

# Greenthumb - A Plant Care App

## Design Document



### About

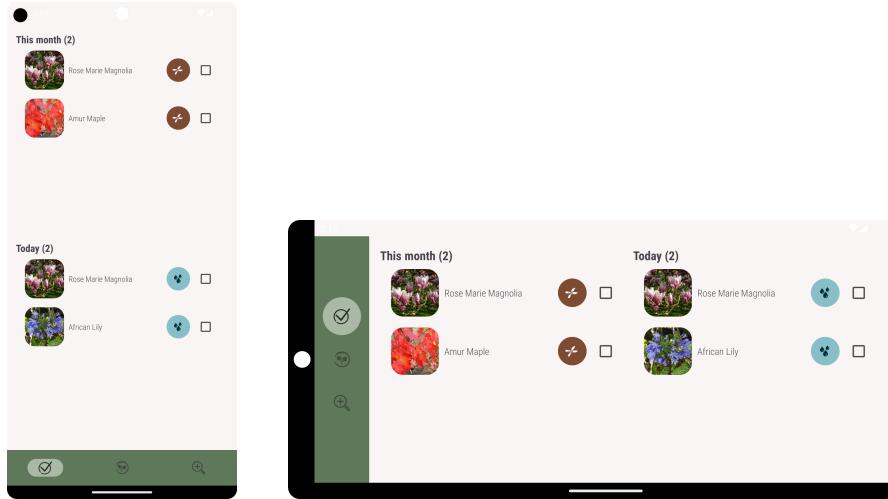
Greenthumb is a user-friendly app designed for beginners. Seamlessly navigate through a vast array of plants by simply searching for their names. Each plant comes with a concise description and essential care details, including ease of maintenance, sunlight and water requirements, and optimal pruning times.

Once you've found the perfect plant companion, save it for future reference. Take control of your plant care routine by scheduling task reminders for crucial activities like watering and pruning and receive notification when it's time.

