

BloomMate

House Gardening with AI Chat using LG Tiun

An Soonho
*dept. Information Systems
Hanyang Univ.*
Seoul, Republic of Korea
ash1823@hanyang.ac.kr

Kim Donghyun
*dept. Information Systems
Hanyang Univ.*
Seoul, Republic of Korea
akainoo@hanyang.ac.kr

Shin Hyunah
*dept. Information Systems
Hanyang Univ.*
Seoul, Republic of Korea
hyunah0923@hanyang.ac.kr

Yoon Yongsung
*dept. Information Systems
Hanyang Univ.*
Seoul, Republic of Korea
mirr001003@hanyang.ac.kr

Abstract—BloomMate is a user-friendly gardening application tailored for beginners or individuals with limited gardening experience, particularly those who have acquired the LG Smart Cottage and intend to optimize their gardening with the LG Tiun. This app simplifies the gardening process by allowing users to register their plants and access a range of personalized guidance to enhance their gardening skills.

The standout feature of BloomMate is the unique capability to communicate with your plants. Through generative AI technology, this app facilitates direct interaction with plants, creating a personalized and engaging experience for users. Moreover, it exploits AI to identify and diagnose plant diseases, offering valuable insights to maintain the health of your pet plants.

Index Terms—gardening, generative AI, Matter, AI

I. ROLE ASSIGNMENT

Below is the table of role descriptions. Due to the small members in the team, some roles are distributed to the same person in order to harmoniously progress the project.

Roles	Name	Task description & etc.
User	Shin Hyunah	She plays the role of a plant grower. She has little experience growing plants and doesn't know how to water and nourish them. Additionally, she often kills her plants because she doesn't have time to take care of them, so she needs a service that takes care of her plants automatically.
Customer	Yoon Yongsung	He is a person who owns a house, and in particular, he has purchased an LG Smart Cottage or has an LG Tiun. He goes to his house on the weekends, enjoys country life, and takes care of plants. He wants software that automatically manages his plants even on weekdays when he is away or on weekends when he cannot be there.
Software developer	An Soonho	As a software developer for our project, he designs and creates APIs and databases, and builds the AI that can implement the requirements of the service we plan.
Development manager	Kim Donghyun	He is responsible for the overall management of our project, distributing each task, setting and managing task deadlines. Additionally, he helps resolve any issues between the backend and frontend.

II. INTRODUCTION

A. motivation

a. Customer Trend Analysis

1) positive effect of pet plant

Pet plant is known for positive effect to give psychological/emotional stability and inspiration about life satisfaction. Especially, it is showing clear tangible results in several areas like overcome stress and depression, therapy various physical illnesses, air purification.[1]

2) The appearance of Green Hobby

As interest in planterior increases, sales of flower gardens/flowers increased compared to the previous year (2019). The search volume of 'planterior' on Naver, a search portal site, also increased significantly during the pandemic compared to pre-COVID-19, especially during the April 2020 pandemic. Also, attention in urban farmer, self-gardening, weekend farm is growth. An indicator that can confirm this is the number of Instagram hashtag posts, with the home gardening tag exceeding 520,000 and the planterior tag exceeding 163,000. [2]

3) Impact of COVID-19

Pet plant has long received attention to relieve loneliness in single-person households or the elderly, and the home gardening market, including pet plant, has grown rapidly as restrictions have arisen on outdoor activities since the COVID-19. Compared to before COVID-19, sales of flowerpots increased by 20%, gardening supplies increased by 48%, and seedlings increased by 92%. March to April 2020, SSG.com saw a 97% increase in home gardening category sales, and Enuri.com saw a 491% increase in garden set sales. Accordingly, the Rural Development Administration predicted that domestic home gardening sales will increase rapidly after COVID-19, reaching 500 billion won in 2023, an eight-fold increase from 60 billion won in 2020. [3]

4) Expanding the market size of plant cultivation machines

The market for plant growers that allow you to grow flowers and vegetables at home becomes larger. The market size, which was 10 billion won in 2019, increased to 60 billion won in 2020 and is expected to reach 500 billion won in 2023. [4]

5) Urban agriculture

The number of 'urban farmers' in Seoul, who farm on rooftops, verandas, and weekend farms, increased 14-fold from 45,000 in 2011 to 640,000 in 2019. During the same period, Seoul's urban agricultural space also increased 6.9 times from 29ha to 202ha. [5]

6) Desired residential space after retirement

In 2021, the real estate platform 'Zigbang' conducted a survey on the residential space desired after retirement, and 'country house, townhouse' ranked first with 38%, beating apartments (35.4%). When comparing housing transactions in the first half of the year by housing type, apartment transactions decreased in June compared to January, while house transactions increased. [6]

Considering these data, interest in horticulture such as home gardening and planterior is steadily increasing, and along with this, interest in weekend farms, vacation homes, country houses, etc. is also steadily growth. Therefore, it can be expected that the demand for gardening or growing plants in country houses will gradually get larger. At the same time, sales of plant-related home appliances will also become greater.

b. Company Analysis

1) Regulations related to the companion plant industry

As interest in pet plants grew, the Seoul Metropolitan Council passed the 'Ordinance on Fostering and Supporting the Pet Plant Industry' at the plenary session in May, laying the foundation for the growth of the pet plant industry and focusing on expanding supporting projects.[7]

2) LG Electronics: Innovation for a Better Life

LG Electronics is currently trying to change from being a company that only sells simple home appliances to a company that provides a lifestyle that can change customers' lives to be smarter through applications such as ThinQ.

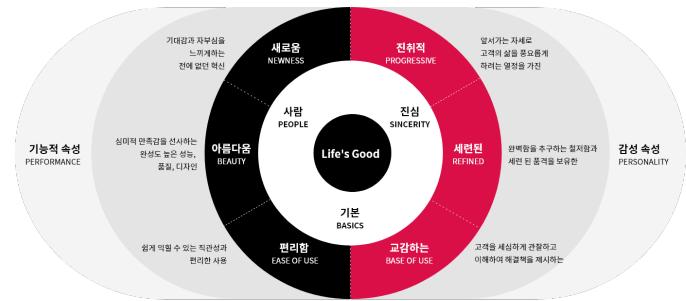


Fig. 1. LG electronics Brand Goal

Customers are always at the center of the LG brand story. LG makes an incredible contribution to the lives of its customers and always strives to make their lives more valuable.

i. PEOPLE

Just as important as a good product is the customer experience using that product. Customers' lives are improved with new products through creativity

and innovation, and a richer life is completed by experiencing products with excellent aesthetic satisfaction. Every product we create must be able to play that role in our customers' lives. Creating a better life is how we do our best for our customers.

ii. SINCERITY

How to make life new and more convenient, our technology always starts with sincerity toward customers and their lives. It is an attitude of constantly making passionate efforts and pursuing sophistication through thorough perfection rather than flashiness, and the most important factor when creating products is sincerity.

iii. BASICS

Rather than being different for the sake of being different or flashy for show, we focus on the core values that customers who use the product expect. The core value refers to offering important value to customers' lives beyond the product's functions. [8]

c. Product Analysis

1) LG Smart Cottage

It is a small modular house in the form of a second house that combines LG Electronics' advanced energy, heating, and cooling technology, and differentiated premium home appliances, with a two-story one-room structure measuring 31.4m². The 4kW solar panel installed on the roof can cover a significant amount of the electricity used per day by two adults, and the heat pump cooling and heating system 'Therma V Monobloc' is installed to significantly reduce energy consumption. A home energy solution was implemented that stores used power in a home ESS system. There is a charger for electric vehicles (EV) outside the Smart Cottage, and the inside is equipped with premium home appliances equipped with various technologies to increase energy efficiency, such as the Objet Collection Wash Tower Compact, a dishwasher, an induction electric range, and a water purifier. Through the 'LG ThinQ' application, you can check home appliances, heating, cooling, and air conditioning (HVAC) systems, and energy storage and consumption. It has a premium interior with a warm atmosphere, with a sophisticated design based on the concept of walnut wood tone and an exterior finished in light beige.



Fig. 2. LG Smart Cottage

2) LG Tiiun



속비업소의 소형 냉장고 만한 크기다 (출처=LG전자 홈페이지)

Fig. 3. LG Tiiun

LG Tiiun is a home plant grower designed to target the home gardening market, which is expanding due to the impact of COVID-19. It can be said to be the culmination of LG's technologies, combining the temperature control technology of the Dios refrigerator, the air conditioning technology of the Whisen air conditioner, the inverter compressor technology of the Tromm dryer, the water supply control technology of the PuriCare water purifier, and the LED lights of LG Display. You can easily and comfortably grow plants by filling them with water 6 times and adding nutrients 6 times in a month. The power consumption for one year is 42.5kWh, which is similar to that of a small refrigerator. LG Tiiun consists of two shelves (upper and lower compartments), and a total of six seed kits can be installed, three each. Seed kits limited to 2 types of herbs, 4 types of leafy vegetables, and 1 type of flower are being sold. Through the LG ThinQ app, you can

check the operation/growing status on your smartphone and receive a notification when nutrients or water are insufficient.

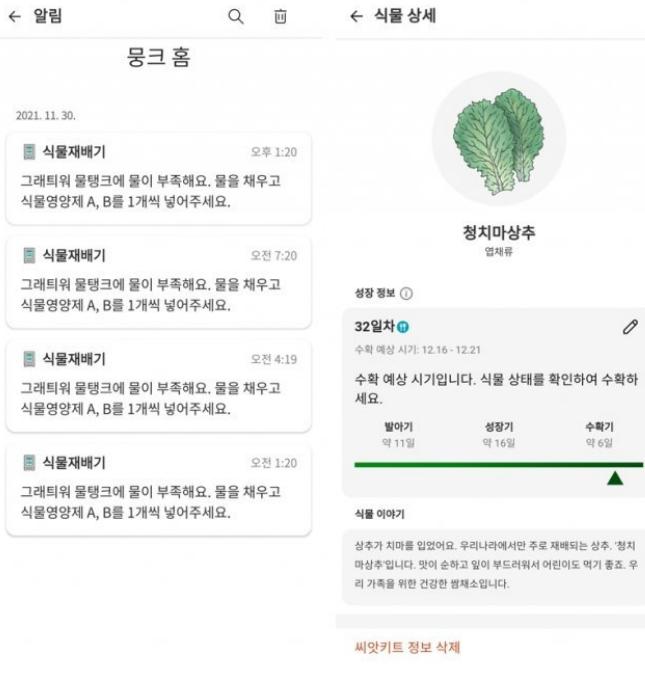


Fig. 4. Example of using ThinQ for LG Tiiun

B. Problem Statement

a. LG Smart Cottage: absence of home turf

People live in house almost enjoy gardening. However, LG Smart Cottage that LG Electronics launched doesn't have home turf. So, we think LG Tiiun for outside to help with LG's brand goal is necessary. Especially in the case of smart cottages, many people buy them as a second home and are unlikely to be there every day. In this case, they may struggle to get information about the condition of the plants, unless they have restrictions except for gardening with Tiiun.

b. Brown Thumb

In abroad, the term, brown thumb, means the people who always kill the plant because of lack of skills. Like this, there are more things to worry about in growing plants than you think, so if you don't pay attention, you often kill the plants, and even if you care a lot, there are many cases where you kill the plants because you don't know how to grow them properly. Therefore, Software, to help all of the plant growers can grow them easily, is essential.

c. Rasing Plants: absence of communication

Although raising pet plants is already known to help with various emotional stability, it was judged that there was less of a feeling of communication with each other compared to animals with which one could actually communicate. In order to grow plants well, constant attention is needed, and for this to happen, it is important to have affection for the plants. Therefore, in order to increase the fun of growing plants, it is necessary to be able to communicate with plants.

C. Solution

a. Meeting the needs of more customers

Considering the aforementioned, the number of people interested in gardening, cottages, etc. will increase. To contribute to increasing the value of life for the growing number of caterers' customers who grow their own gardens in cottages, we have adopted LG Tiiun which can be used in the outside garden of LG Smart Cottage as a target appliance.

b. Brown thumb turns into green thumb with LG

We have designed a software that allows people who are not talented in plant care to easily grow plants. To briefly introduce the features, customers can register their plants with the app and it will adjust the appropriate temperature, humidity, light, ventilation, etc. according to the plant species and automatically spray water and nutrients. Through the camera, the current health status of the plant (pest infestation, wilting, etc.) can be identified using AI, so that the customer can come and check the plant before it dies.

c. Build a bond with plants

One of the disadvantages of growing plants is that customer can only get an idea of their condition by looking at the wilting of their leaves or the appearance of fruits or flowers, and can't communicate with them directly like animals. These disadvantages reduce the fun of growing plants. To overcome this, we devised a feature that allows customer to communicate with plants using generative AI. If customers can communicate with their plants, they will be able to build a bond with them and give them more affection.

