



SMART SPROUT

Document Title: Deliverable 4

Team Name: Smart Sprout

Project Name: Smart Home-Garden System

Group Number: Group 9

Team Member & Student ID:

- Aditi Patel n01525570
- Birava Prajapati n01579924
- Darshankumar Prajapati n01574247
- Zeel Patel n01526282

Table of Content

Table of Content.....	2
Brief description of the project.....	4
Comparison with two other apps.....	5
Work Completed by each member.....	6
Unit Test Cases.....	7
Sprint goals:.....	8
Sprint Stories and Tasks:.....	9
Sprint Retrospective.....	11
C4 Model:.....	12
C4 System Context Diagram:.....	12
Key Interactions:.....	12
C4 System Container Diagram:.....	13
Design Pattern.....	14
1. Builder Pattern.....	14
Code Example:.....	14
Explanation:.....	14
2. Dependency Injection.....	14
Code Example:.....	15
Explanation:.....	15
Coding work progress since deliverable 3.....	16
Feedback Screen.....	17

Name	Student Id	Github Id	Signature	Efforts
Aditi Patel	n01525570	AditiPatel5570		100%
Birava Prajapati	n01579924	BiravaPrajapati9924		100%
Darshankumar Prajapati	n01574247	DarshanPrajapati4247		100%
Zeel Patel	n01526282	ZeelPatel6282		100%

- GitHub Repo Link: <https://github.com/DarshanPrajapati4247/SmartHomeGardenSys>

Brief description of the project

Our project, **Smart Sprout**, is a smart home gardening app designed to simplify plant care by integrating sensor data. It helps users monitor their plants' needs, such as moisture and temperature. The app automates plant care by offering notifications, weather updates, and a user-friendly interface for managing and tracking plant health, making it easy for users to care for their plants effortlessly.

What issue your product will solve?

The **Smart Sprout** app addresses the challenge of keeping plants healthy by automating and simplifying plant care. It helps users monitor and adjust the conditions necessary for plant growth, such as soil moisture and temperature without requiring constant attention. This solution eliminates the guesswork, saves time, and makes plant care more accessible, especially for people who may struggle with maintaining optimal plant conditions or are new to gardening.

Comparison with two other apps

	Smart Home Garden System	PlantNet	Flora
Links		Website: https://identify.plantnet.org/ App: https://play.google.com/store/apps/details?id=org.plantnet&pampaignid=web_share	Website: https://florasense.com/ App: https://play.google.com/store/apps/details?id=com.florasense&referrer=utm_source%3Dweb%26utm_medium%3Dhbanner
Features	<ul style="list-style-type: none"> • Plant identification and diagnosis using photos. • Automated Watering • Automated data updates • Sensor-based monitoring • Supports French • Reminder notifications. 	<ul style="list-style-type: none"> • Plant identification using photos • Extensive plant species database • Geographical filtering for localised results • Community-contributed database • Multi-language support • Educational plant information 	<ul style="list-style-type: none"> • Plant Identification • Personalized Care Guides • Watering Reminders • Community Engagement • Rewards & Badges
Pros	<ul style="list-style-type: none"> • Saves time and effort in plant care. • Ensures optimal conditions for plant growth. • Convenient remote monitoring. • Free to use. 	<ul style="list-style-type: none"> • Collaborative contributions from a global user base • Free to use without subscription • Encourages environmental awareness • User-friendly interface 	<ul style="list-style-type: none"> • Extensive plant identification • Customizable care guides • Rewards system • Active community
Cons	<ul style="list-style-type: none"> • Initial setup can be expensive because of the sensors. • Requires stable internet connection. • Limited plant species for now. 	<ul style="list-style-type: none"> • Requires active internet connection for full functionality • Accuracy depends on high-quality photos • Lacks real-time monitoring features 	<ul style="list-style-type: none"> • Some features may require internet access • Relies on user input for some data tracking • Limited to the plants in the database

In our app, we've integrated sensor-based data that gives users real-time information on when and how to care for their plants, based on their specific needs. Unlike other apps, which don't offer this feature, our system makes plant care effortless. By automatically tracking plant conditions, we ensure users can maintain healthy plants without the guesswork or extra effort. This hands-off approach makes plant care much more intuitive and convenient.

Work Completed by each member

Aditi Patel: I have fully implemented the menubar in the Smart Sprout app, ensuring that it functions smoothly with the rest of the application. In addition to this, I enhanced the feedback screen feature by adding more functionality and improving the user experience. To further strengthen the app, I created and added 2 test cases to ensure the stability and reliability of the features. I also made changes to the Home fragment UI, improving its visual appeal and overall usability. Lastly, I implemented a refresh scroll feature, allowing users to easily update the weather data displayed on the screen, ensuring real-time updates and a seamless experience.

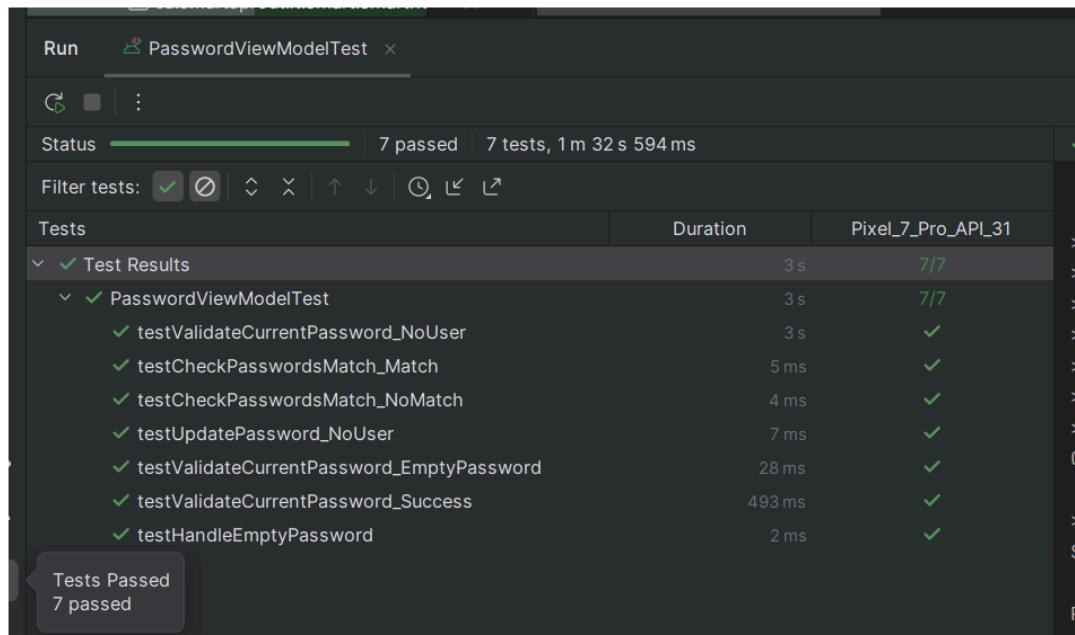
Birava Prajapati: I've made significant progress in enhancing the Smart Sprout app's functionality by implementing dynamic notification permission handling, and ensuring real-time updates in the app when system settings change. Password validation using regex was introduced to enforce strong password requirements, improving security. In the account settings, users can now update their profile pictures with real-time previews, edit their names with proper validation, and securely change their passwords with current password verification. I also enhanced user interactivity by adding functionality to one of the Floating Action Buttons (FAB) and resolved syncing issues with the notification toggle, ensuring real-time feedback for user actions. Additionally, I tested critical features like password matching and update workflows, including a test to verify the app correctly identifies mismatched passwords and handles cases where no user is logged in, ensuring robust and user-friendly functionality.

