

DogAlone: Your Dog’s Caretaker

Smart home Solution for Dogs Left Alone at Home

Junhyeong Choi
dept. Automotive Engineering
Hanyang University
Seoul, Korea
james200074@ hanyang.ac.kr

Junhyeong Byun
dept. Mechanical Engineering
Hanyang University
Seoul, Korea
jhjs1202@hanyang.ac.kr

Dongryul Lee
dept. Sport science
Hanyang University
Seoul, Korea
ga0212kimi@naver.com

Seoyeon Kim
dept. Tourism
Hanyang University
Seoul, Korea
2001seoyeon@gmail.com

Chaeyeon Jun
dept. Information System
Hanyang University
Seoul, Korea
oandand03@gmail.com

Abstract—In modern households, dogs have become more than just pets—they are now regarded as members of the family. As of 2022, the number of registered dogs in Korea has reached approximately 3.02 million. However, a significant number of dogs are left alone for more than 5 hours a day on average, experiencing issues such as loneliness, separation anxiety, and excessive barking. To address these challenges, many dog owners resort to measures like leaving the TV or lights on or using automatic feeders. However, these solutions often fall short of resolving the root problem, and financial constraints can make sending dogs to daycare an impractical option. To solve this issue, our team developed DogAlone, an application that analyzes a dog's barking to determine its emotional state and provides tailored smart home solutions. According to research, a dog’s bark conveys emotional and behavioral intentions depending on its frequency and rhythm. Using this insight, our application employs real-time AI analysis to monitor a dog's condition and recommend appropriate responses. Through DogAlone, we aim to alleviate the difficulties of managing dogs left alone at home and enhance the well-being of both pets and their owners.

Keywords—dog care, AI-based vocalization analysis, emotion detection, smart home integration

TABLE 1: ROLE ASSIGNMENTS

Name	Role & Description	
	Role	Description
Dongryul Lee	Design & Frontend Developer	As a Frontend developer, I used React Native in the Node.js environment to build an intuitive and visually appealing app interface. By using the expo tools, I simulated and tested the app in real time to ensure a smooth and dynamic user experience. My work focused on creating diverse and user-friendly UI components and app pages to increase usability and improve overall user journey. After develop UI components and app pages, I linked up our app with databases and AI. I'm also project manager in front-end development team. UI component and

Name	Role & Description	
	Role	Description
		page development role distribution and schedule management.
Junhyeong Choi	Project Manager & Back-end Developer & AI Developer	As an AI developer, I built a system to classify dog barking sounds by collecting labeled audio datasets, converting them into spectrograms, and training a deep learning model with TensorFlow to analyze the dog's emotional state. As a back-end developer, I implemented Node.js code to process AI analysis results and store them in a MySQL database, leveraging AWS for remote database management. I also analyzed stored data and generated reports. As a project manager, I oversaw development, coordinated team tasks, and ensured timely progress by managing resources and addressing challenges to achieve project goals effectively.
Junhyeong Byun	Project Manager & Back-end Developer & AI Developer	As a Backend and AI Developer, I focused on implementing robust server-side infrastructure and integrating AI functionalities. For backend development, I utilized Node.js to build core features such as user registration, login, and information retrieval, ensuring secure user management with Firebase Authentication. I also designed and managed relational databases using MySQL, enabling efficient data storage and retrieval to support the application's functionality. Additionally, I developed and tested APIs to ensure seamless communication between the client and server, playing a critical role in the Backends' stability and performance. On the AI side, I contributed by developing a system to analyze and classify dog barking sounds. I collected labeled audio data from Kaggle, processed the audio into spectrogram images, and trained a deep learning model using TensorFlow to classify the audio effectively. As the Project Manager (PM), I led the team by creating detailed schedules, breaking down the development process

Name	Role & Description	
	Role	Description
		into manageable milestones, and assigning specific tasks to each member. I defined the technical stacks and timelines for each feature, ensuring efficient progress and alignment across the team. My contributions to backend development, AI integration, and team management were instrumental in the project's successful completion.
Seoyeon Kim	Document & Back-end Developer & AI Developer	My main role was to integrate Backend and AI development seamlessly by writing code that connected the two effectively. As a backend developer, I used Node.js with Express to develop core functionalities for real-time sound data collection. I built a server that collects sound data in real-time, preprocesses it, and sends it to the AI model while integrating it with the Frontend. As an AI developer, I transformed audio data obtained from Kaggle and YouTube into spectrogram images and trained a CNN model using TensorFlow. I analyzed the emotional meanings of different sounds and matched each type of bark to a corresponding emotion. In addition to development, I was responsible for project documentation, which naturally led me to coordinate the tasks and schedules of all team members, effectively managing the overall project process.
Chaeyeon Jun	Document & Front-end Developer	As a front-end developer, I designed and implemented an intuitive and user-friendly app interface using React Native. I used expo tools to quickly solve problems with real-time testing, ensuring that the user experience remained efficient and seamless. I designed and optimized core components, and systematically designed navigation structures and screen transitions to enable users to quickly find information and complete tasks with ease. This simplified the user journey and increased the utilization of app with intuitive design. In addition, I worked closely with the team to continuously improve UI/UX and completed a stable structure that is easy to maintain by reflecting feedback. In addition, by writing the document, I clarified the implementation process, configuration, function, and usage of the app to intuitively convey the app's information.

I. INTRODUCTION

A. Motivation

In modern households, dogs have become more than just pets; they have established themselves as important members of the family. As of 2022, the number of registered dogs in South Korea reached 3,025,859, meaning that there is roughly one dog for every 16 people in the country. As the number of households with dogs increases, proper management and protection of these pets have become a significant social concern. Various applications providing features such as walking management and health monitoring for dogs have emerged, indicating that the pet market is growing in the mobile sector as well.

In reality, many dogs spend long periods alone at home and do not receive adequate care. According to the '2021 Korea Pet Report' published by KB Financial Group, 75.3% of pet owners reported leaving their pets alone at home. The average time dogs are left alone is 5 hours and 22 minutes per day, which amounts to about a quarter of the day without their owners. Dogs left alone can experience loneliness, and in severe cases, they may suffer from separation anxiety, excessive barking, or potty issues. Experts and experienced dog owners warn that if a dog must be left alone for long periods, reconsidering adoption may be necessary, as it can have detrimental effects on the pet's well-being.

To address this, many pet owners take measures such as leaving the TV or lights on or installing automatic feeders. Despite these efforts, many people still feel guilty about leaving their dogs alone, worrying about the loneliness their pets will experience while waiting for their owners to return. Sending a dog to a pet daycare is another option, but it can be financially burdensome, leading many dog-owning households to face difficult decisions.

To help solve this problem, our team has designed an application that analyzes a dog's barking to assess its emotional state and provide appropriate smart home solutions. According to the paper "Communication in Dogs," dogs' barking can be distinguished by frequency, which reveals their emotional state. For example, low-frequency barking expresses aggression or alertness, while high-frequency barking indicates anxiety or distress. Dogs' barking patterns, including frequency and rhythm, provide contextual information that allows humans to interpret their emotional state and behavioral intentions.

Building on the characteristics of dogs' barks, we have developed an AI that can analyze their vocalizations to determine what emotions they are expressing. Through real-time sound analysis, our application, DogAlone, identifies the dog's emotional state and offers suitable smart home solutions. With this application, we aim to address the difficulties faced by many pet owners in managing their dogs when they are left alone.

B. Problem Statement

1 As the number of single-person households and DINKs (dual-income, no-kids families) increases, the number of households with pets has also risen significantly. However, it is difficult for busy owners, who are occupied with work or school, to manage their dogs throughout the day. As a result, nearly 75% of dogs are left alone at home, and no suitable solutions have yet been found to address this issue.

2 Both Samsung and LG have introduced various smart home appliances featuring pet care modes, but these are far from providing real-time management of pets. Simply being able to control devices remotely or monitor pets via cameras does not alleviate the concerns of dog owners.

3 Research into analyzing animal sounds using AI has progressed over time, and several apps have been developed to interpret dog barks. Despite the technological advancements, there have been no efforts to address social issues related to pet dogs through simultaneous interpretation of their vocalizations.

4 Dog barks express various emotions depending on their characteristics. Analyzing these barks in real-time and providing results helps facilitate communication between the dog and its owner, even when they are apart. This goes beyond simply monitoring the dog's condition; through a smart home system, appropriate actions can be taken based on the dog's emotional state.