D. Research on Any Relative Software

a. Groo

Groo is an application for plant care scheduling, community, plant books, and plant-related purchases. You can enter your experience with plants and where you grow them, and it will recommend the best plants for your space. You can also take photos of your plants to learn about their types and care, and use the plant guide to find out what to look out for. You can also communicate with other people who are interested in

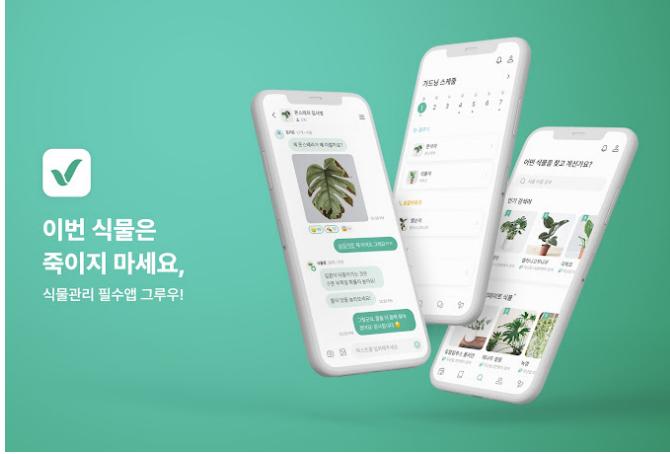


Fig. 5. Groo

growing plants through the community and buy gardening supplies. What makes it different from other plant growing applications is that it uses vision AI to check the health and disease of plants.



Fig. 6. Diagnosis of plant health using vision AI

b. Planta

Planta is an application that helps you grow plants, similar to Groo. You can add information about your plants to the app and it will automatically remind you when to water, fertilize, spray, clean, etc. based on an algorithm. You can even take a photo of your plant and get the information you need by identifying the plant's variety. You can also get plant advice, keep a diary to track your plants' growth, and more. One of the unique features is that the app also recognizes where you live and recommends plants that are suitable for your hardiness zone and adjusts the watering schedule according to the local weather.



Fig. 7. Planta

c. PlantLink

PlantLink is an application that allows you to manage plants by connecting the Soil Module, an IoT device. Through the IoT device, you can record changes in soil humidity, illumination temperature, etc. to create a growth report. It is possible to talk to plants through chatbots and conduct psychological analysis through these conversations to provide mental health reports and receive plant management advice, but it is currently not working smoothly.



Fig. 8. PlantLink

d. Greg

Greg is the zero-guesswork plant care app & community. It makes growing indoor plants super easy and fun. If you scan any plant, Greg's plant vision will identify the species, tell you all about it, measure the pot, and even the distance to the nearest window. And it provides highly-personalized plant care, a custom watering plan based on each plant's species, size, plus the actual details of your home environment. Moreover, you can communicate with other veteran growers for answers to any plant questions you have.

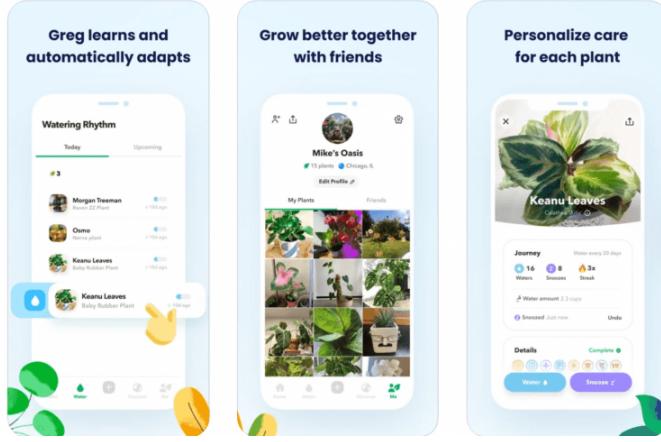


Fig. 9. Greg

III. REQUIREMENT ANALYSIS

A. Create an account

When you first install and run the application, you'll see a page where you can sign up or log in. If you click Sign Up, we'll ask you a few questions to get your information and create your account. The following information is required from users when signing up.

- Name
- Picture
- Location of home
- The size of the garden
- ID and Password
- The serial number of users' LG Tiiun

After signing up, you'll be taken to the login page. You can edit all the information you received during the signup process on your My Page.

B. Log in

Logging in is an essential step for every member to access and manage their own garden on BloomMate, and it's the gateway to utilizing BloomMate's full range of features. The login process involves entering the ID and password you provided during registration. Once successfully logged in, you'll be directed to a screen that allows you to either add new plants or view the plants you have previously registered, depending on whether you have any plants registered. The login requirement is a fundamental aspect of the BloomMate platform. It empowers users to have a personalized gardening experience, enabling them to manage their garden with ease. The authentication through the use of an ID and password is crucial for maintaining account security and ensuring the protection of sensitive information.

C. Add own plant

Logged-in users who don't have any registered plants will be taken to the plant registration screen. The plant registration

process is essential for managing plants and chatting with them. To register a plant, users need to enter some information. The following is a list of information required to register a plant.

- Name of plant
- Plant variety
- Photo of plant
- Planted date

At this stage, you have the following constraints.

- Limited plant types
 - Provide application services within a limited plant to provide optimal personalized information.
- Limited number of plants based on the size of your garden - Warns that planting more plants than the size of your garden can accommodate may result in poor gardening.

After entering the above information, the user can view their plants on the registered plants screen.

D. See the registered plants

Your registered plants will appear in a list, showing the name of the plant you entered when you registered it and how many days it has been with you based on the date you planted it. Clicking on each plant will take you to a page where you can view the plant's details. If there are no plants, the user will see a screen with a button to add a plant.

E. View details of a plant

You can click on each plant to view its details, and the first thing you'll see is a photo and name of the plant you've registered. Also, you can start chatting with the plant by clicking the chat button. The plant details page contains the following information.

- Name of plant
- Plant variety
- Planted date
- Basic information based on plant variety:
 - Appropriate temperature
 - Appropriate illuminance
 - Soil moisture
 - Cultivation difficulty
 - Flowering season
 - Watering cycle
 - Caution
 - etc.

You can also take a photo to detect the current status of a plant by clicking checkup button.

F. Chat with your plant

Users can converse with their plants using ChatGPT, a popular generative chat AI. This AI assumes the role of the

plant, providing regular updates on its condition and growth. In addition to standard plant status and growth records, users can engage in daily conversations with their green companions. ChatGPT shares insights into the plant's current state, addresses questions and concerns, and offers care advice. Critical information, such as plant status and growth records, is automatically delivered daily and accessible on the chat screen. Users can easily search for past information by date in the chat history. This feature fosters a stronger connection between users and their plants, enhancing the gardening experience.

G. Examine plants through photos

Users have a convenient way to assess the health of their plants by simply taking a photo. When users capture an image of their plant, leaves and all, an AI model steps in to evaluate the plant's current health status and check for any potential diseases. This feature not only informs users about the plant's overall condition but also provides guidance on how to manage any issues that may arise. If there are any health concerns, the AI not only identifies the issue but also offers detailed information about the specific disease, if present. Furthermore, it goes the extra mile by suggesting effective methods for addressing the problem, whether it's a common issue that can be easily resolved or a more complex condition requiring special care.

H. My page

My page is where you can view your membership information. You can take the following actions from here.

- View user's name and picture
- Modify account information
- View about BloomMate

IV. DEVELOPMENT ENVIRONMENT

A. Choices of Software Development Platform

a. Development Platform

1. Windows

Windows is an extremely popular and widely used operating system for software development, known for its versatility and user-friendly interface. Currently, Windows is the pre-installed OS on most computers in South Korea, far outpacing other OSes in terms of share. It offers a multitude of features and supports a vast array of programming languages and tools, making it the go-to choice for developers worldwide. With Windows, developers have the flexibility to utilize various development environments and frameworks, empowering them to create applications efficiently and effectively. Whether it's desktop applications, web development, or mobile app development, Windows provides the necessary tools and resources to meet the demands

of any project. Additionally, Windows continually evolves and updates its platform, ensuring that developers have access to the latest technologies and advancements in the software development field. In conclusion, Windows is an indispensable platform for software development, offering a comprehensive ecosystem and empowering developers to bring their ideas to life.

2. MacOS

MacOS is a popular operating system favored by many developers, especially those in the creative and design fields. Known for its sleek and intuitive user interface, MacOS offers a seamless development experience. It provides a wide range of powerful tools and frameworks, making it an excellent choice for building applications for various platforms. With MacOS, developers can take advantage of tools like Xcode, which is the primary integrated development environment (IDE) for Apple's platforms. This is especially important when the application we want to develop must also support IOS. To develop for iOS, Xcode is essential, and Xcode can only run on MacOS. Xcode offers a comprehensive set of tools and resources for developing iOS, macOS, watchOS, and tvOS applications. Additionally, MacOS has a strong integration with other Apple products and services, allowing developers to create a cohesive ecosystem across different devices. Overall, MacOS provides a robust and efficient development environment for developers looking to create software for Apple.

b. Tools and Language

1. JavaScript(JS)

JavaScript is a widely-used programming language known for its versatility and its ability to create dynamic and interactive web applications. It serves as the backbone of modern web development and is supported by all major web browsers. With JavaScript, developers can add functionality to websites, handle user interactions, and manipulate web page content. It is a high-level, interpreted language that allows for rapid development and prototyping. After the introduction of Google's V8 engine-based Node.js, JavaScript can now be executed in a non-browser environment as well. JavaScript also benefits from a vast ecosystem of libraries and frameworks, which enhance its capabilities and simplify the development process. These libraries and frameworks are published by npm, the Node Package Manager, and developers can easily download and utilize them in their projects. Some notable features of JavaScript

include its single-threaded nature, prototype-based language structure, and its status as a script language. JavaScript is regularly updated by ECMA (European Computer Manufacturers Association), with the updated version being referred to as ECMAScript. Recent ECMAScript versions support object-oriented programming and functional programming. In summary, JavaScript is an essential language for web development, empowering developers to create engaging and interactive user experiences on the web.

2. TypeScript(TS) [9]

TypeScript is a superset of JavaScript that provides static typing, making it a powerful tool for building robust and scalable applications. It was developed by Microsoft and first published to the public in 2012. With TypeScript, developers can catch errors during compile-time, resulting in fewer bugs and improved code quality. It offers advanced features such as interfaces, generics, and decorators, which enhance code organization and maintainability. TypeScript is fully compatible with JavaScript, allowing developers to gradually migrate their existing JavaScript codebases to TypeScript. It also seamlessly integrates with popular JavaScript libraries and frameworks, making it a versatile choice for web development. Nowadays, companies are compelled to adopt TypeScript instead of JavaScript, regardless of project size, due to its numerous benefits. Overall, TypeScript is a valuable addition to the development environment, offering enhanced type safety and productivity for developers.

3. yarn [10]

Yarn is a JavaScript package manager that helps manage project package dependencies and facilitates package sharing among developers. Due to security and performance issues experienced with npm as their projects grew larger, Facebook developed Yarn as a replacement. Yarn offers improvements in performance and security compared to npm. One key advantage of Yarn is that it installs packages in parallel, unlike npm which installs them sequentially, resulting in faster installation times. Yarn also utilizes caching, further speeding up package installation. Additionally, Yarn utilizes semantic versioning to specify package versions. For this project, we have chosen to use Yarn due to its fast performance. This is especially important as we will be installing various frameworks and libraries for frontend development.

4. Eslint/Prettier [11]

ESLint performs automated scans of your JavaScript and TypeScript files to detect common syntax and style errors. Prettier scans your files for style issues and automatically reformats your code to ensure consistent rules for indentation, spacing, semicolons, and quotes (single vs double). We use these tools on our teams for the following reasons:

- They promote consistency by enforcing the same rules for everyone.
- They save time in code reviews by allowing us to focus on code structure and semantics instead of style issues.
- They catch errors, with ESLint being particularly effective at detecting syntax errors and basic type errors like undefined variables. Although Prettier doesn't catch errors, it still contributes to code quality.
- Changes made by these tools are automatically applied to code when the file is saved in Visual Studio Code.

While setting up these tools requires an initial time investment, the time-saving benefits accumulate over time.

