Gardener's Best Friend

Chase Duclos
Hollow Rock, Tennessee
wescducl@ut.utm.edu

Lucy Gauldin
Martin, Tennessee
bildgaul@ut.utm.edu

Vrushank Mali
Martin, Tennessee
vrujmali@ut.utm.edu

Shakira Perry
Martin, Tennessee
shaaperr@ut.utm.edu

ABSTRACT

Gardener's Best Friend is an android only mobile app for people who want to maintain a journal tracking the well-being of their plants. Within this digital journal, gardeners can meticulously record plant-related information, including health status, watering schedules, and sunlight preferences. The primary goal of this app is to help users maintain their plants' schedule, strengthen their connection with their gardens, and enhance their gardening knowledge by capturing progress photos and documenting their plant journey in a personal journal. Gardener's Best Friend aims to enhance the interaction that gardeners have with their plants, providing users with tools to improve the overall wellness of their gardens.

The app offers further information about each of the user's plants, such as its known preferences regarding sunlight exposure retrieved from a comprehensive database. The database should give the user some insight into their plants' specific needs, ensuring that users can optimize their gardens' care routine. The digital journal even employs a notification system, reminding the user when they need to water their plants, based on the user provided watering schedule. In addition to these features, the app utilizes user location and weather conditions to determine if watering is necessary on a particular day. Overall, Gardener's Best Friend intends to cater to the average novice and expert gardeners alike, looking to keep records of their plants while offering guidance on how to take care of them.

1. INTRODUCTION

This comprehensive application is designed to offer a diverse array of features, catering to the needs of plant enthusiasts and individuals keen on maintaining a detailed record of their plant care journey. A prominent function within the app is the inclusion of a diary feature, empowering users to document and track various aspects of their plant's health and growth. To ensure seamless data management, the application is equipped with the capability to leverage either local storage or a database for the secure storage of plant-related information.

In order to enhance the visual aspect of the diary entries, the application seamlessly integrates with the device's camera or gallery, providing users with the flexibility to either capture real-time images of their plants or incorporate existing photos into their diary entries. This visual element not only adds a personal touch to the documentation process but also serves as a valuable visual reference for the user's plant care journey.

Among the innovative features currently in the planning phase is the potential integration of the Google API. This exciting addition allows users to capture an image of their plant, send it to the Google API, and receive insightful information, including the scientific name of the plant. This integration not only facilitates educational aspects of plant care but also adds a layer of sophistication to the user experience.

Further enriching the functionality of the application, discussions are underway regarding the incorporation of a calendar system. Users can opt for either the built-in calendar or an internal calendar within the app. This feature is intended to serve as a robust reminder system, prompting users to make diary entries or attend to essential plant care activities such as watering, ensuring a proactive approach to plant maintenance.

Recognizing the significance of environmental factors in plant care, the application explores the integration of location services. By harnessing location data, the app aims to connect with weather forecasting services, providing users with real-time and localized weather information. This feature empowers users to make informed decisions about their plant care routines based on the specific climatic conditions in their geographical area.

In essence, this application goes beyond a mere plant diary, evolving into a comprehensive tool that amalgamates technological advancements and user-friendly features to create a holistic platform for plant enthusiasts, fostering a deeper connection between users and their green companions.

2. BACKGROUND

The reader should have a general understanding of app development and familiarity with Kotlin and Java, so that the report can delve into advanced concepts of Android development. It can focus on intricate aspects of the Android platform, such as its architecture and components. The report may also touch upon the utilization of Android Studio, assuming the reader's comfort with its features and plugins. Advanced user interface (UI) design principles can be explored, building on the reader's foundational knowledge. Additionally, discussions on frameworks and libraries like Retrofit, Dagger, and Jetpack may be introduced without providing extensive explanations. By tailoring the content to this audience, the report aims to provide insights that go beyond the basics, offering valuable information for developers already well-versed in Android app development.

3. REQUIRED FUNCTIONS

Our application is designed to cater to the diverse and dynamic needs of plant enthusiasts, requiring a rich set of functionalities to enhance the user experience. Foremost among these features is the ability to effortlessly add and showcase photos, empowering users to visually document the journey and flourishing of their plants. The diary entry function is equally indispensable, serving as a digital journal for users to record detailed observations, growth milestones, and personalized insights, creating a narrative that evolves with each plant's unique story. To ensure optimal and mindful plant care, a fundamental requirement is the integration of a user-friendly watering schedule feature. This feature allows users to set and manage watering routines, ensuring that each plant receives the right amount of hydration at the right times.