5 Organizing the data of dog barking analysis by day, week, or month can yield meaningful results. This allows owners to understand the emotions their dog most frequently experiences and think of appropriate ways to respond to these feelings.

C. Research on Related Software

i) iPet

iPet is a dog translation app that uses machine learning data to infer the meaning of a dog's sounds. By pressing the record button within the app, users can record their dog's barking for 7 seconds. The app translates the dog's sounds into human language and displays them. In addition, users can register their dog's profile to calculate calorie intake and food portions, and it includes features to calculate the calories burned and distance covered during walks. The app also displays the days walked in a calendar view. While the main feature is dog translation, it includes various useful tools for dog care.

ii) BowWow

BowWow is a dog translation app that records a dog's barking and converts it into human language. It also allows users to convert their own speech into sounds that their dog can understand. This translation takes place in a chat format, creating the feeling of a real conversation with the dog. In addition to the translation feature, the app offers AI dog profile creation, a unique AI dog karaoke function to create personalized songs with your dog, a service information feature for nearby pet-related services, and a feature that provides information about the dog's habits.

iii) 펫카

Petca is a smart home app for pets launched by LG U+. Through Petca, users can control various devices such as the home CCTV 'AI Momca', a pet training device called 'PetToy', a snack robot, sleep lights, and more. It serves as an AI-based pet care assistant, designed to ease the concerns of pet owners who worry about their pets being left alone. Users can view live footage of their home via a home cam, and can also save, record, or attempt to communicate with their pets. In addition, users can remotely control various smart home devices.

iv) Barkio: Dog Monitor & Camera

Barkio is a dog monitoring app that turns two smartphones into a smart pet camera, allowing users to monitor their dog in real-time. When users leave home, they can view live video of their dog, record it, or take snapshots. The app allows users to listen to the dog's barking and track the dog's movements through a motion detection feature. Additionally, the app includes features like communication with the dog, two-way video, activity logs, and separation anxiety detection. The biggest advantage is that unused phones or tablets at home can be used as home cams.

II. REQUIREMENT

A. Introduction

We have designed the app to be simple and user-friendly, allowing even those who have never used it before to intuitively understand how to use it. Most of the key features can be controlled from the main screen, and users can easily navigate to other screens using the navigation bar at the bottom.

B. Sign Up and Login

Users can quickly complete the sign-up process by entering their email address, a secure password, and their dog's name and breed. Registered users can log in to the DogAlone app easily using their email address and password. This grants access to personalized dog profiles, where important information about the dog can be viewed and managed.

C. Real-Time Barking Analysis

Our app effectively collects dog barking sounds through a connection with an external microphone. This advanced voice recognition technology captures the unique sounds of dogs, such as barking, whining, and growling. By analyzing these sounds with AI, the app helps identify the dog's emotional state and allows for appropriate responses.

D. Barking Alert Notifications

If a dog's barking is detected while the app is not in use, users will receive a push notification on their smartphone. Once logged into the app, users can view this new alert and check any undetected barking information. This ensures that users can access information about their dog whenever necessary.

E. Emotion-Based Smart Home Solutions

The app continuously monitors the dog's emotional state through barking sounds and displays it to the user. It also provides appropriate actions through connected smart home devices to enhance the dog's well-being. Here are the recommended services based on the dog's emotional state:

- i) Excitement
To alleviate excitement, the dog needs plenty of exercise and playtime. If the environment is overstimulating, actions to minimize this need to be taken. When excitement is detected, the system can activate the snack robot, turn off music, and adjust the lights to a softer tone to help calm the dog's atmosphere for the pet.
- ii) Fear
A fearful dog requires a safe space, so the home should be perceived as a quiet and secure environment. When fear is detected, calm music is played, the temperature is raised, and the snack robot provides positive reinforcement to help the dog overcome its fear.
- iii) Contentment
In situations where the dog is feeling content, it can be made more comfortable by playing soft music through the speaker, showing videos of other animals on the TV, and maintaining clean indoor air with an air purifier.
- iv) Anxiety
When the dog is anxious, it's important to provide consistent attention and create a quiet space. Communication can be facilitated through the speaker, the lights can be set to warm tones, and videos of the dog's favorite shows can be played on the TV.
- v) Loneliness
To address loneliness, toys can be used to keep the dog entertained when alone. The snack robot can be activated, videos can be played on the TV, and the speaker can play the owner's voice to comfort the dog.
- vi) Pain
When the dog is in pain, it's essential to calm the dog and provide treatment quickly. The app can display nearby veterinary clinic information, play calming music, adjust the lighting to a soft tone, and help the user take appropriate action as soon as possible.

F. Dog Meal Control

To ensure that the dog's meals are provided even when the owner is not at home, we have included a feature to control the automatic pet feeder through the app. The feeder can be turned on or off from the main screen, and users can input the meal schedule and portions. The feeder will automatically provide the dog with meals at the designated time and amount.

G. Smart Home Device Control

In addition to controlling devices based on barking recognition, users can register smart home devices they own and control them remotely. By clicking on a device from the list provided in the app and entering the serial number, the device can be registered. Users can make precise adjustments beyond simply turning devices on or off.

H. Barking Report

The app stores the analysis results of the dog's barking sounds and allows users to view this data in daily, weekly, or monthly intervals. These analysis results are displayed in bar graphs for easy interpretation. Additionally, detailed reports provide insights such as the most frequently detected emotions and how many times they were detected.

I. User Manual

The app provides a user manual that offers specific information about the app's intended design, types of emotions, and compatible smart home devices. If users have any questions while using the app, they can refer to the user manual to resolve their doubts.

III. DEVELOPMENT ENVIRONMENT

A. Choice of software development platform

i) Development platform

Windows 11: Windows provides an optimized environment for both frontend and backend development. For the frontend, it offers intuitive user interfaces and various development tools, improving work efficiency and compatibility with programs needed for web design and UI development. On the backend, it integrates seamlessly with Microsoft technologies to support server application development and database management. Additionally, Windows ensures stability and reliability even in complex server environments, such as cloud services and API integration. These features provide a development environment that meets both productivity and stability requirements for web and application development.

ii) Language & Framework

· JavaScript

JavaScript is a widely used programming language for web and mobile app development. Through frameworks like React Native, Ionic, and Expo, it offers a cross-platform environment that allows for simultaneous development of iOS and Android apps using a single codebase. Thanks to asynchronous processing and support for various libraries, it enables efficient data handling and flexible user interface design. It also simplifies API integration and real-time data processing, making it ideal for creating fast and scalable apps.

· Python

Python is one of the most widely used programming languages in AI and machine learning, offering powerful data processing libraries and an intuitive syntax that is convenient for both researchers and developers. Python integrates well with various AI frameworks and libraries (e.g., TensorFlow), making it an ideal choice for model development and experimentation.

· Node.js

Node.js is a runtime environment designed to run JavaScript on the server side. It is built on Google's V8 engine and offers fast and efficient performance. By using the same language (JavaScript) for both frontend and backend, it increases development productivity. With

event-driven asynchronous processing, it is ideal for handling large-scale concurrent requests. Additionally, through npm (Node Package Manager), developers can utilize numerous open-source libraries to further enhance productivity.

- React Native

React Native is a cross-platform application development framework based on JavaScript. It allows for efficient development of both iOS and Android apps using a single codebase. Using React's component-based development approach, it enables the creation of efficient and reusable UIs, providing a user experience close to native applications. The active community and abundant libraries help developers implement necessary features quickly, reducing development time and making maintenance easier.

- Expo

Expo is an open-source framework that supports mobile app development based on React Native. It simplifies the development process by minimizing the initial setup and complex native code tasks. Through the Expo Go app, developers can instantly see code changes and easily deploy and test their apps. It also provides various APIs and modules (e.g., camera, local storage) by default, enabling the implementation of features without extra configuration. This fast and simplified development environment, with numerous built-in features, makes app development efficient.

- Flask

Flask is a lightweight web application framework developed in Python, commonly used for API server implementation. It is simple and modular, equipped only with necessary functionalities, and can be expanded as needed. With various plugins, users can easily add features like databases, authentication, and migration. Flask allows for writing clear and concise code, making it suitable for both rapid and efficient development of small prototypes and large-scale applications.