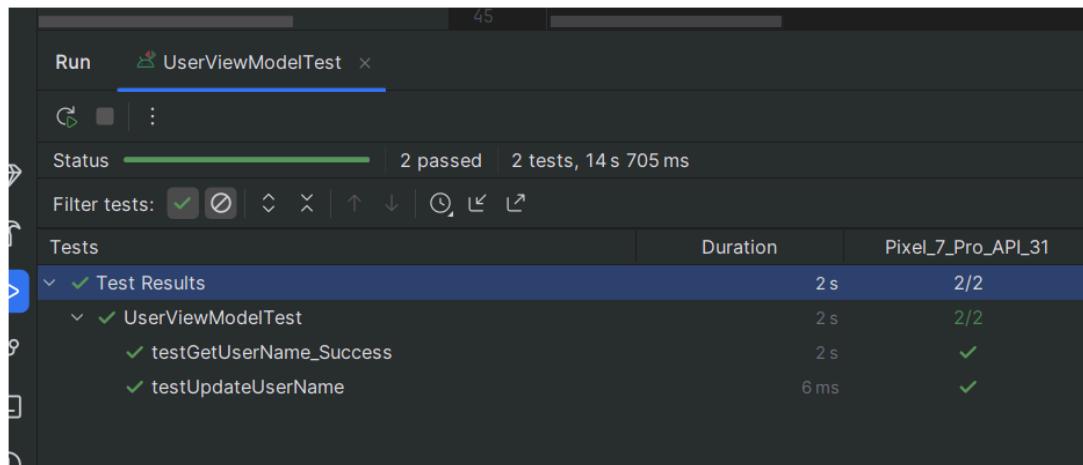
Darshan Prajapati: Since Deliverable 3, I have made good progress in my coding and testing work. I implemented a search feature using the Perennial Plant API, allowing users to search for plants by name and see the results in a list. I also worked on testing important functions, like validating current passwords and checking if two passwords match, to ensure the app is secure and works correctly. These updates make the app easier to use and more reliable.

Zeel Patel: I worked on integrating Google Sign-In functionality and revamped the UI to ensure a more user-friendly experience. I implemented regex for phone validation to ensure correct input format and added password validation with a toggle to hide or reveal the password for better security. Additionally, I introduced the 'Remember Me' checkbox to streamline login sessions for users. To further enhance the user experience, I redesigned the registration screen UI, making it more intuitive and visually appealing.

Unit Test Cases

- **assertEquals** is there to make sure that the value I'm expecting matches what's actually returned. For example, when checking if two passwords match, it confirms that the system properly identifies them as equal or not.
- **assertTrue** is used to verify that a specific condition is correct. For instance, when I test if the username is updated successfully, this ensures that the updated name is actually what it's supposed to be.
- **assertFalse** is great for confirming that something isn't true. For example, when validating an empty password, this checks that the system correctly recognizes it as invalid.
- **assertNotNull** helps to make sure that something exists and hasn't been missed or left out. For example, in LiveData tests, it's used to ensure that the value isn't null before running further checks on it.





Sprint goals:

Name	Assignee	Status	Start Date
↳ Sprint4 - SmartSprout Core Features and Enhancements ⓘ 8	BP AP DP ZP	IN PROGRESS	Nov 7
↳ As a user, I want a clear and secure registration screen where I can enter my full name, email, password, and phone nu... ⓘ 6	ZP	IN PROGRESS	Nov 7
↳ ✓ As a user, I want a menu bar with options to access Settings, Notifications, unit conversion, and a help section, so I can... ⓘ 5	AP	DONE	Nov 7
↳ ✓ As a user, I want to search for plants through the search screen and view the results in a scrollable list, so that I can eas... ⓘ 8	DP	DONE	Nov 7
↳ ✓ As a user, I want to update my profile picture, full name, and password from the Profile Settings screen, so that my inf... ⓘ 13	BP	DONE	Nov 7
↳ ✓ As a team leader, I want to compare SmartSprout with two similar apps to showcase its unique advantages and improv... ⓘ 5	DP AP BP	DONE	Nov 7
↳ ✓ As a developer, I want to create and run unit tests for a key Java class in the app, ensuring that the core functionality is... ⓘ 5	DP AP BP	DONE	Nov 7
↳ ⓘ As a user, I want to provide feedback on the app by filling out a form with my details and rating, and I want to receive ... ⓘ 5	ZP AP	IN PROGRESS	Nov 7
↳ ⓘ As a user, I want a secure login screen that enforces a strict format for credentials, so I can be assured that my account... ⓘ 4	ZP	IN PROGRESS	Nov 7