5. React-Native [12]

React Native is an open-source UI software framework developed by Meta Platform, Inc. It enables developers to use the React framework along with native platform capabilities to build applications for Android, Android TV, iOS, macOS, tvOS, Web, and Windows. Unlike web apps, React Native communicates with the Native Thread through native bridges, resulting in optimized performance. In terms of hybrid frameworks, Flutter has an advantage in making it easier for web developers to write React Native code. Previously, there was a perception that Flutter was much faster and lighter. However, with the introduction of Fabric in React Native 0.68, which supports faster communication between Android and iOS, that perception has become a thing of the past. That being said, React Native does heavily rely on other libraries, even for core features like the camera function in mobile applications, which requires the use of third-party libraries. Nevertheless, due to the aforementioned advantages, it is difficult to find a better option than React Native for creating fast and cross-platform applications that support both Android and iOS.

6. React Query [13]

React Query is a library that simplifies the process of fetching, caching, and keeping your

React application's server state synchronized and updated. Unlike other data fetching methods that require complex and verbose code, React Query provides a simple and intuitive API that can be used within React Components. Previously, managing state in React applications lacked clear distinction, leading to confusion when writing code. Specifically, determining which state should be continuously fetched from the network and which state is solely dependent on the client required a thorough understanding. However, with the introduction of React Query, you can easily differentiate between server state management and client state management in your React application. This distinction enables efficient collaboration and simplifies cache operations, loading operations, and the implementation of difficult asynchronous operations in network state management.

7. Recoil [14]

In React, it can be challenging to directly pass data between two different components unless they have a parent-child relationship. In such cases, you need to send the data to the parent component and then pass it back to the component that requires it. This process, known as props drilling, can make it difficult to keep track of the props being passed around. That's why having a global state management library is crucial for modern React applications. React's own Context API requires implementing it from scratch and requires significant effort to prevent unnecessary re-rendering. On the other hand, Redux, which was the most popular choice until 2020, has a steep learning curve and a complex architecture that can make it challenging to understand the code flow quickly. In contrast, Recoil is relatively easy to use if you understand the basic syntax of React. It is also free from re-rendering issues and provides various optional features for global state management. Additionally, Recoil is lightweight and does not significantly impact application performance.

8. Python [15]

Developed in 1991 by programmer Guido van Rossum, Python is known for its readability and easy syntax, making it quicker to learn compared to other programming languages. As a result, it has gained popularity among non-programmers and is utilized in various fields such as statistics, data analysis, modeling, deep learning, and artificial intelligence. Python's intuitive and user-friendly syntax makes it highly recommended for beginners in programming. Additionally, it is the most widely adopted

language, meaning that many libraries necessary for application development are readily available. Therefore, we have chosen Python for building the backend and training machine learning models.

9. Django [16]

Django is a popular and widely used web application framework written in Python. It offers robust features and components for efficient website and web application development. One of its key advantages is simplifying and accelerating the development process. Django provides pre-built components for user authentication, an admin panel, forms, and file uploads. User authentication is made simple with a few lines of code, ensuring secure sign up, log in, and log out. The admin panel allows easy content and data management. Django's form handling system simplifies the implementation of complex forms with validation and error handling. Handling file uploads is effortless with Django's built-in capabilities. Additionally, Django has a vast ecosystem of libraries and extensions, such as djangorestframework for creating RESTful APIs. Overall, Django is a versatile and powerful framework for building robust and scalable web applications, suitable for both beginners and experienced developers.

10. SQLite [17]

Since SQLite doesn't run as a server process, it's a ready-to-use database. It's lightweight but has all the necessary features. It supports transactions and is platform-agnostic. Since it's a locally run DB, it's also highly portable. However, it is not an isolated environment, so the server must also handle the burden of the DB. Additionally, you can't authorize users, only assign basic access rights based on the operating system. We decided to use SQLite because it is supported by Django and our application doesn't have a complex database structure or require various features of a database management system.

11. AWS EC2 [18]

Amazon Elastic Compute Cloud (Amazon EC2) offers scalable computing capacity on-demand in the Amazon Web Services (AWS) cloud. By using Amazon EC2, you can accelerate application development and deployment while reducing hardware costs. Instead of purchasing physical servers, our team has decided to utilize a cloud system for its economic benefits and flexibility for future expansion. With Amazon EC2, you can deploy multiple virtual servers, configure security and networking, and manage storage.

It allows you to increase capacity (scale up) to handle compute-intensive tasks such as monthly or annual processes or spikes in website traffic. Conversely, when usage decreases, you can decrease capacity (scale down). One of the great advantages of Amazon EC2 is that it is free for the first year. Since our application needs to be built urgently and will not run for more than a year, Amazon EC2 is an excellent choice.

12. Jupyter NoteBook [19]

Jupyter Notebook is a popular open-source web application that provides an interactive environment for writing code, visualizing data, and documenting. It supports various programming languages, including Python, R, and Julia. Jupyter Notebook is organized into code cells and markdown cells. In code cells, you can write and run Python code, while in markdown cells, you can write documents, formulas, or insert images. This combination of code and documentation is especially useful for projects like data analytics and machine learning. One of the advantages of Jupyter Notebook is the ability to run Python code line by line and view the results, which is helpful for debugging and testing. The output is immediately displayed below the code, allowing you to see visualized graphs, tables, and more. Additionally, Jupyter Notebook can be easily used online. Google Colab, provided by Google, is a cloud-based Jupyter Notebook environment. For our project, we will utilize Jupyter Notebook to run machine learning tasks using Python. Machine learning often involves writing lengthy code, and it is crucial to check and debug the results along the way. Jupyter Notebook is the ideal tool for this purpose.

13. Tensorflow / Tensorflow light [20]

TensorFlow is a library created by Google that provides features for implementing deep learning programs. It is primarily implemented in C++, but supports multiple languages including Python, Java, and Go. However, Python is prioritized as it is most suitable for our application. Machine learning is a complex field, but with machine learning frameworks like TensorFlow, implementing a machine learning model has become more accessible and less complicated. These frameworks simplify tasks such as data acquisition, model training, making predictions, and refining results. Additionally, TensorFlow Lite is a mobile library designed for deploying models on mobile devices, microcontrollers, and other edge devices. When running a machine learning or deep learning model through TensorFlow Lite,

it produces a small file of less than 3MB. This allows us to use trained models on mobile without any capacity constraints. In our application, we will use TensorFlow Lite to ensure immediate model availability on mobile devices.

14. Material Design [21]

UI/UX is crucial in modern mobile and web applications, and a unified UI can greatly enhance the user experience. This unified UI is achieved through the use of a design system. However, implementing a design system on your own can be challenging. Therefore, it is common practice to utilize publicly available design systems, with Google's Material Design being the most prominent one. Material Design is a design approach that combines the benefits of flat design with the use of shadows to create a sense of depth. It also provides a comprehensive design system that includes typography and primary-secondary color schemes. This design can be easily implemented in the Figma library and seamlessly integrated into code using react-native-paper.

15. Git [22]

Git is an effective version control system that allows you to manage versions and incorporate changes and updates seamlessly. But before delving into Git, let's first discuss what a "version control system" is. It is a system that records changes to a file over time and enables you to retrieve that file later when needed. When working on a document, there are usually multiple revisions and updates from the initial draft to the final version. Along the way, we often rename the file to "final," "_final," "finalized," and so on, thus overwriting the previous versions. This can make it difficult to go back to a specific point in time and understand the changes that were made. However, with a version control system, this becomes possible. By using a version control system, you can manage multiple versions of the same information. This allows you to track changes over time and identify the individuals who made them. You can easily revert back to previous versions or the original version, and quickly identify the person responsible for any issues that may arise.

16. Cloudinary

Cloudinary is a cloud-based media management platform that enables you to efficiently manage, optimize, and distribute images and videos. This service is particularly useful for effectively managing and optimizing the media assets used

in your web and mobile applications. While AWS's image server could have been an option, we considered the fact that our machine learning models run within Django. Using AWS's static server would have compromised server capacity and performance. Therefore, we made the decision to utilize Cloudinary as a completely new static server for images. Now, the AWS server only needs to store the image download addresses from Cloudinary in its database.

B. Software in use

1. Visual Studio Code



Fig. 10. Visual Studio Code

Visual Studio Code is a lightweight text editor developed by Microsoft, which is excellent for web development. Nowadays, it offers a wide variety of powerful plugins that make it as lightweight and powerful as other IDEs. Our team exclusively uses Visual Studio Code for all of our code development. For front-end development, there are plugins available for React-Native and for seamlessly transferring work from Figma to code. Furthermore, there are plugins for Eslint and Prettier that automatically apply code formatting when you save a file, greatly enhancing the overall developer experience. In addition to front-end development, Visual Studio Code is also used for backend and machine learning tasks. The Python and Jupyter Notebook plugins are highly regarded, and the debugging capabilities are excellent.

2. Xcode



Fig. 11. Xcode

XCode is an integrated development environment (IDE) developed by Apple. It is essential for developing an iOS application using React Native. While we write the code in Visual Studio Code, we need to compile it as

an iOS application using XCode.

3. Github

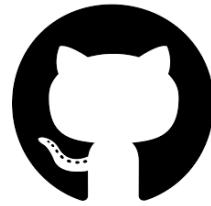


Fig. 12. Github

Github is a program that supports projects using Git, acting as a remote control center for Git. It serves as a platform for version control and collaboration among developers, offering a cloud-managed version control system. Git and Github are often used interchangeably for collaboration in modern software development. However, the capabilities of Github go beyond that. Firstly, Github is the preferred platform for open-source software, providing access to the source code of various tools used by our team. Additionally, any issues or bugs with open libraries can typically be found on Github. Github also offers other collaborative features. Pull requests allow us to review work in different Git branches before merging them. Furthermore, Github actions simplify the process of implementing continuous integration and continuous deployment (CI/CD). In our team, we utilize Github actions to check for errors in TypeScript on the frontend and to instantly reflect code changes to AWS EC2 on the backend.

4. Figma



Fig. 13. Figma

Figma is an application for UI/UX design, available as a cloud-based SaaS program. One of its main advantages is its real-time nature, allowing users to see immediate modifications to the UI within Figma. This eliminates the traditional process of a planner or developer reviewing a designer's file, requesting changes, waiting for updates, and then reviewing and modifying it again. Now, anyone can receive instant confirmation and immediate feedback, regardless of their job title. Another great aspect of Figma is the extensive collection of plugins and communities available. Many people have created useful design components and shared

them with the Figma community. This allows users to leverage high-quality UIs without having to create them from scratch. Lastly, Figma offers the functionality to create prototypes. These prototypes can be used to demonstrate button interactions within Figma, providing developers and planners with a clearer understanding of the application flow.

5. Slack

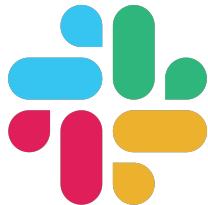


Fig. 14. Slack

Slack is a communication tool that offers several advantages. One of the biggest advantages is the ability to create separate channels for different teams and topics. This helps to avoid crowded conversations and keep discussions organized. Additionally, within a channel, you can use the thread feature to have focused discussions on specific topics. Threads allow users to send messages within a message they've sent, creating a threaded conversation. Another powerful feature of Slack is huddles. Huddles enable instant online meetings with just a push of a button. During a huddle, users can collaborate by writing on a shared computer screen using their mouse, making it easier to work together. Lastly, Slack offers a variety of plugins. One notable plugin is the Github plugin, which allows users to automatically post messages to Slack using tags, such as code review requests.

6. Notion



Fig. 15. Notion

Notion is a web-based SaaS application that functions as a wiki. One of its advantages is the ability to create articles in the form of MD files and see real-time changes. Recently, with various updates, it has become a valuable tool for managing meeting minutes and projects.

7. Overleaf



Fig. 16. Overleaf

Overleaf is an online platform that facilitates collaborative writing and editing of LaTeX documents. It offers a user-friendly interface for creating scientific and technical documents such as research papers, reports, and thesis papers. With Overleaf, multiple team members can work on the same document simultaneously, enabling seamless collaboration and change tracking. Additionally, it provides built-in features for managing references, equations, tables, and figures, making it a preferred choice for researchers and academics. This document was written using Overleaf's IEEE specification.

8. Postman



Fig. 17. Postman

Postman is a SaaS (Software as a Service) tool that facilitates faster and easier API development. It is a platform that enables you to test, document, and share your developed APIs. It is particularly useful for testing RESTful APIs. When it comes to testing backend functionalities like authorization, headers, and caching, it can be challenging. However, Postman automates these tasks and allows for easy customization, making testing a breeze. Additionally, Postman allows you to share tested URLs, simplifying collaboration between front-end and back-end developers.

