Final Report

Product Name: PetSentry

Date: 29.12.2023

Team Number: 2

Team Members:

Student ID	Name and Surname	E-Mail
21901461	Alkım Ege Akarsu	ege.akarsu@ug.bilkent.edu.tr
21901873	Doğa Diren	doga.diren@ug.bilkent.edu.tr
21801616	Muhammet Oğuzhan Gültekin	gltkn.oguz@gmail.com
21902836	Vefa Övünç Özer	ovunc.ozer@ug.bilkent.edu.tr

1. Executive Summary

This project aims to create a product that can assist cat and dog owners in taking care of their pets while they are not home. Many owners feel the need to check what their pets are up to when they are away. Furthermore, the food and water supplies of the pet may need to be refilled if the owner is unable to come home for prolonged periods.

With PetSentry, we have developed a solution for these and some other challenges faced by pet owners. Utilizing off-the-shelf components, cloud services, and an Android app, we have created a fully remote pet care system. Using PetSentry, through the Android app, the user can watch their pet in real-time, fill their food and water bowls with one tap, and optionally, configure the product so that the food and water bowls automatically refill to the desired amount. Additionally, the user has the option to view past videos of their pet and listen to the recordings that have been automatically created when PetSentry has detected loud noises.

To conclude, we have achieved our goal of creating a product that can provide convenience and peace of mind to pet owners who are away from home. PetSentry is a solution that combines hardware, software, and cloud services to offer a comprehensive and user-friendly pet care system. We believe that PetSentry has the potential to significantly improve the quality of life of both pets and their owners.