B. Software in Use

- Visual Studio Code (VS Code)

Visual Studio Code (VS Code) is an open-source code editor that is lightweight, fast, and offers a variety of features, making it popular among developers. It includes essential tools to improve development efficiency, such as code auto-completion, debugging tools, an integrated terminal, and Git integration. Additionally, with over 10,000 extensions, developers can customize their development environment by adding language support, code formatters, debuggers, and more. Its strong extensibility and support for various languages and platforms make it suitable for most development projects.

- Google Colab

Google Colab is a cloud-based Jupyter Notebook environment that allows Python code to be written and executed directly in the web browser. This environment is highly useful for machine learning, deep learning, and data analysis tasks. It integrates with Google Drive to store and share files and connects to Google Cloud Platform for utilizing hardware accelerators such as GPUs and TPUs. This enables fast processing of large datasets and complex models. Additionally, Colab comes with

TensorFlow and other libraries pre-installed, making them easily accessible for users.

- Postman

Postman is a platform for API development and testing. It helps developers design and debug various APIs, including RESTful, GraphQL, and SOAP. Through its intuitive interface, developers can easily create requests, review responses, set up authentication, and manage environment variables. It also supports API test automation, documentation, and team sharing, making the development and collaboration process smoother. Postman improves productivity by simplifying API testing and providing systematic management of complex requests and workflows.

- AWS (Amazon Web Services)

Amazon Web Services is a cloud computing platform offered by Amazon. It provides a wide range of cloud-based services, from virtual servers to AI and machine learning services. It is characterized by flexibility and scalability, allowing users to increase or decrease the resources they need. AWS delivers services quickly across the globe via a global network, making it convenient for global expansion.

- TensorFlow

TensorFlow is an open-source machine learning library that supports model design, training, and deployment. It processes numerical computations using data flow graphs and represents data through tensors (multi-dimensional arrays). TensorFlow supports various machine learning algorithms, including deep neural networks, and can be applied to fields like image and speech recognition, natural language processing, and robotics.

- MySQL

MySQL is an open-source relational database management system designed to handle data storage and management efficiently. Based on SQL (Structured Query Language), it offers high reliability and performance, making it widely used in web applications. MySQL provides fast processing speed and scalability, making it ideal for projects that require stable data management.

- Firebase

Firebase is a cloud-based development platform provided by Google. It helps developers create mobile applications quickly and efficiently, without writing server-side code. Firebase allows real-time data synchronization and updates, and is especially useful for developing applications that require sorting and searching data.

- Figma

Figma is a tool that facilitates seamless collaboration between designers and developers, especially in UI/UX design for software development projects. Team members can access and modify the same project in real-time, enabling quick design feedback and edits. Figma reduces communication gaps between design and development, making the workflow more efficient.

- GitHub

GitHub is a source code management service based on the distributed version control system Git. It helps developers manage projects and collaborate effectively. GitHub provides tools for tracking code changes, creating

branches, merging, and managing the code repository centrally. Its intuitive interface and vast community make collaboration easy, while various integrations help improve development efficiency.

- Overleaf

Overleaf is a cloud-based LaTeX document editor used for writing technical papers, scientific reports, and presentations. With real-time collaboration, multiple users can edit documents simultaneously. It provides an intuitive interface, allowing users to start without dealing with complex LaTeX configurations. Overleaf's templates and auto-compilation features help reduce document writing time and improve efficiency.

C. Developer's computer resource

TABLE 2: Developer's environment

Name	Computer resource	
Chaeyeon Jun	Operating System	Microsoft Windows 11 Home 23H2
	CPU	Intel(R) Core(TM) i7-8565U CPU @ 1.80GHz
	RAM	8GB
	GPU	Intel(R) UHD Graphics 620
Dongryul Lee	Operating System	Microsoft Windows 11 Home
	CPU	13th Gen intel(R) Core(TM) i5-1340p CPU @ 1.90Ghz
	RAM	16G
	GPU	Intel(R) Xe Graphics
Seoyeon Kim	Operating System	Microsoft Windows 11 Pro
	CPU	11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz
	RAM	16.0GB
	GPU	Intel(R) Iris(R) Xe Graphics
Junhyeong Byun	Operating System	Microsoft Windows 10 Enterprise
	CPU	Intel(R) Core(TM) i7-8565U @ 1.80GH
	RAM	16GB
	GPU	Intel(R) UHD Graphics 620 외 Mirage Driver
Junhyeong Choi	Operating System	Microsoft Windows 11
	CPU	Intel(R) Core(TM) i5-8500U CPU @ 3.00GHz
	RAM	16GB
	GPU	NVIDIA GeForce GTX 1060 3GB

D. Cost Estimation

In the DogAlone app development environment, both Colab and AWS cloud services may incur costs depending on the use case. For Colab, costs arise when handling large datasets or training complex deep learning models. For

smaller datasets or simpler machine learning models, there are typically no costs involved. AWS cloud services might not incur costs during the testing phase or when server and database usage is low. However, if large amounts of data need to be stored, API calls are frequent, real-time data synchronization is required, or high traffic needs to be handled, additional costs may arise. In conclusion, costs for both Colab and AWS depend on the scale of the tasks and resource usage.

IV. SPECIFICATIONS

A. Loading Page

The loading page is displayed while the app loads data. It appears when the app is first launched or when new data needs to be fetched during a page transition. To keep the user engaged, animations like a spinning circle and a dog running were used. The app name and logo are displayed in the center. The app logo is a paw print in yellow, symbolizing that the app is for dogs.

B. Login Page

The login page handles user authentication. Firebase authentication is integrated to ensure secure and efficient user management, while MySQL is used to design the relational database for the login system. If the user has already registered, they can log in by entering their ID and password. Login is only possible if the ID and password match exactly. If the user has not yet registered, they can click the "Sign Up" button at the bottom of the login page to go to the registration page. On the registration page, users can create an account by entering an ID, password, dog's name, and dog breed. The ID must be in the form of an email. If the user attempts to register with an email that is already in use, the registration is denied with a message saying "This email is already registered." Users can also log in through Google integration. The app will log the user out if it hasn't been used for a full day. After a successful login, the user is immediately redirected to the main page (C).

C. Main Screen

Upon entering the app, users are directed to the main page. Pressing the home icon (first from the left on the bottom bar) also leads to the main page. The main page allows users to monitor their dog's status in real-time and control relevant smart home solutions. The main page offers access to the key features of the app.

i) Welcome Message and Notification

At the top of the screen, a greeting message like "Hello, [dog's name]'s owner!" is displayed. To the right of this message is a bell-shaped "New News" button, which shows the recent changes in the dog's emotional state along with timestamps. For example: "13:40 | Choco felt anxiety!".

ii) Dog's Status and Service Recommendations

On the left side of the box, there is a dog illustration along with the dog's name, breed, and current emotional state, displayed in a speech bubble. The emotional state is determined by analyzing sounds when the user presses the real-time dog sound analysis button (3-c). On the right side, smart home services are recommended based on the

dog's emotional state. These service suggestions are shown based on the user's registered smart home devices. Smart home services suggested for each emotion are as follows:

- Excitement (bark): Turn off house music, turn on soft lighting, activate treat robot
- Fear (growl): Play quiet music, increase room temperature, activate treat robot
- Contentment (grunt): Play quiet music, play nature videos, turn on air purifier
- Anxiety (whimper): Connect to speaker for call, turn on soft lighting, play animal videos
- Loneliness (howl): Play owner's voice on speaker, play animal videos, activate treat robot
- Pain (yip): Show nearby veterinary clinic information, play quiet music, turn on soft lighting

When each button is pressed, the corresponding smart home solution is triggered. For example, if the dog feels anxious, the buttons for "Connect to speaker for call", "Turn on soft lighting", and "Play animal videos" are shown. Pressing the "Connect to speaker" button allows communication with the dog through the home speaker.

iii) Real-time Dog Sound Analysis

Before pressing the 'rec' button, the message reads "Try detecting real-time dog sounds!". Pressing the 'rec' button turns it red and starts recording, showing the time in mm:ss format. The message changes to "Real-time dog sound detection in progress!". On the front-end, a connection is made with the backend using socket.io. When an audiostream request is received, the backend invokes a pre-built dog emotion analysis AI model to analyze the incoming sounds in real time and send the results back to the front end. The front end then updates the dog's emotional state and the recommended services in the "Dog's Status and Service Recommendations" section(C-ii)