Furthermore, our application necessitates the capability to seamlessly query an external API for plant information. This integration provides users with access to a wealth of data, including detailed care guidelines, botanical facts, and insights sourced from the API. This not only empowers users with knowledge but also enriches their understanding of the plants under their care. In essence, these required functionalities collectively form the backbone of our application, creating a holistic and intuitive platform for plant enthusiasts to cultivate a deep connection with their green companions, supported by a wealth of information and features that promote both expertise and joy in the art of plant care.

4. TECHNOLOGY

The technological foundation of our cutting-edge plant care application represents a carefully curated ensemble of tools, each contributing distinctive features to ensure a comprehensive, feature-rich, and technologically advanced user experience. At the core of our development efforts is Android Studio, an indispensable integrated development environment (IDE) renowned within the Android app development community. Beyond its user-friendly interface, Android Studio boasts an extensive suite of features that lays a robust foundation for efficient coding, meticulous debugging, and thorough testing, guaranteeing the delivery of a high-caliber application.

In the realm of programming languages, our unwavering commitment to excellence led us to embrace Kotlin, a language endorsed by Google for Android development. Beyond its syntactical elegance, Kotlin introduces a modern and concise coding style that significantly enhances code readability and reduces boilerplate code. The language's seamless interoperability with existing Java code not only facilitates a smooth integration process but also positions our application for long-term scalability and maintainability, ensuring the adaptability of our codebase to evolving needs.

For the crucial task of local data storage, our choice is SQLite, a venerable relational database management system. Acknowledged for its lightweight nature and efficiency in managing structured data, SQLite emerges as an optimal choice for our mobile application. Its reliability in performance ensures swift and secure data storage and retrieval, contributing to a seamless user experience without compromising on the integrity of stored data.

In our relentless pursuit of a dynamic and enriched user experience, the integration of the Perenual API represents a pivotal leap forward. This collaboration elevates our application beyond the confines of a conventional plant diary. Users can now access real-time, in-depth information about their plants, including intricate scientific details. The Perenual API transforms our application into a dynamic hub of plant-related knowledge, positioning it at the forefront of technological innovation in the domain of plant care applications. This dynamic integration not only caters to the informational needs of users but also infuses a layer of sophistication into their plant care journey.

To orchestrate seamless collaboration and foster an agile development process, GitHub takes center stage as our version control and collaboration platform. GitHub's robust suite of tools encompasses version control, issue tracking, and code review features, fostering a collaborative and transparent environment within our development team. This strategic choice not only underscores our commitment to delivering a product but also emphasizes our dedication to continuous improvement and innovation in the realm of plant care applications.

In essence, our extensive technological stack, comprising Android Studio, Kotlin, SQLite, the Perenual API, and GitHub, unfolds as a meticulously selected array of tools. This strategic amalgamation, enriched with robust features, not only empowers our development team but also ensures that our end-users embark on a technologically advanced, seamless, and gratifying plant care journey. In setting new standards for mobile applications catering to plant enthusiasts, we aspire to redefine the landscape with an application that transcends expectations, offering a harmonious blend of functionality and innovation.

5. REQUIREMENTS FOR APPLICATION USE

To fully enjoy the features of Gardener's Best Friend, the user's mobile device must meet a few requirements. These requirements exist to enhance the user's overall experience, ensuring that every function implemented works seamlessly.

Firstly, the user's mobile device must at least run on API level 28. This minimum API requirement guarantees compatibility with the technologies integrated into Gardener's Best Friend. If this requirement is met, users can enjoy all the features without any technical limitations. API level 28 and above provide the necessary framework for the app to perform optimally, delivering a smooth and responsive experience.