ⓘ This is a group with tasks from other Lists

Name	Assignee	Status	Start Date	Due date	T-Shirt Sizing	Priority
↳ Sprint4 - SmartSprout Core Features and Enhancements ⓘ 8		Today		Today	XS S M L XL XXL	High
↳ ⓘ As a user, I want a clear and secure registration screen where I can enter my full name, email, password, and phone nu... ⓘ 6		Today		Today	XS S M L XL XXL	Critical
↳ ✓ As a user, I want a menu bar with options to access Settings, Notifications, unit conversion, and a help section, so I can... ⓘ 5		Today		Today	XS S M L XL XXL	Medium
↳ ✓ As a user, I want to search for plants through the search screen and view the results in a scrollable list, so that I can eas... ⓘ 8		Today		Today	XS S M L XL XXL	Critical
↳ ✓ As a user, I want to update my profile picture, full name, and password from the Profile Settings screen, so that my inf... ⓘ 13		Today		Today	XS S M L XL XXL	High
↳ ✓ As a team leader, I want to compare SmartSprout with two similar apps to showcase its unique advantages and improv... ⓘ 5		Today		Today	XS S M L XL XXL	Medium
↳ ✓ As a developer, I want to create and run unit tests for a key Java class in the app, ensuring that the core functionality is... ⓘ 5		Today		Today	XS S M L XL XXL	Critical
↳ ⓘ As a user, I want to provide feedback on the app by filling out a form with my details and rating, and I want to receive ... ⓘ 5		Today		Today	XS S M L XL XXL	High
↳ ⓘ As a user, I want a secure login screen that enforces a strict format for credentials, so I can be assured that my account... ⓘ 4		Today		Today	XS S M L XL XXL	Critical

Sprint Stories and Tasks:

Name	Assignee	Status	Start Date	Due date	T-Shirt Sizing	Priority
✓ Sprint4 - SmartSprout Core Features and Enhancements 8	BP AP JP ZP	DONE	Nov 7	Today		High
✓ As a user, I want to search for plants through the search screen and view the results in a scrollable list, so that I can eas...	DP	DONE	Nov 7	Today		Critical
✓ Set up the Plant API connection in the application to fetch plant data.	DP	DONE	Nov 7	Today		Critical
✓ Parse the API response to retrieve plant details such as name, type, and description.	DP	DONE	Nov 7	Today		High
✓ Store the fetched plant data in a local list for display on the search screen.	DP	DONE	Nov 7	Today		High
✓ Bind the fetched plant data to the RecyclerView adapter for display.	DP	DONE	Nov 7	Today		Critical
✓ Implement real-time search functionality to filter the plant list as the user types.	DP	DONE	Nov 7	Today		High
✓ Handle edge cases such as: =	DP	DONE	Nov 7	Today		Critical
✓ Add error handling for API connectivity issues, displaying an appropriate error message when the API call fails.	DP	DONE	Nov 7	Today		High
✓ Test the RecyclerView to ensure it scrolls smoothly.	DP	DONE	Nov 7	Today		Medium
✓ Sprint4 - SmartSprout Core Features and Enhancements 8				Today		High
✓ As a user, I want to search for plants through the search screen and view the results in a scrollable list, so that I can eas...				Today		Critical
✓ Set up the Plant API connection in the application to fetch plant data.				Today		Critical
✓ Parse the API response to retrieve plant details such as name, type, and description.				Today		High
✓ Store the fetched plant data in a local list for display on the search screen.				Today		High
✓ Bind the fetched plant data to the RecyclerView adapter for display.				Today		Critical
✓ Implement real-time search functionality to filter the plant list as the user types.				Today		High
✓ Handle edge cases such as: =				Today		Critical
✓ Add error handling for API connectivity issues, displaying an appropriate error message when the API call fails.				Today		High
✓ Test the RecyclerView to ensure it scrolls smoothly.				Today		Medium
✓ Sprint4 - SmartSprout Core Features and Enhancements 8						High
✓ As a user, I want to update my profile picture, full name, and password from the Profile Settings screen, so that my inf...						High
✓ Add an option on the Profile Settings screen to change the profile picture.						High
✓ Implement functionality to upload a new profile picture: =						Low
✓ Save picture in the shared preference.						Medium
✓ Fetch the new profile picture and display it across all relevant screens in the app.						High
✓ Add an editable field for the user's full name on the Profile Settings screen.						Low
✓ Validate the new name (e.g., ensure it's not empty or too short).						Medium
✓ Update the full name in the Firebase database upon confirmation.						High
✓ Ensure the updated full name reflects on all relevant screens.						High
✓ Add a field for the current password and two fields for the new password and confirm password.						High
✓ Validate the new password according to the following criteria: =						High
✓ Compare the new and confirm password fields to ensure they match.						Medium
✓ Authenticate the current password before allowing changes.						High
✓ Update the password in Firebase Authentication and notify the user of a successful update.						Critical
✓ Sprint4 - SmartSprout Core Features and Enhancements 8						High
✓ As a user, I want to update my profile picture, full name, and password from the Profile Settings screen, so that my inf...						High
✓ Add an option on the Profile Settings screen to change the profile picture.						Low
✓ Implement functionality to upload a new profile picture: =						Medium
✓ Save picture in the shared preference.						Medium
✓ Fetch the new profile picture and display it across all relevant screens in the app.						High
✓ Add an editable field for the user's full name on the Profile Settings screen.						Low
✓ Validate the new name (e.g., ensure it's not empty or too short).						Medium
✓ Update the full name in the Firebase database upon confirmation.						High
✓ Ensure the updated full name reflects on all relevant screens.						High
✓ Add a field for the current password and two fields for the new password and confirm password.						High
✓ Validate the new password according to the following criteria: =						High
✓ Compare the new and confirm password fields to ensure they match.						Medium
✓ Authenticate the current password before allowing changes.						High
✓ Update the password in Firebase Authentication and notify the user of a successful update.						Critical

Name	Description	Assignee	Status	Start Date
✓ Sprint4 - SmartSprout Core Features and Enhancements 8		BP AP DP ZP	DONE	Nov 7
✓ As a user, I want a menu bar with options to access Settings, Notifications, unit conversion, and a help section, so I can...	5	AP	DONE	Nov 7
✓ Design and implement the Menu Bar layout, ensuring it's accessible from every screen.		AP	DONE	Nov 7
✓ Add navigation to the Settings screen when the Settings option is selected.		AP	DONE	Nov 7
✓ Implement navigation to the Notification screen for viewing recent alerts.		AP	DONE	Nov 7
✓ Develop the Unit Conversion feature that toggles between Celsius and Fahrenheit and stores the user's choice in Share...		AP	DONE	Nov 7
✓ Implement the Help option.		AP	DONE	Nov 7

Name	Description	Assignee	Status	Start Date	Due Date	T-Shirt Sizing	Priority
✓ Sprint4 - SmartSprout Core Features and Enhancements 8			Today				High
✓ As a user, I want a menu bar with options to access Settings, Notifications, unit conversion, and a help section, so I can...	5		Today				Medium
✓ Design and implement the Menu Bar layout, ensuring it's accessible from every screen.			Today				Medium
✓ Add navigation to the Settings screen when the Settings option is selected.			Today				Low
✓ Implement navigation to the Notification screen for viewing recent alerts.			Today				Medium
✓ Develop the Unit Conversion feature that toggles between Celsius and Fahrenheit and stores the user's choice in Share...			Today				Medium
✓ Implement the Help option.			Today				Medium

