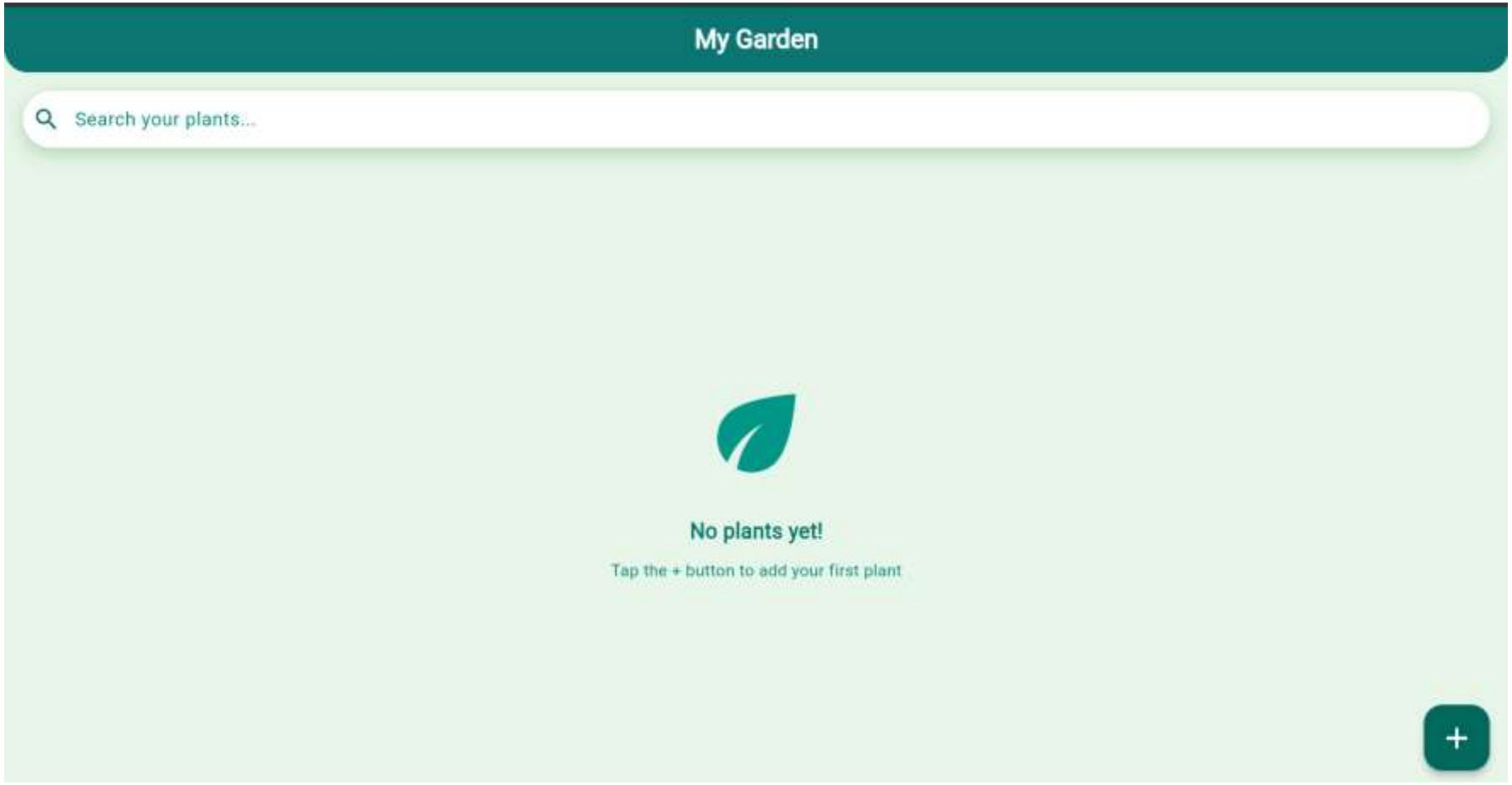