9. ChatGPT



Fig. 18. ChatGPT

ChatGPT is an AI-powered tool that enables real-time conversations with an AI. For GPT-3.5, it is trained on data up to 2021, while GPT-4 is trained on more recent data. ChatGPT has revolutionized generative AI,

offering enhanced capabilities for tasks such as report generation, article summarization, problem-solving, and even coding. In our team, we will utilize ChatGPT to simulate plant conditions, allowing the AI to engage with users as if it were a plant.

10. DBML & DBDocs



Fig. 19. DBML& DBDocs

DBML is a Domain Specific Language (DSL) for defining database structures. This text-based language allows you to explicitly describe tables, columns, indexes, foreign key relationships, and more. DBML is designed to create visual and intuitive database structures. It allows developers to define database schemas without having to write complex SQL syntax. dbdocs is a web-based tool that generates documentation based on database schemas written in DBML. dbdocs allows you to create beautifully documented websites directly from DBML files. This tool can help you visualize your database design, share it with team members, and help non-technical stakeholders understand the structure. When a DBML file is uploaded through the dbdocs.io service, then dbdocs processes it to provide detailed documentation of your database schema, including tables, relationships, indexes, and more. This document includes search capabilities, relationship diagrams, table details, and more, making it easy to navigate and understand your database structure.

11. Android Studio



Fig. 20. Android Studio

Android Studio is an integrated development environment (IDE) for Android, developed by Google. Similar to Xcode, Android Studio is essential software as it handles the final compilation of an Android application.

12. Flipper

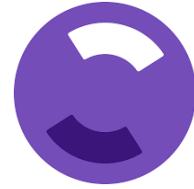


Fig. 21. Flipper

In React Native, Flipper is a platform used for debugging and development purposes. It is an open-source tool developed by Facebook and offers various features for debugging and performance profiling of mobile applications. Flipper itself provides three types of debugging: checking the DOM tree structure, monitoring network communication, and inspecting image information. One of the powerful aspects of Flipper is its plugins. It offers a wide range of plugins that allow you to debug libraries essential for developing BloomMate, such as react-navigation, recoil, and react-query. You might wonder if React Native already provides its own Chrome debugging. However, some libraries, especially those related to react-native-reanimated, may not work with Chrome debugging enabled due to compatibility issues with UI Threads. In such cases, Flipper can still be used as it debugs at the JavaScript level.

C. Team's Development Environment

TABLE I
TEAM'S DEVELOPMENT ENVIRONMENT

Name	Environment
Kim Dong Hyun	MacOS Monterey 12.5 react-native 0.72.4 python 3.9.6
Shin Hyun Ah	Windows 11.22.2 react-native 0.72.4
Yoon Yong Sung	Windows 11.22.2 react-native 0.72.4
An Soonho	Windows 11.22.2 python 3.10.7

D. Cost Estimation

To create BloomMate, we take advantage of a diverse range of cost-effective programs that are available for free. This helps us minimize the expenses related to the development process. However, it's important to mention that using the ChatGPT API does come with a cost. Nevertheless, there is a provision that allows you to enjoy a free quota of approximately \$18, which means you can ask around 300,000 questions without any charges. Therefore, you can be confident that the expenses associated with running the

application will be relatively low.

E. Task Distribution

TABLE II
TASK DISTRIBUTION

Tasks	Name	Descriptions
Frontend Developer	Shin Hyunah, Yoon Yongsung, Kim Donghyun	Front-end developers utilize languages like React Native and TypeScript to create applications. They are responsible for designing the interfaces that users interact with, such as tapping buttons and swiping through screens. Their primary objective is to create a user experience that is both accessible and engaging, while adhering to the specified design. Additionally, front-end developers are responsible for transferring user-entered information to the backend developers. The reason for having three front-end developers is that two of them write code for each screen, while the third developer reviews and optimizes the code for the screens that users see. This organizational structure requires effective teamwork, clear role allocation, excellent communication skills, and collaborative synergy.
Backend Developer	An Soonho	Backend developers are responsible for designing the database and application architecture, as well as writing the APIs used by frontend developers. When working with APIs, backend developers need to be able to receive information from application users through the frontend and provide the correct return value to the API. They also need to design APIs that interact with the backend to leverage generative AI and machine learning features, and make them accessible to application users. This role requires a strong understanding of the central database and software structure, and ensuring that software development aligns with that structure.
UI-UX Designer	Yoon Yongsung	The UI-UX designer, using Figma, is responsible for determining how the application screens are presented to users. This role involves deciding which screens will be more engaging and comfortable for users to use. As a UI-UX designer, the goal is to create a design that keeps users engaged and encourages them to return to the application. Once the UI-UX decisions are finalized, they can be communicated to the front-end developers.

		It is important for the UI-UX decisions to not only look good, but also consider the needs of both the application user and the front-end developer who will be building the screens. The design should also allow the back-end developer to design the database. This role relies on effective collaboration with developers.
Product Designer	Shin Hyunah	A product designer is someone who designs products with a user-centered perspective. They must have the ability to put themselves in the shoes of the user, find the pain points within the service as it currently exists, explore what they would like to see added, and come up with ideas. They are responsible for creating the overall framework of the product or service. Product designers are also responsible for communicating with the rest of the team to find out how they can take their ideas further and finalize their direction. Product designers should focus on the user usability of the product or service, exploring features that users really need and modifying the design of the product.
AI Developer	Kim Donghyun	A machine learning software developer works with algorithms, data, and artificial intelligence. Their role involves researching, building, and designing artificial intelligence software specifically for machine learning purposes. They primarily focus on applying artificial intelligence systems to various applications. The responsibilities of this role include collecting, cleaning, and preprocessing data to extract meaningful value. They then use this data to train models and deploy them in software. Additionally, the machine learning software developer must appropriately implement machine learning algorithms into software functions, conduct experiments and tests of AI systems, and determine the most suitable models for the application's functions. They are also responsible for designing and developing machine learning systems, as well as performing statistical analysis.

V. SPECIFICATIONS

A. Landing Page

The Landing Page is the first screen that users see when they open the application. It prominently displays the BloomMate logo and a condensed representation of the application's identity. Additionally, it includes two buttons that allow users to navigate to the SignUp Page and the Login Page. It is important that the design of each button aligns with our established design system and clearly indicates their respective functions.

BloomMate



식집사를 위한 친구

로그인
회원가입

Fig. 22. Landing Page

B. SignUp Page

Users can begin using BloomMate via the SignUp Page. This is a crucial step as BloomMate offers personalized plant management and chat services. During the signup process, users are required to fill in six fields: name, ID, password, Tiiun product key, garden size, and address. Each field is presented on a separate page, and none can be skipped.

Fig. 23. Sign Up Page: Example of Next button enabled

An arrow icon in the top left corner of each page indicates "back." Clicking this icon will either return you to the Landing Page (if you're on the first step, entering your name), or to the previous step. A Call-To-Action (CTA) Button is located at the bottom of each page. This button only activates once all fields are correctly filled in. The button will read "Sign up" on the last step and "Continue" on all others. Upon pressing the activated Register button on the final step, a Register Post API request is sent to the server. If received successfully, the user is directed to the Login Page.

The constraints and peculiarities for each input item are as follows: (1) The name should be between 2 to 5 characters. Input is accepted directly from the user via the keyboard. (2) The username should contain 5 to 20 characters, allowing only

English letters and numbers. Input is accepted directly from the user via the keyboard. (3) The password should contain 8 to 20 characters, allowing only English letters and numbers. A password re-entry is required for confirmation. Input is accepted directly from the user via the keyboard. (4) The Tiiun product key should begin with 'tiun' and be exactly 8 characters long. Input is accepted directly from the user via the keyboard. (5) For the garden size, use the toggle button to select either Large, Medium, or Small. Selecting another size will deselect the current one. Each garden size has a plant limit indicated by the number of pots in the image - Large has 7, Medium has 5, and Small has 3 pots. (6) For the address, we use the address input API provided by Kakao. It only provides the street address and displays the smart cottage photo when the address is entered.

Fig. 24. Sign Up Page: Input Field Requirements

C. Login Page

Fig. 25. Login Page

The Login Page of BloomMate allows users to personalize their experiences and utilize core features. Login is essential because it enables the server to identify the user making the request. The login process requires a username and password, each on a separate page, and both must be filled in. The

Login button at the bottom is initially disabled, but it becomes enabled when the username and password are correctly entered without errors. If you press the enabled login button, a login POST API request is sent to the server. If successful, the server provides a token, stored via the Set-Cookie command in the header, which identifies the user. You are then directed to the Plant List Page. On the top left of this page, there's an arrow icon pointing left. Pressing this icon returns you to the Landing Page.

The constraints and peculiarities for each input item are as follows: (1) For the ID, you are allowed to enter a minimum of 5 and a maximum of 20 characters, using only English letters and numbers. User input is accepted directly through the TextInput via the keyboard. (2) For the password, you must enter between 8 and 20 characters, again using only English letters and numbers. User input is similarly accepted directly through the TextInput via the keyboard.

D. Plant List Page

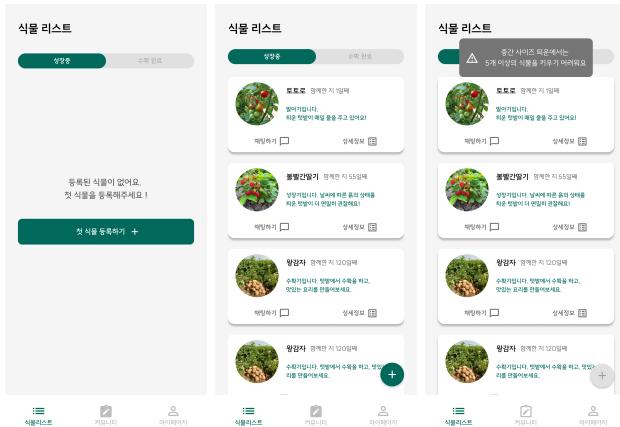


Fig. 26. Plant List Page: Growing Tab

The Plant List Page, which appears immediately after a user logs in, displays a list of plants currently growing in Tiiun on SmartCottage or those previously grown and successfully harvested. The plant list is retrieved from the server via a GET request. This page, found on the first tab, is one of three screens where the bottom tab bar is visible. The icon on the first tab should be in the primary color, with the other two icons in the disabled color. The page should be titled "Plant List" at the top. Below that, there are two top tabs: 'Growing' and 'Harvested'.

For the "Growing" Tab: (1) If no plants are currently growing, an 'Add Plant' button should be displayed in the center of the screen, accompanied by a text prompt to add a plant. Clicking the 'Add Plant' button will navigate to the 'Plant Add Page'. (2) If plants are growing, the screen should display a scrollable list of plants, each presented as a card. These cards contain six elements: a plant picture, a nickname, the number of days spent with the plant, a growth stage guide, a 'Chat with Plant' button, and a 'Plant Details'

button. Clicking on a plant card navigates to the 'Plant Detail Page'. The 'Chat with Plant' button opens the chat feature, while the 'Plant Details' button leads to the 'Plant Detail Page'. A floating button at the bottom of the scroll directs to the 'Plant Add Page' when clicked. (3) If the maximum number of plants has been reached, a message should appear stating that no more plants can be added. This message may be overlooked if the list of plants is full, requiring the user to scroll to see the bottom. If the 'Add Plant' floating button is clicked in this case, the button should change to a disabled color and a toast message should display, indicating that adding more plants is not permitted.



Fig. 27. Plant List Page: Completed Tab

For the "Completed" Tab: (1) If you don't currently have any harvested plants, write a text in the center of the screen stating that you don't have any harvested plants and encouraging you to grow your plants to harvest them. (2) If you do have any harvested plants, the body of the screen should be a scrolling page. Within the scrolling screen, show a list of harvested plants in card format, one by one. Nothing happens when the plant card button is pressed. Inside the plant card, there are three pieces of information: a picture of the plant, the plant's nickname, and the date it was grown.

E. Community Page

The Community Page is a resource for BloomMate users. It provides answers to questions, solutions to problems, and necessary information. The aim is to reduce user bounce rates and enhance farming experiences with accurate information from smartCottage. Plant list items are fetched from the server using GET requests. The bottom tab bar, one of three screens displayed, is located on the first tab. Icons on the second tab should be in the primary color, while the remaining two icons should be in the disabled color. The Community Page should feature the title 'Plant Buttler Community' at the top. Directly underneath, there should be two main tabs: 'Questions and Answers' and 'Expert Articles'.

