

# Deliverable 3

**Team Name:** FurGuardian

**Project Name:** Pet Wellness App

**Team Group:** 8

**Student Names and ID's:**

Justin Chipman – N01598472

Imran Zafurallah - N01585098

Zane Aransevia - N01351168

Tevadi Brookes - N01582563

Name	Student Id	Github Id	Signature	Effort
Justin Chipman	N01598472	Chipman8472	Justin C	100%
Imran Zafurallah	N01585098	ImranZafurallah5098	Imran Z	100%
Zane Aransevia	N01351168	Zayne917	Zane A	100%
Tevadi Brookes	N01582563	TevadiBrookes2563	Tevadi B	100%

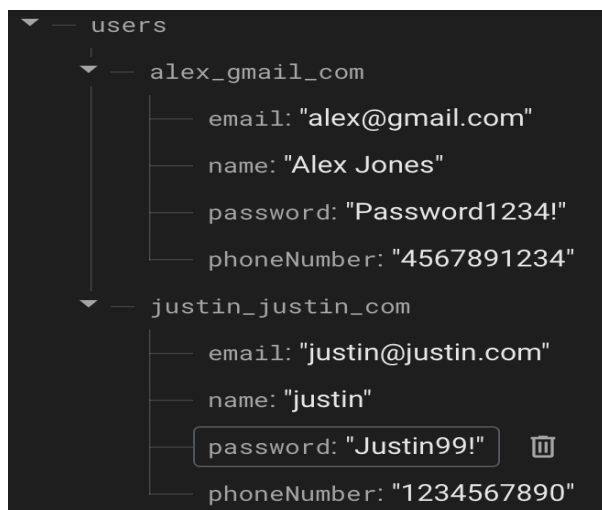
### **Project Scope**

The technical scope of the FurGuardian project encompasses both the development of the

Android app and the integration of hardware components. This involves creating a fully functional and user-friendly application that connects to various hardware sensors, processes pet data, and offers real-time monitoring and interaction.

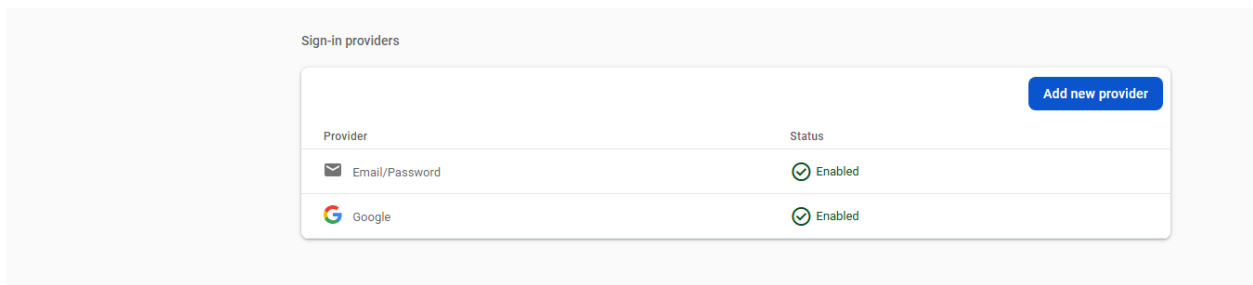
**GitHub Link:** <https://github.com/Chipman8472/FurGuardian.git>

### **Account Creation in Database**



```
▼ — users
  ▼ — aaa_bbb_com
    email: "aaa@bbb.com"
    name: "Haki"
    password: "Admin101!"
    phoneNumber: "1234567890"
```

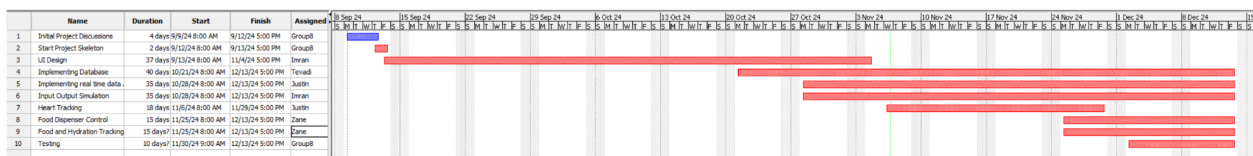
## Screenshot showing Users Gmail



## Screenshot showing how feedback is stored in Database

```
▼ — -0AygA6oy16zgL5rcyd6
  comment: "Wooww!"
  deviceModel: "sdk_gphone64_arm64"
  email: "tev@bro.com"
  name: "Tevadi Brookes"
  phone: "6159157045"
  rating: 5
  feedback
    comment: "Perfection"
    deviceModel: "Google sdk_gphone64_arm64"
    email: "sen.iii@gmail.com"
    name: "Tevadi Brookes"
```

## Gantt Chart



## Monday Scrum Board

<

## Sprint 3

**Tevadi Brookes** for sprint 3 he has developed a **Customer Feedback Screen** for the app to enable users to submit feedback directly within the app. The design includes input fields for the user's name, phone number, email, comments, and a RatingBar for satisfaction ratings. Tevadi ensured that all essential fields must be completed before submission, using input validation to improve data quality and user experience. The device model is captured programmatically for backend insights, though it is hidden from the user's view on the screen.