2. Functional Specifications - Summary [1]

2.1. Mobile app

2.1.1. Compatible with Android OS [2].

2.1.2. Bluetooth [3] and Wi-Fi [4] connections for data transfer.

2.1.3. User authentication via email and password. Multiple users can access the same physical product.

2.1.4. View the live stream from the physical product.

2.1.5. Send commands (fill water and fill food), set food weight, and set operation mode (automatic or manual).

2.1.6. View sensor data (water level and food weight).

2.1.7. Push notifications when a new event (loud noise, low water level, etc.) occurs.

2.1.8. View all past events and listen to past loud noise events.

2.2. Physical product

2.2.1. Read mass and water level data.

2.2.1.1. Read water level by using an ultrasonic sensor.

2.2.1.2. Read the mass of the bowl by using a weight sensor.

2.2.2. Feed water or food according to the command taken from the user.

2.2.2.1. Feed water via the water pump.

2.2.2.2. Feed food by using the helical screw conveyor that is turned with a DC motor.

2.2.3. Capture video via the Raspberry Pi camera.

2.2.4. Obtain sound signals via microphone.

3. Design - Summary [5]

3.1. Hardware Design

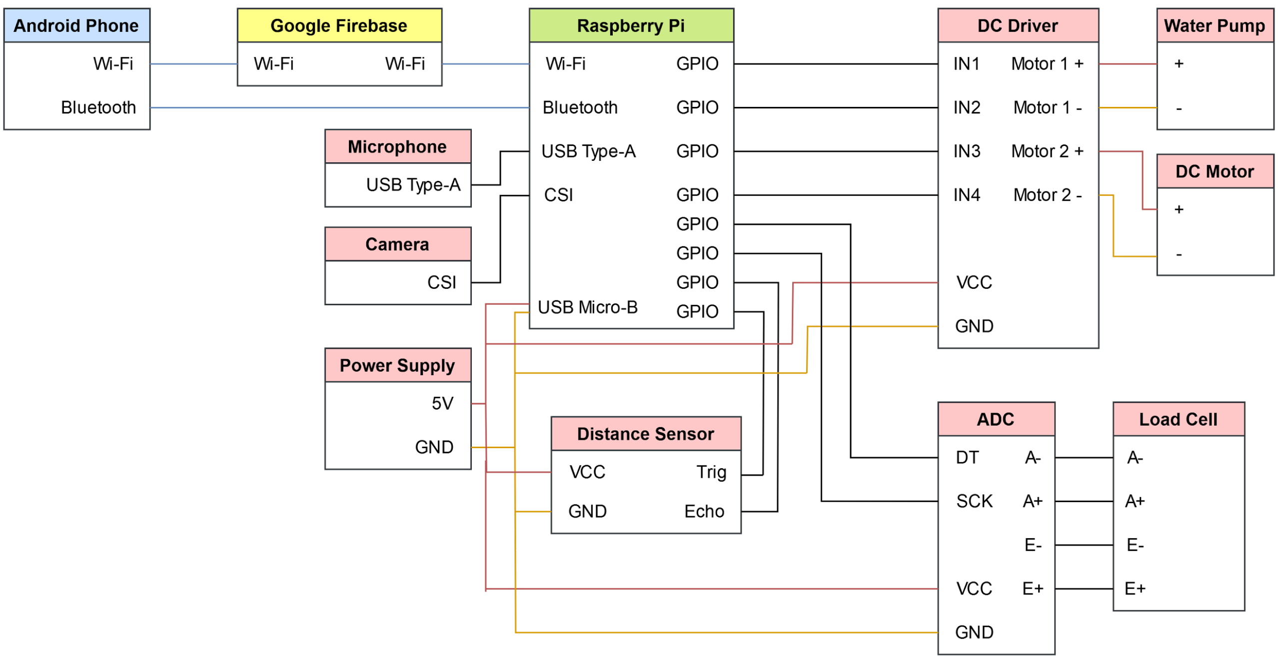

Figure 1: System hardware block diagram.

3.1.1. **Android Phone:** Runs the PetSentry app.

3.1.2. **Google Firebase:** Provides authentication, database, and storage services. [6]

3.1.3. **Raspberry Pi:** Runs the necessary local code and interfaces with connected components. [7]

3.1.3.1. **Power Supply:** Supplies the power needed to run the physical components of the system.

3.1.3.2. **Microphone:** Captures the environment sound for high noise detection.

3.1.3.3. **Camera:** Captures video for the live stream.

3.1.3.4. **Distance Sensor:** Reads the water level distance in the water bowl. [8]

3.1.3.5. **DC Driver:** Supply the voltage to the DC water when needed. [9]

3.1.3.6. **Water Pump:** Transfers water from a water container into the water bowl.

3.1.3.7. **DC Motor:** Turns the helical screw conveyor to feed food to the food bowl.

3.1.3.8. **ADC:** Converts analog signal from load cell to a digital signal. [10]

3.1.3.9. **Load Cell:** Reads mass of food in the bowl.

