

Deliverable 1

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

Contents

Team Contract	1
Project Background and Description	4
Software Aspect	5
Screen Flow	5
Post Interview Feedback Implementation	6
Cloud Database READ/WRITE Plan	6
Project Scope	7
Integrate Software and Hardware.....	8
Project Theme Structure.....	9
Screenshots.....	11

Team Contract

FurGuardian

[Company Address]

[City, ST ZIP Code]

Please negotiate, sign, scan and include as the first section in your Deliverable 1.

Please note that if cheating is discovered in a group assignment each member will be charged with a cheating offense regardless of their involvement in the offense. Each member will receive the appropriate sanction based on their individual academic honesty history.

Please ensure that you understand the importance of academic honesty. Each member of the group is responsible to ensure the academic integrity of all of the submitted work, not just their own part. Placing your name on a submission indicates that you take responsibility for its content.

For further information read Academic Honesty Policy on <https://humber.ca/legal-and-risk-management/policies/search-by-students.html>.

Team Member Names (Please Print)	Signatures	Student ID	Github Id
Project Leader: Justin Chipman	<i>Justin C</i>	N01598472	Chipman8472
Zane Aransevia	<i>Zane A</i>	N01351168	Zayine917
Tevadi Brookes	<i>Tevadi B</i>	N01582563	TevadiBrookes2563
Imran Zafurallah	<i>Imran Z</i>	N01585098	ImranZafurallah5098

By signing this contract, we acknowledge having read the Humber Academic Honesty Policy as per the link below.

<https://academic-regulations.humber.ca/2018-2019/17.0-ACADEMIC-MISCONDUCT>

Responsibilities of the Project Leader include:

- Assigning tasks to other team members, including self, in a fair and equitable manner.
- Ensuring work is completed with accuracy, completeness and timeliness.
- Planning for task completion to ensure timelines are met
- Any other duties as deemed necessary for project completion

What we will do if . . .

Scenario	Accepted initials	We agree to do the following
Team member does not deliver component on time due to severe illness or extreme personal problem	JC TB ZA IZ	a) Team absorbs workload temporarily

Scenario	Accepted initials	We agree to do the following
Team member cannot deliver component on time due to lack of ability	JC TB ZA IZ	b) Team helps member
Team member does not deliver component on time due to lack of effort	JC TB ZA IZ	b) Team "fires" team member by not permitting his/her name on submission
Team member does not attend team meeting	JC TB ZA IZ	a) Team proceeds without him/her and will assign work to the absent member
An unforeseen constraint occurs after the deliverable has been allocated and scheduled (a surprise test or assignment)	JC TB ZA IZ	a) Team meets and reschedules deliverable
Team cannot achieve consensus leaving one member feeling "railroaded", "ignored", or "frustrated" with a decision which affects all parties	JC TB ZA IZ	a) Team agrees to abide by majority vote

Scenario	Accepted initials	We agree to do the following
Team members do not share expectations for grade desired	JC TB ZA IZ	a) Team will elect one person as "standards-bearer" who has the right to ask that work be redone. "Justin Chipman"
Team member behaves in an unprofessional manner by being rude or uncooperative	JC TB ZA IZ	c) Team agrees to avoid use of all vocabulary inappropriate to the business setting
Team member assumes or requests that his/her name be signed to a submission but has not participated in production of the deliverable	JC TB ZA IZ	a) Team agrees that this is cheating and is unethical
There is a dominant team member who is content to make all decisions on the team's behalf leaving some team members feeling like subordinates rather than equal members	JC TB ZA IZ	b) Team will express subordination feelings and attempt to resolve issue
Team has a member who refuses to participate in decision making but complains to others that s/he wasn't consulted	JC TB ZA IZ	c) Team discusses the matter at team meeting

Project Background and Description

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

The goal of FurGuardian is to create a comprehensive pet wellness app that offers pet owners an all-in-one solution for managing their pets' health, feeding, and activity levels, both in real-time and remotely. The final vision includes an intuitive and feature-rich Android app that provides:

Health monitoring

Medical record management

Remote feeding and monitoring

Wellness alerts

Software Aspect

The Android app will be developed using Java with Firebase as the backend database for real-time data storage and retrieval. It will interact with various hardware components via Bluetooth or Wi-Fi.