Recording stops after 1 hour, and a push notification is sent to the user with the message "Recording has stopped, would you like to start it again?". If the user presses the button showing the recording time again, it stops recording and resets the button to 'rec'. The recorded sound is sent to the backend in WAV format, where it undergoes preprocessing into a spectrogram before being analyzed by the AI.

iv) Feeding Control

If a smart feeder is registered, the feeding time and amount can be set so that the feeder operates automatically. The on/off button on the right allows users to control the smart feeder. Clicking the arrow leads to the feeder settings page (C-v). Multiple feeders can be added. To add a new feeder, click the "+" button to set its name and serial number.

v) Smart Feeder Settings

Registered smart feeders can be set to provide food at specific times and days, with a set amount. Clicking the "+" button allows adding a feeding schedule with a selected time, day, and feeding amount. The feeding amount is based on the specific settings of the feeder. After setting the schedule, users can save it, and the schedule is added to the list. The saved schedule can be turned on/off or edited. When the feeder provides food,

the user receives a push notification saying, "The smart feeder has fed your dog!"

vi) Home Environment

If no smart home devices are registered, users can click the "+" button to add devices and control them. Devices that can be registered include "LG Air Conditioner, LG Air Purifier, LG Smart TV, AI Speaker, Lighting, LG U+ Treat Robot." Clicking the "+" button opens a popup where users can select the device type, enter the serial number, and complete the registration. Registered devices can be controlled through the main screen or the smart home services page. When lights and air conditioners are registered, a box for "Lighting/Temperature" is automatically created to display the current status of lighting and temperature in the home. These boxes show the device status but cannot modify it. Below this are two device status boxes, which allow turning devices on/off. To view all devices, users can click the "Go to Smart Home Services" button to access Smart Home Services page(G).

D. Edit Dog Information

By pressing the menu icon (4th from the left on the bottom bar), users can select the "Edit Dog Information" button to access the dog information popup. Here, users can modify the dog's name and breed, which were entered during registration. Clicking the saved dog name and breed allows editing them. After making changes, clicking the "Save Changes" button closes the popup, and the updated information is displayed.

E. Notices

By pressing the menu icon (4th from the left on the bottom bar), users can select the "Notices" button to view a popup containing important updates, such as app features or improvements. The popup can be closed by clicking the "X" button.

F. Contact Us

By pressing the menu icon (4th from the left on the bottom bar), users can select the "Contact Us" button to view a popup with an email address for inquiries. The popup can be closed by clicking the "X" button.

G. Smart Home Services

This page allows users to view all registered devices. While the main screen only allows on/off control for up to two devices, this page allows more detailed control of all devices. Users can turn devices on/off and adjust specific settings for each device.

- Air Conditioner: Adjust temperature and change mode
- Speaker: Change channel and adjust volume
- TV: Change channel and adjust volume
- Air Purifier: Change mode and adjust intensity
- Treat Robot: Display last operation time and adjust scheduled operation time
- Lighting: Change mode and adjust brightness

H. Dog Cry Report

By pressing the chart icon (2nd from the left on the bottom bar), users can access the Dog Cry Report page. It displays the frequency of each emotion in a bar graph format.

The default date is set to today, but users can select a specific date using the calendar button to view the graph. Dates can be selected up to 3 months back. Without selecting a date, users can view the most recent data by day, week, or month. Below the graph, a report is displayed with advice on the most frequent emotion, time of occurrence, and recommendations. If no recording has been made, the message “Start recording!” is shown. Advice for each emotion includes:

- Excitement (bark):
 1. Calm Down: When your dog is excited, speak calmly to them. Raising your voice or reacting excessively can make the situation worse.
 2. Avoid Attention: Avoid giving your dog what they want immediately when they're excited. Reduce stimuli to help them calm down.
 3. Reduce Stimuli: Minimize external stimuli (sounds, people, other animals, etc.) to help your dog feel more at ease.
- Fear (growl):
 1. Provide Comfort: Create a warm and stable environment for a fearful dog. Use a soft voice to reassure them.
 2. Avoid Overstimulation: It's important to avoid overwhelming your dog when they are scared. Make sure too many people or animals don't approach them.
 3. Seek Help: If your dog continues to feel fearful, consider seeking help from a professional.
- Contentment (grunt):
 1. Positive Feedback: When your dog is satisfied, praise them. Saying things like “Good job!” or “Feeling good?” can provide positive reinforcement.
 2. Play Together: Enjoy the moment with your dog by engaging in activities like walks or playing with toys to enhance their happiness.
 3. Maintain a Comfortable Environment: Keep a peaceful and comfortable environment when your dog is content, as it helps them feel relaxed and happy.
- Anxiety (whimper):
 1. Create a Stable Environment: When your dog is anxious, create a calm space for them to feel comfortable. Use soft lighting or a secure area.
 2. Reassure with Your Voice: Your voice can be a great comfort to an anxious dog. Speak softly or approach them gently to help calm their nerves.
 3. Distract with Play: Redirect your dog's attention through play to help alleviate anxiety.
- Loneliness (howl):
 1. Spend Time Together: When your dog feels lonely, spend as much time as possible with them. Walks or playtime can help reduce their sense of isolation.
 2. Comfort with Your Voice: Hearing your voice can provide reassurance during moments of loneliness. Call or send a voice message to ease their anxiety.
 3. Provide Toys: Offer toys to help alleviate loneliness when your dog is alone.
- Pain (yip):
 1. Visit the Vet: If your dog seems to be in pain, take them to the vet for professional care. Quick treatment is essential.

2. Provide a Comfortable Environment: Create a quiet and comfortable space for your dog to rest. Use soft bedding or a warm blanket for extra comfort.
3. Regular Check-Ups: Continuously monitor your dog's condition and seek further veterinary care if necessary. Regular check-ups are important for ongoing recovery.

I. User Guide

By pressing the book icon (3rd from the left on the bottom bar), users can access the DogAlone User Guide page. It includes information about the app.

V. ARCHITECTURE DESIGN

A. Overall architecture

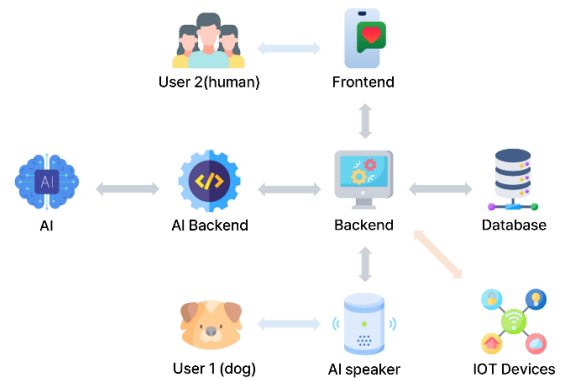


Fig 1. Overall architecture

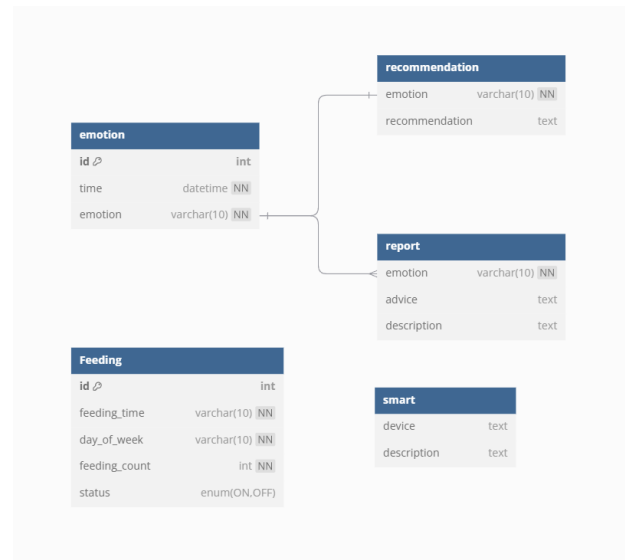


Fig 2. Database

DogAlone is divided into three main modules: Frontend, Backend, and AI.

The first module is the Frontend. It is responsible for designing the screens and implementing interactions to ensure that users can intuitively understand and easily use the app's functions. Developed using React Native, it ensures smooth operation across various devices and provides a user-friendly UI that allows users to monitor their dog's condition

in real-time and control smart home devices. The frontend module visually represents the emotional state of the dog, analyzed from its barking sounds, and presents the information clearly on the report screen. The report shows the frequency of each emotional state using graphs, and users can easily view the analysis results through the “Today's Bark Report.” Based on this data, the app provides personalized solutions or allows users to control smart home devices directly, helping users manage their dogs more effectively.