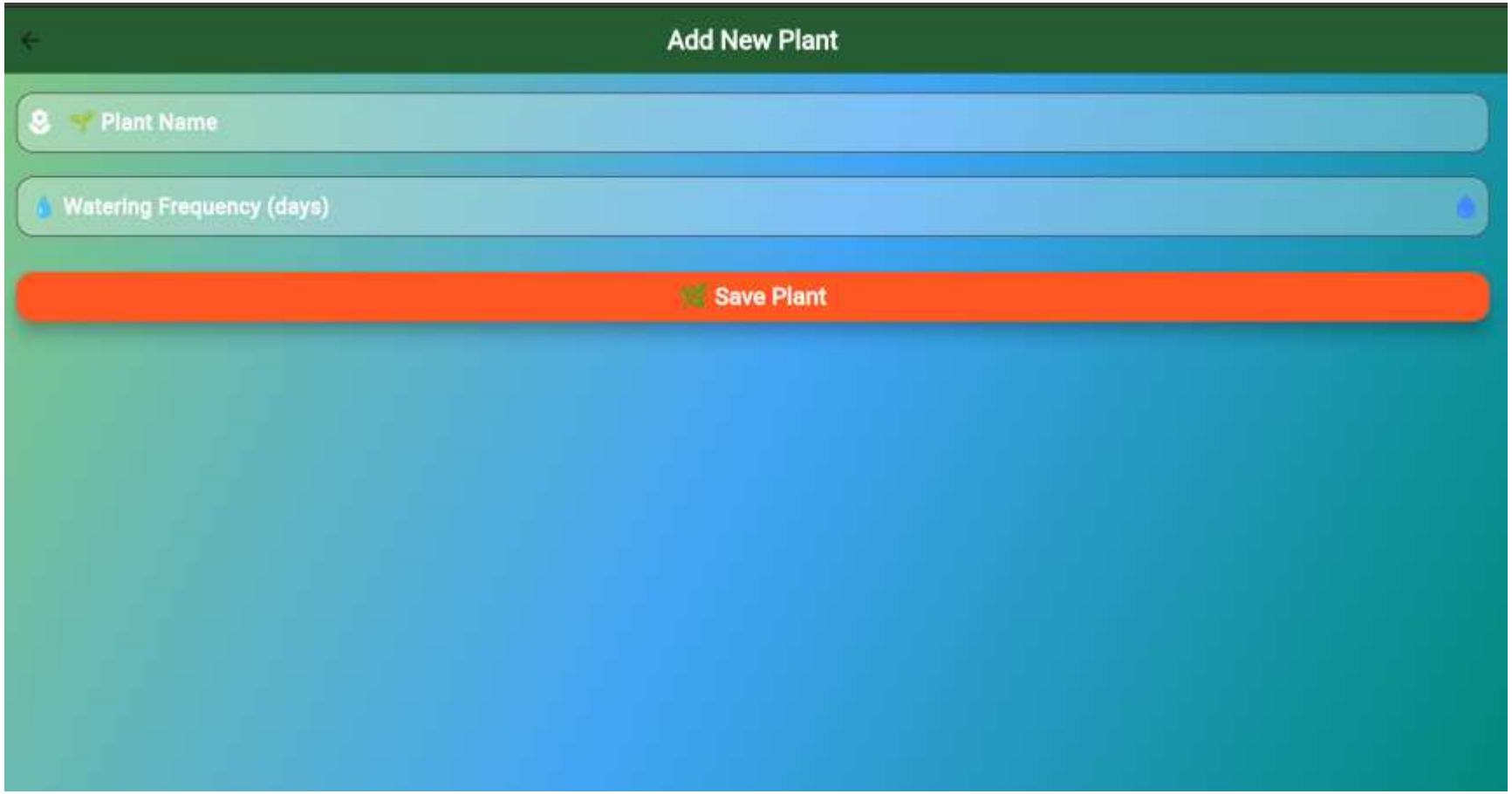
PlantCare App