Name	Description	Assignee	Status	Start Date
● Sprint4 - SmartSprout Core Features and Enhancements 8		BP AP DP ZP	IN PROGRESS	Nov 7
● As a user, I want a clear and secure registration screen where I can enter my full name, email, password, and phone nu...	6	ZP	READY TO START	Nov 7
✓ Implement an Email field to act as the username and validate it with a proper email format using Regex.		ZP	DONE	Nov 7
✓ Remove any existing username (userId) field and ensure the email is used as a unique identifier for each user.		ZP	DONE	Nov 7
✓ Implement a Password field with following format requirements.		ZP	DONE	Nov 7
● Add a Confirm Password field and verify it matches the Password field with matched indication.		ZP	IN PROGRESS	Nov 7
● Add a Phone Number field that validates a 10-digit phone format using Regex.		ZP	IN PROGRESS	Nov 7
● Test the registration flow, ensuring all validations (Regex, field requirements) are working correctly and that the registr...		ZP	IN PROGRESS	Nov 7

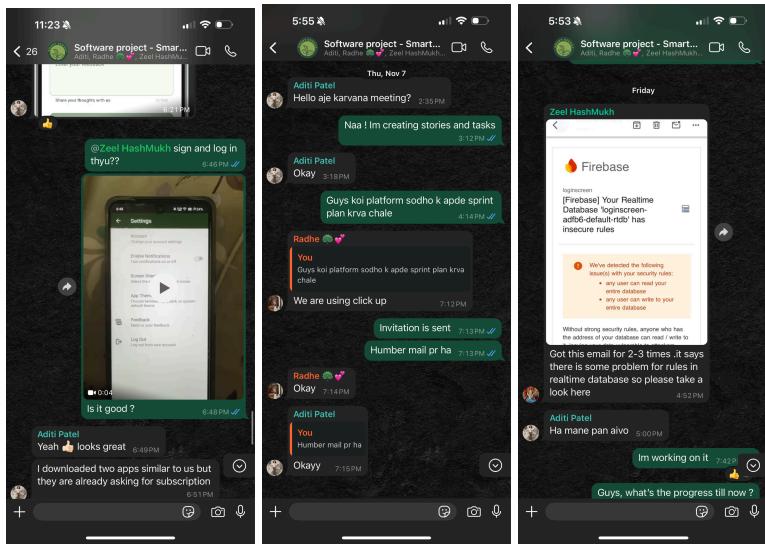
Name	Description	Assignee	Status	Start Date	Due Date	T-Shirt Sizing	Priority
● Sprint4 - SmartSprout Core Features and Enhancements 8			Today				High
● As a user, I want a clear and secure registration screen where I can enter my full name, email, password, and phone nu...	6		Today				Critical
● Implement an Email field to act as the username and validate it with a proper email format using Regex.			Today				High
● Remove any existing username (userId) field and ensure the email is used as a unique identifier for each user.			Today				Low
● Implement a Password field with following format requirements.			Today				Medium
● Add a Confirm Password field and verify it matches the Password field with matched indication.			Today				Medium
● Add a Phone Number field that validates a 10-digit phone format using Regex.			Today				Medium
● Test the registration flow, ensuring all validations (Regex, field requirements) are working correctly and that the registr...			Today				Critical

Name	Description	Assignee	Status	Start Date
● Sprint4 - SmartSprout Core Features and Enhancements 8		BP AP DP ZP	IN PROGRESS	Nov 7
● As a user, I want to provide feedback on the app by filling out a form with my details and rating, and I want to receive ...	5	ZP AP	IN PROGRESS	Nov 7
● Implement input validation for the feedback form to ensure valid phone numbers and other required fields.		ZP	READY TO START	Nov 7
● Display a progress bar during feedback submission, with a 5-second delay to simulate processing time.		AP	DONE	Nov 7
● On successful feedback submission, show an AlertDialog with an OK button confirming that the form was submitted.		ZP	READY TO START	Nov 7
● Clear the form input fields upon submission and restrict further submissions to once every 24 hours.		AP	DONE	Nov 7
● Disable the submit button and display a countdown timer showing the remaining time until the next available submiss...		AP	DONE	Nov 7
● As a user, I want a secure login screen that enforces a strict format for credentials, so I can be assured that my account...	4	ZP	IN PROGRESS	Nov 7
● Implement username validation to check that the input is a valid email format.		ZP	DONE	Nov 7
● Implement password validation using Regex to enforce the following rules.		ZP	READY TO START	Nov 7
● Test and verify credentials against the database on the cloud to ensure they are valid.		ZP	IN PROGRESS	Nov 7
● Display error messages for invalid credentials, guiding the user to correct any mistakes.		ZP	READY TO START	Nov 7

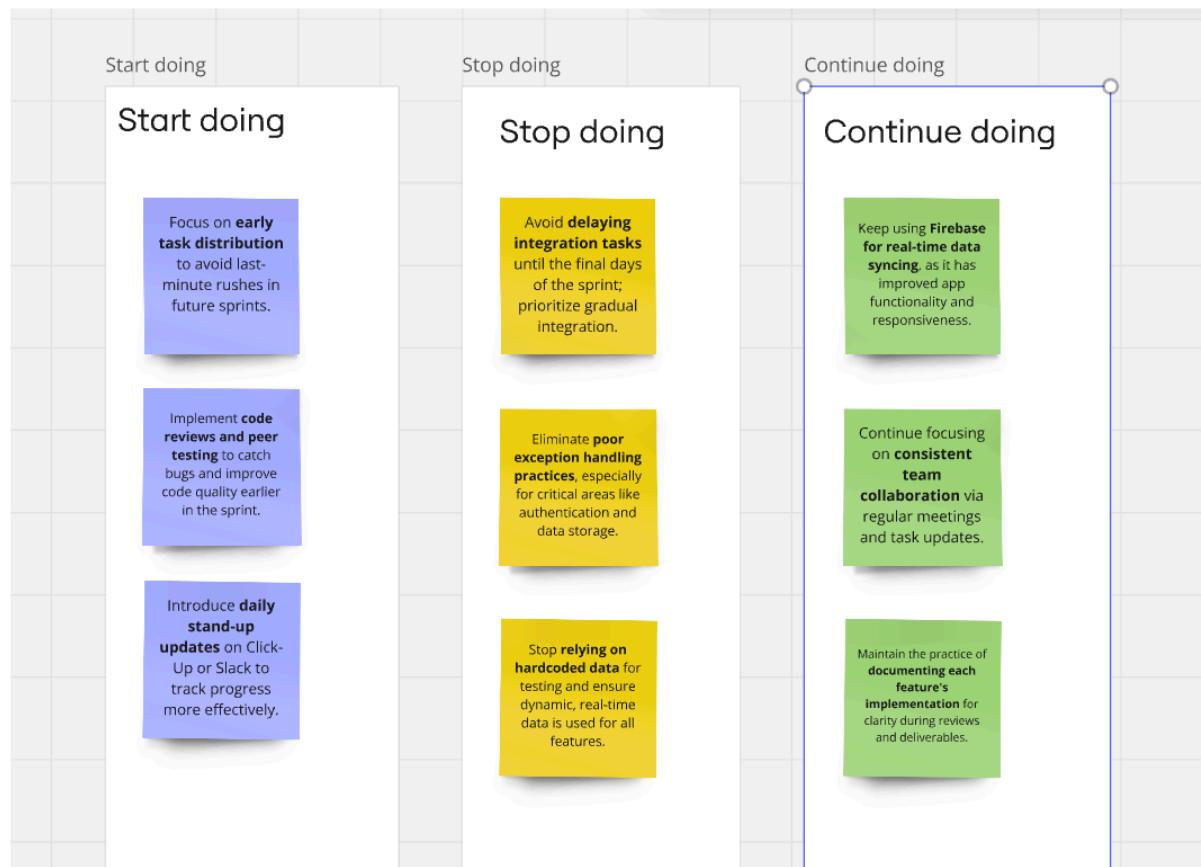
● This is a group with tasks from other Lists