The second module is the backend, which connects to the database and AI. The backend of DogAlone consists of a main server implemented with Node.js, an AI connection server implemented with Flask, and a database implemented with MySQL. The main server provides functionalities such as login, emotion-based recommendation service retrieval, feeding schedule creation/editing/deletion, real-time sound data reception, and bark report retrieval. The Node.js server stores user registration information and retrieves login data. When a specific emotion analysis result is obtained, it automatically fetches the recommendation service. If a new feeding schedule is created, it is stored in the database, and users can modify or delete it. This information can always be viewed when accessing the feeding schedule page. Additionally, SocketIO is used to send audio data from the frontend to the Flask server, where the analysis results are received and stored in the database. The bark sound analysis database allows users to retrieve daily, weekly, and monthly analysis results and provides reports with advice based on the most frequent bark sound class.

The Flask server connects with the main server to receive real-time bark sounds, and it retrieves the file paths of these sounds. These sounds are converted into spectrograms, and a pre-trained AI model is used to analyze them. The analysis results are then sent back to the main server, allowing them to be stored in the database. All backend components are deployed on Amazon Web Services (AWS). The main server and Flask server are implemented on AWS Compute Cloud (Elastic EC2), and the database is connected to AWS Relational Database Service (RDS).

The third module is AI, which is responsible for analyzing the audio data sent by the backend and returning the results. The AI is based on a Convolutional Neural Network (CNN) model. It receives the preprocessed spectrogram data from the backend and determines which of the six barking sound types is most similar.

To train the AI, audio data in WAV format is first converted into spectrograms. A spectrogram visually represents frequency changes over time and is useful for extracting features from sound. The CNN model goes through several learning processes to recognize characteristic spectrogram patterns for each sound type. The model learns to identify patterns in the dataset and, upon receiving input spectrogram data, predicts which sound type it corresponds to. The sound type dataset was collected from Kaggle datasets and audio extracted from YouTube videos. The publicly available datasets on Kaggle included various sound types, making them useful resources for model training. Additional data was gathered by extracting audio from YouTube videos that contained specific sound types.

Through this process, the AI can analyze real-time audio data and accurately determine the emotional state represented by the sound the user sends. The AI then returns this analysis

result to the backend server, enabling the system to provide an appropriate response to the user.

B. Directory Organization

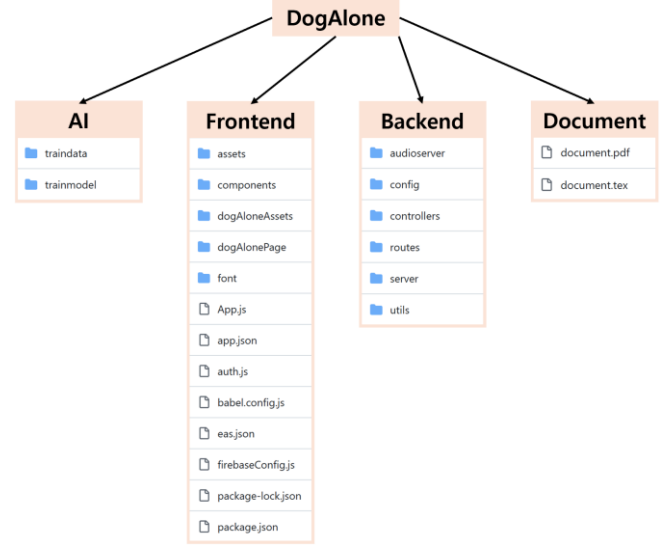


Fig 3. Directory

TABLE 2: Frontend Directory Organization

Directory	File/Names	Library
DogAlone/ frontend	assets components dogAloneAssets dogAlonePage font App.js app.json auth.js babel.config.js eas.json firebaseConfig.js package.json package-lock.json	react @react- navigation/native @react- navigation/stack expo-web-browser firebase/app firebase/analytics @env
DogAlone /frontend/assets	Adaptive-icon.png favicon.png icon.png splash.png	
DogAlone /frontend /dogAlonePage	DocuPage.js Loading1.js Login.js MainPage.js RegisterPage.js ReportPage.js smartHome.js FoodPage.js	react react-native @expo-google- fonts/inter @react- navigation/native expo-web-browser expo-av socket.io- client expo-file- system
DogAlone /frontend/font	Inter- VariableFont_opsz,w ght.ttf	

DogAlone /frontend /components	BackButton.js DayMonthButton.js DocuButton.js DragBar.js ExpandableBar.js HomeButton.js LGairCard.js LGlightCard.js LGpetCard.js LGSpeakerCard.js LGtvCard.js LGwaterCard.js LoginButton.js MenuButton.js OnOffButton.js RecommendButton.js ReportButton.js inputBox.js	react react-native @react- navigation/native @react-native- community/datetimepicker
DogAlone /frontend /dogAloneassets	air.png alarm.png arrow.png calendar.png closeicon.png docu.png dog.png dogbox.png home.png light.png loadCon.png loadingdog.png loadinglogo.png logo.png menu.png miniLight.png miniMusic.png miniVideo.png offbutton.png onbutton.png pet.png recordingblue.png recordingred.png report.png report1.png report2.png report3.png report4.png report5.png speaker.png textbubble.png tv.png water.png	

TABLE 3: Backend Directory Organization

Directory	File names	Library
DogAlone/backend/audioserver/	audio_data audio_analyzer.py	flask os numpy librosa matplotlib tensorflow datetime
DogAlone/backend/config/	db.js firebaseConfig.js	mysql2
DogAlone/backend/controllers	authController.js	
DogAlone/backend/routes	routes.js	express

DogAlone/backend/server	audioStreamServer.js server.js	socket.io http axios fs path
DogAlone/backend/utills	add.js recommendation.js report_daily.js report_monthly.js report_weekly.js	socket.io-client mysql2 express

TABLE 4: AI Directory Organization

Directory	File names	Library
DogAlone/AI/trainmodel	train_image.py train_model.py processed_files.txt dog_emotion_model.h 5	os librosa matplotlib numpy tensorflow
DogAlone/AI/traindata	train_png train_wav	

C. Module 1: Frontend

i) Purpose

The frontend module is the interface designed to allow users to easily and effectively utilize the service. This module enables various functions such as emotion analysis of dogs and smart home device control. It sends user input data to the server and presents the processed results in a visually understandable format. Through this process, DogAlone users receive personalized solutions based on their current situation. Moreover, the module utilizes a component-based architecture to improve code reusability, and enhances maintainability and scalability. The user interface adopts a simple and intuitive design, enabling users to easily navigate and utilize all the features within the app.

ii) Functionality

Users can create an account through email or a social account and record their dog's barking sounds to analyze its emotions. Based on the analyzed data, the system can control LG smart home devices, and the emotional data can be viewed through visual graphs that display daily, weekly, or monthly trends. Additionally, users can check data for specific periods via a calendar or timeline, and access the app's user manual for useful information. The frontend communicates with the backend API to store user data and synchronize the analysis results in real-time, allowing users to quickly access the latest data.

iii) Location of source code

:<https://github.com/pata1202/DogAlone-Your-Dog-s-Caretaker/tree/main/frontend>

iv) Class components

[DogAlone/Frontend]

- App.js: The main entry point of the app, managing the overall navigation structure and routing between screens.

- `babel.config.js`: Initializes Firebase Authentication and exports authentication objects for easy access in other modules.
- `firebaseConfig.js`: Initializes Firebase services, including Authentication, Firestore, Storage, and Cloud Messaging, and allows easy expansion of Firebase features within the app.