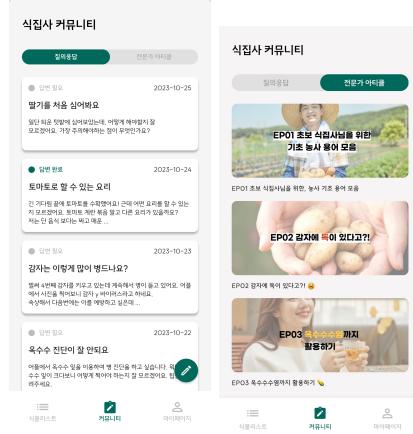


Fig. 28. Community Page

For the "Questions and Answers" Tab: The main part of the Q&A screen is a scrolling page, with questions displayed in a card format, sorted by the most recently asked. Tapping on a question card directs you to the Question Detail Page. Each question card includes four details: answer completeness, question date, question title, and question body. The question body has an 80-character limit; if it exceeds this, it's truncated with ellipses. At the bottom of the scroll screen, there's a floating button. Pressing this button navigates to the Add Question page.

For the "Expert Articles" Tab: The article screen should be a scrollable page where articles are displayed in a card format. Each card contains two pieces of information: a thumbnail image and a title. Tapping on an article will allow you to view it in WebView.

F. Question Add Page

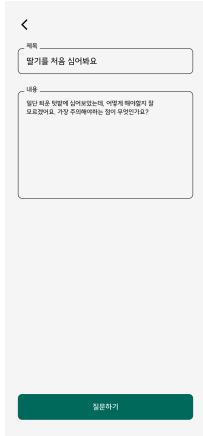


Fig. 29. Question Add Page

The Question Add Page is where users can post their inquiries. Users can ask about their BloomMate experience, report issues, or pose questions about their plants. Each question includes a title and content, both provided by the user

as TextInput. However, the content is entered in a multiline TextInput. Each TextInput features an appropriate placeholder. The Ask Question button is located at the bottom. Initially disabled, it becomes enabled when both title and content contain at least one character. An arrow icon on the top left of the page serves as a back button, leading back to the Community Page.

G. Question Detail Page



Fig. 30. Question Detail Page

The Question Detail Page is where you can see the details of your question. You'll find professional answers from the team behind BloomMate. First, the question is formatted like a question card in the question list. If the answer is complete, it will be shown along with the date of the answer. If the answer is long, it might be a scrolling screen. In the top left corner of the page, you'll see an arrow icon that indicates left. Pressing the arrow icon will take you back to the Community Page.

H. MyPage

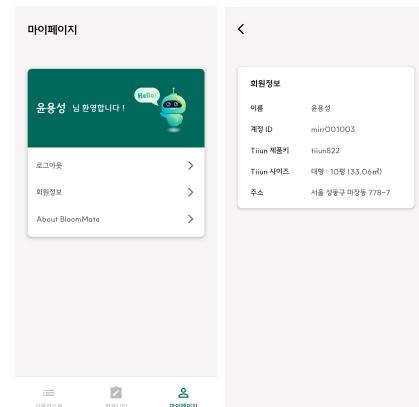


Fig. 31. MyPage

MyPage, a feature of BloomMate, provides information tailored to the user. It is found on the third tab of the bottom

bar, which is visible on three screens. The icon on the second tab should be in the primary color, while the other two icons should be in the disabled color. At the top, MyPage should be titled 'MyPage', followed by a welcome message in the form '*Welcome userName*'. MyPage offers three functions: Logout, View Membership, and About BloomMate.

The Logout function, when clicked, deletes the user's cookie information and redirects the user to the landing page.

The View Membership function, upon selection, navigates the user to a new screen where they can review their membership details. This screen displays five types of information: name, ID, activated product key, garden size, and address.

Lastly, the About BloomMate function directs the user to a Notion webview that provides an introduction to BloomMate.

I. Plant Add Page

BloomMate is an app designed to assist you in growing plants in Tiiun. To use it, you need to register your plants on the Plant Add Page. The plant addition process requires input in four fields: photo, variety, nickname, and planting date. Each field is located on a separate page, with the photo field split across two additional pages. All fields are mandatory and cannot be skipped. All pages feature an arrow icon in the top left corner, indicating a return function. Clicking this icon will either take you back to the Plant List Page if you're at the first step, or to the previous step. Each page also includes a Call-To-Action (CTA) button at the bottom. This button will only be activated once all fields on the page are filled out correctly. The button text reads "Add Plant" on the final step and "Continue" on all other pages. On the final step, pressing the activated Register button will trigger a server request to the Add Plant Post API. Upon successful receipt, you will be redirected to the Login Page.

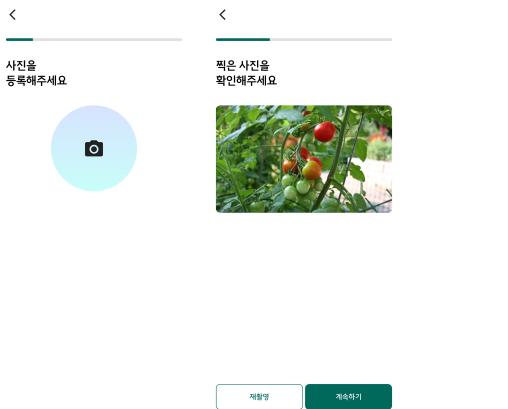


Fig. 32. Plant Add Page: add Picture

Now let's explain each input: (1) For photos, once the user takes a photo, it is instantly registered with the photo server using a POST request. The URL received from the photo server is then used in the final POST request. (2) Regarding plant varieties, users can choose one of the four options provided in the bottom-sheet via a toggle button. The options are strawberry, tomato, potato, and corn. These varieties are used in the AI model. They were chosen for their suitability in home gardens due to their reasonable growth heights. The toggle button allows you to switch between varieties. If another variety is selected while one is already active, the first one will be deselected. (3) For nicknames, the user should enter a minimum of 1 character and a maximum of 5 characters. TextInput allows direct input from the user through the keyboard. (4) To choose the planting date, users can select a date from the calendar provided in the bottom-sheet. Dates later than the present are not selectable. The selected and pressed states should be clearly marked.

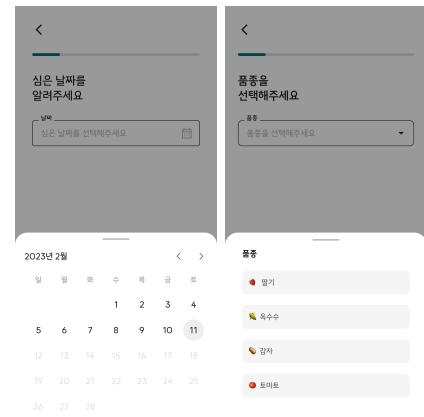


Fig. 33. Plant Add Page: Fields

J. Plant Detail Page



Fig. 34. Plant Detail Page

Users want to access information about the plants they are growing. To fulfill this need, they should be able to view plant details on the Plant Detail Page. All plant details are obtained from the server through a GET request. Plant information can be categorized into two types: details and growth information. Details remain the same for each plant variety. The Plant Detail Page displays eight types of details: variety, temperature, humidity, light level, flowering time, watering frequency, difficulty, and precautions. The growth information indicates the current stage of the plant based on the planting date: germination, growing, or harvesting. We have included a graph to provide a quick overview of the growth information. At the bottom of the page, there are buttons for different scenarios: a 'Harvest' button during the harvest season and a 'Diagnose' button for other situations. Clicking the Harvest button triggers a PATCH request to the server and redirects you to the Plant List Page. On the Plant List Page, the plant is no longer considered a growing plant as it has been harvested. Alternatively, you can click the Diagnose button, which will take you to a page where you can diagnose the plant. In the top-right corner, there is a button that navigates to the Plant Edit Page. The top-left corner of the page contains a left arrow icon. Clicking the arrow icon will return you to the Plant List Page.

K. Plant Edit Page



Fig. 35. Plant Edit Page

Users may want to update information about their plants, especially if they're growing them and need to take pictures of them frequently as they grow. The Plant Edit Page is designed for this purpose. There are two things that can be updated: a nickname and a photo. The nickname is accepted as a TextInput, and if it is empty, an error will be thrown. The photo can be updated by taking a picture. At the bottom is the Update button, which is activated if at least one item exists that is different from the existing one and there are no errors. If you press an active update button, it will send a PATCH request to the server, and if it is successful, you will be taken to the Plant Detail Page. In the top left corner of the page,

you will see an arrow icon indicating left. Pressing the arrow icon will take you back to the Plant Detail Page.

L. Plant Diagnosis

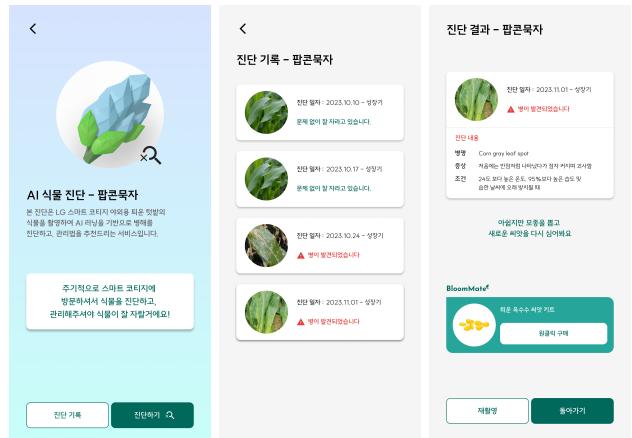


Fig. 36. Plant Diagnosis

One of BloomMate's main features is AI plant diagnostics, which serves two purposes. Firstly, it helps users, especially those with limited plant-growing experience, determine if their plants have any diseases by diagnosing their health. The AI diagnosis simplifies the decision-making process. Secondly, by encouraging users to regularly diagnose their plants when visiting SmartCottage, it promotes a sense of care and nurtures a loving relationship between users and their plants.

The plant diagnosis process begins on the information page, where users can learn about the benefits of using AI to diagnose their plants. It also reminds them to periodically visit SmartCottage to monitor their plants. From the information page, users can navigate to the Make a Diagnosis page and the Diagnosis History page.

The Make a Diagnosis page functions as a camera app. Users take a picture of their plant using the app, and the diagnosis is performed within their phone using the TensorFlow Lite model. After the diagnosis is complete, users are directed to the diagnosis results page. The diagnosis result will indicate whether the plant is (1) sick or (2) healthy. For each diagnosis, the following information is provided: the photo taken, the date of diagnosis, and the plant's growth stage (germination/growing). If the plant is diagnosed as sick, users will receive guidance on the disease, symptoms, and conditions. Unfortunately, if the plant is diseased, it will need to be replanted. Additionally, the generative AI will automatically provide a link to purchase customized Tiuun seeds that are most suitable for the user's current plant. By tapping on the link, users can make the purchase, and a confirmation message will be displayed. At the bottom of the page, there are options to reshoot the picture or return to the diagnosis guide page.

The Diagnostic History page displays a scrolling screen where users can view their previous diagnosis records. Each

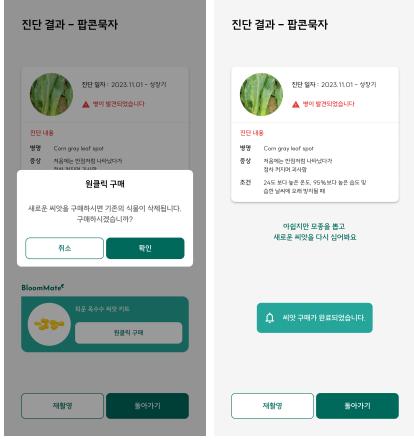


Fig. 37. Plant Diagnosis - One Click Purchase

diagnosis history is presented in a card format, showing the photo taken, the date of diagnosis, and the plant's growth stage. By tapping on a diagnosis history card, users can view the diagnosis log page in the same format as the diagnosis results page.