Take screenshots showing the team discussion, topics discussed, and team members. Provide screenshots for three different dates, i.e. WhatsApp, Discord, ...etc.

Team Collaboration

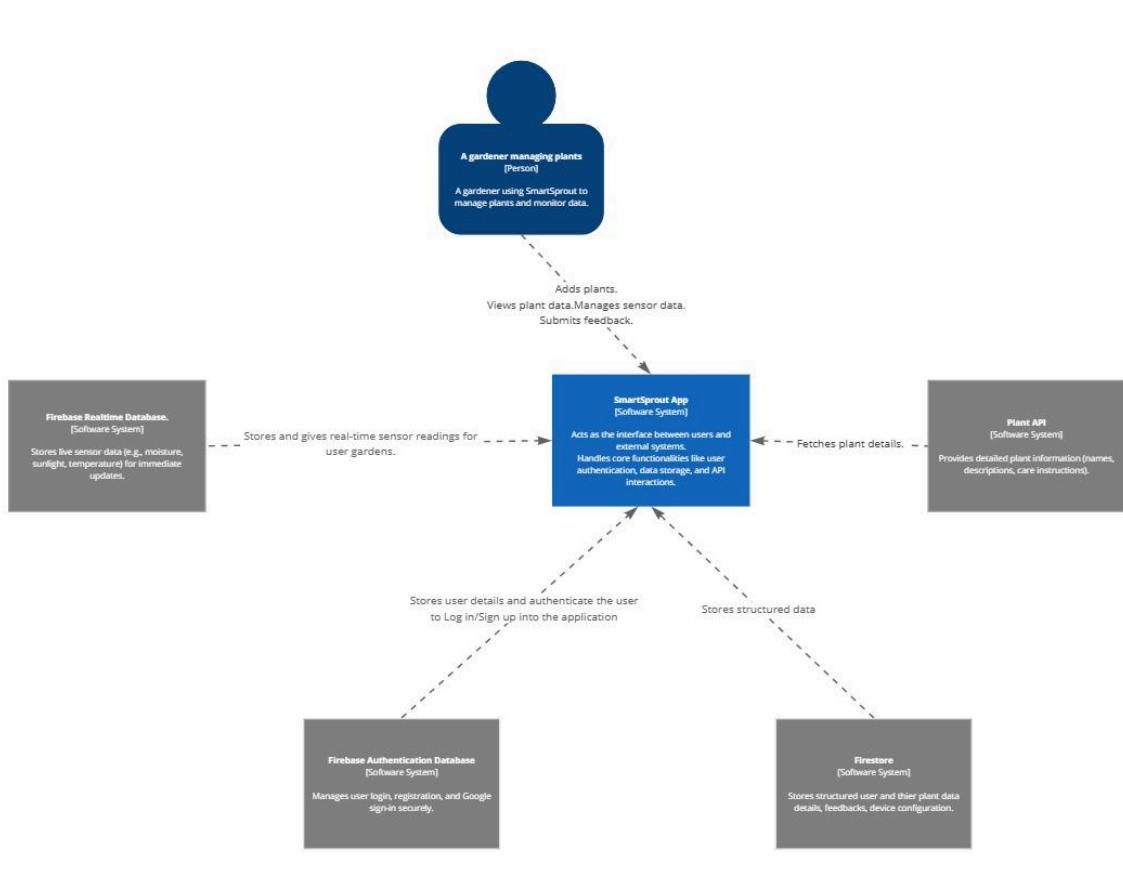


Sprint Retrospective



C4 Model:

C4 System Context Diagram:

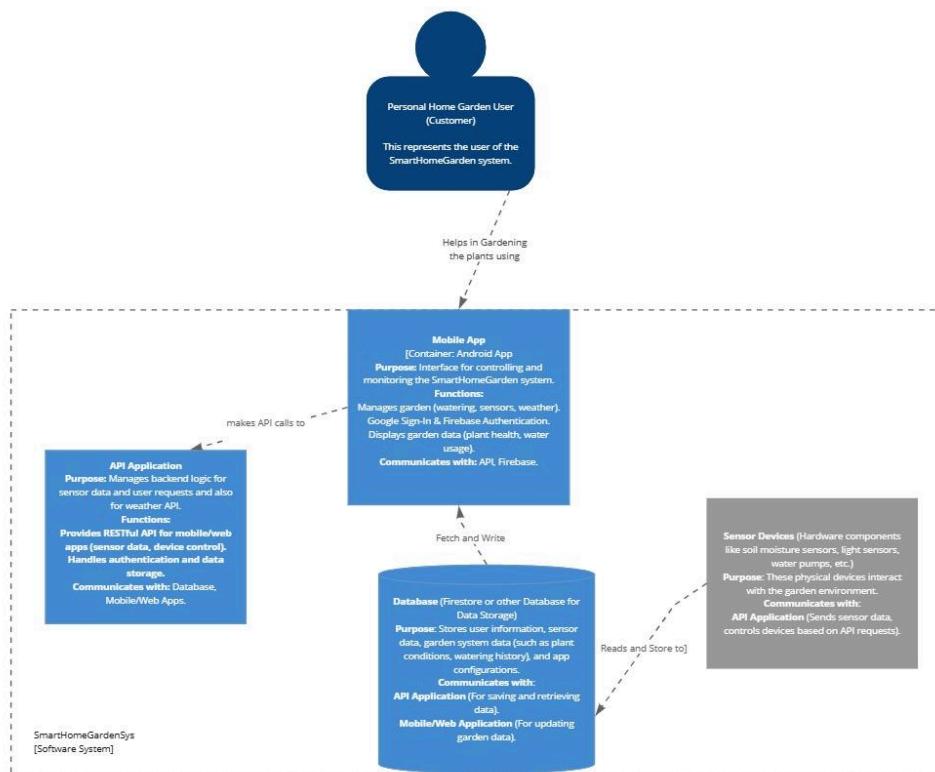


Key Interactions:

- The **End User** interacts with the **SmartSprout App** to manage plants and monitor data.
- The **SmartSprout App** fetches real-time sensor data from the **Firebase Realtime Database** and structured plant data from **Firestore**.
- The **Plant API** supplies detailed plant information for the app.
- **Firebase Authentication** ensures secure user login and registration.

This diagram illustrates how the app integrates with various systems to provide a seamless experience for the user while ensuring data synchronization and secure authentication.

C4 System Container Diagram:



Key Interactions::

- **User**: Interacts with the **Mobile App** to manage the garden.
- **Mobile App**: Communicates with the **API Application** for data and device control, and with the **Database** to display garden data.
- **API Application**: Handles backend logic, manages data storage, and provides RESTful APIs for the **Mobile App** and **Sensor Devices**.
- **Database**: Stores user and sensor data, and works with the **API Application** for data retrieval and storage.
- **Sensor Devices**: Collect data and control devices, interacting with the **API Application** for commands.

Design Pattern

1. Builder Pattern

The **Builder Pattern** is used in the `showImagePickerDialog()` method to simplify the construction of an `AlertDialog`. This dialog is used to give users a choice between picking an image from the camera or the gallery.

Code Example:

```
new AlertDialog.Builder(this)
    .setTitle(R.string.choose_option)
    .setMessage(R.string.select_camera_or_gallery)
    .setPositiveButton(R.string.camera, (dialog, which) -> {
        // Open camera logic
    })
    .setNegativeButton(R.string.gallery, (dialog, which) -> {
        // Open gallery logic
    }).show();
```

Explanation:

- The `AlertDialog.Builder` allows incrementally adding different components to the dialog, like the title, message, and buttons, without cluttering the code with a single large constructor.
 - Once everything is configured, the `show()` method finalizes the dialog and displays it.
 - This makes the code cleaner and easier to understand, especially if additional options are added later.
-

2. Dependency Injection

Dependency Injection plays a crucial role in the Smart Sprout app's design, especially through the use of `ViewModelProvider`, which simplifies the management of `ViewModel` lifecycles for classes like `AccountSettingsViewModel`, `UserViewModel`, and `PasswordViewModel`. Additionally, it's seen in the way external resources, such as `SharedPreferences`, are passed into classes like `AccountSettingsViewModel`. This approach ensures that dependencies are handled efficiently and keeps the code clean and modular.

Code Example:

```
public AccountSettingsViewModel(Application application) {
    super(application);
```

```

        sharedPreferences =
    application.getSharedPreferences(PREFS_USER_PROFILE,
Context.MODE_PRIVATE);
    loadProfileData();
}

viewModel = new ViewModelProvider(this).get(AccountSettingsViewModel.class);
passwordViewModel = new ViewModelProvider(this).get(PasswordViewModel.class);
userViewModel = new ViewModelProvider(this).get(UserViewModel.class);

```

Explanation:

1. SharedPreferences:

- In `AccountSettingsViewModel`, the `Application` object is passed into the constructor, allowing `SharedPreferences` to be accessed without needing to manually set them up inside the class.
- This keeps the responsibilities clear: `SharedPreferences` manage data persistence, while the `ViewModel` focuses on handling the app's logic.
- For instance, `AccountSettingsViewModel` uses `SharedPreferences` to save and retrieve the profile image URI and username, ensuring that user data stays consistent across app sessions.

2. ViewModelProvider:

- The `ViewModelProvider` takes care of creating and managing the lifecycle of `ViewModel` instances like `AccountSettingsViewModel`, `PasswordViewModel`, and `UserViewModel`.
- This ensures that the same `ViewModel` instance persists across configuration changes, such as when the screen rotates, without requiring extra code from the developer.
- For example, in `AccountSettingsActivity`, the `AccountSettingsViewModel` is injected using `ViewModelProvider`, which handles updates like profile image URLs. This allows the activity to focus purely on UI logic while leaving data management to the `ViewModel`.

Coding work progress since deliverable 3

Fully implemented the menubar, enhanced the feedback screen with added functionality, and created 10 test cases. Updated the Home fragment UI for improved usability and added a refresh scroll feature for real-time weather updates.