PRESENTED BY BALAKUMARAN.B


My Garden
Search your plants...
No plants yet!
Tap the + button to add your first plant
+

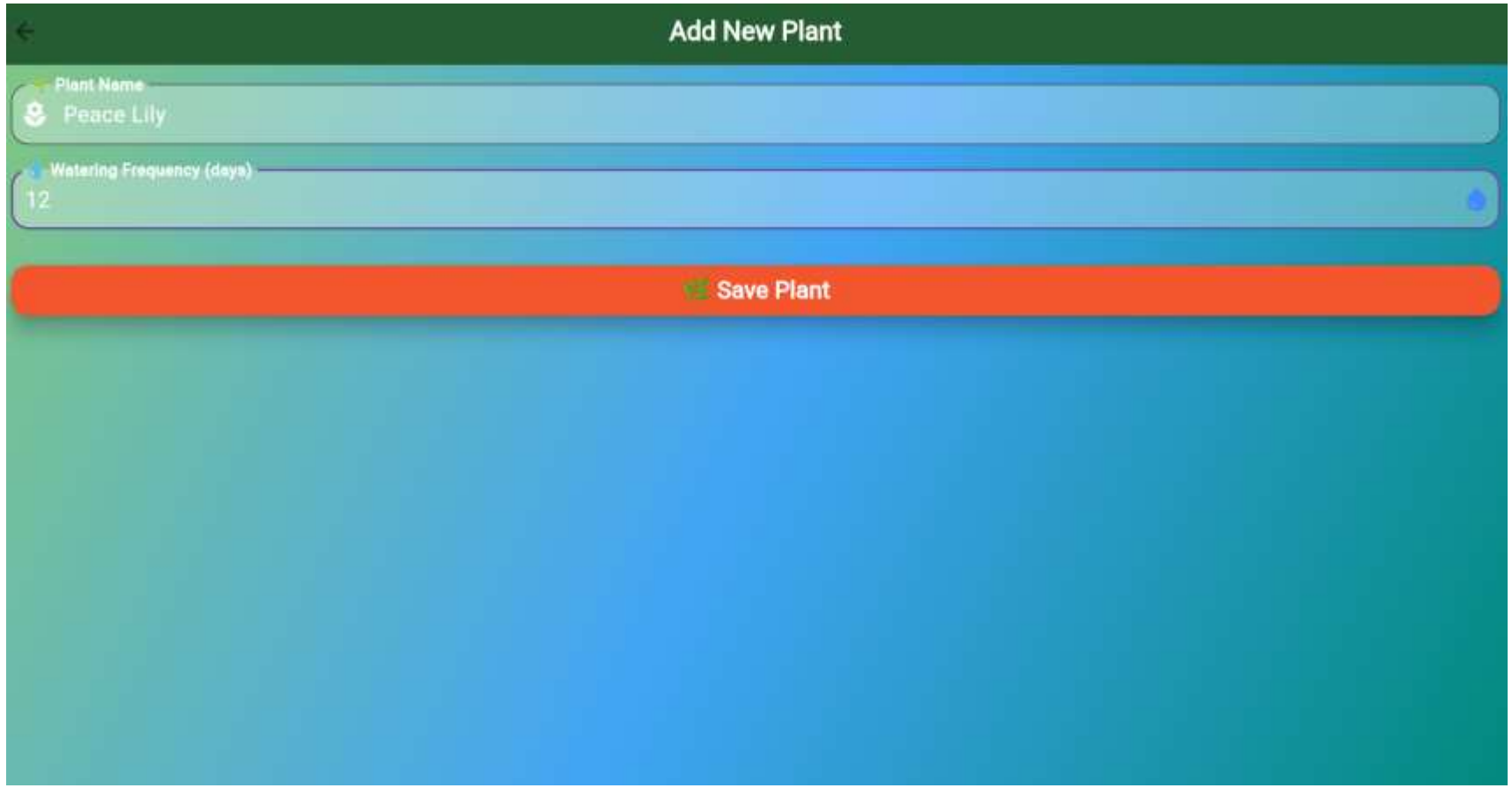
WIDGETS

- Used Stateless and Stateful widgets efficiently.
- Reusable custom widgets for plant cards, buttons, and dialogs.
- Implemented ListView, GridView, Stack, and Column properly.
- Optimized widget tree to avoid unnecessary rebuilds.
- Used Inherited Widget/Provider for efficient state management.