M. Chat With Plant

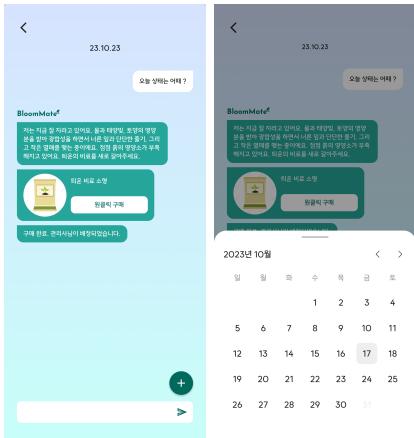


Fig. 38. Chat With Plant

The chat feature is the second core component of BloomMate. Each day, when a user requests to start a chat, the server synthesizes all available information and generates a script to send to the generative AI. This script essentially prompts the generative AI to assume the role of a plant. The server utilizes six types of information to create the script: plant details, plant nickname, plant planting date, today's weather, Tiuun soil information, and diagnostic information. Today's weather is obtained from the weather API based on the address provided during the signup process. As for Tiuun soil information, it is currently not accessible, so we will randomly generate a good or bad status. Regarding diagnostic information, we only retrieve the date of the last diagnosis, not the actual content. We assume that this date represents the last time the user visited SmartCottage. If it has been

more than a week, we will include a situation in the script where the plant expresses a desire to see the user.

Let's recap the chat flow with the plant:

1. The user clicks the "Start today's chat" button.
2. The server generates a script to obtain today's report from the generative AI and sends it to the client.
3. The client checks the report and, if the soil condition is bad, generates a link for fertilizer recommendations from the generative AI.
4. The user is then free to chat with the generative AI, which has been trained with scripts.

Additional specifications:

1. The above flow applies when there are no ongoing chats for the day. If there is already an active chat, users can continue without the need for the "Start" button.
2. Today's chat ends at midnight every day.
3. There is a floating button located at the bottom left of the screen. Tapping on the previous report will open a bottom-sheet with a calendar. By selecting a past date from the calendar, users can view the chat history for that specific day.

VI. ARCHITECTURE DESIGN & IMPLEMENTATION

A. Overall Architecture

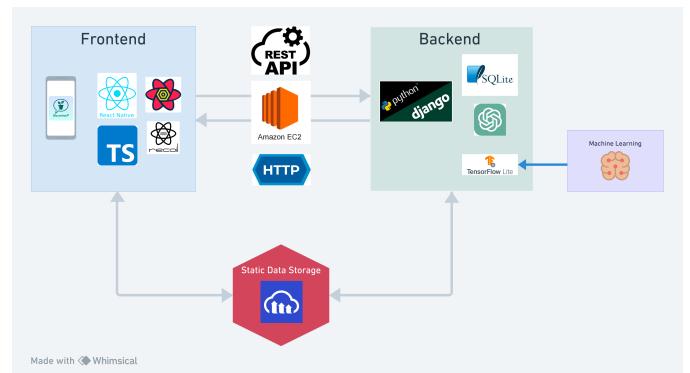


Fig. 39. Application Architecture

BloomMate consists of four main modules: Frontend, Backend, Static Data Storage, and Machine Learning. Let's take a closer look at each module.

The Frontend module refers to the part of the mobile application that is used by the end user. We prioritize meticulous code writing, good UI/UX design, and bug-free functionality in the frontend. BloomMate's frontend is built using react-native with Typescript. We utilize react-query for server status management and recoil for internal global status management. Additionally, we use react-native-calendar for the calendar feature and react-hook-form for input form management. With the BloomMate mobile application, users

can perform various functions such as adding plants, checking plant information, diagnosing plants, chatting with plants, accessing expert articles, and asking questions.

The Static Data Storage module is accessible from both the frontend and backend. This module serves as an externalization of certain database functions, specifically for storing and loading static data, images, videos, etc. BloomMate utilizes cloudinary for this purpose. Despite being a free service, cloudinary offers a generous 25GB of storage and simplifies the process of uploading and downloading images. Here's how it works: the frontend uploads a photo to cloudinary and receives a downloadable URL as a response. The frontend then passes this URL to the backend, which stores it in its own SQLite database. When the frontend receives a request using the URL, it fetches the URL from the database and retrieves the corresponding image from cloudinary. In BloomMate, the frontend uploads images in two cases: when adding a plant and when diagnosing a plant.

Please note that the third module, which will be discussed in more detail, covers the process of storing and loading static data in the backend's SQLite database.

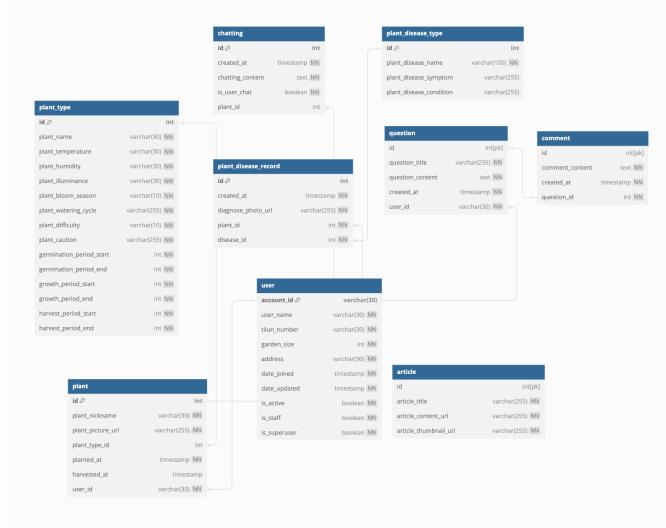


Fig. 40. Database Entity Relationship Diagram

The third module is the Backend module, which serves as the server handling user requests. What sets BloomMate apart is that the database is built within the Backend module itself. This is made possible by using Django, a technology that includes its own DB called SQLite. Looking at the SQLite part, depicted in the ERD above, it consists of 9 databases. Among these, the article table is an independent table not connected to any other table. Within Django, there is code that also interacts with chatGPT. We provide chatGPT with information about the plant and the user, assigning the plant's role. It then sends the phrases you write to chatGPT, retrieves

the answers, and sends them back to you. The TensorFlow Lite model created in the fourth module is also stored in the backend, along with the necessary code to execute it. After the user takes a photo, the photo URL is stored in the second module, cloudinary, and this URL is used for diagnosis.

The fourth module is the Machine Learning module. Its purpose is to create a model capable of image classification using TensorFlow's ResNet50. When designing BloomMate, our team focused on whether or not the plant's owner is present in the SmartCottage. For each scenario, the core functionality is as follows: if the owner is not in the SmartCottage, generative AI is used to interact with the plant; if the owner is in the SmartCottage, a picture of the plant is taken and AI is used to diagnose it. This Machine Learning Repository is intended for the AI used in the second scenario. It involves training the AI using a Custom Dataset and converting it to TensorFlow Lite so that it can be utilized in real projects.

B. Directory Organization

TABLE III
DIRECTORY ORGANIZATION - FRONTEND 1

Directory	File Name	Library
BloomMate/ BloomMate-FE	.bundle .github .vscode android ios node_modules patches src .env .eslintrc.js .gitignore .prettierc.js .watchmanconfig app.json babel.config.js bsconfig.json Gemfile Gemfile.lock index.js jest.config.js metro.config.js package.json README.md react-native.config.js tsconfig.json yarn.lock	react react-native patch-package postinstall-postinstall
BloomMate-FE/ ios	BloomMate.xcodeproj BloomMate.xcworkspace BloomMateTests .xcode.env link-assets-manifest.json Podfile Podfile.lock	<React/RCTBridge.h> <ReactRCTBundle- URLProvider.h> <React/RCTRootView.h> <React/RCTLinking- Manager.h> <React/RCTConvert.h>
BloomMate-FE/ patches	react-native-action- button+2.8.5.patch react-native- paper+5.10.4.patch	patch-packages

TABLE IV
DIRECTORY ORGANIZATION - FRONTEND 2

Directory	File Name	Library
BloomMate-FE /android	app build gradle build.gradle gradle.properties gradlew gradlew.bat link-assets-manifest.json local.properties settings.gradle	gradle kotlin
BloomMate-FE /src/screen	root.navigator.tsx about-bloom-mate.screen.tsx article-webview.screen.tsx community-qna-detail.screen.tsx community-qna-post.screen.tsx landing.screen.tsx login.screen.tsx plant-add.screen.tsx plant-cht.screen.tsx plant-detail.screen.tsx plant-detail-edit.screen.tsx plant-diagnosis-intro.screen.tsx plant-diagnosisis-list.screen.tsx plant-diagnosisis-log.screen.tsx plant-diagnosis-result.screen.tsx primary-primary-community.screen.tsx primary-my-page.screen.tsx primary-plant-list.screen.tsx primary-plant-harvesed-list.screen.tsx primary-plant-current-list.screen.tsx primary-community-article.screen.tsx primary-community-qna.screen.tsx signup.screen.tsx user-info.screen.tsx	@actbase/react-daum-postcode @gorhom/bottom-sheet @hookform/resolvers @mobily/stacks @react-native-community/hooks @react-navigation/bottom-tabs @react-navigation/material-top-tabs @react-navigation/native @react-navigation/stack axios color dayjs eslint-config-prettier eslint-plugin-import eslint-plugin-prettier eslint-plugin-react eslint-plugin-unused-imports lodash lottie-react-native moti react react-hook-form react-native react-native-action-button react-native-calendars react-native-error-boundary react-native-gesture-handler react-native-image-picker react-native-linear-gradient react-native-mmkv react-native-pager-view react-native-paper react-native-safe-area-context react-native-screens react-native-shimmer-placeholder react-native-tab-view react-native-toast-message react-native-webview react-query recoil rooks yup

TABLE V
DIRECTORY ORGANIZATION - FRONTEND 3

Directory	File Name	Library
BloomMate-FE /src/assets	img/img.asset.ts lottie/ about-BloomMate.json mutation-lottie.json plant-add-lottie.json plant-delete-lottie.json plant-harvest-lottie.json	lottie-react-native
BloomMate-FE /src/atoms	button.atom.tsx divider.atom.tsx icon.atom.tsx image.atom.tsx modal.atom.tsx point-linear-gradient.atom.tsx skeleton.atom.tsx suspender.atom.tsx text.atom.tsx text.util.ts text-input.atom.tsx toast.atom.tsx	react react-native @mobily/stacks color lodash react-native-paper react-native-vector-icons react-native-fast-image react-native-modal react-native-linear-gradient react-native-shimmer-placeholder react-native-toast-message
BloomMate-FE /src/providers	mutation-indicator query-client recoil ui	react react-native @mobily/stacks react-native-flipper react-query-native-devtools react-native-reanimated react-native-mmkv-flipper-plugin
BloomMate-FE /src/layouts	basic-layout CTA-section dialog loading-pae modal-header scroll-view	react react-native @mobily/stacks react-native-keyboard-aware-scroll-view @react-native-community/hooks
BloomMate-FE /src/hooks	get-account-info-query.hook.ts get-article-list-query.hook.ts get-plant-chatting-query.hook.ts get-plant-detail-query.hook.ts get-plant-diagnosis-record-detail-query.hook.ts get-plant-diagnosis-record-list-query.hook.ts get-plant-list-query.hook.ts get-question-detail-query.hook.ts upload-image-library.hook.ts upload-photo.hook.ts	react react-query react-native-image-picker axios

TABLE VI
DIRECTORY ORGANIZATION - BACKEND 1

Directory	File Name	Library
BloomMate	.git .vscode .env .gitignore db.sqlite3 manage.py requirements.txt test2.jpg	
BloomMate /BloomMate_backend	__init__.py asgi.py settings.py urls.py wsgi.py	os django.core.asgi pathlib datetime dotenv django.contrib django.urls
BloomMate /accounts	__init__.py admin.py apps.py models.py serializer.py tests.py urls.py views.py	django.contrib django.apps django.db django.utils rest_framework django.urls rest_framework_simplejwt
BloomMate /articles	__init__.py admin.py apps.py models.py serializer.py tests.py urls.py views.py	django.contrib django.apps django.db rest_framework django.urls rest_framework_simplejwt
BloomMate /chatting	__init__.py admin.py apps.py models.py serializer.py tests.py urls.py utils.py views.py	django.contrib django.apps django.db django.utils rest_framework django.urls rest_framework_simplejwt openai requests random django.utils geopy.geocoders django.conf django.shortcuts django.utils

TABLE VII
DIRECTORY ORGANIZATION - BACKEND 2

Directory	File Name	Library
BloomMate /community	__init__.py admin.py apps.py models.py serializer.py tests.py urls.py utils.py views.py pagination.py permissions.py	django.contrib django.apps django.db django.urls rest_framework django.https rest_framework_simplejwt
BloomMate /plants	__init__.py admin.py apps.py models.py serializer.py tests.py urls.py views.py model.tflite	django.contrib django.utils django.apps django.db rest_framework django.urls tensorflow numpy os keras

TABLE VIII
DIRECTORY ORGANIZATION - MACHINE LEARNING

Directory	File Name	Library
BloomMate /BloomMate-ML	.vscode dataset tensorflow-lite-models test-case .gitignore model.h5 model.tflite README.md resnet-tflite-test.ipynb resnet-all-datasets.ipynb training-result.png	tensorflow numpy keras pathlib matplotlib
BloomMate-ML /dataset	Corn Common rust Corn Gray leaf spot Corn Healthy Corn Northern leaf blight Potato Early blight Potato healthy Potato Late blight Strawberry healthy Strawberry leaf scorch Tomato Early blight Tomato healthy Tomato Late blight Tomato Target Spot Tomato Yellow Leaf Curl Virus	

C. Module1 - Frontend

1) Purpose

We chose to use TypeScript-based React Native for the development of BloomMate. Let's first discuss the reasons behind this decision. React Native is a cross-platform framework developed by Facebook that allows for simultaneous development of iOS and Android applications. One of our team members had prior experience with React Native at a previous company, and the rest of the team had JavaScript development experience, making it a logical choice for us. The decision to adopt TypeScript was motivated by the need for more stable development. JavaScript, being dynamically typed, has a higher probability of unexpected bugs at runtime. This becomes particularly problematic when working with react-navigation, an essential component for React Native, as it becomes unclear what parameters and navigation the screen inherits when TypeScript is not used. To mitigate these drawbacks, we developed BloomMate using React Native with TypeScript. In the frontend, we perform the following key functions: (1) Receiving input from users (2) Passing the input to the server (3) Fetching data from the server.