For data handling, he integrated Firebase Realtime Database, with all feedback saved to a dedicated "feedback" collection. This includes the user's personal details, comments, rating, and device model, stored securely for future analysis. The feedback submission process is user-friendly, providing confirmation messages upon successful submissions and error alerts if issues arise. This new feature not only enhances customer engagement within the FurGuardian app but also provides valuable feedback data for continuous improvement of the service.

**Justin Chipman** developed the **Login and Registration System** for the app, focused on providing users with a seamless and secure login experience. In addition to traditional account creation, he incorporated Google authentication, allowing users to log in with Gmail without needing a separate account within the app. The login screen also includes a "Remember Me" checkbox, which allows users to stay signed in across sessions without the need to re-enter credentials, while securely managing user sessions without storing sensitive credentials locally on the device. The login screen has also been integrated with a link directing users to the registration screen for easy access.

The **Registration Screen** ensures secure and validated user data collection, featuring fields for the user's name, phone number, email, password, and password confirmation. Each input field includes appropriate validation to guarantee data integrity and usability; for example, phone numbers accept only numeric input, and passwords must be confirmed to prevent entry errors. These enhancements provide a robust foundation for user account management, elevating both security and ease of use in the app.

**Imran Zafurallah** enhanced the **Settings and Emergency Contacts** features of the app. He implemented functionality for users to add and manage emergency contacts directly within the app, improving the overall safety features for pet owners. Additionally, Imran developed functionality to allow users to update key settings, such as changing the location within the app. This adds flexibility for users for any veterinarians in their proximity, as well as to personalize their experience and ensures the app meets individual preferences.

In addition to these technical improvements, Imran took responsibility for editing and refining the **Gantt Chart**, ensuring that it accurately reflects large project tasks and milestones. He also updated the Monday dashboard by adding new tasks, marking completed tasks, and ensuring the project's progress is tracked efficiently. However, to maintain control over task assignments and due to membership limitations, Imran opted to limit the number of individuals with access to the Monday dashboard to just two members. This ensures streamlined communication, with the two members responsible for managing tasks and informing the remaining group members of their duties.

In addition, Zane developed a system for adding and **managing pet profiles** within the app. He included features to upload custom images and input the pet's breed and type of animal. Zane ensured that users could easily switch between different pets on the home screen, allowing for a more personalized experience. The pet profiles will persist even after the app is closed, ensuring seamless usage.

	Implement Menu	4		Done	Get-8	High	! Oct 1 - 8
	Subitem		Owner	Status	Date	+	
	Pet profile database	+		Done			
	Reminders	+		Done			
	Account database	+		Done			
	Settings	+		Done			
	+ Add subitem						

	Implement Customer Feedback			Done			✓ -
	Subitem		Owner	Status	Date	+	
	+ Add subitem						

	Log In Screen	4		Done	✓ Get-8	High	Action items	✓ Oct 1 - 8
	Subitem		Owner	Status	Date	+		
	connection to firebase	+		Done				
	creation of user logins	+		Done				
	UI design	+		Done				
	snackbar notifications	+		Done				
	+ Add subitem							

## Daily Standup

### Sprint 3

November 3<sup>rd</sup>

<b>Justin Chipman</b>	<b>Imran Zafurallah</b>	<b>Zane Aransevia</b>	<b>Tevadi Brookes</b>
<b>What did you do yesterday?</b>	<b>What did you do yesterday?</b>	<b>What did you do yesterday?</b>	<b>What did you do yesterday?</b>
Started developing the login and registration screens, focusing on UI layout and basic field validation.	Went over what I was assigned to do in Sprint 3.	Started designing the health and home screen UI with no implementations.	Work on the initial design to the feedback page
<b>What will you work on today?</b>	<b>What will you work on today?</b>	<b>What will you work on today?</b>	<b>What will you work on today?</b>
Implementation of login and registration screens.	Finished implementing SharedPreferences in the settings screen for Dark Theme preference.	Polish all screen UIs to ensure they meet production standards and address any remaining layout issues for consistency.	Finished implementing SharedPreferences in the settings screen for Dark Theme preference.
<b>Any Blockers?</b>	<b>Any Blockers?</b>	<b>Any Blockers?</b>	<b>Any Blockers?</b>
Not at this time	Not at this time	Not at this time	Not at this time

