



JOURNAL ON COMMUNICATIONS

ISSN:1000-436X

Scopus®

REGISTERED

www.jocs.review

PetFeed Planner: An Integrated AI and IoT Platform for Automated Food Analysis and Personalized Pet Nutrition Management

Kimbohyun[1], Kim semin[2], Byeongchang Kang[3], Khant Ko Ko[4], Seungjae Lee[5*]

[1,2,3,4]*Department of Computer Engineering, Sunmoon University, Republic of Korea*

Abstract: *The rapid growth in pet-owning households has significantly increased the demand for systematic pet health and nutrition management. However, pet owners frequently rely on subjective experience rather than scientific guidelines when determining daily feeding amounts, often leading to issues such as obesity, malnutrition, or nutritional imbalance. Furthermore, general users frequently struggle to interpret complex nutritional information and ingredient lists on commercial pet food packaging. To address these challenges, this study proposes PetFeed Planner, a comprehensive mobile platform that integrates artificial intelligence (AI) and the Internet of Things (IoT) for automated food analysis and personalized feeding management. The proposed system empowers users to analyze pet food products via image recognition, receive tailored feeding recommendations based on individual pet profiles, and seamlessly track longitudinal feeding records. Additionally, an IoT-equipped smart feeder with integrated load cell sensors automatically measures actual consumption and synchronizes real-time data with the mobile application for advanced monitoring and statistical analysis. The platform was engineered using React Native for the mobile frontend, Go for the backend server, Python FastAPI for AI-driven analysis services, Oracle Database for persistence, and an ESP32 microcontroller with HX711 sensors for IoT hardware integration. Throughout the development lifecycle, critical software engineering hurdles—such as AsyncStorage structure optimization, centralized exception handling, and the harmonization of manual versus IoT-generated feeding records—were successfully resolved to enhance system stability, maintainability, and data consistency. Comprehensive experimental testing and a user study involving 36 participants (yielding an overall satisfaction score of 4.00 out of 5.00) verified the robust performance and practical utility of core features. Ultimately, this research demonstrates the high feasibility and effectiveness of fusing AI and IoT technologies to build an advanced, user-centric pet feeding management ecosystem.*

Keywords: **Pet Nutrition Management, AI-Driven Food Analysis, IoT Smart Feeder, Image-Based Food Analysis**

1. Introduction

1.1 Background

The global pet industry has experienced substantial growth over the past decade, accompanied by a steady increase in the number of households keeping companion animals. As pets are increasingly regarded as family members, pet owners have become more attentive to their animals' nutrition, health, and overall well-being. Among the various aspects of companion animal care, appropriate nutritional management and regular feeding are particularly important because they are closely associated with maintaining healthy body conditions and supporting long-term health.

Despite growing awareness of pet health, determining an appropriate daily feeding amount remains challenging for many pet owners. In practice, feeding decisions are often based on personal experience, intuition, or generalized recommendations rather than on individualized nutritional requirements. Inappropriate feeding quantities may

consequently result in excessive weight gain, inadequate nutrient intake, or other nutritional imbalances. Therefore, a systematic approach that considers the individual characteristics of each pet is needed to support more appropriate feeding management.

Another challenge is the difficulty of interpreting commercial pet food information. Pet food products typically provide ingredient lists, guaranteed nutritional analyses, feeding guidelines, and other product information. However, these data can be difficult for general users to interpret, particularly when they need to determine whether a particular product is appropriate for an individual pet. The problem becomes more complex when factors such as age, body weight, activity level, body condition, and health status need to be considered simultaneously.

Recent advances in artificial intelligence (AI), computer vision, and the Internet of Things (IoT) provide opportunities to address these challenges. AI-based image analysis can assist in extracting and interpreting information from product images, while intelligent recommendation techniques can support personalized decision-making based on user- and pet-specific information. At the same time, IoT devices equipped with sensors can continuously collect physical data and transmit it to software platforms in real time. These capabilities provide a technological foundation for developing integrated systems that connect food information, individualized recommendations, and actual feeding behavior.

However, these technologies are often implemented as separate functions. Existing pet management applications generally focus on record keeping, scheduling, or basic health management, whereas smart feeders primarily emphasize automated food dispensing and remote control. Consequently, food analysis, personalized feeding recommendations, feeding records, and actual feeding measurements are not always integrated into a single management framework.

