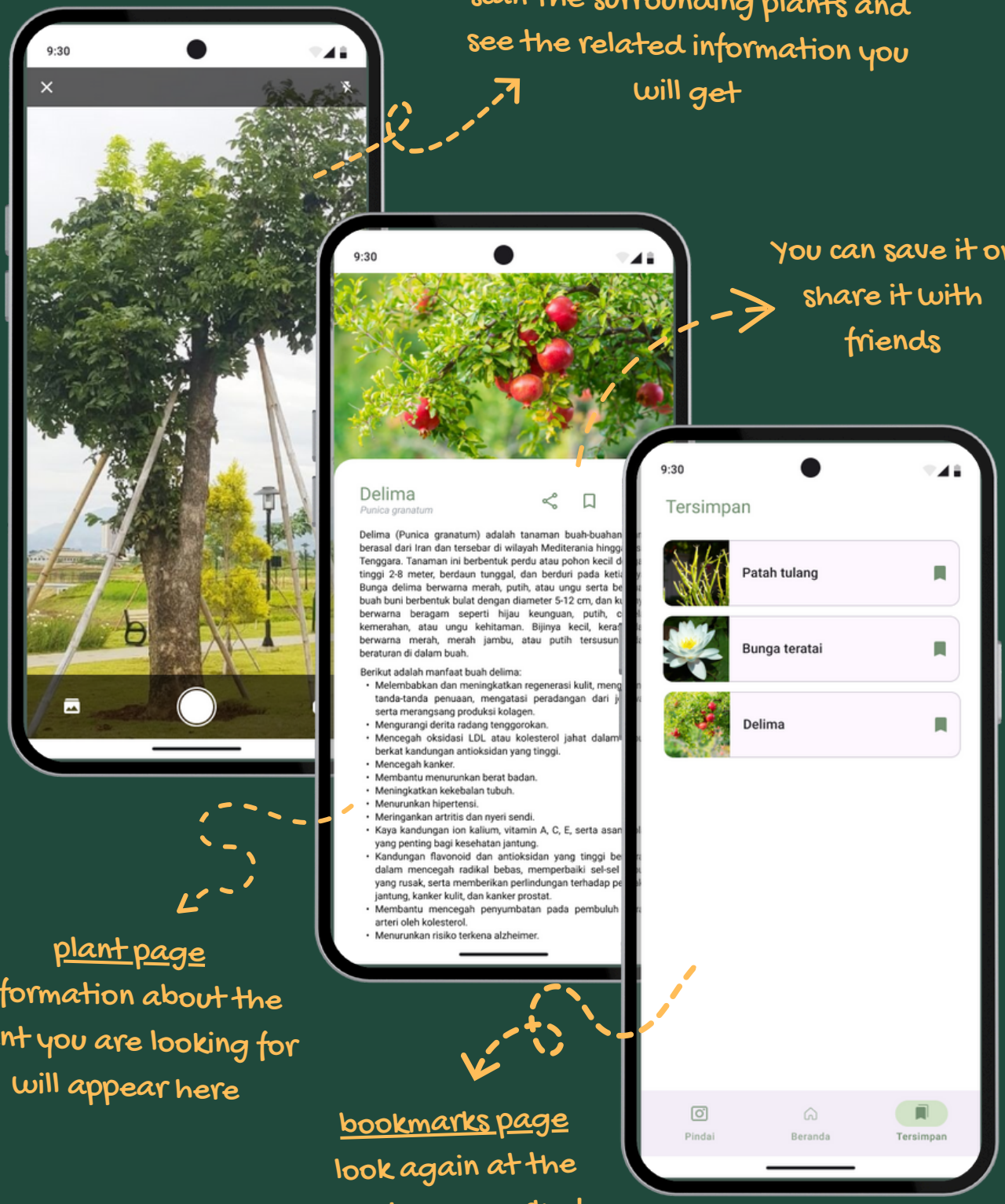
welcome page
new users will see this page

search result page
displays a list of plants relevant to the user's search

Application Design

scan page

scan the surrounding plants and see the related information you will get



You can save it or share it with friends

plant page

Information about the plant you are looking for will appear here

bookmarks page

look again at the plants you marked

Research Question and Objectives

Sometimes people are curious and want to know what plants are around them, but they never really know. Therefore, the research question arises: How to develop a plant image recognition system using machine learning that can identify plant from images and provide relevant information about them.

The aim of the research is to build an application that can help users recognize plants around them.

Research Methodology

1. **Data Collection**

Data collection involves gathering images of plants from PlantNet-300K, the internet, and capturing images oneself. The collected data is then analyzed and processed to separate and label them based on plant names, toxicity, and their benefits.

2. **Data Preparation**

Pre-processing of data is carried out, including resizing, normalization, and augmentation. The dataset is then divided into training, validation, and testing sets.

3. **Model Training**

Supervised learning methods are employed to train the machine learning model. An appropriate model architecture (defined on Method and Technologies section), and the model is trained using the processed dataset until achieving satisfactory accuracy and error rates.

4. **Model Evaluation**

The trained model is tested using a previously unseen testing dataset. Performance metrics such as accuracy, error percentage, and prediction result are measured to evaluate the model's performance.

5. **Application Development**

An Android-based application is developed, allowing access to the user's camera. The machine learning model is integrated within the API that will be called by android app to get the prediction result. The API implemented using microservices and can handle user request with auto scaling.