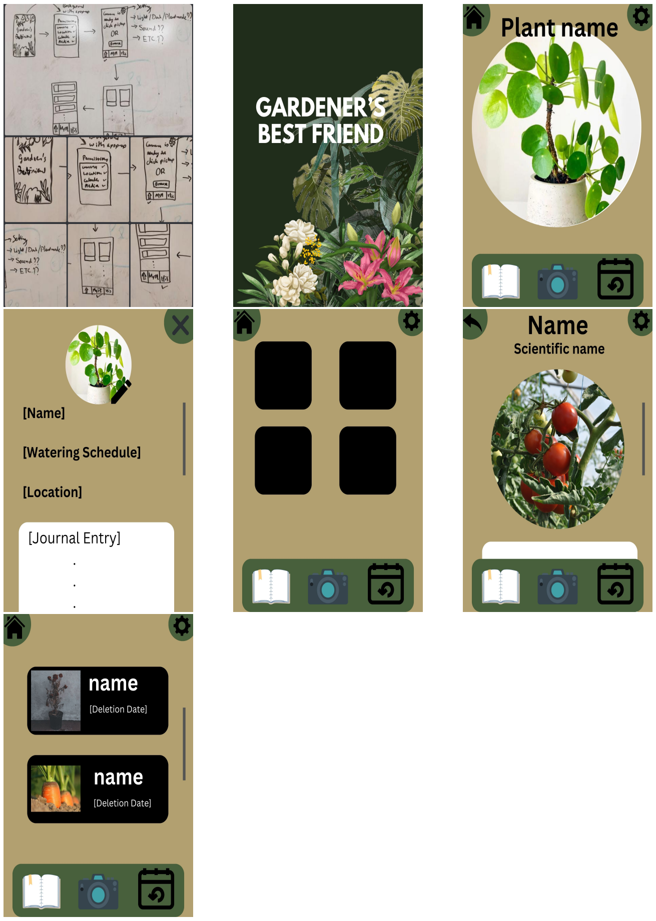
Secondly, the presence of a functional camera on the mobile device is a crucial requirement for utilizing the app's journaling feature. Gardener's Best Friend relies on the device's camera, allowing users to capture and log the progress of their plants.

Additionally, the mobile device must have a reliable method of storing data for the app's retrieval. This requirement guarantees that the pictures taken by the device's camera can be efficiently saved and accessed by Gardener's Best Friend. Adequate storage capacity is essential for maintaining a comprehensive record of the user's gardening journey.

Lastly, a device with internet access is required to use the Perenual API. The API relies on internet access to retrieve data to present the user with information about the plants they have added to their journal.

By meeting these specified requirements, users can unlock the full potential of Gardener's Best Friend, transforming their mobile devices into digital journals for growing healthy and thriving gardens.

6. MOCKUPS



7. IMPACT

We are hoping our application will have a positive influence on our environment and can help people take care of their gardens better. We hope users can have more flourishing gardens through the app keeping track of users plants and helping them to maintain a proper watering schedule with the least amount of plants been thrown out due to severe neglect. With the help of the Perenual API, we can provide additional information which can result in better planting and help users to provide necessary nutrition.

8. KEY TERMINOLOGY

The Perenual API not only introduces the users to additional insights about their plants, it also introduces them to a specialized agricultural vocabulary. To enhance the comprehension of the information provided by the API, it is highly beneficial for users to familiarize themselves with any potential new terms. The data retrieved by the API contains four distinct cycles that users might encounter.

The first term is perennial. Perennial defines plants that persist for more than two years, typically flowering and producing seeds repeatedly during their life cycle. These plants often withstand seasonal changes, regenerating and blooming year after year, contributing to a longer and more sustained presence in a garden or natural environment.

The second term is annual. Annual refers to plants that complete their entire life cycle within a single growing season. These plants germinate, grow, flower, produce seeds, and die off within a year. Unlike perennials, annuals are known for having a short life cycle. A clear understanding of annuals aids users in strategizing and maintaining a garden that showcases a wide range of changing blooms throughout the year.

Name	Primary Actor
1	Open Add Plant Page
2	Select Plant Image Using Camera
3	Select Plant Image Using Existing Photo
4	Add/Edit Weekly Reminders
5	Add Journal Entry Date
6	Add Journal Entry
7	Confirm New Journal Entry
8	View Home Screen
9	View Current Plants
10	View Specific Plant
11	Edit Plant/Add New Entry
12	View Plant History

Table 1. Plant Use Case IDs

The next term is biennial. Biennial refers to plants that take two years to complete their life cycle. In the first year, they grow leaves and stems. In the second year, they flower, produce seeds, and finish their life cycle before dying off. The seeds biennial plants produce may be the start of new plants, but there is no guarantee.

Lastly, the user may encounter the term biannual. Biannual refers to plants that complete their life cycle every two years. This term is less common than the previously discussed ones, and it signifies a unique growth pattern where the plant goes through all stages of its life cycle biennially. Biannual plants might germinate, grow, flower, produce seeds, and complete their life cycle every two years.