To address this gap, this study proposes ‘PetFeed Planner’, an AI- and IoT-integrated platform for personalized pet food analysis and feeding management. The proposed platform combines image-based food analysis, pet profile management, personalized feeding recommendations, feeding history, feeding alarms, and sensor-based feeding measurement within a unified mobile application. By connecting recommended feeding information with actual feeding data collected through an IoT smart feeder, the system is designed to support more systematic and data-driven pet feeding management.

1.2 Motivation

Existing pet management applications provide useful functions such as feeding schedules, feeding records, vaccination information, and general health diaries. However, these applications often provide limited analytical support for interpreting pet food information or generating individualized feeding recommendations. In addition, feeding records may depend primarily on manual user input, which can introduce inconsistencies or omissions in the recorded data.

Commercial smart feeders provide another useful approach by automating food dispensing according to predefined schedules. Nevertheless, their primary focus is generally on the physical delivery of food rather than on comprehensive nutritional management. In particular, automated feeding systems may not directly incorporate information about the nutritional characteristics of a food product, the individual characteristics of a pet, or historical feeding patterns when determining appropriate feeding strategies.

These limitations motivated the development of an integrated system that connects food analysis, personalized recommendation, feeding management, and IoT-based

measurement. In the proposed system, users can obtain information about pet food through image-based analysis rather than manually entering extensive product information. Pet-specific attributes, including species, age, body weight, body condition score (BCS), and relevant health information, are incorporated into the feeding management process. Established nutritional concepts, including Resting Energy Requirement (RER) and Daily Energy Requirement (DER), provide a basis for generating individualized feeding recommendations.

The integration of IoT technology further enables the system to monitor actual feeding activity. The proposed smart feeder uses an ESP32 microcontroller and weight sensors to measure feeding amounts and transmit the collected data to the backend server. These data are subsequently synchronized with the mobile application, allowing users to compare planned or recommended feeding amounts with actual feeding records. Such integration can reduce reliance on manually entered records and provide a more continuous view of feeding behavior.

From a software engineering perspective, the development of an integrated platform also introduces challenges associated with data consistency, local storage management, exception handling, and the integration of heterogeneous data sources. In particular, manual feeding records and automatically generated IoT records may describe the same feeding event, potentially resulting in duplicate or inconsistent information. Therefore, mechanisms for centralized storage management, unified exception handling, and record integration are required to maintain system reliability and data consistency.

Accordingly, the motivation of this study is not only to develop a functional pet feeding application but also to demonstrate how AI, mobile computing, and IoT technologies can be integrated into a unified architecture for personalized pet nutrition management. The proposed system aims to provide users with an end-to-end workflow extending from pet food information analysis and individualized feeding recommendations to actual feeding measurement and historical monitoring.

1.3 Contributions

The main contributions of this study are summarized as follows:

1. Integrated AI- and IoT-Based Platform:

An end-to-end pet feeding management platform was designed and implemented by integrating a mobile application, backend services, AI-based food analysis, database management, and an IoT smart feeder.

2. Personalized Feeding Management:

A feeding recommendation module was developed using pet-specific information, including species, age, body weight, body condition, and health-related attributes, together with established nutritional concepts.

3. Image-Based Pet Food Analysis:

An AI-based image analysis function was implemented to assist users in obtaining and interpreting pet food information from product images.

4. IoT-Based Feeding Measurement:

An ESP32- and load-cell-based smart feeder was developed to measure actual feeding amounts and transmit the collected data to the backend system in real time.

5. Integration of Manual and Automated Records:

A data integration mechanism was implemented to reconcile manually entered feeding records with IoT-generated records and reduce duplicate or inconsistent feeding data.

6. Software Architecture and Reliability Improvements:

Centralized local-storage management and unified exception-handling mechanisms were introduced to improve maintainability, reliability, and scalability as the application evolved.

7. System and User Evaluation:

The implemented system was evaluated through functional testing and a user study involving 36 participants to examine the feasibility, usability, usefulness, and user satisfaction of the proposed platform.

2. Related Works

2.1 AI-Based Image and Multimodal Information Analysis

Image-based information analysis has become an important component of intelligent application systems. Conventional computer vision approaches, including convolutional neural networks (CNNs), have demonstrated their effectiveness in extracting visual features and performing image classification and recognition tasks. More recently, multimodal large language models (MLLMs) have extended these capabilities by jointly processing visual and textual information, enabling systems to interpret images and generate contextual descriptions or structured information [1].