3.2. Software Design

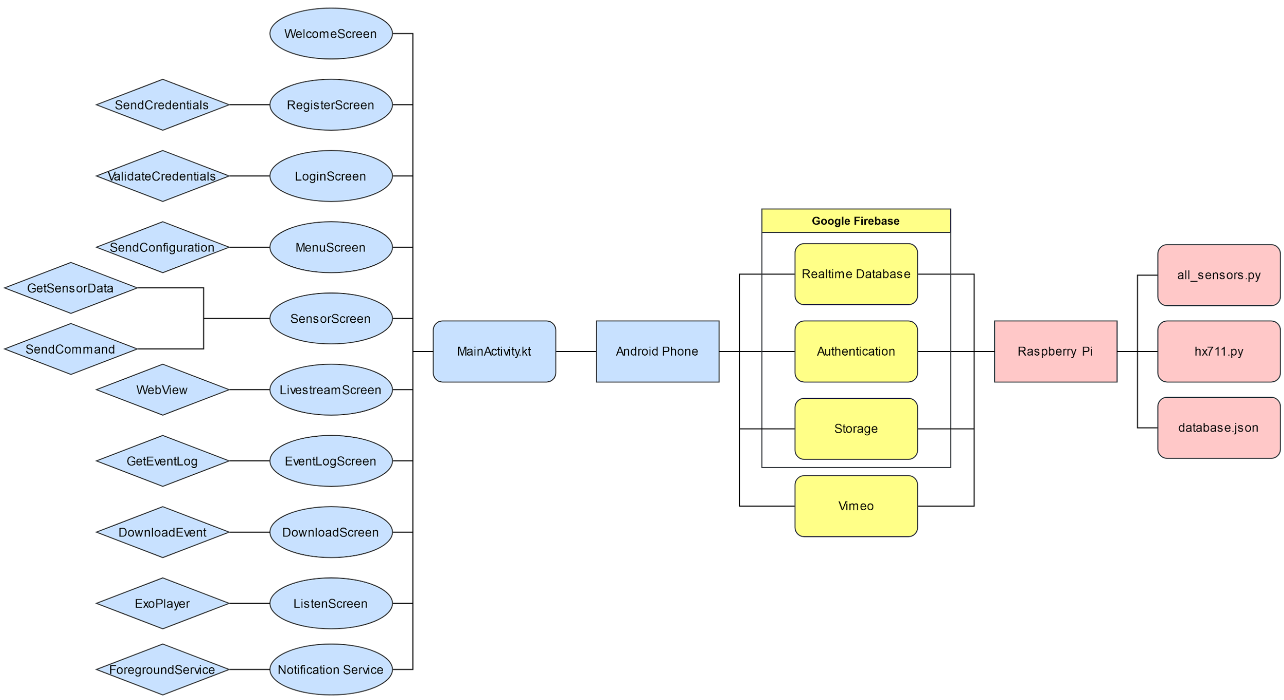

Figure 2: System software block diagram.

3.2.1. **Android Phone:** PetSentry app written in Kotlin [11]. UI uses the Jetpack Compose toolkit [12].

3.2.1.1. **User Authentication:** Firebase authentication is used in the “Register” and “Login“ screens.

3.2.1.2. **Sensor Data and Commands:** The Firebase real-time database is used to get sensor data and send commands.

3.2.1.3. **Livestream:** The Android system web view is used to display the Vimeo [13] Livestream page.

3.2.1.4. **Event Log:** Firebase real-time database is used to get the event log, Firebase storage is used to get the recordings, and ExoPlayer [14] is used to play the ".wav" files downloaded from the storage.

3.2.1.5. **Notifications:** A foreground service is used to listen to changes in the event log and send the most recent change as a notification.

3.2.2. **Google Firebase:**

3.2.2.1. **Realtime Database:** The database is used to get the data that are taken from the sensors implemented on Raspberry Pi.

3.2.2.2. **Authentication:** The Firebase database stores the user's ID and information.

3.2.2.3. **Storage:** The database stores the sound data that is captured by the microphone connected to Raspberry Pi to be used for high noise detection.

3.2.3. **Vimeo:** Vimeo is used to broadcast the live stream video captured by the Raspberry Pi camera. Live video is sent to the RTMP link provided by Vimeo, and the live stream is watched from the output link provided by Vimeo.

3.2.4. **Raspberry Pi:**

3.2.4.1. **Load Cell Calibration:** Set up the load cell unit with precise specifications for optimal functionality

3.2.4.2. **Firebase Access:** Authorize access to the database and create a direct, stable link for data exchange.

3.2.4.3. **Sensors, Actuators, and Firebase:** Ensure coordinated working of all components (camera, microphone, ultrasonic sensor, load cell, water pump, and DC motor), control their activities, and collect data from each and send them to Firebase.

4. Components and Cost

Description	Cost (in TRY)
Raspberry Pi 3 Model B+ 4GB RAM	1,800 - 2,000
L298N DC Motor Driver Board	60 - 100
Submersible Water Pump and Tube	50 - 80
DC Motor With Plastic Reduction Gears	30 - 50
HC-SR04 Ultrasonic Distance Sensor	30 - 50
Raspberry Pi Camera Module	350 - 400
USB Microphone	80 - 100
Load Cell	80 - 100
Total	**2,480 - 2,880**

Table 1: Components used and their respective costs.

5. Task Distribution

Alkım Ege Akarsu:

- Everything related to the Android app. Including but not limited to:
    - Firebase integration:
        - User authentication
            - Register and log in via email and password.
        - Realtime database
            - Get sensor data and event logs, and send commands.
        - Storage
            - Download ".wav" files of loud noise events.
    - Access to the live stream and previously recorded live streams from Vimeo.
    - Real-time notifications.
    - User interface and user experience design.
    - Bluetooth communications with Raspberry Pi.
- Raspberry Pi:
    - Bluetooth communications with Android app.
    - Run PetSentry-related code at startup (using the cron service).

Doğa Diren:

- Hardware implementation
    - Implementing a part of the hardware system such as the microphone.
- Sound processing
    - Writing the Python code to detect and record the high-volume noise.