Understanding the distinctions among perennial, annual, biennial, and biannual plants is crucial for users harnessing the potential of the Perenual API. By familiarizing themselves with this specialized agricultural vocabulary, users are able to enhance their experience with the app.

9. AUDIENCE

The reader should have an understanding of Gardener's Best Friend's target audience so that the report can best understand the app and why it was designed. The app is aimed towards the average person who wish to take care of a garden of plants, but may not be able to properly track and care for each and every plant to the best of their ability. Gardener's Best Friend allows the audience to improve their plant-taking abilities by allowing the user to keep a diary of their plant by writing any actions the user performed that day or any inconsistencies that the user noticed. The app also provides the user information about each plant that allows each user to make adjustments to their usual plant-taking activities that can benefit each plant.

10. USE CASES

These are the use cases for the app. The table containing each ID can be found at Table 1.

Use Case ID: 1 Actor: User
Description: User wants to add a plant to the journal
Precondition: The user must be at the Home screen
Scenario: The user presses the "Add Plant" button, which is a leaf with a plus next to it
Result: The system opens the add plant page

Use Case ID: 2
Actor: User
Description: User wants to add an image of their plant using the camera
Preconditions:

- The user must have given the app permission to use the camera
- The user must be on the add plant or edit plant screens

Scenario:

1. The user presses the large image frame at the top of the screen below the "Confirm" button
2. The user presses the button labeled "Use Camera"
3. The user takes a picture of their plant using the camera

Result: The picture taken by the user is displayed on the image frame that the user pressed

Use Case ID: 3
Actor: User
Description: Use wants to add an image of their plant using a existing photo in their gallery
Preconditions:

- The user must have given the app permission to view their images
- The user must be on the add plant or edit plant screens

Scenario:

1. The user presses the large image frame at the top of the screen below the "Confirm" button
2. The user persses the button labeled "Use Gallery"
3. The user selects an image of their plant from the photo gallery

Result: The picture selected by the user is displayed on the image frame that the user pressed

Use Case ID: 4
Actor: User
Description: The user wants to add or edit the weekly reminders of their plant
Precondition: The user must be on the add plant or edit plant screens
Scenario:

1. The user presses the field labeled "Reminders (Optional)"
2. The user selects the days they want to get reminders for their plant

Result: The user's new reminders are selected

Use Case ID: 5
Actor: User
Description: The user wants to add a date to their plant's journal entry
Precondition: The user must be on the add plant or edit plant screens
Scenario:

1. The user selects the field above the Journal Entry field, which contains the required date format
2. The user enters the date of their journal entry in the specified date format

Result: The user's journal date is entered

Use Case ID: 6
Actor: User
Description: The user wants to enter a journal entry for their plant
Precondition: The user must be on the add plant or edit plant screens
Scenario:

1. The user selects the large field labeled "Write your diary entry..."
2. The user enters the journal entry for the plant

Result: The user's journal entry is entered

Use Case ID: 7
Actor: User
Description: The user wants to confirm their new plant or their new journal entry
Preconditions:

- The user must be on the add plant or edit plant screens
- The user must have provided an image and entered a journal entry date and a journal entry

Scenario: The user presses the button labeled "Confirm"
Result: The user's new plant is added or their new journal entry is entered and the user is at the Home screen

Use Case ID: 8
Actor: User
Description: The user wants to go to the home screen
Precondition: The user must be on the My Diary or History screens
Scenario: The user presses the left menu button at the bottom of the screen, which is a globe with leaves
Result: The user is at the home screen

Use Case ID: 9
Actor: User
Description: The user wants to view their current plants by accessing the My Diary screen
Precondition: The user must be on the Home or History screens
Scenario: The user presses the middle menu button at the bottom of the screen, which is a diary with a pencil
Result: The user is on the My Diary screen and all added plants are shown

Use Case ID: 10
Actor: User

Description: The user wants to view a specific plant's page
Preconditions:

- The user must be on the My Diary screen
- The user must have added a plant

Scenario: The user presses the plant entry containing the plant they want to view
Result: The user is viewing the selected plant's page

Use Case ID: 11
Actor: User
Description: The user wants to edit their plant's reminders or add a new diary entry
Precondition: The user must be on the page of the plant they want to edit or add a new diary entry for
Scenario: The user presses the pencil icon at the bottom right of the plant's image
Result: The user is on the edit plant screen

Use Case ID: 12
Actor: User
Description: The user wants to view a history of their deleted plants
Precondition: The use must be on the Home or My Diary screens
Scenario: The user presses the bottom right menu button at the bottom of the screen, which is an hourglass timer containing a substance that looks like the Earth
Result: The user is at the History screen

11. CHALLENGES

In the course of our project development, we encountered a myriad of challenges that required diligent problem-solving and creative solutions. One notable hurdle pertained to obtaining the necessary permissions for various functionalities, including camera usage, gallery access, and calendar integration. Navigating the intricate landscape of permissions proved to be a multifaceted task, demanding careful consideration of user privacy and security concerns.