Add New Plant
Plant Name
Watering Frequency (days)
Save Plant

Stateless & Stateful Widgets

- Stateless widgets for static UI components like headers and buttons.
- Stateful widgets for dynamic features like plant growth tracking.
- Used setState() correctly to update UI when needed.
- Efficient use of StatefulBuilder in dialogs where needed.
- Used ValueNotifier & ChangeNotifier for managing state.


Add New Plant
Plant Name
Peace Lily
Watering Frequency (days)
12
Save Plant

Styling & Theming

- Used Material Theme for consistent colors and fonts.
- Created a dark/light mode toggle for better user experience.
- Implemented custom fonts & typography to enhance readability.
- Used BoxDecoration, Shadows, and Gradients for UI elements.
- Maintained theme consistency across all screens.


My Garden
Search your plants...
Monstera Deliciosa
Health: Good
0/7 days
Hydrated
WATER NOW
Snake Plant
Health: Excellent
0/14 days
Hydrated
WATER NOW

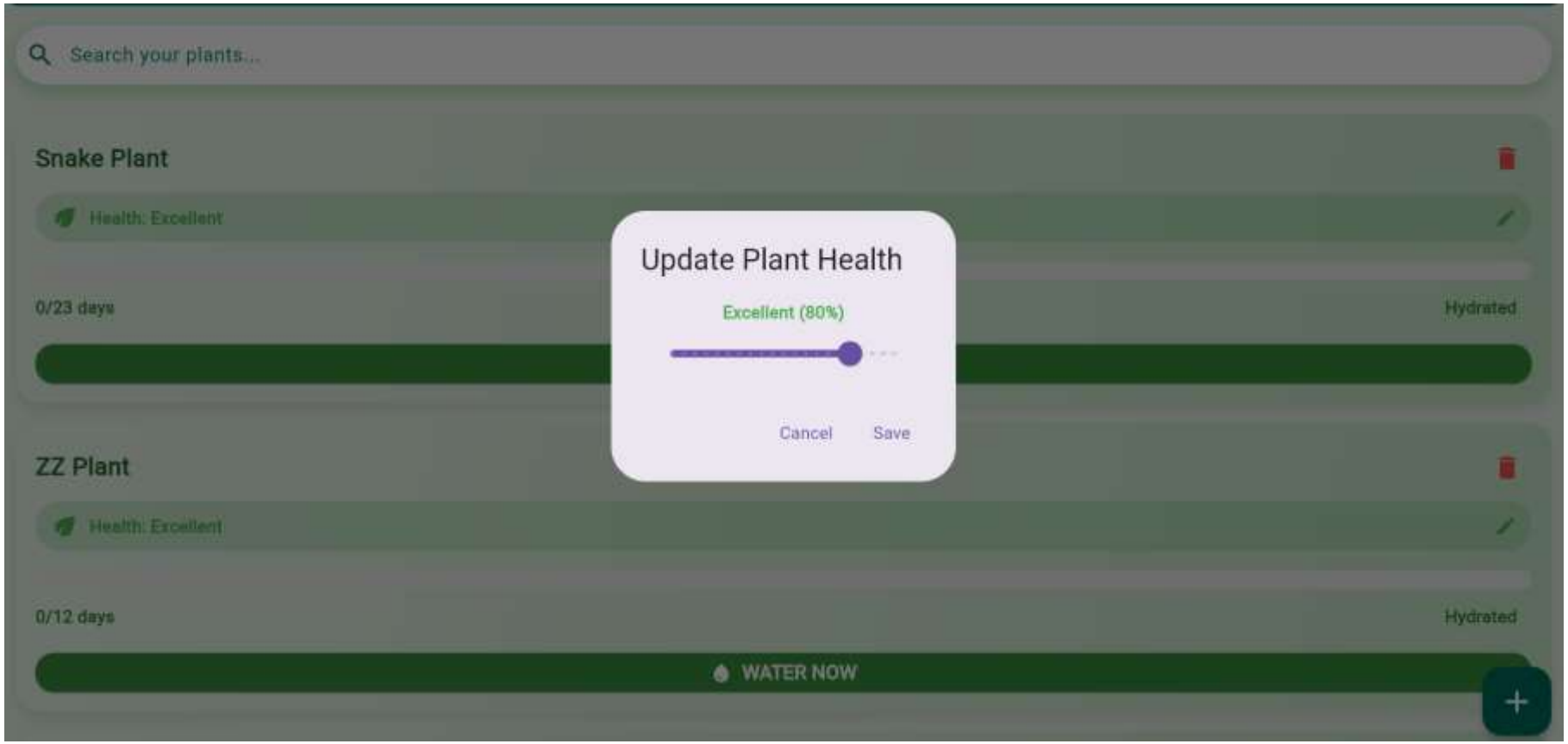
Layouts & Views

- Used Column, Row, Stack, and Expanded for proper UI structuring.
- Implemented Responsive UI using MediaQuery and LayoutBuilder.
- Used SliverAppBar & CustomScrollView for smooth scrolling.
- GridView for displaying plants in a structured manner.
- Proper padding and spacing for a clean, uncluttered UI


My Garden
Search your plants...
ZZ Plant
Health: Needs Care
0/14 days
Hydrated
WATER NOW
Pothos
Health: Needs Care
0/7 days
Hydrated
WATER NOW

