Simplifying Plant Care with Plantiful: A Comprehensive Plant Care App

1st Karol Raffay
SPŠ Brno, Purkyňova
Brno, Czech Republic
k.raffay@protonmail.com

2nd Robin Jarůšek
SPŠ Brno, Purkyňova
Brno, Czech Republic
robinjarusek@gmail.com

3rd Valon Mavriqi
SPŠ Brno, Purkyňova
Brno, Czech Republic
valon.m@seznam.cz

4th Tomáš Caha
Department of Telecommunications
Brno University of Technology
Brno, Czech Republic
tomas.caha1@vut.cz

***Abstract*—Caring for plants is not easy for many people. It requires knowledge, it takes time and each plant has different requirements. Platinful is a mobile app developed by three high school students designed to simplify plant care. The app is connected to an external plant recognition API, Plant.id, which provides the app with comprehensive plant care information, watering intervals, and general recommendations for indoor and outdoor plants. This article discusses the development of the Plantiful app, its features, and its potential impact on conservation efforts.**

***Index Terms*—plant care, mobile application, plant recognition, watering reminder, plant health**

I. Introduction

Plants are an integral part of our environment, but not everyone knows how to look after them properly. For one thing, a layperson may not recognise a plant in order to know how to look after it properly. Secondly, people are pressed for time and can simply forget about watering plants.

Our goal was to create a mobile app that users can keep close at hand at all times, making it easier and more accessible to care for plants so that fewer plants die and more are planted. In the app, we provide users with all the information they need to care for their plants - recommended room temperature, sun exposure, watering intervals and the amount of water a given plant needs, all by connecting to the external API called Plant.id.

In the following sections, we will go over our technical solution where we describe the developed mobile app and connections to external services including the technology behind it and our vision for the future in this paper.