2) Functionality

All features available on BloomMate can be accessed on the frontend. To get started, simply sign up and log in by providing the necessary information (all user inputs are handled with error handling implemented). Once logged in, you can create your own nickname and attach a photo to add a plant. From there, you can freely interact with your plant and even diagnose its condition by examining its leaves if it appears to be in poor health. Additionally, you have the opportunity to ask questions to experts and read high-quality articles for further knowledge and guidance.

3) Location of source code:

<https://github.com/BloomMate/BloomMate-FE>

4) Class Component

Each folder under the src directory has a specific role assigned to it. You can determine the role based on the words between the dots in the module name. Let me first explain about the atom module. The term "atom" is short for "atomic" and is inspired by the atomic design system. Although it cannot create functionality on its own, it plays a crucial role in terms of UI as it represents the most fundamental UI components. Although not all of the atom components, I have tried to follow Google's Material Design system for most of them (although there have been compromises in practical issues).

- `button.atom.tsx` : Buttons are components that allow you to trigger specific actions when they are

pressed. The specific action is received through onPress. Buttons follow the material design guidelines faithfully and can be implemented in three modes (outlined, contained, text). The text inside the button can be obtained through children, and icons can also be displayed. When the button is pressed, a layer of color covers the entire button, indicating that the button is in a pressed state.

- `divider.atom.tsx` : A divider serves as a UI element that separates other elements. It is a thin line in a light gray color. If you want a thicker divider, you can set the "heavy" parameter to true.
- `icon.atom.tsx` : Icons are a valuable tool for conveying information through visual symbols. In react-native, you can use icons by utilizing the react-native-vector-icons library. This library supports various icon sets, and we have decided to use the material icon set. To use material icons, simply import the package from react-native-vector-icons without any additional steps. When using an icon on a typical screen, you only need to provide three values: name, size, and color. It is worth noting that modern icons are treated as fonts, allowing you to use the Icon component within Text. This simplifies styling for many publishing tasks.

D. Module2 - Backend

1) Purpose

The role of the backend is to build and manage back parts such as applications and databases. It builds server logic to return appropriate outputs according to user requests, and manages databases to store, retrieve, update, and manage input and output data. Backend designs the architecture of the data flow in the back of the application, helping to organize the structure of it. Backend also develops and manages APIs to allow the application to communicate with users. It maintains security to prevent unauthorized access to user information and to prevent direct hacking or errors that may occur. We chose Django for our backend database management for the following reasons Django is written in Python, so a lot of functionality is built in. This allows developers to focus more on the core functionality of the application and not waste time on the basics. Django also simplifies working with databases through its object-relational mapping (ORM), which allows developers to easily manipulate databases without having to write SQL themselves. In fact, its support for sqlite as a database allows code written in Django to store data in the correct format for sqlite. Django has an administrator interface that makes it easy for developers to see what's actually being stored and manage the site when they deploy the backend part.

2) Functionality

The BloomMate application allows users to perform various activities related to plants. BloomMate processes the information which user enters into the application and stores it in a database. When the user needs information, it returns useful data to the user on demand. Users can register, manage, diagnose and chat with their plants. All information generated when using these features is automatically stored in SQLITE. In the case of plant diagnostics, the user can use a function of AI running on the backend to determine the presence of a disease. User can also use Chat GPT-4 to feel the dialog with the plant based on the plant information registered by the user, weather information based on the address location information, etc. All these user inputs help to make chatting with the plant feel natural.

3) Location of source code:

<https://github.com/BloomMate/BloomMate-BE>

4) Class component

- accounts

- views.py : The View role in Accounts folder represents server-side management of the functionality available to users when they first connect to the BloomMate application.
- Sign up : Users can register if they have never used the BloomMate application before. The BloomMate application uses a REST API and also uses JWT tokens. Before using any feature of the application, the user must verify that he is an authorized user using a JWT token, but in the case of registration, this is not necessary because the user has not yet registered. Therefore, all users are free to use the function and once all the required information has been entered, the backend will return the information entered by the user, the message "Registration Success" and a status of 200.

- plants

- views.py : The role of View in Plants folder describes how users can register and see plant information, diagnose their plants, edit plant information (like the picture of the plant), and delete plants.
- Plant Disease Record : The diagnosis of plant diseases is done using the leaf parts of the plant. The diagnosis of the plant is done by the AI, which will be explained in detail in the AI section below. On the backend, application can determine whether a plant has a disease or not through a function that calls the AI. The backend first uses JWT tokens to determine if the user has properly registered with the application. Then it checks if the plant user wants to diagnose is the one he registered. Only plants registered by the user can be diagnosed. In the backend, the storage capacity is reduced

by storing the URL processed by the frontend in the database instead of directly storing the photos taken by the user for disease identification. Users can verify the diagnosis using AI by making a POST request with the plant ID and the URL of the photo. Users can also retrieve the diagnosis results for each plant they grow by making a GET request. The user can know the growth level and disease name of the plant through the request. When storing plant disease records in the database, the user can also check the name of the plant set by the user in connection with the plant database through the plant ID stored together.

- chatting

- utils.py : For chatting with plants, we have implemented several functions that are separate from the view. The information needed to chat with a plant is the plant information, the weather at the address you entered, the soil information provided by Tune, and the date of the last visit.
- generate_chatgpt_response : This function connects to Chat GPT-4 and receives chat responses. We chose to use Chat GPT version 4 because it has been trained with more recent information than its predecessor and has the ability to better understand and generate more advanced languages spoken by humans. In keeping with the concept of chatting with plants, the default system prompt in Chat GPT is set to the plant itself, based on the plant information provided by the user. The function brings the history of the user's conversations with the plant by date and enters it into the assistant prompt, allowing the conversation to continue naturally. It also pulls in weather information, soil conditions, and the name of the user who registered the plant. The user only needs to type a phrase like "How is the plant today?" but in the backend, the server enters other information, such as above, to get the desired answer. The backend returns the answer along with the soil condition, whether the user is chatting, and the plant ID.

VII. USE CASES

Primary Tab - Plant List

After logging in, users can easily check and manage their plant list through the Primary Plant List tab. This tab provides a comprehensive view of the plants at a glance.

1) Plant Add

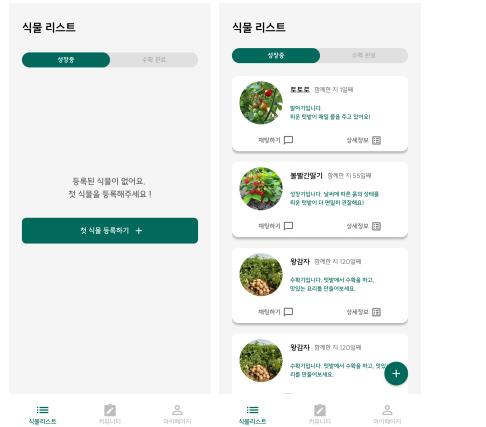


Fig. 41. Plant List

- Plant addition mainly occurs in two scenarios. The first scenario is when a user signs up and logs in for the first time. In this case, the user can view the initial state of the plant list screen, and add a plant using the 'First Plant Registration' button. The second scenario is when the user has already completed the membership registration and registered one or more plants. In this case, the user can add a new plant through the 'Add Plant' floating button located at the bottom right of the plant list screen.
- When adding a plant, the user must enter information such as photo, species, nickname, and planting date. This information helps users to manage their plants more effectively.

2) Harvesting & Harvested Tab

- At the top of the plant list screen, there are two tabs, 'Harvesting' and 'Harvested'. These tabs help users easily understand the current status of their plants. The 'Harvesting' tab displays plants that are in the 'Germination', 'Growth', or 'Harvest' periods, while the 'Harvested' tab displays the plants that the user has already harvested.

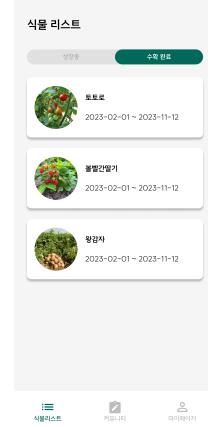


Fig. 42. Plant List - Harvested

3) Plant Details



Fig. 43. Plant Details

- Users can check the detailed information of each plant by clicking the 'Plant Detailed Information' button at the bottom of each plant component. In this screen, users can update the plant's photo and nickname, check basic information about the plant's species and its current growth status. In the growth information section, users can visually check which growth period the plant is currently in and how far that period has progressed through a Progress bar. Also, users can more easily manage the plant's status through the Primary color phrase that changes depending on the growth degree, and the 'AI Diagnosis' or 'Harvest' button at the bottom.

4) Plant Update



Fig. 44. Plant Update

- Users can update the plant's photo and nickname on the plant's detailed information screen. Users click the 'Update' text button located at the top right of the photo. They are then taken to a page where they can update the nickname and photo. If the user does not want to make a change, they can return to the plant's detailed information screen by clicking the back arrow icon at the top left.
- To update the nickname or photo, the user modifies the desired item and then clicks the 'Update' button at the bottom. The user then returns to the plant's detailed information screen, where they can see that the changes have been reflected.

5) AI Diagnosis

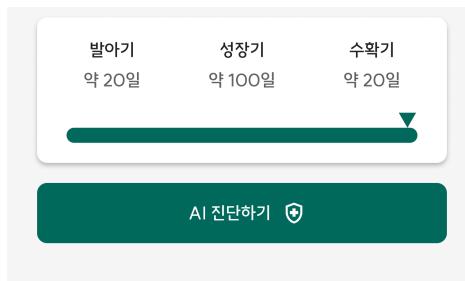


Fig. 45. AI Diagnosis Button

- The button at the bottom of the plant's detailed information screen is displayed as the 'AI Diagnosis' button when the plant is not in the harvest period.
- When the user clicks the diagnosis button, they can check the diagnosis record of the plant and move to the screen where they can diagnose it. Users can get information about the current status of the plant through chatting and growth information. Based on this information, users can visit the LG Smart Cottage and diagnose the status of the plant planted in the Tiun garden through AI diagnosis.

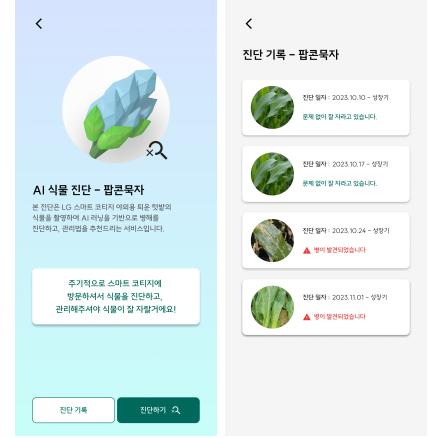


Fig. 46. Diagnose Intro & History

The diagnosis-related function screen includes the plant's nickname so the user knows they are only dealing with that plant.