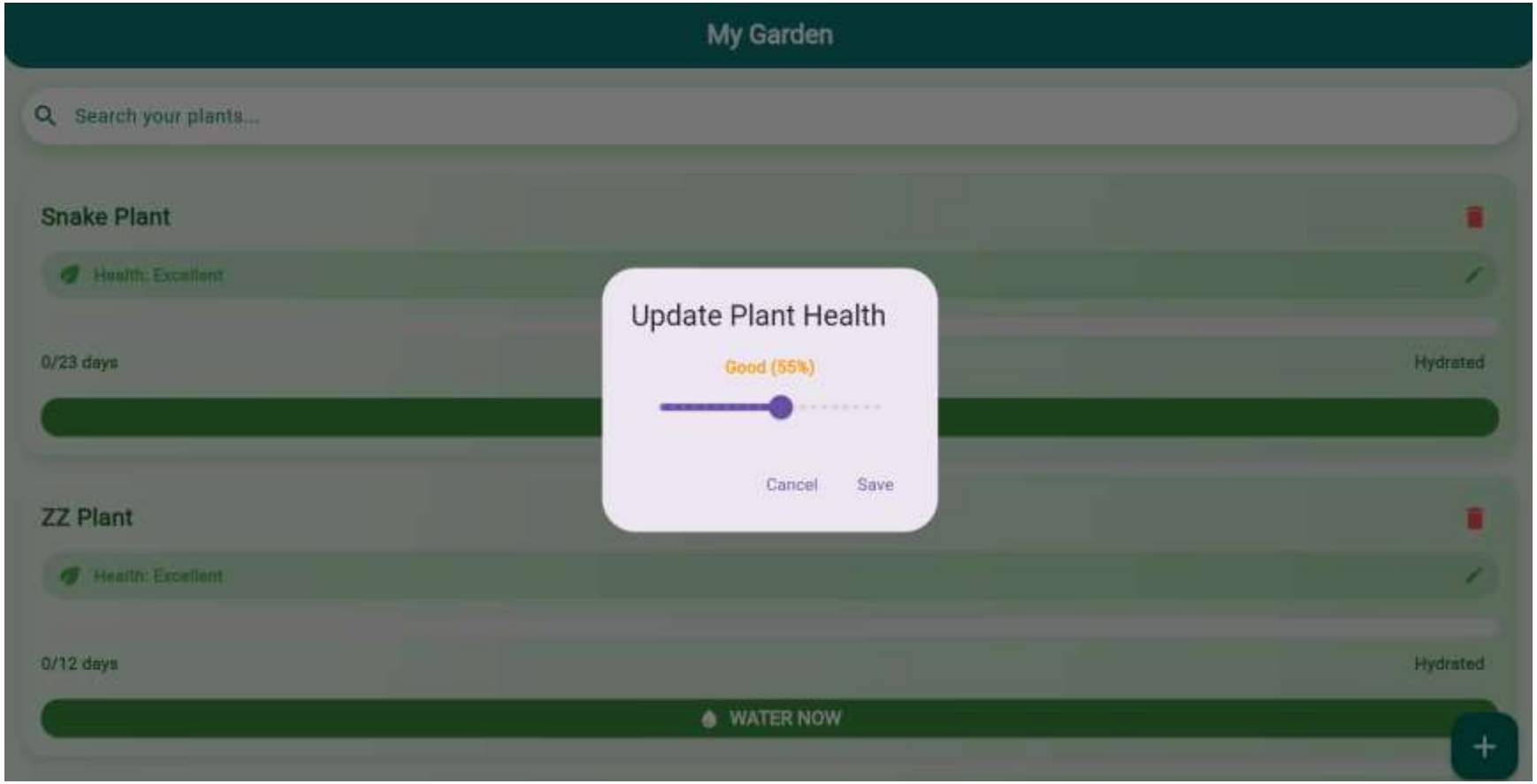
Navigation

- Implemented Navigator.push() & pop() for screen transitions.
- Used Named Routes for better navigation structure.
- Applied PageView & BottomNavigationBar for smooth transitions.
- Used Drawer and Tabs for easy access to features.
- Implemented Deep linking for direct access to specific plant details.


Search your plants...
Snake Plant
Health: Excellent
0/23 days
Hydrated
Update Plant Health
Excellent (80%)
Cancel
Save
ZZ Plant
Health: Excellent
0/12 days
Hydrated
WATER NOW