### UI

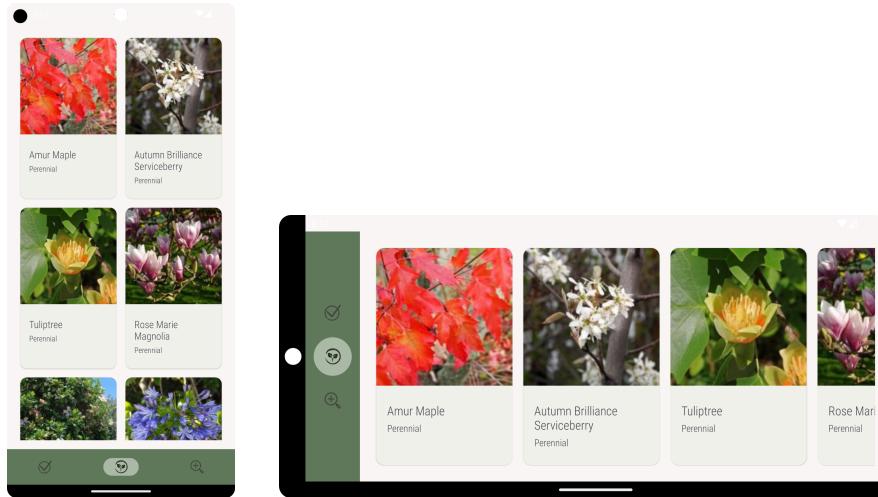
#### Home

Shows watering tasks for current day and pruning tasks for current month



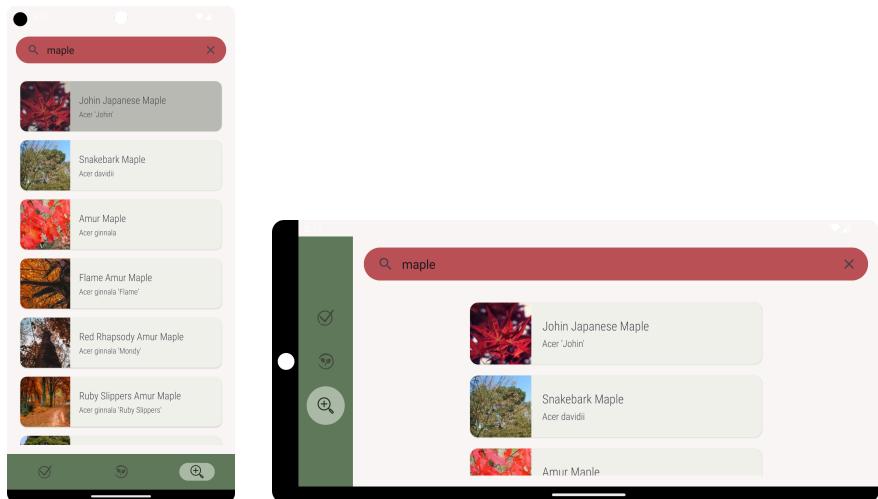
# My Plants

Shows saved plants



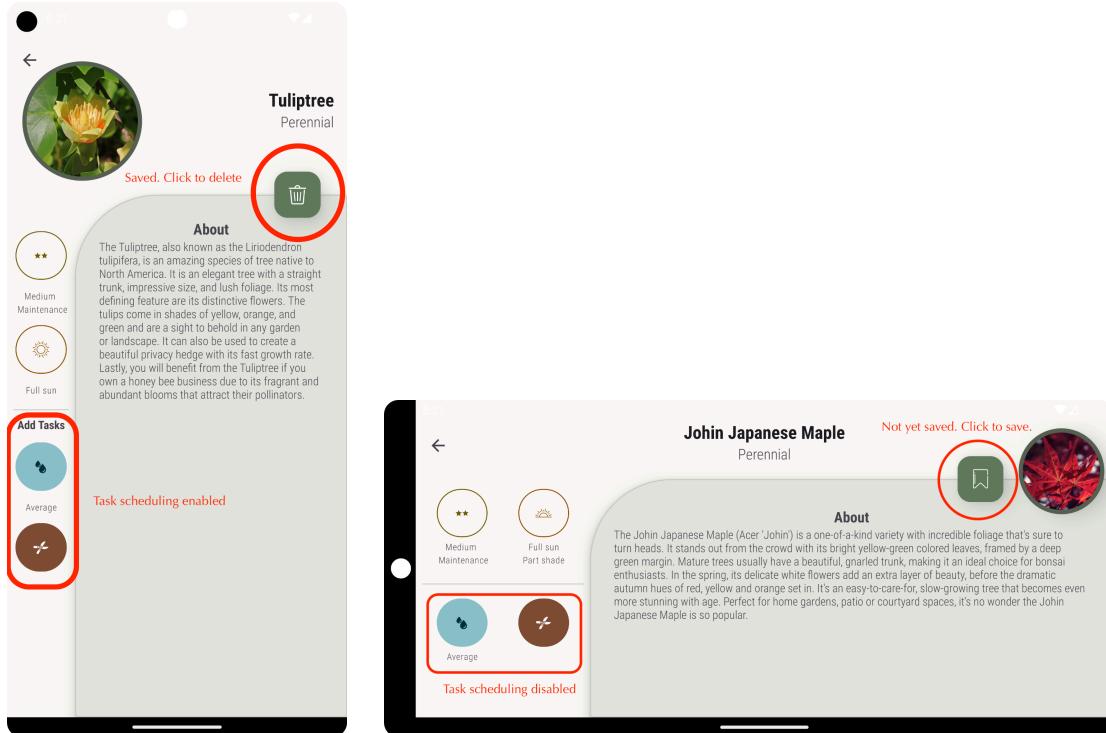
# Search

Shows search results



## Plant Details

Shows selected plant results. This screen is shared by Search (not saved) and My Plants(saved). The UI reflects the saved status of the plant appropriately.



## Usage

Add below API key in Greenthumb\gradle.properties

```
perennialAPIKey=sk-xRVf653aad45855ed2724
```

**Important:** The provided key supports the free tier which has access to only the first 3000 plant species. So not all plant names may return valid results.

Below are some search strings that are known to work

1. Maple
2. Fir
3. Birch
4. Cypress
5. Dogwood
6. Lily
7. Magnolia

Start by going to the search screen and saving plants to create task reminders.

# Functional Requirements

## Search

- Search by Name
  - Fetches a list from a remote end point.
  - Selecting an item opens the details screen.
    - FAB icon: 
    - Options:
      - Save Plant
      - Go Back
        - Up or back button returns to the search screen.
        - Previous search results remain displayed.

## My Plants

- Fetches plants from the Room DB.
- Selecting an item opens the details screen.
  - FAB icon: 
  - Options:
    - Delete Plant
    - Go Back
      - Up or back button returns to the My Plants screen.
      - If deleted, the list doesn't contain the deleted plant.

## Plant Details

- Save Plant
  - Saves the plant to the Room DB.
  - Updates the FAB icon to .
  - Enables UI to set/update tasks.
- Delete Plant
  - Shows a warning alert to delete the plant and related tasks.
    - Continue:
      - Deletes the plant from the DB.
      - Updates FAB icon to .
      - Deletes tasks associated with that plant (FK relationship).
      - Disables UI to set tasks.