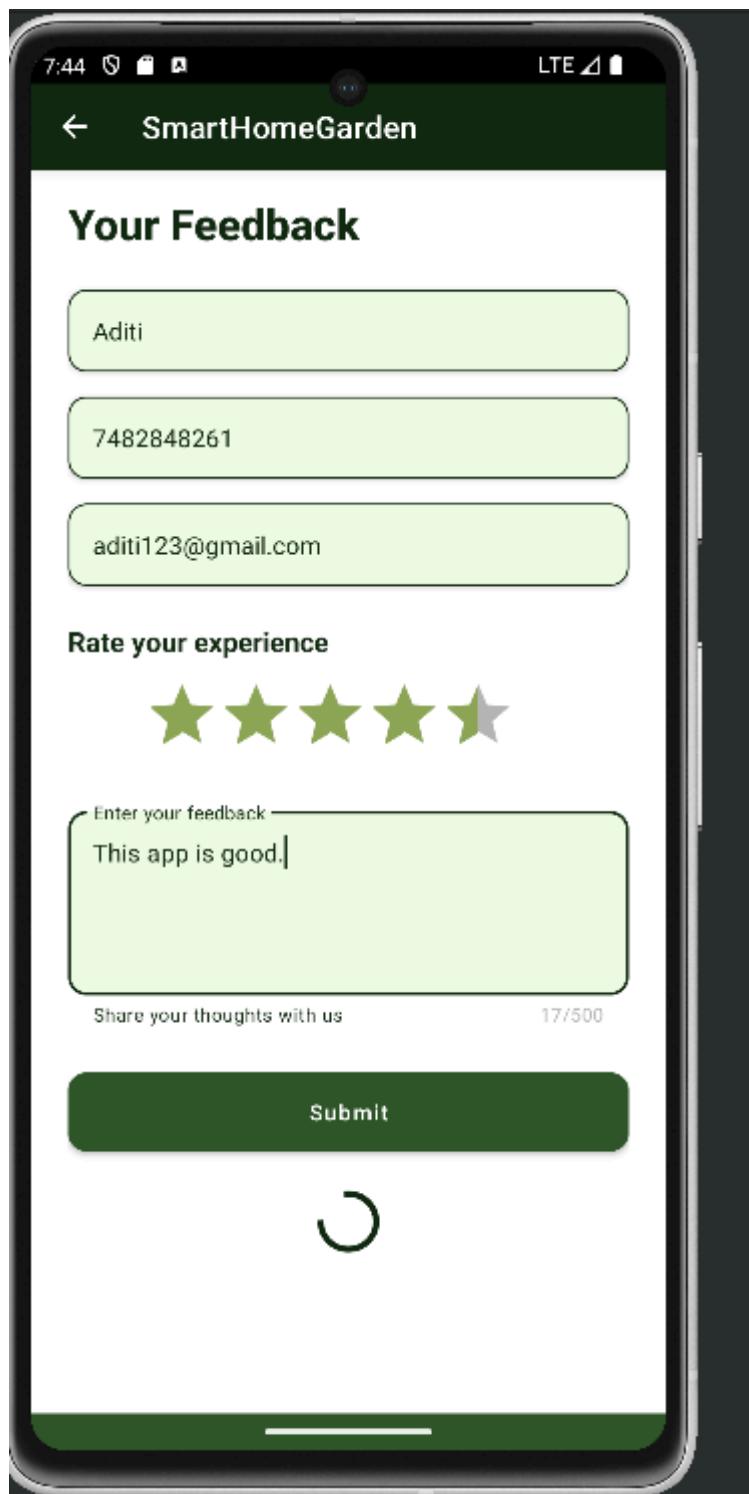
Implemented dynamic notification permission handling with real-time updates, password validation using regex, and dynamic profile updates (profile picture, name, and password). Enhanced user interaction with a Floating Action Button and resolved notification toggle syncing issues. Conducted thorough testing for password workflows and user actions.

Developed a search feature using the Perennial Plant API for plant name searches and tested critical functions like password validation and matching to ensure security and reliability.

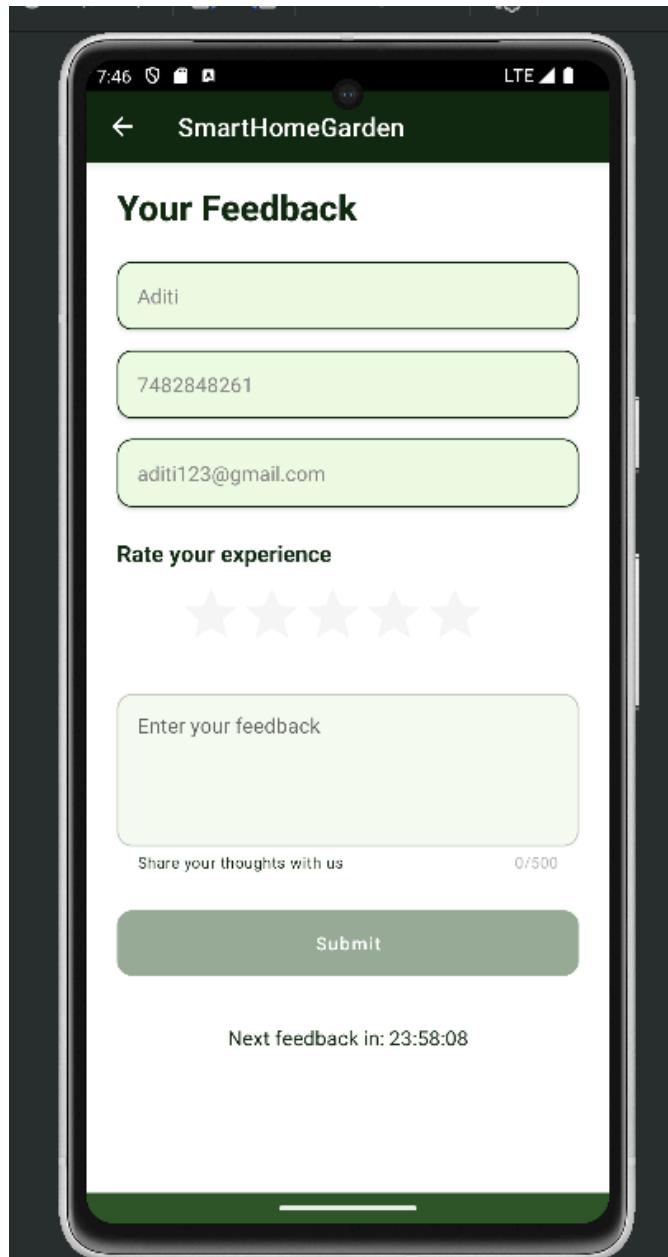
Integrated Google Sign-In and redesigned the registration UI for a better user experience. Added regex for phone and password validation, along with a toggle for password visibility. Implemented a 'Remember Me' checkbox for streamlined login sessions.

Feedback Screen

Progress Bar:



Feedback screen greyed out after the successful feedback.