November 4<sup>th</sup>

<b>Justin Chipman</b>	<b>Imran Zafurallah</b>	<b>Zane Aransevia</b>	<b>Tevadi Brookes</b>
<b>What did you do yesterday?</b>	<b>What did you do yesterday?</b>	<b>What did you do yesterday?</b>	<b>What did you do yesterday?</b>
Login and registration screens.	Finished implementing SharedPreferences in the settings screen for Dark Theme preference.	UI designed and fixed previous issues.	Design of the feedback screen.
<b>What will you work on today?</b>	<b>What will you work on today?</b>	<b>What will you work on today?</b>	<b>What will you work on today?</b>
Add Gmail sign-in functionality, implement the "Remember Me" feature, and validate all input fields, especially email and phone number formats.	Begin working on other functionalities in the settings fragment and Emergency Contacts in the top menu.	Implemented writing to Firestore for sensor data and feedback data, including device model information from the feedback screen.	Work on the feedback screen, adding fields for name, phone, email, and comment, along with star rating functionality. Ensure device model is saved to Firestore.
<b>Any Blockers?</b>	<b>Any Blockers?</b>	<b>Any Blockers?</b>	<b>Any Blockers?</b>
Facing some challenges integrating Gmail sign-in and ensuring compatibility across devices.	None to speak of	Not at this time	Not Currently



**November 5<sup>th</sup>**

<b>Justin Chipman</b>	<b>Imran Zafurallah</b>	<b>Zane Aransevia</b>	<b>Tevadi Brookes</b>
<b>What did you do yesterday?</b>	<b>What did you do yesterday?</b>	<b>What did you do yesterday?</b>	<b>What did you do yesterday?</b>
Added Gmail sign-in functionality, implemented the "Remember Me" feature.	Begin working on my task in Settings functionality and Emergency Contacts	Implemented writing to Firestore for sensor data and Pet Profile.	Worked on the functionality of the feedback screen.
<b>What will you work on today?</b>	<b>What will you work on today?</b>	<b>What will you work on today?</b>	<b>What will you work on today?</b>
Complete the runtime permission implementation for accessing device sensors, ensuring permissions are requested when functionality is needed.	Completed settings screen functionality, including saving user-selected themes to SharedPreferences and retaining preferences across sessions.	Finished functionality for Pet profile and health fragment, simulating the proper readings.	Implemented writing to Firestore for feedback data, including device model information from the feedback screen.
<b>Any Blockers?</b>	<b>Any Blockers?</b>	<b>Any Blockers?</b>	<b>Any Blockers?</b>
Some issues with sensor permission handling	Some issues with data being stored in shared pref.	Not Currently	Not at this time

## Two different design principles used in the code

**Single Responsibility Principle (SRP):** Each class has a single responsibility and does not overlap with another class unless to share resources. For instance in our code, the **Splash Screen Activity** is dedicated solely to managing splash screen functionality such as displaying introductory visuals or initiating app loading procedures without additional logic related to user authentication or other app components. Similarly, the **Login Screen** is responsible only for user authentication, maintaining clean boundaries between its purpose and other app features.

**Don't Repeat Yourself (DRY) Principle:** The DRY principle ensures that code is concise, modular, and free of redundancy by promoting reusable methods and components. In FurGuardian, we adhered to the DRY principle by centralizing commonly used features, such as Firebase database interactions and form validation methods. For instance, reusable methods were created for user input validation across the registration and **Feedback Screen**, minimizing duplicate code and making maintenance more efficient. This approach not only reduces errors but also simplifies future updates, as changes to a single method apply universally across the app.

**Open/Closed Principle (OCP)** Each class is open for extensions but close for all modifications.

## Coding work progress since deliverable 2

Since Deliverable 2, we have made several enhancements to improve the functionality and user experience of the application. A **Customer Feedback Screen** has been implemented and integrated with Firebase Realtime Database to securely capture and store user feedback. The **Login Screen** has been enhanced to support Google authentication, allowing Gmail users a seamless login experience. Additionally, an option has been added to enable users to invite others to their account, granting them shared access to pet data.

Further updates include expanded functionality within the **Settings Fragment**, offering users more control over their app preferences. A **Pet Profile** feature has also been added, allowing users to manage profiles for multiple pets efficiently. These upgrades collectively enhance the app's usability, accessibility, and personalization for users managing multiple pets and sharing information with trusted individuals.

## Document what runtime permission you have implemented

```
<uses-permission android:name="android.permission.POST_NOTIFICATIONS" />
```

## Runtime permission

```
private void saveReminder() {
    String reminderName = reminderNameEditText.getText().toString();
    if (reminderName.isEmpty()) {
        Toast.makeText(getApplicationContext(),
getString(R.string.please_enter_a_reminder_name), Toast.LENGTH_SHORT).show();
        return;
    }

    // Check notification permission
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.TIRAMISU) {
        if (ActivityCompat.checkSelfPermission(requireContext(),
Manifest.permission.POST_NOTIFICATIONS) != PackageManager.PERMISSION_GRANTED) {
            requestNotificationPermission();
            return;
        }
    }
}
```

Document the two main functionality of your app that was implemented in this release.

## Home Screen Functionality



## Pet Profile Functionality