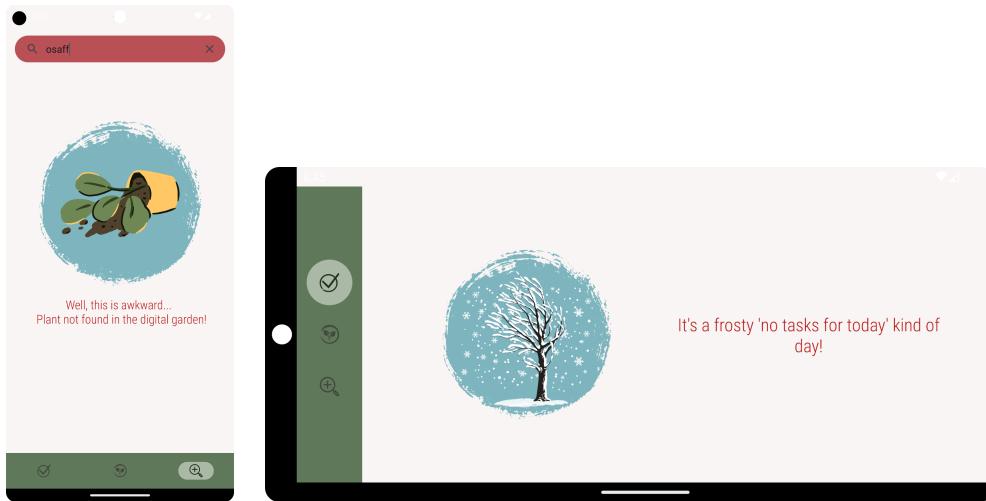
- Add Task
  - Toggle switch to set up a task.
    - Switch on:
      - Opens scheduling dialog to provide time and day/month, with expected schedule if provided by the remote API or current day/month and current time otherwise.
      - Creates and saves a task for that plant and task type in the DB.
      - Edit button visible to adjust the default schedule.
    - Switch off:
      - Deletes the task.
- Edit Task
  - UI shows existing schedule.
  - Opens scheduling dialog to provide time and day/month, with existing schedule
  - Updates task in the DB with the new schedule.
- Delete Task
  - Deletes the task from the DB.
  - Edit button becomes invisible.

## Home

- Fetches tasks from the DB.
- Mark Complete
  - Updates the task as completed.
  - Displays a strike-through in the UI.
  - Decrement the total task count in the header.

## Default and Error States

The app incorporates a robust error messaging system to facilitate user understanding and navigation in the event of a failure. Additionally, when the absence of tasks or plants is encountered, the application gracefully defaults to a status display.



## Background Jobs and Notifications

- Schedule periodic-work request that runs nightly at 2.00 AM
  - Clears previous day tasks or previous month task
  - Schedule tasks for today or this month
- If a new task is added after 2.00 AM, schedule one-time work request if it is later on the same day
- Check if task exists before sending notification
- Send notification when worker runs

# Rubric Key Areas

Success Criteria	App Features
<b>Android UI/UX</b>	
Build a navigable interface consisting of multiple screens of functionality and data.	Contains 4 fragments controlled by Navigation Controller
Construct interfaces that adhere to Android standards and display appropriately on screens of different size and resolution.	Uses ConstraintLayout and RecyclerView. Supports portrait and landscape modes.
Animate UI components to better utilize screen real estate and create engaging content.	Uses MotionLayout in Plant Details fragment 1. To show Add Tasks label when a plant is saved 2. In landscape only - To show task reminder section when watering/pruning reminder buttons are clicked
<b>Local and Network data</b>	
Connect to and consume data from a remote data source such as a RESTful API.	Uses Retrofit and Moshi to fetch plant data from Perenual via coroutines ( <a href="https://perenual.com/docs/api">https://perenual.com/docs/api</a> )
Load network resources, such as Bitmap Images, dynamically and on-demand.	Uses Glide to load plant images. Displays loading animation and shows placeholder image if loading fails.
Store data locally on the device for use between application sessions and/or offline use.	Uses Room to store saved plants and tasks. All data operations are performed using coroutines.
<b>Android system and hardware integration</b>	
Architect application functionality using MVVM.	Uses MVVM architecture for all fragments. Also uses Repository pattern for data access.
Implement logic to handle and respond to hardware and system events that impact the Android Lifecycle.	Requests for Internet and Notification runtime permissions Gracefully handles orientation changes
Utilize system hardware to provide the user with advanced functionality and features.	Uses Notification service