[DogAlone/frontend/components]

- `BackButton.js`: Implements a "back" button using the `useNavigation` hook to navigate the user to the previous screen.
- `DayMonthButton.js`: Allows users to filter data by day, week, or month, with visual emphasis on the selected option to clearly indicate the current selection.
- `DocuButton.js`: A button that redirects users to the "documentation page" for further information. When clicked, it navigates to the `DocuPage` screen.
- `DragBar.js`: Implements a draggable slider interface that allows users to adjust values, such as volume or brightness.
- `ExpandableBar.js`: A collapsible bar with an animation effect, expanding or collapsing when clicked.
- `LGairCard.js`: Controls LG air conditioner functionality, including turning it on/off, and checking and adjusting the current temperature.
- `LGlightCard.js`: Manages LG lighting controls, including turning lights on/off, adjusting color and brightness.
- `LGpetCard.js`: Controls the LG pet care device, allowing users to check and adjust the last and next scheduled feeding times.
- `LGspeakerCard.js`: Manages LG speaker controls, including power, channel, and volume adjustments.
- `LGtvCard.js`: Controls LG TV functionality, including channel and volume adjustments.
- `LGwaterCard.js`: Manages the LG water dispenser, allowing users to manage its operating status and schedule the next watering time.
- `LoginButton.js`: A login button that allows users to log in or navigate to the login page when clicked. The button's action can be dynamically defined using the `onPress` attribute.
- `MenuButton.js`: A menu button for easy access to various features, such as editing dog information, managing LG smart home devices, viewing announcements, and inquiries.
- `OnOffButton.js`: A button that toggles a specific setting between "on" and "off."
- `RecommendButton.js`: A button that executes a recommended action, such as playing music or adjusting lighting.
- `ReportButton.js`: A button that navigates to the `ReportPage` screen.
- `inputBox.js`: A custom text input component that allows users to enter data within various UI forms in the app.

[DogAlone/frontend/dogAlonePage]

- `DocuPage.js`: A documentation page where users can learn about the main features of the app, such as dog emotion analysis and smart home solutions.

- `Loading1.js`: A loading screen displayed when the app is launching, featuring animated elements such as the DogAlone logo and a moving dog image.
- `Login.js`: The login page that allows users to log in via email and password or through Google authentication. The app uses `expo-auth-session` and Firebase Authentication for Google login.
- `RegisterPage.js`: A registration page for users to create an account and input dog information. Upon successful registration, users are redirected to the login page.
- `MainPage.js`: The main screen where users can view real-time dog status, control smart home devices, analyze emotions, and receive customized solutions based on emotion analysis.
- `ReportPage.js`: A report page that visualizes emotion data from dog barking analysis. Users can filter data by day, week, or month and view the results in bar graph format.
- `smartHome.js`: A page for managing and controlling smart home devices, including cards for devices like speakers, TV, lights, air purifiers, pet care, and water dispensers.

D. Module 2: Backend

i) Purpose

The backend is responsible for managing the server and database, processing user requests generated from the client-side, and returning the necessary information based on those requests. It handles data storage, management, and retrieval, providing essential functionality to support smooth and intuitive user experiences. The backend stores data created by user actions in the database and retrieves the necessary data when requested.

The backend of the DogAlone: Your Dog's Caretaker project is implemented using Node.js and Flask. The main server handles user authentication, pet information storage and retrieval, AI analysis results management, and more. Firebase Authentication is used to securely authenticate users, while Firestore is used for storing pet information. The audio analysis server analyzes uploaded audio data through machine learning models to predict the emotional state of the dog. Flask, a Python-based framework, is suitable for serving machine learning models and performing data analysis, enabling efficient and flexible data processing.

In the DogAlone architecture, the main server has two primary functions. First, it communicates with the client application (React Native), processes user requests, and returns appropriate responses. Second, it communicates with the audio analysis server to transmit audio data to the Flask server and stores the returned emotional analysis results. The audio analysis server analyzes the dog's behavior and emotions through machine learning models, providing useful feedback to the user. This enhances communication and interaction between users and their pets, helping effectively manage the pet's state.

ii) Functionality

DogAlone: Your Dog's Caretaker manages the dog's condition and behavior, helping users communicate effectively with their pets.

- Main Server

The main server provides secure user authentication using Firebase Authentication and manages user and pet information using Firestore. Pet details such as name and breed can be stored and retrieved, and the AI-powered emotion recommendation system and bark report analysis offer personalized advice. The server allows the management of feeding schedules, including the storage, retrieval, modification, and deletion of feeding times, days, and amounts. The smart device management feature enables users to check and modify the status of registered devices. Through communication with the Flask-based audio analysis server, the main server processes AI analysis data and stores and returns the results, providing real-time interaction with the client via RESTful APIs.

- **Audio Analysis Server**

The audio analysis server processes uploaded .wav files by converting them into spectrograms and uses a TensorFlow-based machine learning model (dog_emotion_model.h5) to predict the dog's emotional state (e.g., 'Bark', 'Growl'). The model returns the emotional state and prediction probability, which is sent to the main server for sharing with the user. Additionally, the server can receive real-time audio stream data from the client, convert it into spectrograms, and analyze it using the machine learning model, allowing for real-time processing.

DogAlone aims to strengthen interaction between users and their pets by analyzing the pet's behavior and emotions, providing valuable insights.

iii) Location of source code

Main Server

:<https://github.com/pata1202/DogAlone-Your-Dog-s-Caretaker/tree/main/backend>

Audio Analysis Server

:<https://github.com/pata1202/DogAlone-Your-Dog-s-Caretaker/tree/main/backend/audioserver>

iv) Class components

a. Main Server (Node.js-based)

[DogAlone/backend/config/]

- db.js: This file sets up and initializes the MySQL database connection, making the connection available for use in other modules.
- firebaseConfig.js: Manages Firebase initialization and configuration, enabling Firebase Authentication and Firestore operations. It loads configuration details from an environment variable file (.env) for use in other modules.

[DogAlone/backend/controllers/]

- authController.js: Handles user data storage in Firestore, manages user login through Firebase Authentication, returns JWT tokens, and provides an API for retrieving user information.

[DogAlone/backend/routes/]

- routes.js: Defines API routes for user registration, login, and information retrieval, and connects these routes to the controllers.

[DogAlone/backend/server/]

- server.js: Responsible for running the server and processing API requests.
- audioStreamServer.js: Handles real-time audio stream processing, converts Base64 data into WAV files, and

sends the data to the AI server for analysis. It also receives the results from the AI server and transmits them back to the client.

[DogAlone/backend/utills/]

- add.js: Stores the analysis results received from the AI server in the MySQL emotion table, processing the data when an "audioResult" event occurs.
- recommendation.js: This is a file that provides recommendation services based on the most recent emotion.
- report_daily.js: This is a file that provides the frequency of each emotion for a specific date and generates a crying report based on the emotion with the highest frequency.
- report_weekly.js: This is a file that provides the frequency of each emotion for a specific week and generates a crying report based on the emotion with the highest frequency.
- report_monthly.js: This is a file that provides the frequency of each emotion for a specific month and generates a crying report based on the emotion with the highest frequency.

b. Audio Server (Flask-based)

[DogAlone/AI/trainmodel/]

- audio_analyzer.py: Uses the AI model (dog_emotion_model.h5) to analyze audio data, generate spectrograms, and predict emotional states. It provides analysis results and prediction probabilities through the /analyze_audio.js endpoint.
- audio_data: A folder where real-time audio data is temporarily stored. The data is deleted after analysis to manage storage space.

E. Module 3: AI

i) Purpose

The AI is responsible for data preprocessing, model creation, and data analysis. Its primary role is to process audio data from dogs and analyze it to determine their emotional state. The collected audio data is converted into spectrograms for AI model training. The model, built using TensorFlow, adopts a Convolutional Neural Network (CNN) architecture to effectively extract features from audio data and accurately perform emotion analysis. The analyzed emotion results are stored in a database through JavaScript integration, allowing the app to provide customized solutions based on the dog's emotional state. With this analysis, users can monitor their dog's emotions and respond appropriately.

ii) Functionality

DogAlone collects audio data through a recording feature in the mobile app. The collected data is analyzed by the AI model, and the emotion analysis results are stored. This emotional data is saved in an AWS database and used to generate daily and monthly reports and recommend appropriate solutions.

iii) Location of source code

<https://github.com/pata1202/DogAlone-Your-Dog-s-Caretaker/tree/main/AI>

iv) Class components

[DogAlone/AI/trainmodel/]

- train_image.py: Converts audio files in the "train_wav" folder into spectrogram images and saves them in the "train_png" folder.