6. **Testing and Application Monitoring**

A series of tests are conducted to ensure the application's performance and reliability. Continuous monitoring of the application post-launch is performed to identify and rectify any issues that may arise.

Method and Technologies

Tech	Description
Python	Python is used for model and microservices development due to its versatility and extensive ML libraries.
Kotlin	Kotlin is chosen for Android app development for its modern syntax and seamless integration with Java.
Yolo V8	YOLOv8 serves as the object detection base model, known for its speed and accuracy.
FastAPI	FastAPI is selected for backend and microservices creation due to its high performance and simplicity.
Docker	Docker containerizes the backend, ensuring consistent deployment across environments.
Kubernetes	Kubernetes implements load balancing and autoscaling for the containerized backend services, ensuring optimal performance.

Time Tables

Task	Div	Progress						
		W 1	W 2	W 3	W 4	W 5	W 6	W 7
System Design Planning & Application Wireframing	UI/UX							
Application Development	FE							
Backend Integration to Application	FE							
ML Model Integration to Application	FE							
Application Testing	FE							
Create Backend & Endpoint Agreement	BE							
Backend Deployment	BE							
Containerization & Implement Load Balancing and AutoScaling	BE							
ML Technology Research	ML							
Collecting & Cleansing Data For Model	ML							
Create ML Model and Architecture	ML							
Testing & Evaluating Model	ML							
ML Model Deployment	ML							
Create proposals & identify business processes	PM							
Monitor and ensure the progress of each division	PM							
Project Documentation	PM							

W - Week
Div - Division

References

[1] M. A. Haq, A. Ahmed, and J. Gyani, "Implementation of CNN for Plant Identification using UAV Imagery," International Journal of Advanced Computer Science and Applications, vol. 14, no. 4, 2023, doi: 10.14569/IJACSA.2023.0140441.

[2] K. Suwais, K. Alheeti, and D. Al_Dosary, "A Review on Classification Methods for Plants Leaves Recognition," International Journal of Advanced Computer Science and Applications, vol. 13, no. 2, 2022, doi: 10.14569/IJACSA.2022.0130211.

[3] S. Anubha Pearline, V. Sathiesh Kumar, and S. Harini, "A study on plant recognition using conventional image processing and deep learning approaches," Journal of Intelligent & Fuzzy Systems, vol. 36, no. 3, pp. 1997–2004, Mar. 2019, doi: 10.3233/JIFS-169911.

[4] J. Huixian, "The Analysis of Plants Image Recognition Based on Deep Learning and Artificial Neural Network," IEEE Access, vol. 8, pp. 68828–68841, 2020, doi: 10.1109/ACCESS.2020.2986946.

[5] P. S. Kanda, K. Xia, and O. H. Sanusi, "A Deep Learning-Based Recognition Technique for Plant Leaf Classification," IEEE Access, vol. 9, pp. 162590–162613, 2021, doi: 10.1109/ACCESS.2021.3131726.

[6] C. Yuan, T. Liu, S. Song, F. Gao, and R. Zhang, "Research on Plant Species Identification Based on Improved Convolutional Neural Network," Phyton (B Aires), vol. 92, no. 4, pp. 1037–1058, 2023, doi: 10.32604/phyton.2023.025343.

[7] S. Anubha Pearline and V. Sathiesh Kumar, "High Performance Ensembled Convolutional Neural Network for Plant Species Recognition," 2021, pp. 526–538. doi: 10.1007/978-981-16-1103-2_44.

[8] S. Chavan, J. Ford, X. Yu, and J. Saniie, "Plant Species Image Recognition using Artificial Intelligence on Jetson Nano Computational Platform," in 2021 IEEE International Conference on Electro Information Technology (EIT), IEEE, May 2021, pp. 350–354. doi: 10.1109/EIT51626.2021.9491893.

[9] S. K. Mahmudul Hassan and A. Kumar Maji, "Identification of Plant Species Using Deep Learning," 2021, pp. 115–125. doi: 10.1007/978-981-15-7834-2_11.

References

[10] D. Zhou, X. Ma, and S. Feng, "An Effective Plant Recognition Method with Feature Recalibration of Multiple Pretrained CNN and Layers," Applied Sciences, vol. 13, no. 7, p. 4531, Apr. 2023, doi: 10.3390/app13074531.

[11] Peralta, J. H. (2023). Microservice APIs: Using Python, Flask, FastAPI, OpenAPI and More. Simon and Schuster.

[12] Niazi, M., Abbas, S., Soliman, A. H., Alyas, T., Asif, S., & Faiz, T. (2023). Vertical Pod Autoscaling in Kubernetes for Elastic Container Collaborative Framework. Computers, Materials and Continua, 74(1), 591–606. https://doi.org/10.32604/cmc.2023.032474

[13] Imdoukh, M., Ahmad, I., & Alfailakawi, M. G. (2020). Machine learning-based auto-scaling for containerized applications. Neural Computing and Applications, 32(13), 9745-9760.

[14] Muddinagiri, R., Ambavane, S., & Bayas, S. (2019, December). Self-hosted kubernetes: Deploying docker containers locally with minikube. In 2019 international conference on innovative trends and advances in engineering and technology (ICITAET) (pp. 239-243). IEEE.

[15] Nguyen, N., & Kim, T. (2020). Toward highly scalable load balancing in kubernetes clusters. IEEE Communications Magazine, 58(7), 78-83.