Muhammet Oğuzhan Gültekin:

- Hardware implementation
    - Implementing the hardware parts such as the water pump and the weight sensor
- Live stream via Vimeo
    - Setting Raspberry Pi to send the live video taken by Raspberry camera to the RTMP link provided by Vimeo (live stream platform).
- Setting Firebase for data transfer and control of the system
    - Firebase is set to get data from Raspberry Pi and update the values in the Realtime Database console. Firebase is also set to send data to Raspberry Pi to be able to control the system.

Vefa Övünç Özer:

- Hardware Implementation
    - Designing and implementing the hardware design of the system and circuit design of the system and components; ultrasonic sensor, load cell, water pump, and DC motor.
- Raspberry Pi
    - Initiate the boot process and configure the system initialization settings of the Raspberry Pi.
    - Establish connections to peripheral devices and implement control through coded algorithms, ensuring synchronization and cohesive functionality.
    - Obtain and process audio data from a microphone for high-level noise detection
- Firebase Connection

- Implement real-time access to the Firebase for the transmission and retrieval of data

6. Professional and Ethical Issues

Further considerations must be made to transform this project into an actual product on the market. In this section, we analyze our project from the perspective of the use of external intellectual property (IP). In the development of the mobile app, many libraries and services were used. The IDE used for development (Android Studio) and the programming language (Kotlin) both have the Apache 2.0 license [15][16]. Apache 2.0 license permits commercial usage [17]. The Android SDK also permits commercial use [18]. The SDK contains most of the libraries used in the project including Jetpack Compose. Another used library is ExoPlayer 2. Just like the IDE and programming language, ExoPlayer 2 uses the Apache 2.0 license [19].

In the end product, all of the Python code was written by the group members. To accomplish the initial sound processing design, a pre-trained model was used to distinguish the bark noises from the ambient noise [20]. Yet with the later changes in the design, we decided to eliminate the machine learning part, hence rendering the outside sourcing of this model obsolete.

The use of Firebase in commercial applications is not restricted, however, the limited functionality of the free plan is highly restrictive [21]. Similar restrictions apply to Vimeo as well [22].

The overall product design was done by the group members yet the basic outside hardware components were utilized to accomplish this project. As we have also stated in the components section we have used Raspberry Pi 3 Model B+ 4GB RAM, L298N DC Motor Driver Board, Submersible Water Pump and Tube, DC Motor With Plastic Reduction Gears, HC-SR04 Ultrasonic Distance Sensor, Raspberry Pi Camera Module, USB Microphone and Load Cell to implement our project. We have not designed nor implemented any of these components and hence we would have to obtain permission from their rightful owners in any product. For the Raspberry Pi, the licensing fell under the Creative Commons Attribution-ShareAlike 4.0 International (CC BY-SA) license [23]. Yet by obtaining this license, the usage of the Raspberry Pi for any commercial launch would be granted. We have not obtained any such license for this project, but for any further use or product launch this part would be crucial. The same is also true for the Raspberry Pi Camera Module. Other than Raspberry Pi, the hardware components may differ according to the ones that are used and since many options are available on the market the proper licensing and the obtention of the proper permissions would be dependent on the utilized product itself.

7. Conclusion

All in all, we have concluded our project accomplishing everything we set out to do and the final product we have developed holds the same functionalities as our initial design. The changes between our initial design and our final product have no impact on the functional promises of the initial design. The two biggest adjustments from our initial plan lay in the processing of the sound and the provider of the cloud system. We ended up using Firebase and Vimeo instead of AWS because these services were easier to implement, free of charge, and more appropriate for our needs. This did not have an impact on the user experience and the overall product functionality. The second change that we incorporated into our final design was recording higher-than-usual ambient sounds instead of performing bark/meow detection using machine learning. This decision was made because we were unable to configure basic machine-learning libraries on our Raspberry Pi. The obtained accuracy using the Raspberry adaptable version of tensorflow, tensorflow lite, was inadequate. To ameliorate this situation, we reverted to the initial suggestion of Prof. Mehmet Alper Kutay in detecting the higher volume and recording any unusual noise when the high noise is detected. With this procedure, we get to notify the user accurately each time and provide additional information concerning the noise through the recordings and the mobile app to the user.