Sensors: Human Detection IR sensor – used to make sure only pets are recognized by the device

DC motor – used to dispense the pet food

RFID sensor – used to detect and differentiate between multiple pets.

Liquid Level Sensor – used to detect if the pets water level is low.

Heart rate sensor – used to detect the pets heart rate.

Scale – to weight food.

Relay – used to dispense treats.

Camera + microphone – for interacting with pets.

Screen Flow

Our app is designed with a streamlined bottom navigation layout for seamless user interaction. Upon launch, users are greeted by a dynamic splash screen, which transitions into the home screen that streams live video and audio directly from a pet's environment. This feature allows users to monitor and interact with their pets in real-time, fostering a stronger bond even from a distance.

The health monitoring screen offers comprehensive insights into your pet's well-being. Key metrics such as heart rate, hydration, and activity levels are displayed in an intuitive, user-friendly interface, empowering owners to keep their pets in peak condition.

Our feeding schedule feature allows pet owners to remotely manage feeding times and dispense food with just a tap. Users can also monitor water levels, ensuring their pets always have access to fresh water. The medical records screen provides quick access to your pet's complete medical history, including vaccination schedules and essential health information, ensuring that vital data is always at your fingertips.

Additionally, the insights screen offers valuable educational content about pet care and nutrition, helping users enhance their understanding of pet wellness and make informed decisions.

Post Interview Feedback Implementation

Feature Expansion: The team added wellness alerts for feeding and health check-ins based on health data, expanding the scope of monitoring.

Improved Usability: Suggestions to enhance the user experience were implemented, such as making the health tracking screen more intuitive and adding easy access to remote feeding controls.

Hardware Integration: Input regarding the RFID sensor for tracking multiple pets and the importance of a water monitoring system was considered to ensure robust functionality.

Cloud Database READ/WRITE Plan

To satisfy the need for reading and writing data to/from the cloud:

Firebase will be used for real-time data storage. Each pet's health data (heart rate, hydration, etc.), feeding history, and wellness metrics will be stored in Firebase Firestore.

The app will write pet data (e.g., weight, feeding time, health records) as the sensors collect the data and store it under a unique pet profile in the cloud.

The app will read this data on the owner's device, presenting it in charts and notifications, giving users real-time insights into their pets' wellness.

Authentication will be handled using Firebase Authentication to ensure secure access to the data.

Data retrieval will be optimized to support offline functionality in case of network issues, allowing users to view the most recent data until a connection is re-established.

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.

Week 1-2: Requirements Gathering & Research

Define app features and hardware components.

Research compatible sensors to ensure alignment with project goals.

Week 3-4: UI/UX Design

Create wireframes and mockups.

Plan the user journey for smooth navigation.

Week 5-6: Backend & Database Setup

Set up Firebase Firestore for data storage.

Configure user authentication and cloud access.

Week 7-8: Hardware Integration

Integrate sensors, motor, camera, and microphone with the app.

Test data transmission between hardware and the app.

Week 9-10: Core Features Development

Implement health tracking, remote feeding, video monitoring, and notifications.

Ensure real-time data syncing with the cloud.

Week 11-12: Testing & Debugging

Conduct unit and integration testing for the app.

Test hardware operation with the app.

Week 13-14: User Testing & Feedback

Test the app with pet owners.

Refine features and UI based on feedback.

Week 15: Final Deployment

Complete the app for submission to the Google Play Store.

Prepare for the app to go live.

Integrate Software and Hardware

The FurGuardian app integrates with hardware components like RFID sensors, DC motors, and heart rate sensors using Bluetooth and Wi-Fi for communication. A microcontroller, such as Raspberry Pi or Arduino, serves as the intermediary, collecting data from sensors and sending it to the app, while commands from the app trigger hardware actions, such as dispensing food. This seamless connection ensures that users can manage their pets' wellness remotely, including live video monitoring and health tracking.

The app processes real-time sensor data, such as heart rate or water levels, and displays it for the user, sending alerts when needed. The app-to-hardware communication allows users to interact with pets remotely, with commands like feeding or monitoring live streams being sent to the microcontroller, which triggers hardware responses. Feedback from the hardware is sent back to confirm actions, ensuring reliability and accuracy.