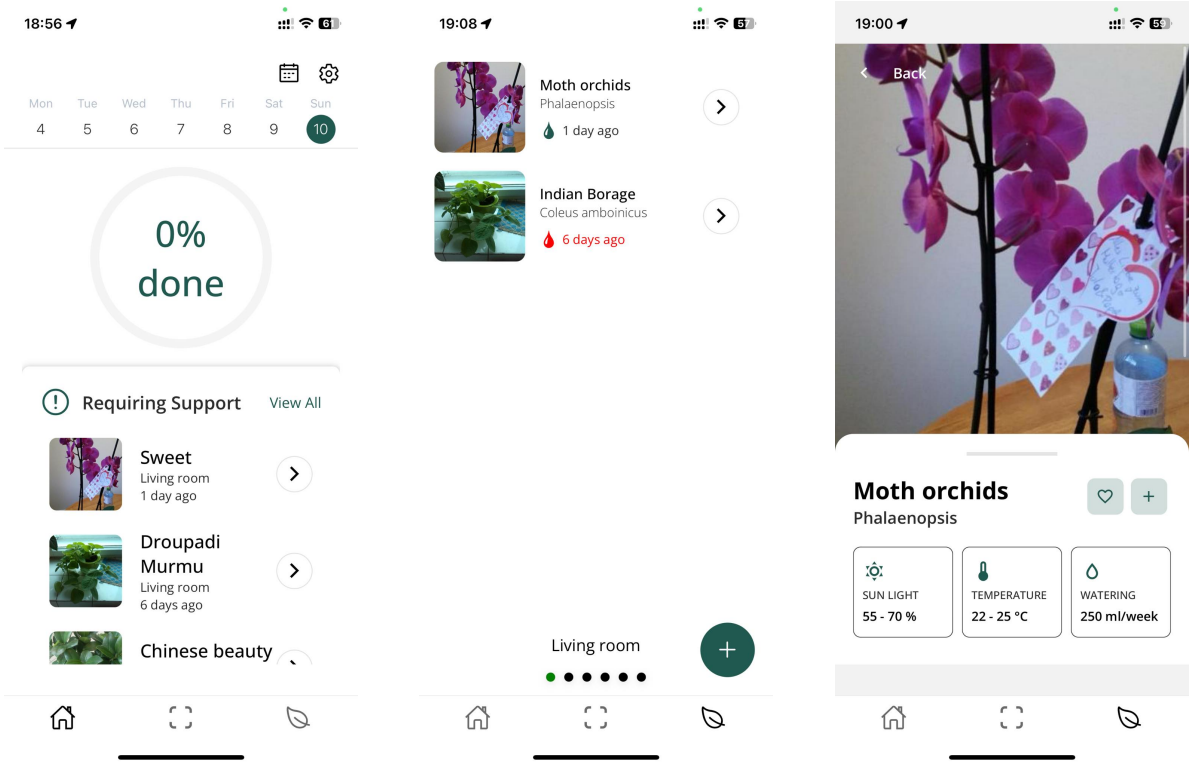
II. Description of developed mobile app

Plant recognition is an essential part of our application. The process of recognising a plant is done through an AI (artificial intelligence) model that is trained on a large amount of specific data (images of plants from different angles, in different lighting conditions, in different climates, with varying health and the corresponding plant's name). To create a model like this ourselves was an impossible solution for us – three high school students; simply gathering all of the necessary images would take ages [1]. So we decided to use Plant.id's API (application programming interface) [2] with their own respective plant database and AI model. Our reasons were their competitive pricing, experience in the industry (launched in 2018), and the fact that they are also a company from Brno.

We have created the following list of key features of our application:

- **Plant and Disease Recognition:** Utilising AI technology, Plantiful can identify a wide range of plants and their potential diseases, providing users with immediate care solutions. See Figure 1a.
- **Personalised Watering Reminders:** Based on the type of plant and its specific needs, the app sends timely watering reminders to users, ensuring optimal hydration levels.
- **Plant Information:** A large database offers valuable information about your plants, including sunlight requirements, soil types, and fertilisation schedules.
- **Categorization by Room:** Users can organise their plants based on their location within their home, allowing for tailored care recommendations that consider the unique environmental conditions of each room. See Figure 1b.
- **Cross-platform:** Our app is designed to make the experience seamless whether you are using an Android or iOS device without sacrificing performance or user experience.
- **Beginner-Friendly:** Designed with simplicity in mind, Plantiful offers an intuitive and easy-to-navigate interface. With step-by-step guides and helpful tips, users can learn at their own pace and gain the confidence needed to care for their plants successfully. See Figure 1c.

We wanted to have a unified code base and ultimately decided to use React Native, which allows us to develop native apps for both iOS and Android without sacrificing performance or code complexity. The downsides are definitely dependencies on third-party libraries and complexity in navigation [3], which is currently split between four different files due to the approach react-navigation uses [4]. Then we had to figure out how to store all of the user data, their plants and rooms, and authorise users. We found a clear favourite, making use of Firebase [5], which is a set of backend cloud computing services and application development platforms provided by Google. It hosts databases, services, authentication, and integration for a variety of applications, including Android, iOS, JavaScript, and many more, which is perfect for our use case.



(a) Home screen. (b) List of plants. (c) Detail of selected plant.

Fig. 1: Screens of Plantiful app

III. Use case example: From Recognition to Care

The objective of this use case example is to demonstrate the application of Plantiful in identifying an unknown plant and setting up a customised care schedule to ensure its optimal health.

The primary actor is a regular user with limited plant knowledge, referred to as "the user" for this scenario. The user has an unidentified indoor or outdoor plant requiring regular care. The whole process can be seen in Fig. 2. When the user wants to identify the plant, he opens the Plantiful application and selects the "Scan Screen". With the device's camera, the user takes a clear image of the plant in question. The application bundles the image along with other necessary data, such as the base64-encoded image, its dimensions, and whether the API should respond with similar images of the scanned plant, into a JSON payload. The payload is sent to the Plant.id API, awaiting a request. The user sees a loading spinner until we get a response from the API (approx. 3 seconds). The response from the Plant.id API contains data for the backend, such as "is_plant", signifying if Plant.id recognized the image as a plant or the probability of a correct prediction, although most of the data, such as the plant's name, description, watering intervals, or if there is a need for fertilising, is for the user himself.

After successful identification, Plantiful prompts the user to select the room or general area the plant is located in or to create a new one. The user then adds the plant to one of the two options mentioned and gets taken to the "Plant's Detail Screen", which details important information such as optimal sunlight exposure or watering frequency. The user can later find all of his plants in the "Plants Screen". Plantiful prompts the user to activate personalised reminders, such as push notifications or general reminders within the application. These reminders are specific to the plant's needs.

IV. Future work

Future work should focus on integrating a health assessment feature to match results of other competitors, full use of Plant.id's data and infrastructure, extensive user testing to ensure the right user experience, and refinement of the app design. Images and plant recognition (meta)data collected from users will be stored securely and anonymously to build our own AI model for plant recognition to reduce dependency on third parties and optimize the app's operational cost per API query.