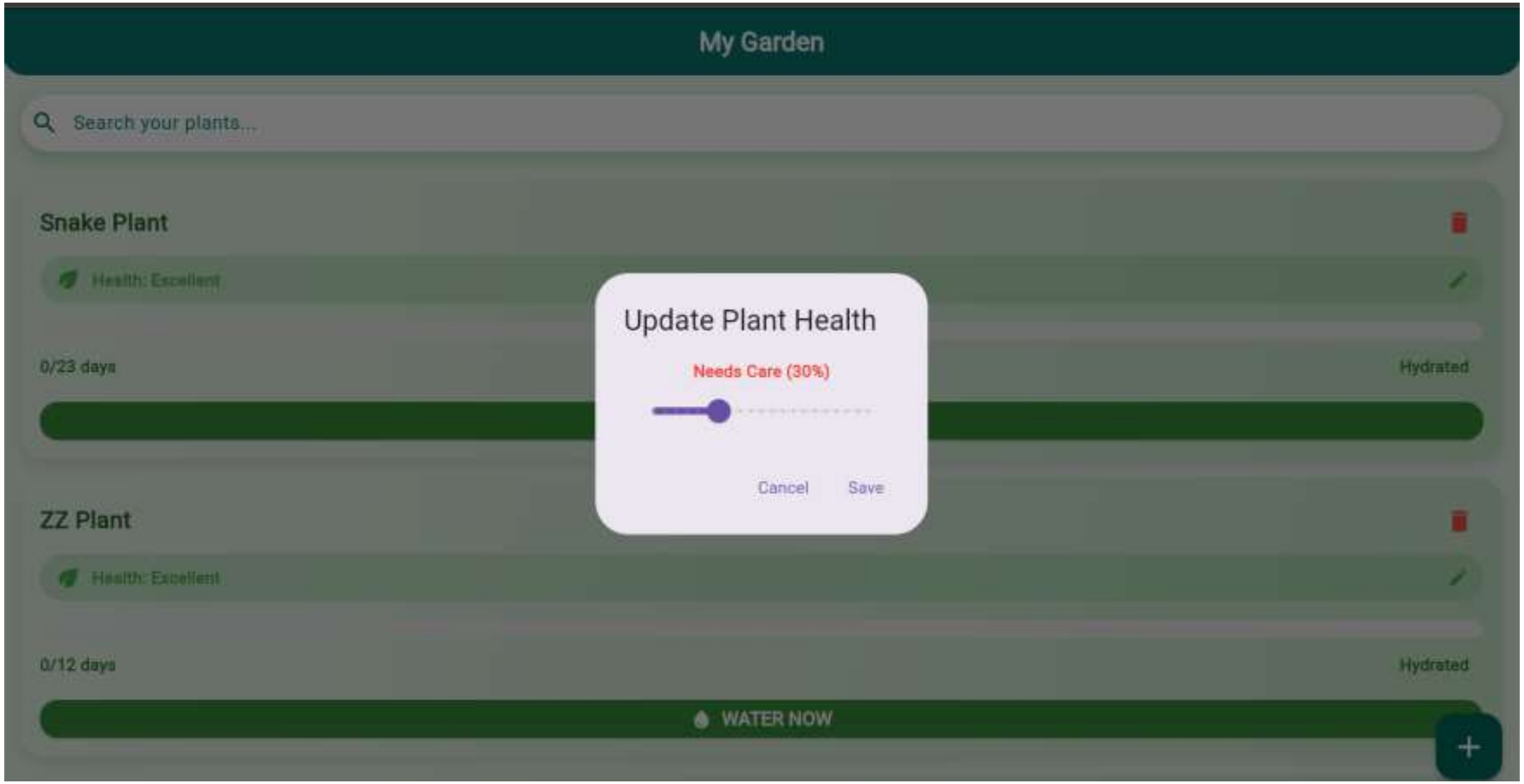
Gestures

- Used GestureDetector for custom touch interactions.
- Swipe gestures for deleting, archiving, or marking tasks.
- Tap interactions for expanding/collapsing plant details.
- Long-press gestures for context menus & additional options.
- Custom animations for tap, hover, and swipe effects


My Garden
Search your plants...
Snake Plant
Health: Excellent
0/23 days
Hydrated
Update Plant Health
Good (55%)
Cancel
Save
ZZ Plant
Health: Excellent
0/12 days
Hydrated
WATER NOW

API Calling

- Used Dio/HTTP package for fetching plant care data.
- Handled API errors using try-catch & error handling UI.
- Implemented Loading & Retry states for API calls.
- Used FutureBuilder to efficiently update UI based on API response.
- Optimized API requests with caching techniques.


My Garden
Search your plants...
Snake Plant
Health: Excellent
0/23 days
Hydrated
Update Plant Health
Needs Care (30%)
Cancel
Save
ZZ Plant
Health: Excellent
0/12 days
Hydrated
WATER NOW

Database

- Used SQLite or Hive for local plant data storage.
- CRUD operations for adding, updating, and deleting plants.
- Implemented SharedPreferences for saving user preferences.
- Optimized queries for faster data retrieval.
- Integrated local database syncing with Firebase for backup.

Firebase Integration

- Firestore for storing plant data in the cloud.
- Firebase Authentication for user sign-in/sign-up.
- Firebase Storage for storing plant images.
- Push Notifications to remind users about plant care.
- Firestore offline mode for data availability without the internet.

Localization

- Used intl package for multiple language support.
- Added support for at least 2-3 languages (English, Tamil, etc.).
- Implemented dynamic text translation based on user settings.
- Used language JSON files for easy text management.
- Ensured text direction and UI adaptability for different languages