To develop and implement our project, we had to gain a deeper understanding of the cloud communication of different subsystems. This process introduced us to the utilization of AWS and Firebase. After completing this project we also have a better sense of use cases of IOT devices and how to integrate them into any further product that we may design. To collect further knowledge on these concepts, we have highly depended on the documents that were provided by our course instructor Prof. Mehmet Alper Kutay, and other open-source projects that preceded our project.

8. References

[1] A. E. Akarsu, D. Diren, M. O. Gültekin, and V. Ö. Özer, “Product Specifications”, PetSentry, Oct. 2023.

[2] “Android,” Android. https://www.android.com/ (accessed Dec. 29, 2023).

[3] “Learn About Bluetooth,” Bluetooth Technology Website, https://www.bluetooth.com/learn-about-bluetooth/ (accessed Nov. 16, 2023).

[4] “Wi-Fi Alliance,” Wi-Fi Alliance. https://www.wi-fi.org/ (accessed Dec. 29, 2023).

[5] A. E. Akarsu, D. Diren, M. O. Gültekin, and V. Ö. Özer, “High-Level Design,” PetSentry, Nov. 2023.

[6] “Firebase | Google’s Mobile and Web App Development Platform,” Firebase. https://firebase.google.com/ (accessed Dec. 29, 2023).

[7] “Raspberry Pi 3 Model B+,” Raspberry Pi, https://www.raspberrypi.com/products/raspberry-pi-3-model-b-plus/ (accessed Nov. 16, 2023).

[8] “Handson Technology User Guide HC-SR04 Ultrasonic Sensor Module,” Handson Technology. Accessed: Dec. 29, 2023. [Online]. Available: https://www.handsontec.com/dataspecs/HC-SR04-Ultrasonic.pdf

[9] “Handson Technology User Guide L298N Dual H-Bridge Motor Driver,” Handson Technology. Accessed: Dec. 29, 2023. [Online]. Available: http://www.handsontec.com/dataspecs/L298N%20Motor%20Driver.pdf

[10] “HX711 Datasheet,” SparkFun Electronics, https://cdn.sparkfun.com/datasheets/Sensors/ForceFlex/hx711_english.pdf (accessed Nov. 16, 2023).

[11] “Kotlin Programming Language,” Kotlin, https://kotlinlang.org/ (accessed Nov. 16, 2023).

[12] “Jetpack Compose UI App Development Toolkit,” Android Developers, https://developer.android.com/jetpack/compose (accessed Nov. 16, 2023).

[13] “Vimeo,” Vimeo. https://vimeo.com/ (accessed Dec. 29, 2023).

[14] “ExoPlayer,” ExoPlayer. https://exoplayer.dev/ (accessed Dec. 29, 2023).

[15] “license.txt,” Android Studio Software Development Team.

[16] udalov, “LICENSE.txt,” GitHub, Aug. 24, 2021. https://github.com/JetBrains/kotlin/blob/master/license/LICENSE.txt (accessed Dec. 29, 2023).

[17] “Apache License Version 2.0,” The Apache Software Foundation, Jan. 2004. https://www.apache.org/licenses/LICENSE-2.0.txt (accessed Dec. 29, 2023).

[18] “Terms and conditions | Android Studio | Android Developers,” Android Developers. https://developer.android.com/studio/terms (accessed Dec. 29, 2023).

[19] ojw28, “ExoPlayer/LICENSE,” GitHub. https://github.com/google/ExoPlayer/blob/release-v2/LICENSE (accessed Dec. 29, 2023).

[20] “Classifying Urban Sounds Using Deep Learning,” GitHub, https://github.com/mikesmales/Udacity-ML-Capstone/tree/master/Notebooks (accessed Dec. 29, 2023).

[21] “Firebase Pricing | Firebase,” Firebase. https://firebase.google.com/pricing (accessed Dec. 29, 2023).

[22] “Pricing plans | From free plans to enterprise solutions,” Vimeo. https://vimeo.com/upgrade-plan (accessed Dec. 29, 2023).

[23] “Licensing,” Raspberry Pi, https://www.raspberrypi.com/licensing/ (accessed Dec. 29, 2023).