Another noteworthy challenge we confronted revolved around the seamless implementation of reminders within our application. Ensuring that reminders occurred reliably and at the appropriate times presented a complex puzzle that demanded meticulous attention to detail and rigorous testing. This aspect of our development process underscored the importance of creating a user experience that not only met but exceeded expectations.

Additionally, our journey in the development realm brought us face to face with compatibility issues between SQLite and Java database. It became apparent that these two database systems posed a compatibility dilemma, requiring a thoughtful evaluation of our database architecture. Ultimately, We decided to cut Java database and use SQLite. The intricacies involved in resolving this issue highlighted the nuanced nature of database management and underscored the significance of choosing the right tools for the job.

In summary, our project's development path was marked by challenges that spanned permissions management, reminder functionality, and database compatibility. Each obstacle presented an opportunity for growth, learning, and ultimately, the enhancement of our application's overall quality and user experience.

12. RESULTS

To wrap this application up, the bullets below are what we have fully working.

- Journal
- Photo records
- Notification (Reminders for watering schedule)
- API for additional plant information
- Database

13. FUTURE WORK

In envisioning the future development of our plant care application, a compelling avenue for expansion lies in the incorporation of location-based services to enhance plant care recommendations. By integrating location services, we aim to seamlessly tie into weather forecasting applications, providing users with real-time and localized weather information pertinent to their geographical location. This innovative feature would empower users to make informed decisions about their plant care routines, aligning them with the specific climatic conditions influencing their plants.

The integration of location services could enable the application to dynamically adapt its recommendations based on factors such as temperature, humidity, and sunlight exposure unique to each user's region. For instance, users in colder climates might receive tailored suggestions during winter months, ensuring their plants receive adequate protection and care against frost. Conversely, users in warmer climates may receive guidance on managing hydration levels during hot summer days.

Furthermore, this feature could facilitate the automatic adjustment of watering schedules and other care routines, creating a personalized and responsive experience. By tapping into real-time weather data, the application could provide timely alerts or notifications, prompting users to take proactive measures in response to sudden weather changes.

Incorporating location-based weather services not only elevates the sophistication of our plant care application but also aligns with our commitment to delivering a holistic and user-centric experience. As we look toward the future, this expansion into dynamic, location-aware plant care features holds the potential to redefine the way users interact with and care for their plants, fostering a more intuitive and adaptive connection between technology and nature.

14. CONCLUSION

In conclusion, this application stands as a sophisticated and user-centric solution for plant enthusiasts seeking a comprehensive tool to enhance their plant care experience. The diverse range of features, from the intuitive diary function to the seamless integration of visual elements through camera and gallery access, demonstrates a commitment to providing users with a personalized and enriching journey in plant maintenance.

The incorporation of the API introduces a layer of intelligence, offering users valuable insights into the scientific details of their plants. This not only educates users but elevates the app to a more dynamic and informative platform.

The calendar system, leveraging the built-in calendar, reflects a thoughtful approach to user engagement. This feature ensures that users stay proactive in their plant care routines, with timely reminders for diary entries and essential activities like watering.

The forward-looking integration of location services adds a practical dimension, connecting users with real-time weather forecasts based on their geographical location. This not only underscores the app's commitment to informed decision-making but also reinforces its adaptability to the specific environmental conditions influencing plant health.

In essence, this application transcends the conventional boundaries of a plant diary, evolving into a multifaceted companion for plant enthusiasts. By amalgamating technological advancements, educational features, and user-friendly design, it not only meets but exceeds the expectations of users, fostering a deeper and more meaningful connection between individuals and their green companions.

15. REFERENCES

1. Freepik
2. Perenual API
3. Notification
4. API Call
5. Image Button
6. Database YouTube Tutorial
7. API Call YouTube Tutorial