- train_model.py: Trains the AI model using data from the "train_png" folder and generates a model file.
 - processed_files.txt: Records previously processed data to manage new data efficiently.
 - dog_emotion_model.h5: AI model file generated using the "train_model.py" script.
- [DogAlone/AI/traindata]
- train_wav: A folder containing audio datasets in WAV format for training. The dataset is categorized into six types: bark, growl, grunt, whimper, howl, and yip, with each category containing 100 data samples.
 - train_png: A folder where the data from the "train_wav" folder is converted and stored in PNG format.

VI. USE CASES

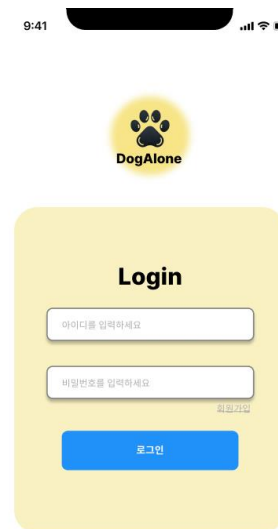
A. Loading



Fig 4. Loading page

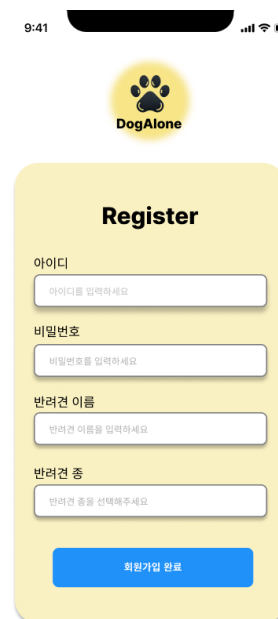
The Loading Page appears when the user launches the app. Once all application elements are loaded, the page automatically transitions to the Login Page.

B. Login



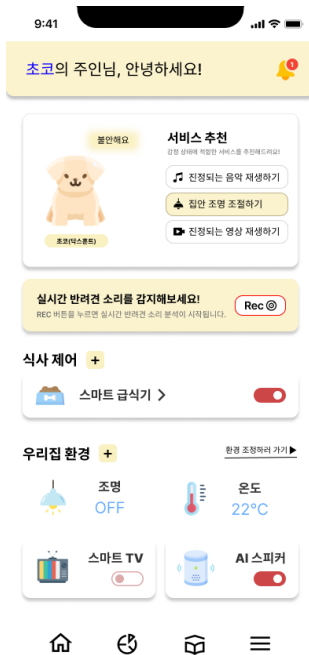
The Login Page allows users to log in by entering their username and password.

C. Register

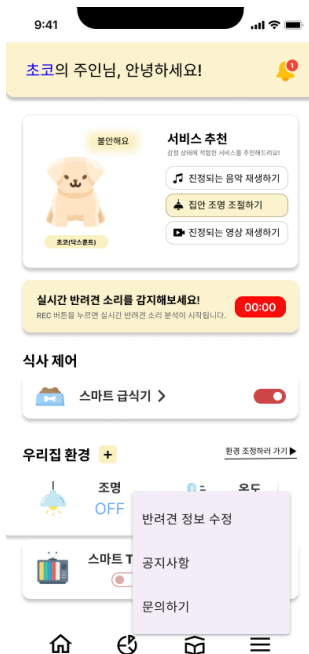


The Register Page enables users to create an account by entering a valid username, password, their dog's name, and breed.

D. Main Page



The Main Page is the central hub of DogAlone's features. It displays the dog's profile registered in the app and allows real-time monitoring of the dog's emotions. Pressing the "Rec" button starts audio recording for emotion analysis. Based on the analysis, users receive recommendations for services such as calming music playback, video streaming, or lighting adjustments. A button to add LG devices is available, allowing users to directly control devices (e.g., on/off or indoor temperature adjustments). Clicking the "Go to Environment Settings" button navigates to the SmartHome Device Page for more detailed controls.



Recording Start: When recording begins, the button changes to a timer to indicate that recording is in progress. Pressing the button again stops the recording, and the audio file is automatically uploaded to the server. If barking is detected in the uploaded file, the AI sends a new alert. It analyzes the

sound, reflects the results on the Report Page, and recommends personalized solutions.

Recording Successful: A popup that appears after successful recording, informing the user that the recording has been saved and emotion analysis will begin.

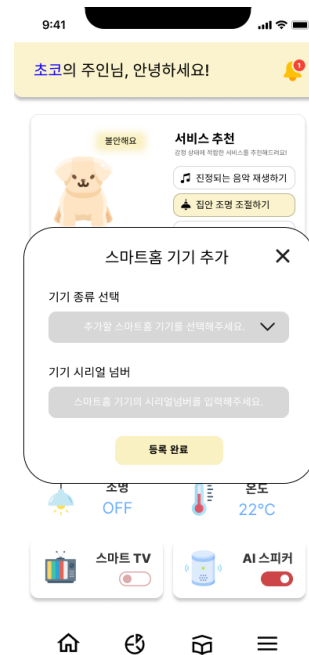
Recording Fail: A popup shown when the app fails to save the recording, indicating that the recording could not be stored, and analysis cannot proceed.

New Alarm: A popup notifying the user of unacknowledged barking incidents. Clicking the bell icon at the top displays the popup.

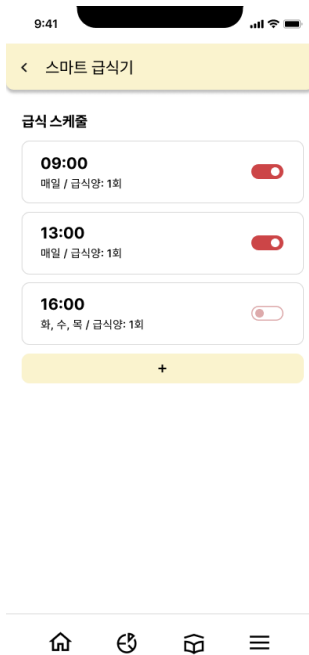
Service Recommendation 1 (Music): A popup notifying users that calming music has started playing via the AI speaker.

Service Recommendation 2 (Light): A popup for adjusting indoor lighting settings.

Service Recommendation 3 (Video): A popup notifying users that calming videos have started playing on the TV or StandbyMe.

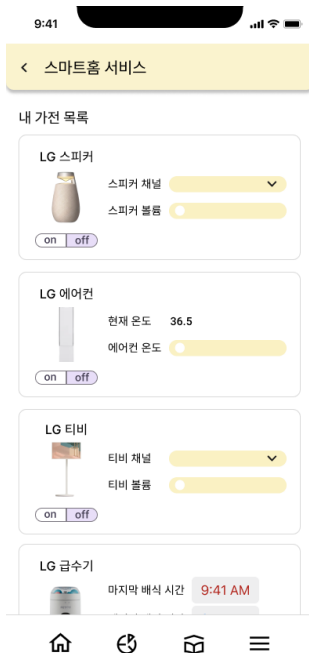


Adding SmartHome Devices: A popup where users can add LG smart home devices. By selecting the device type and entering the serial number, registration is completed.



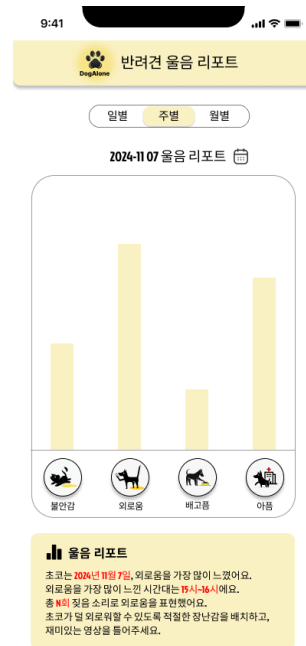
Smart Food Machine Page: A page for adjusting the schedule and feeding frequency of the smart feeder. Users can add a new schedule by clicking the "+" button and selecting the time, day, and feeding frequency. Added schedules can be toggled on or off easily.

E. SmartHome Page



The SmartHome Device Page allows users to manage LG devices in detail, displaying a list of registered appliances at a glance. Each device can be turned on/off and controlled using its respective features.

