# RootShare

**Nir Shitrit | 313526642 | nirvsgov@gmail.com**

### A. Application Goal Definition

**RootShare** is a social platform designed for urban gardeners and plant enthusiasts. The application's main purpose is to allow users to document their plant growth, share their collections with the community, and offer cuttings or plants for trade/giveaway. To enhance the user experience, the app enriches user posts with professional botanical data (such as watering schedules and sunlight requirements) fetched automatically from an external REST API.

### B. Application Functional Requirements Specification

The requirements are prioritized into "Must Have" (core features for the final submission) and "Should Have" (future improvements).

**Prioritized Functional Requirements (Must Have):**

1. **User Management & Authentication:**
    - **Registration:** Users must register using an email and password (implemented via **Firebase Authentication**).
    - **Login/Auto-Login:** Existing users can log in. The app will automatically identify the user upon the next launch without requiring re-authentication.
    - **Logout:** Users can securely log out of the application.
    - **User Profile:** Users can view and edit their profile details, including their display name and profile picture.
2. **Social Interaction & Content Creation:**
    - **Create Post:** Users can upload a new post containing an image (from the gallery or camera) and a free-text description.
    - **External API Integration:** When creating a post, the user inputs the plant type (e.g., "Monstera"). The app fetches data from an external botanical API (e.g., Perenual API) and automatically displays relevant care info (Watering, Sunlight, Difficulty) within the post.

- **Main Feed:** A global list displaying posts from all users. Each item shows the user's image, plant photo, description, and the fetched API data.

3. **Personal Content Management:**
    - **My Posts:** A dedicated screen where the user can view only the posts they have uploaded.
    - **Edit/Delete:** Users can edit their existing posts (update text or image) and delete posts they no longer wish to share.
4. **Data Persistence & Architecture (CRITICAL):**
    - **Architecture:** The app will be built using **MVVM** architecture, utilizing **ViewModel** and **LiveData**.
    - Navigation & UI: The application will be designed as a Single Activity Application. It will utilize Fragments for all screens and the Android Navigation Component (NavGraph) to manage navigation. Data passing between fragments (e.g., passing a plant object to the full-view screen) will be implemented using SafeArgs to ensure type safety.
    - **Remote Data Source:** All user data and posts will be stored remotely using **Firebase Firestore** (or Realtime DB) and **Firebase Storage** (for images).
    - **Local Data Source (Cache):** The app will implement a local database using **Room (SQLite)**.
        - **Requirement Compliance:** Firebase's native offline persistence features will be **disabled**.
        - **Implementation:** All posts and user objects fetched from the network will be saved into the Room database. The UI will observe the Room database (Single Source of Truth) to ensure data is available offline. Images will be cached using a standard library (e.g., Picasso/Glide) in conjunction with the Room database.

**Future Improvements (Should Have):**

- **Advanced Filtering:** Ability to filter the feed by plant category or care difficulty.
- **Direct Contact:** A feature allowing users to contact the post owner directly via email.

- **Map View:** Displaying posts on a map based on the user's location.