To ensure smooth functionality, the team will test hardware components individually, debug communication protocols, and check the app's real-time data processing. This approach guarantees reliable integration between the app and hardware, providing a comprehensive solution for pet wellness.

Project Theme Structure

Theme: Develop the "FurGuardian" Pet Health Monitoring & Remote Feeding App









- **Epic 1: Heart Rate Monitoring and Activity Tracking**
 - **Story 1:** Design and implement heart rate tracking UI
 - Task 1: Create UI components for displaying heart rate data in real-time.
 - Task 2: Add integration points to receive simulated heart rate data.
 - Task 3: Implement logic for refreshing heart rate display at regular intervals.
 - **Story 2:** Develop heart rate data processing logic
 - Task 1: Simulate and generate heart rate data.
 - Task 2: Implement algorithms to interpret activity levels based on heart rate fluctuations.
 - Task 3: Store the data in Firebase for historical tracking.
 - **Story 3:** Create activity insights for pet owners
 - Task 1: Analyze stored heart rate data to detect periods of high/low activity.
 - Task 2: Provide insights on the pet's activity trends over time (daily, weekly).
 - Task 3: Design an insights dashboard for owners to easily view activity levels.
- **Epic 2: Food & Hydration Monitoring and Remote Feeding**
 - **Story 1:** Create UI for food and hydration tracking
 - Task 1: Design food and water level indicators within the app.
 - Task 2: Integrate with simulated food dispenser data for real-time tracking.
 - Task 3: Implement thresholds to alert owners when food/water levels are low.
 - **Story 2:** Implement food dispenser control (remote/scheduled)
 - Task 1: Set up UI controls to allow remote food dispensing.

- Task 2: Add logic for scheduling automated feedings.
 - Task 3: Simulate food dispensing events and monitor their success in real-time.
- **Story 3:** Store and retrieve food/hydration data from Firebase
 - Task 1: Implement Firebase structure for storing food and hydration data.
 - Task 2: Write Java code to sync the dispenser data with Firebase.
 - Task 3: Implement retrieval of historical data to show trends over time.
- **Epic 3: Health Insights and Medical Records Storage**
 - **Story 1:** Create Firebase structure for storing medical records
 - Task 1: Design a Firebase schema for holding pet medical records.
 - Task 2: Write Java logic to add, update, and delete records in Firebase.
 - Task 3: Implement security measures for medical record access (authentication).
 - **Story 2:** Implement health insights algorithm
 - Task 1: Create logic to combine heart rate, activity, food, and hydration data into health insights.
 - Task 2: Develop algorithms that highlight significant health changes (e.g., dehydration, lethargy).
 - Task 3: Display health recommendations and alerts based on insights.
 - **Story 3:** Design health insights UI
 - Task 1: Design a user-friendly health insights dashboard.
 - Task 2: Integrate real-time and historical health insights into the dashboard.
 - Task 3: Allow owners to receive notifications for abnormal health events (e.g., low activity).


Screenshots











Manage access


[Add people](#)

<input type="checkbox"/> Select all	Type ▾
<input type="text" value="Find a collaborator..."/>	
<input type="checkbox"/>  Hak11 haki11 • Collaborator	
<input type="checkbox"/>  ImranZafurallah5098 Collaborator	
<input type="checkbox"/>  TevadiBrookes2563 Collaborator	
<input type="checkbox"/>  ZaneAransevia-n01351168 Zayne917 • Collaborator	

Account has been made and paid. Just waiting on verification.

 **Google Play Console**

-  Home
-  Inbox
-  Policy status
-  Users and permissions
-  Order management
-  Download reports
 - Reviews
 - Statistics
 - Financial
-  Developer account
-  Associated developer accounts
-  Activity log
-  Settings

 **FurGuardian**
Personal account • Account ID: 6019219565316114513

Finish setting up your developer account

To publish apps on Google Play, finish setting up your developer account. This helps Google to verify your identity, protect your account, and keep Google Play users safe.

- **Verify your identity**
Upload an official document so that Google can verify your identity. Do not edit or manipulate documents as this will cause verification to fail, which can affect the availability of your apps. The verification process may take a few days. [Get started](#)
- **Verify that you have access to an Android mobile device**
Sign in to the Play Console mobile app to verify that you have access to a real Android mobile device. [View details](#)