Yin et al. [1] reviewed the development of multimodal large language models and highlighted their ability to integrate information from multiple modalities, particularly images and natural language. Such capabilities are relevant to pet food analysis because information required for dietary management is distributed across visual and textual elements, including product packaging, ingredient lists, nutritional information, and feeding guidelines. An AI-based multimodal approach can therefore reduce the burden of manually entering and interpreting product information.

In the context of pet care, however, image analysis should not be considered an isolated recognition task. The extracted food information needs to be combined with characteristics of the individual pet to support meaningful feeding decisions. Accordingly, the present study employs AI-based image analysis as one component of an integrated pet feeding management framework rather than as a standalone image classification system. The proposed approach uses food images to obtain product-related information and subsequently connects the analysis results with pet profiles and feeding management functions.

2.2 Pet Nutrition and Personalized Feeding Management

Appropriate feeding management requires consideration of the nutritional requirements and physiological characteristics of individual animals. The National Research Council (NRC) provides established nutritional guidelines for dogs and cats, including recommendations related to energy and nutrient requirements across different life stages and activity conditions [2]. These guidelines provide an important scientific basis for developing systematic approaches to companion animal nutrition.

Clinical nutrition literature further emphasizes that nutritional management should consider the individual characteristics and health status of animals. Hand et al. [3] describe nutritional assessment and feeding planning as processes that require consideration of the animal's condition, nutritional requirements, and clinical

circumstances. These principles indicate that feeding recommendations should not rely solely on generalized feeding quantities but should incorporate individual characteristics whenever possible.

Based on these principles, personalized feeding management can be formulated using parameters such as species, age, body weight, activity level, and body condition. In particular, energy requirement calculations, including Resting Energy Requirement (RER) and Daily Energy Requirement (DER), provide useful foundations for estimating appropriate energy intake [2], [3]. Body Condition Score (BCS) can also be used as an important indicator when assessing whether an animal's current body condition is appropriate and when adjusting feeding strategies.

Recent application-oriented research has attempted to incorporate these concepts into digital pet management systems. Choi et al. [7] proposed an app-based pet obesity management feeding system that combines feeding management with activity measurement and uses activity information to support feeding recommendations. This approach demonstrates the potential of integrating digital applications and sensor-derived behavioral information for pet feeding management.

Nevertheless, existing approaches generally focus on specific aspects of the problem, such as feeding automation, activity monitoring, or obesity management. They do not necessarily provide an integrated workflow connecting pet food information analysis, individualized nutritional recommendations, feeding records, and actual food consumption. The present study addresses this limitation by incorporating pet-specific information and nutritional principles into a unified feeding management platform.

2.3 IoT-Based Pet Feeding and Management Systems

The Internet of Things (IoT) has enabled physical devices to collect, transmit, and process data through network-connected systems. In companion animal care, IoT technologies have been applied to automatic feeders, activity monitoring, remote surveillance, and health-related applications. Pico-Valencia and Holgado-Terriza [4] systematically reviewed research on the Internet of Pets and reported that IoT technologies are increasingly being applied to feeding, monitoring, surveillance, and other aspects of companion animal care. Their review indicates that remote feeding and monitoring remain important areas of IoT-based pet care research.

Several studies have specifically investigated IoT-enabled automatic feeding systems. Lee and Kim [6] presented an implementation of a smart IoT system with an automatic pet feeder, demonstrating the feasibility of connecting an automatic feeding device with network-based monitoring and control. Similarly, Castillo Rangel et al. [5] developed an IoT-based automatic pet feeding system that enables users to monitor feeding through a mobile device and incorporates sensors for managing feeding quantities. These studies demonstrate the feasibility of combining automatic feeding hardware, sensors, network communication, and mobile applications.

Although such systems improve convenience and enable remote management, their primary emphasis is generally placed on automatic food dispensing and monitoring. The relationship between the physical feeding process and higher-level nutritional analysis or personalized dietary recommendations remains relatively limited. In particular, feeding measurements collected from IoT devices are not always integrated with food-product analysis and individualized nutritional planning.

The present study extends this direction by integrating an IoT smart feeder with AI-assisted food analysis and personalized feeding management. The proposed feeder uses an ESP32 microcontroller and load-cell-based weight measurement to obtain actual feeding

amounts and transmit the data to the backend server. These measurements are then integrated with feeding records maintained by the mobile application, enabling users to monitor the relationship between recommended and actual feeding behavior.

2.4 Mobile Application and System Architecture

Mobile applications provide an effective interface for integrating multiple pet-care functions into a single user environment. In the proposed system, React Native was adopted as the mobile application framework, enabling the development of a cross-platform user interface [8]. The application incorporates pet profile management, food analysis, feeding records, feeding alarms, feeding history, and AI-assisted services within a unified environment.