V. Conclusion

Plantiful can accurately recognize different plants based on data obtained from the external Plant.id API and our own user data. Users can create a list of their own plants, which is stored in cloud storage to their account, and get the information they need to care for their plants, receive watering reminders through our app. The development process was described in Chapter 2, the whole application is built on the React Native framework, which allows us to develop cross-platform for iOS

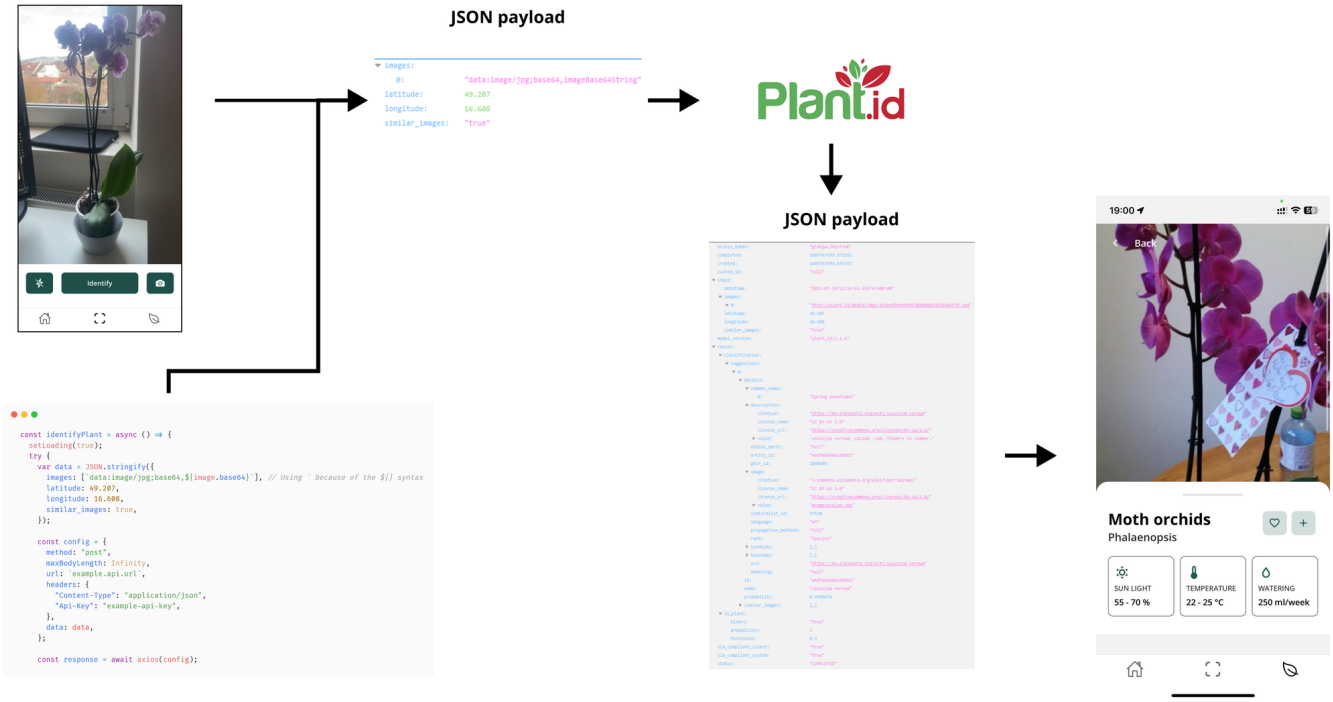


Fig. 2: Use case: Plant identification using Plantiful app

and Android operating systems. Next, the basic idea and its transformation into its current state was described. In Chapter 3, the process of taking a picture of a plant, recognizing it, processing the response from the API and using the data in our application was demonstrated. In the future, it would be good to use more information from the API and focus on reducing the amount of API requests.

Development of this app started in 2022 as part of the Red Hat summer camp competition and after that it secured a year-long internship at Kyndryl. Lastly, we participated in the JA EXPO 2023 competition, winning the Best Business Pitch and Best STEM Project categories.

Acknowledgment

We would like to thank Kindwise, who develops and operates the Plant.id API, without whose data this application would not work. They kindly reached out to us and offered to increase our API usage credits to make our development work smoother.

References

[1] REYES, Angie K.; CAICEDO, Juan C.; CAMARGO, Jorge E. *Fine-tuning Deep Convolutional Networks for Plant Recognition*. CLEF (Working Notes), 2015, 1391: 467-475.

[2] *Plant.id - Plant identification app*. Accessed March 10, 2024. [Online]. Available: https://plant.id/

[3] *Navigating Between Screens · React Native*. Accessed March 10, 2024. [Online]. Available: https://reactnative.dev/docs/navigation

[4] *React Navigation — React Navigation*. Accessed March 10, 2024. [Online]. Available: https://reactnavigation.org

[5] *Firebase — Google's Mobile and Web App Development Platform*. Accessed March 10, 2024. [Online]. Available: https://firebase.google.com