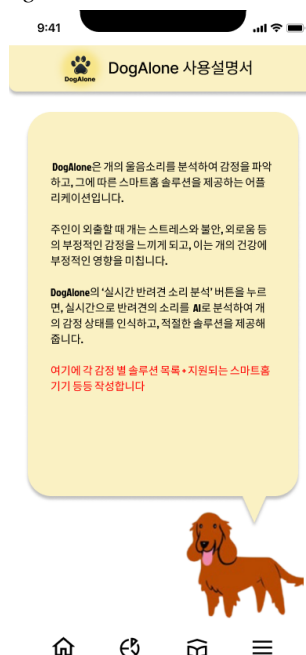
F. Report Page



The Report Page provides an overview of the dog's barking analysis results through graphs and summary reports. The displayed emotions include excitement, fear, satisfaction, anxiety, loneliness, and pain, represented by a graph based on the frequency of each emotion. A summary report below the graph highlights the most frequent emotions, their timeframes, and suggested solutions. This helps systematically understand the dog's emotional state and respond appropriately, enhancing the dog's overall well-being.

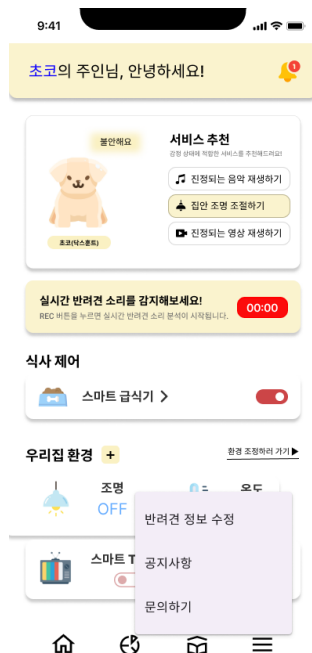
Calendar: A popup allowing users to select a date within the past six months. Users can view daily, weekly, or monthly historical graphs for the selected date.

G. Document Page



The Document Page displays the app's user guide, providing detailed explanations to help users easily understand and utilize its features. The guide includes DogAlone's purpose, key features, smart home solutions for each emotion, and a list of supported smart home devices.

H. Menu



Menu List: Users can choose from additional tasks such as editing their dog's profile, checking notices, or contacting support.

Modify Dog Profile: A popup for editing the dog's name and breed entered during registration.

Notice: A popup displaying updates on new features and the latest announcements.

Contact Us: A popup providing an email address for user inquiries about the app.

VII. DISCUSSION

Throughout the app development process, I faced numerous challenges and learned a great deal. The difficulties could largely be divided into two areas: frontend/backend/AI integration and AI development.

First, after completing the individual development of the frontend, backend, and AI, we began the integration process. However, once we started integrating, it became evident that the implementation methods we had envisioned were different, and the data formats being transmitted were inconsistent. Fixing these discrepancies took a significant amount of time. The most challenging aspect was connecting the Node.js server with the Flask server. Since the environment where the AI model was stored differed from the one running the Flask server, we had to repeatedly save and load the AI model. Resolving errors caused by these environmental differences required a lot of time and effort,

but it taught me various debugging techniques and underscored the importance of proper environment configuration.

Additionally, transferring the AI model's results from the backend to the frontend involved serializing the data into JSON format and interpreting it on the frontend to update the UI. This process was more complex than anticipated. Solving issues like corrupted or misinterpreted data highlighted the critical importance of thorough design for data formats and communication protocols.

Although the integration process was far more complicated and challenging than I had expected, it taught me the importance of communication among team members and meticulous initial planning. For future projects, I resolved to clearly define the data flow, communication methods, and implementation details from the early stages and to share feedback regularly throughout the development process. This experience not only helped me tackle technical challenges but also served as a valuable lesson in the importance of collaboration and communication.

Our team also put a great deal of effort into developing the AI model to analyze dog barks and predict emotional states. The first significant challenge was establishing criteria for distinguishing and categorizing the barks. While dog barks might seem like simple sounds, we found through phonetic studies and research papers that acoustic characteristics such as pitch, duration, and frequency could carry different meanings. Based on this, we categorized dog barks into six emotional states: excitement (bark), fear (growl), satisfaction (grunt), anxiety (whimper), loneliness (howl), and pain (yip).

Data collection was another major challenge. Since publicly available datasets on dog barks were scarce, we relied on Kaggle datasets and extracted sounds from YouTube videos to create our dataset. By converting these sounds into spectrograms, we could more clearly identify differences in pitch, duration, and frequency. This allowed us to design and train a CNN model effectively.

The data labeling process also required significant effort. Connecting the physical properties of the sounds to their emotional meanings from a phonetic perspective was particularly critical. Unlike human language, dog barks represent a much broader emotional spectrum. Thus, we carefully analyzed the context of the sounds and their frequency characteristics to organize the data systematically.

This project made me realize the paramount importance of data quality in building AI models. Working with specialized data like dog barks highlighted the direct impact of understanding acoustic features on model performance. Using spectrograms to visualize the data further emphasized the critical role of phonetic analysis. Without a phonetic approach, there was a risk of misinterpreting barks as mere noise or overlooking subtle emotional nuances. Through the process of labeling data, I gained a deeper understanding of the essence of the sounds and learned how to reflect this in the AI model. This project reinforced the idea that combining phonetics with AI technology leads to more precise and meaningful outcomes.

By the end of the project, I felt that I had not only advanced my research and technical skills but also acquired the attitude and know-how needed to push both fields forward. This

experience has inspired me to explore the potential of voice data as a new means of communication. I aim to develop technologies that address various problems by leveraging audio data.

REFERENCE

- [1] 매일 6 시간씩 나홀로 집에...“반려동물도 외로워요”:
<http://m.kwangju.co.kr/article.php?aid=1620641700720429314&page=2>
- [2] 반려동물 앱 42 개로 확인한 집사 분석:
<https://blog.dighty.com/trend/?bmode=view&idx=110937732>
- [3] [주간 데이터동향] 반려가구 600 만 시대, 펫 케어 앱 누가 얼마나 이용하고 있나?(2024) :
<https://www.banronbodo.com/news/articleView.html?idxno=23095>
- [4] App Store(iPet) :
<https://www.chosun.com/economy/science/2024/01/11/BK5742F32BATLKAGXV7T7KA7BU/>
- [5] <https://apps.apple.com/kr/app/ipet/id1463256910>
- [6] Google playstore (Bowwow- 애견통역기, 애견사진관, 애견노래방):
<https://play.google.com/store/apps/details?id=kr.co.appsolution.bowwow>
- [7] LG 유플러스, 반려동물 케어 스마트홈 '펫토아' 출시:
<https://www.hankyung.com/article/202210057295i>
- [8] Google playstore (Barkio: 강아지 모니터&카메라):
<https://play.google.com/store/apps/details?id=com.tappytaps.android.barkio&hl=ko>
- [9] Communication in Dogs. Marcello Siniscalchi. Department of Veterinary Medicine, Section of Behavioral Sciences and Animal Bioethics, University of Bari “Aldo Moro”, 70121 Bari, Italy, 2018
- [10] The vocal communication of canines. Seong Chan Yeon. College of Veterinary Medicine, Institute of Animal Medicine, Bioacoustics Bank, Gyeongsang National University, Jinju, Republic of Korea. 2007
- [11] A barking emotion recognition method based on Mamba and Synchrosqueezing Short-Time Fourier Transform. Choujun Yang, Shipeng Hu, Lu Tang, Rui Deng, Guoxiong Zhou, Jizheng Yi, Aibin Chen. School of Computer Sciences and Mathematics, Central South University of Forestry and Technology, No. 498 shaoshan South Road, Tianxin District, ChangSha, 4100000, HuNan, China, School of Advanced Interdisciplinary Studies, Central South University of Forestry and Technology, No. 498 shaoshan South Road, Tianxin District, ChangSha, 4100000, HuNan, China, School of Electronic Information and Physics, Central South University of Forestry and Technology, No. 498 shaoshan South Road, Tianxin District, ChangSha, 4100000, HuNan, China, 2024
- [12] Dog emotion recognition from images in the wild: DEBIw dataset and first results. Fernanda Hernández-Luquin, Hugo Jair Escalante, Luis Villaseñor-Pineda, Verónica Reyes-Meza, Luis Villaseñor-Pineda, Humberto Pérez-Espinosa, Verónica Reyes-Meza, Hugo Jair Escalante, Benjamin Gutierrez-Serafín, ACI '22: Proceedings of the

Ninth International Conference on Animal-Computer Interaction, 2023

[13] A Study on Dog-emotion judgment method Based on Deep Learning. Mingu KimO, Seha Kim, Yujeong Go, Hyunseo Lee, Joonho Park(Corresponding Author). Department of Aeronautical Software Engineering, Kyungwoon University. 2022