For application navigation, Expo Router was adopted to organize the application into modular screens and navigation flows [9]. Its file-based routing approach supports the separation of application functions into logically organized screens, which is consistent with the modular architecture adopted in the proposed system.

Local data management is another important consideration in mobile applications. AsyncStorage provides asynchronous persistent key-value storage for React Native applications and can be used for storing application-level data locally [10]. In the early development of the proposed system, however, independently defined storage keys across multiple screens resulted in duplication and inconsistent data management. Therefore, the present study introduced centralized storage-key management to improve maintainability and reduce potential conflicts between user and pet-related data.

These implementation technologies provide the software foundation of the proposed platform, but they are not themselves the primary research contribution. Rather, they support the integration of AI analysis, personalized feeding management, and IoT-based measurement within a unified architecture.

2.5 Research Gap

The reviewed studies demonstrate significant progress in three related areas: AI-based multimodal information analysis [1], scientific approaches to companion animal nutrition [2], [3], and IoT-enabled pet feeding and monitoring systems [4]–[7]. However, these research directions have largely developed independently.

AI-based approaches provide opportunities to extract and interpret information from images, but food analysis alone does not determine whether a product or feeding amount is appropriate for a specific animal. Nutritional management research provides established principles for determining feeding requirements, but these principles are not necessarily integrated with real-time feeding data. IoT-based smart feeders can automate food dispensing and collect feeding-related measurements, but their functionality is often centered on device control rather than comprehensive nutritional management.

Therefore, a research gap remains in the integration of **AI-based food information analysis, personalized nutritional recommendation, and IoT-based measurement of actual feeding behavior** within a single platform. The proposed PetFeed Planner addresses this gap by combining these components with a mobile application and backend infrastructure. The resulting architecture enables the complete workflow from food-product analysis and individualized feeding recommendations to automated feeding measurement, record synchronization, and historical monitoring.

This integrated approach distinguishes the proposed system from existing solutions and provides a foundation for more data-driven and personalized pet feeding management.