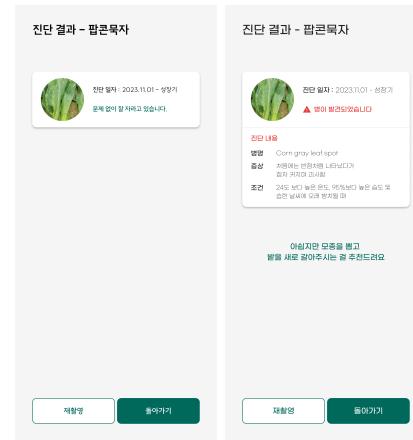


Fig. 47. Diagnose Result Example

- When the user clicks the diagnosis button, they can take a photo directly or select a photo from the gallery for diagnosis. The photo selected by the user is analyzed based on AI machine learning, and this is used to diagnose the status of the plant. Depending on the diagnosis results, the user can get information on whether the plant is healthy or diseased.
- If a disease is found in the plant as a result of the AI diagnosis, the user receives detailed information about the disease diagnosis. This information includes the type of disease, cause, and treatment methods. In addition, the user is recommended to replace the seedlings. For this case, a one-click purchase button is provided on the screen to order new seeds for the same species.
- If the user clicks this button, a dialog appears notifying the user that the existing plant will be

deleted. If the user clicks the 'Confirm' button in the dialog, new seeds of the same species are automatically ordered. After the order is completed, the user returns to the plant list screen and can confirm that the original plant has been deleted.

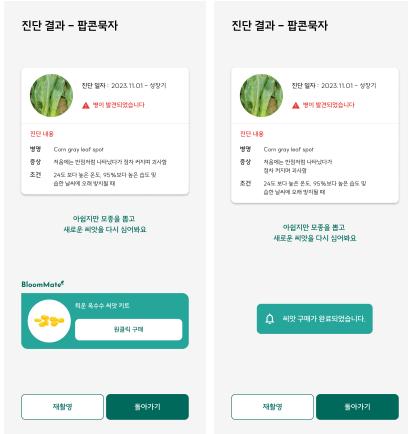


Fig. 48. Diagnose Result - Oneclick Purchase

6) Harvest

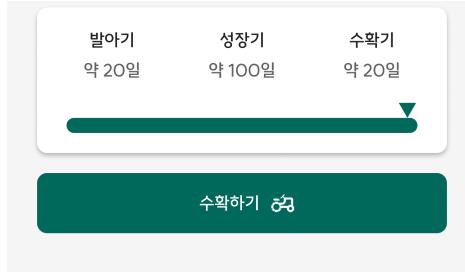


Fig. 49. Harvesting Button

- The button at the bottom of the plant's detailed information screen is displayed as the 'Harvest' button when the plant is in the harvest period. When the user clicks this button, the plant is harvested, and afterwards, the user can confirm that the plant has been deleted from the plant list screen. The harvested plant moves to the 'Harvested' tab in the top tab of the plant list screen.

7) Chatting with GPT

- Today's Chat in the plant list tab:

Users can move to the chat screen by clicking the 'Chat' button on the plant component. When the user clicks the 'Check Today's Report' button in this screen, BloomMate (Chat GPT-based chatbot) analyzes the current weather, temperature, humidity, etc. based on the detailed information of the plant as a plant and the user's location information, and tells the state of the plant. In addition, BloomMate provides information about the soil condition through Tiiun. If the soil condition is poor, the user can

purchase Tiiun-specific fertilizer through the 'One-click' button.

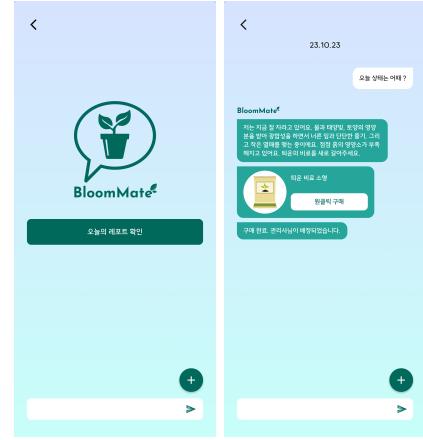


Fig. 50. Chat - Today's report

• Previous Chat :

Users can check previous chat records. When the user clicks the 'calendar-month' icon labeled 'Previous Report' in the floating button on the right bottom of the screen, a calendar modal is displayed. Through this calendar, users can check the chat of the previous date. If there is no chat record on the date selected by the user, they can see the message, "There is no conversation on the date you selected!"

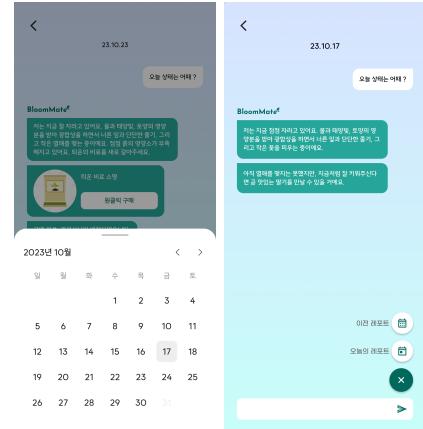


Fig. 51. Chat - Previous Chat

• Return to Today's Chat :

Users can return to today's chat window by clicking the 'calendar-today' icon labeled 'Today's Report' in the floating button.

In this screen, users can check today's report and today's conversations. Users can freely chat with the plant. For example, users can send messages like "How do you feel today", "I miss you", etc., and the plant responds appropriately to these.

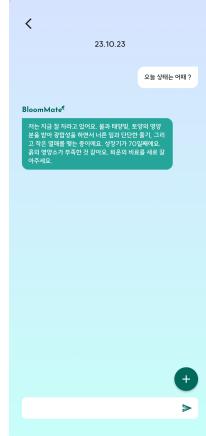


Fig. 52. Chat - Return to Today's Chat

9) Expert's Article



Fig. 54. Expert's Article

Primary Tab - Community

After login, users can obtain various information about plants through the Primary Community tab. This tab includes two main sections: 'Q & A' and 'Expert Article'.

8) Q & A

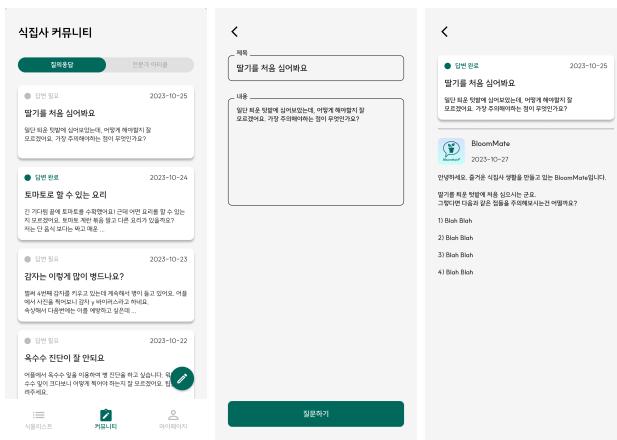


Fig. 53. Q & A

- In the Q & A tab, users can ask various questions. Users can write a new question through the floating button, which includes a pencil-shaped icon located at the bottom right.
- Based on the information obtained from chat or diagnosis process, users can ask general questions about the plant or methods of caring for Tiun. When the user clicks the floating button, they are directed to a text input form screen to write their question. BloomMate provides an answer for each question as soon as possible, and the answer status is displayed at the top left of each question component. Through this, users can easily check the status of their question.

- In the Expert Article tab, users can read expert articles in a sequential episode format. Users can read articles containing a variety of information, including detailed information about the breeds that can be grown in Tiun, cultivation and management methods, and ways to utilize the crops.

Primary Tab - Mypage

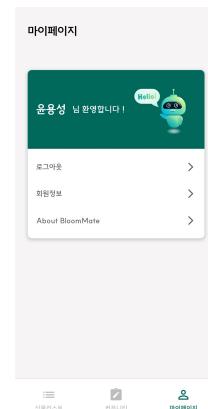


Fig. 55. My page

After login, users can log out and check personal information and information about BloomMate through the Mypage tab. This tab includes three main sections: 'Logout', 'User Info', and 'About BloomMate'.

10) Logout

Users who want to logout can click the 'Logout' button. When this button is clicked, the user's authentication information is removed from the system, and the user is automatically redirected to the landing page where they can login and register.

BloomMate[®]



식집사를 위한 친구

로그인

회원가입

Fig. 56. After Logout - Landing Page

11) User Info



Fig. 57. Info

Users can check the member information entered through the registration process. When the user clicks 'User Info', they are directed to a page where they can check the user's name, ID, Tiiun product key, Tiiun size, and address information. If the user wishes, they can return to the Mypage tab at any time by clicking the back arrow icon at the top left.

12) About BloomMate



Fig. 58. About BloomMate

Users can click 'About BloomMate' to check information about the team that developed BloomMate. On this page, users can refer to the background of BloomMate app development, demo videos, and related links about the development process. Also, users can check the technology stack used, core features, and information about the team members.

REFERENCES

- [1] J. Young-seon, H. So-young, and P. Y. Jin, "Effects of companion plants on mental health of elderly women living alone in the city," *Journal of the Korea Institute of Garden Design*, vol. 6, no. 3, pp. 218–227, 2020.
- [2] H. F. R. Institute. (2020, May) Changes in consumer behavior brought by covid-19. [Online]. Available: <http://www.hanaif.re.kr/boardDetail.do?hmpeSeqNo=34411>
- [3] C. Da-hyun. (2021, April) [the trend is indoor plants] part 1: Perfect for relieving pandemic blues and stabilizing the emotions of the elderly. [Online]. Available: <https://www.ajunews.com/view/20210401172057744>
- [4] SPC-magazine. (2023, June) [trend] the trend of communicating with 'leaves' in companion plants.
- [5] K. Yangjin. (2020, November) Urban farmers in seoul increased 14 times in 8 years, agricultural space expanded 7 times. [Online]. Available: <https://www.hani.co.kr/arti/area/capital/972100.html>
- [6] C. Dong-soo. (2021, November) Top housing preferences after retirement: 'detached houses' and 'townhouses' lead the way. [Online]. Available: <https://www.donga.com/news/Economy/article/all/20211108/110143322/1>
- [7] Y. Kwang-won. (2023, May) Seoul enacts 'pet plant industry promotion and support ordinance'. [Online]. Available: <https://www.mediacan.com/news/view/818763>
- [8] LG Electronics. (n.d.) Lg brand guidelines. [Online]. Available: <https://www.lge.co.kr/company/info/ci>
- [9] "TypeScript," <https://www.typescriptlang.org/>, 2023.
- [10] V. Authors. (2016) Yarn: A new package manager for javascript. Accessed on 2023-10-16. [Online]. Available: <https://engineering.fb.com/2016/10/11/web/yarn-a-new-package-manager-for-javascript/>
- [11] E. Elliott. (2023) Streamline code reviews with eslint prettier. Accessed on 2023-10-16. [Online]. Available: <https://medium.com/javascript-scene/streamline-code-reviews-with-eslint-prettier-6fb817a6b51d>
- [12] (2023) React native. Accessed on 2023-10-16. [Online]. Available: <https://reactnative.dev/>
- [13] Tanstack. (2023) tanstack/query. Accessed on 2023-10-16. [Online]. Available: <https://tanstack.com/query/v3/>
- [14] (2023) Recoil. Accessed on 2023-10-16. [Online]. Available: <https://recoiljs.org/>
- [15] (2023) Python quick reference. Accessed on 2023-10-16. [Online]. Available: <https://library.gabia.com/contents/9256/>
- [16] (2023) Django tutorial - django girls. Accessed on 2023-10-16. [Online]. Available: <https://tutorial.djangogirls.org/ko/django/>
- [17] AboutJoo. (2022) Django 50 questions 50 answers (5) - velog. Accessed on 2023-10-16. [Online]. Available: <https://velog.io/@aboutjoo/Django-50-%EB%AC%B8-50%EB%8B%B5-5>
- [18] A. W. Services. (2023) Amazon ec2 documentation. Accessed on 2023-10-16. [Online]. Available: <https://docs.aws.amazon.com>
- [19] WikiDocs. (2023) Python gui programming (pyqt) - wikidocs. Accessed on 2023-10-16. [Online]. Available: <https://wikidocs.net/195613>
- [20] Oracle. (2022) What is tensorflow? - oracle developer. Accessed on 2023-10-16. [Online]. Available: <https://developer.oracle.com/ko/learn/technical-articles/1481879245868-120-what-is-tensorflow>
- [21] G. D. Korea. (2014) This is material design - google developers korea blog. Accessed on 2023-10-16. [Online]. Available: <https://developers-krgoogleblog.com/2014/07/this-is-material-design.html>
- [22] hxyxneee. (2021) What is git? - velog. [Online]. Available: <https://velog.io/@hxyxneee/git%EC%9D%B4%EB%9E%80-%EB%AC%B4%EC%97%87%87>