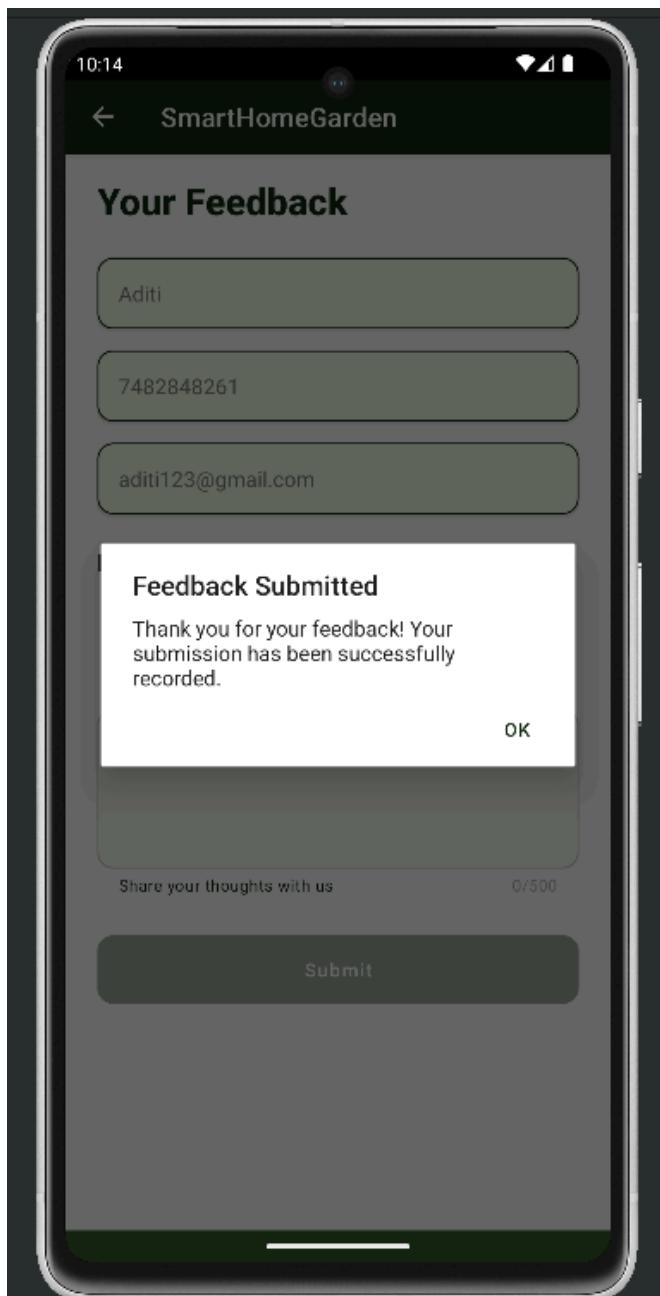


How you meet this requirements:

After the user submits feedback, the `submitFeedback` method disables the input fields and the submit button by calling `setFieldsEnabled(false)`, which grays out the fields and reduces their opacity. It then stores the submission time in `SharedPreferences` to track when the user can submit feedback again. A countdown timer is displayed using `startCountdownTimer`, which calculates the remaining time until the next submission and

updates the `timerTextView` every second. Once the cooldown period ends, the fields are re-enabled and the timer is hidden, allowing the user to submit feedback again.

Alert Dialog box:



Senor data:

The screenshot shows the Firebase Realtime Database console. On the left, there's a sidebar with project settings like Generative AI, Build with Gemini, Project shortcuts, Authentication, Firestore Database, and the selected Realtime Database. The main area is titled "Realtime Database" with tabs for Data, Rules, Backups, Usage, and Extensions. A message at the top says "Need help with Realtime Database? Ask Gemini". Below it, a warning message states "Your security rules are not secure. Any authenticated user can steal, modify or delete data in your database." with links to Learn more and Dismiss. The data view shows a hierarchical structure: notifications, sensors, moisture: 4.9, sunlight: 89.97, and temperature: 27. At the bottom, it says "Database location: United States (us-central1)".

Feedback:

The screenshot shows the Firestore console. On the left, there's a sidebar with collections: (default), feedbacks, and 9YSVtKhEtivFxihc7Ydv. Under feedbacks, there's a sub-collection named users. The main area shows a list of document IDs: 9YSVtKhEtivFxihc7Ydv, HXzbuGUxq3eyKXJBpELr, OpG99RH8QeCU15x6252, PP01U7oX21n5Q1kspJ3K, V1BleI4w8BYMc132wIn3, XyrkREX2k0D0leN1Ccer, dxFNWLtfR10Ejg50tc34, drZqyU1z9t5yXd0jW5hz, gKi6o1riQBbaUqsSUWQtm, gLxtKF3uREsNlxuMJ0qn, 1xw1o9k0YetpHX1PZzwX, phrAMNUsvSD9cmnERDOh, spfXJWBGwRbTGPhPew2L, and ur40XRRtXtnEeC17bAjx. To the right, a detailed view of the document 9YSVtKhEtivFxihc7Ydv is shown, containing fields: description: "Great", deviceModel: "sdk_gphone64_x86_64", email: "aditi123@gmail.com", name: "Aditi", phone: "7482848261", and rating: 4.5.

 (default)	 feedbacks		 HXzbuGUxq3eyKXJBpELr	
+ Start collection	+ Add document		+ Start collection	
feedbacks >	9YSVtKhEtivFxihc7Ydv		+ Add field	
users	HXzbuGUxq3eyKXJBpELr >			
	OpG99RHM8QeCU15x6252		description: "Very nice Application. Good luck team."	
	PP01U7oX21n5Q1kspJ3K		deviceModel: "sdk_gphone64_x86_64"	
	V1BleI4w8BYMc132wIn3		email: "birava@123.com"	
	XyrkREX2k0D0leNlCcer		name: "Birava"	
	dXFNWLtfR10Ejg50tc34		phone: "123456789"	
	drZqyU1z9t5yXdQjW5hz		rating: 5	
	gKi6o1riQBbaUqSUWQtm			
	gLxtKF3uREsN1xuMJ0qn			
	1xw1o9k0YetpHX1PZzwX			
	phrAMNUsvSD9cmnERD0h			
	spfXJWBGwRbTGPhPew2L			
	ur40XRRtXtnEeC17bAjx			

 (default)	 feedbacks		 XyrkREX2k0D0leNlCcer	
+ Start collection	+ Add document		+ Start collection	
feedbacks >	9YSVtKhEtivFxihc7Ydv		+ Add field	
users	HXzbuGUxq3eyKXJBpELr			
	OpG99RHM8QeCU15x6252		description: "App is good!"	
	PP01U7oX21n5Q1kspJ3K		deviceModel: "sdk_gphone64_arm64"	
	V1BleI4w8BYMc132wIn3		email: "aaa@ddd.com"	
	XyrkREX2k0D0leNlCcer >		name: "Darshan"	
	dXFNWLtfR10Ejg50tc34		phone: "2352465737"	
	drZqyU1z9t5yXdQjW5hz		rating: 0	
	gKi6o1riQBbaUqSUWQtm			
	gLxtKF3uREsN1xuMJ0qn			
	1xw1o9k0YetpHX1PZzwX			
	phrAMNUsvSD9cmnERD0h			
	spfXJWBGwRbTGPhPew2L			
	ur40XRRtXtnEeC17bAjx			