3. System Architecture

3.1 Overall System Architecture

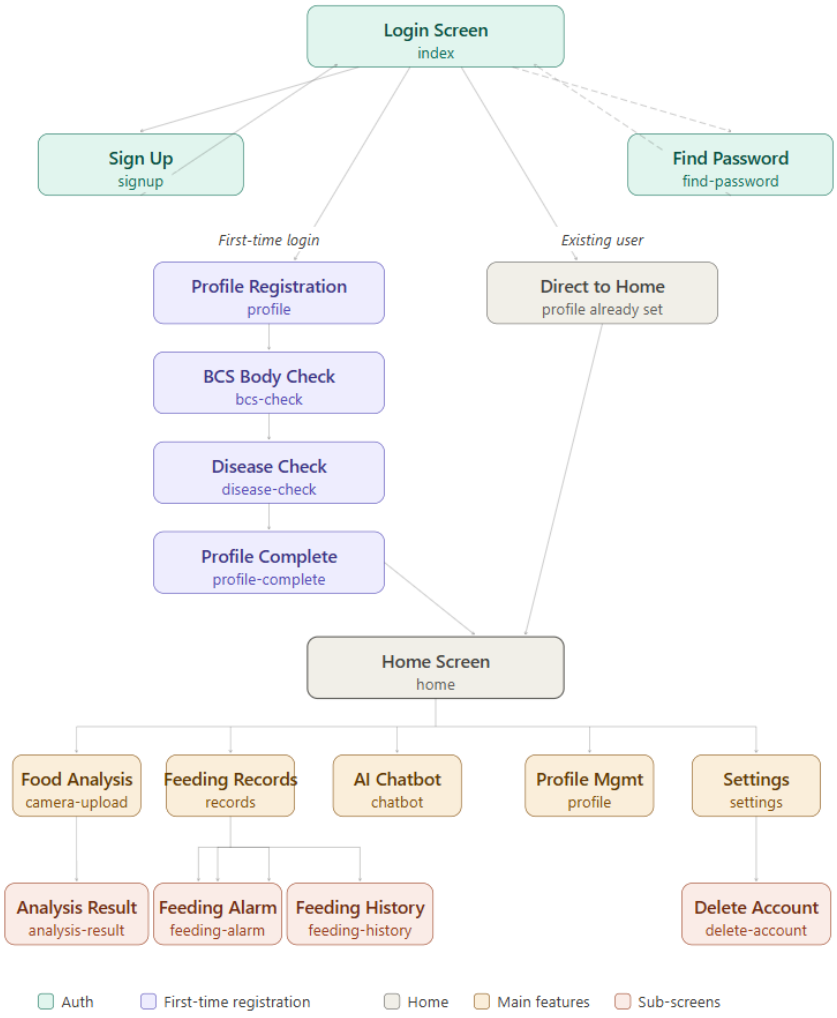


Figure 1. Overall System Architecture

The proposed *PetFeed Planner* was designed as an integrated platform for AI-assisted pet food analysis, personalized feeding management, and IoT-based monitoring. The overall architecture consists of five major components: (1) a mobile application, (2) a backend server, (3) an AI analysis server, (4) a relational database, and (5) an IoT-based smart feeder. These components were designed as separate but interconnected modules to support functional separation, data consistency, and future system expansion.

The mobile application serves as the primary user interface and provides access to the major functions of the system, including user authentication, pet profile management, pet food analysis, feeding recommendations, feeding records, feeding alarms, feeding history, and AI chatbot services. The application communicates with the backend server through RESTful APIs, allowing user and pet-related data to be managed independently from the presentation layer.

The backend server was implemented using Go and functions as the central coordination layer of the system. It provides RESTful APIs for user management, pet profile management, feeding records, and communication with external services. By separating backend services from the mobile application, the architecture reduces the dependency between the user interface and business logic and facilitates future modification or expansion of individual services.

A dedicated AI analysis server was implemented using Python FastAPI. The server receives pet food images from the mobile application and communicates with the Gemini API to perform AI-assisted image and information analysis. The resulting information is returned to the application through the backend service layer. This separation enables AI-related processing to be maintained independently from the core application logic.

The Oracle Database serves as the persistent data-management layer. It stores user accounts, pet profiles, feeding records, AI analysis results, chatbot histories, and IoT-generated feeding measurements. A relational database structure was adopted to maintain data integrity and reduce unnecessary duplication among related entities.

The IoT subsystem consists of an ESP32 microcontroller, an HX711 load-cell amplifier, and weight sensors. The weight sensors measure changes associated with food consumption, while the ESP32 processes the measurements and transmits the collected data to the backend server through Wi-Fi. The transmitted measurements are subsequently stored in the database and synchronized with the mobile application.

As illustrated in Figure 1, the resulting architecture establishes a data flow between the physical feeding environment and the digital management platform. Food-related information is analyzed through the AI service, pet-specific information is maintained through the application and database, and actual feeding activity is obtained through the IoT smart feeder. The integration of these components allows the system to connect planned feeding management with observed feeding behavior.

3.2 Screen Flow and Functional Module Design

The mobile application was designed around a modular screen-flow architecture to provide users with an organized and consistent interaction process. The overall user flow begins with account authentication, followed by pet registration and access to the main feeding-management functions.

After successful authentication, users can register individual pets and provide information required for personalized feeding management. The pet profile serves as the central source of pet-specific information used by subsequent functions. Users can then access the major functional modules, including feeding records, feeding history, feeding alarms, AI-based food analysis, pet profile management, and the AI chatbot.

The application navigation was implemented using Expo Router. Related functions were grouped into logically independent screens to minimize unnecessary navigation and simplify interaction between modules. For example, feeding records and feeding history are connected as related functions, while food analysis and pet profile management provide information required for personalized feeding decisions.

3.3 Frontend Architecture Design

The frontend architecture was designed using a modular and layered approach. React Native was used to implement the mobile application, while Expo Router was adopted for screen navigation. Functional components were separated according to their respective responsibilities to reduce coupling between independent modules.

The frontend architecture can be conceptually divided into three layers: the presentation layer, the service layer, and the local-data management layer. The presentation layer is responsible for rendering application screens and handling user interactions. The service layer manages communication with backend APIs and provides interfaces for functions such as authentication, feeding-record management, AI analysis requests, and IoT-related data retrieval. The local-data management layer handles persistent application data and user-specific settings.

AsyncStorage was used to maintain local application data, including feeding records, alarm configurations, and application settings. During system development, storage keys were subsequently centralized to avoid duplicated or inconsistent key definitions across individual screens. User- and pet-related data were also logically separated to reduce the possibility of conflicts when multiple pets or accounts were managed.

This modular structure improves maintainability by separating presentation logic from application services and data management. It also provides a foundation for future functional expansion without requiring substantial modification of unrelated application modules.

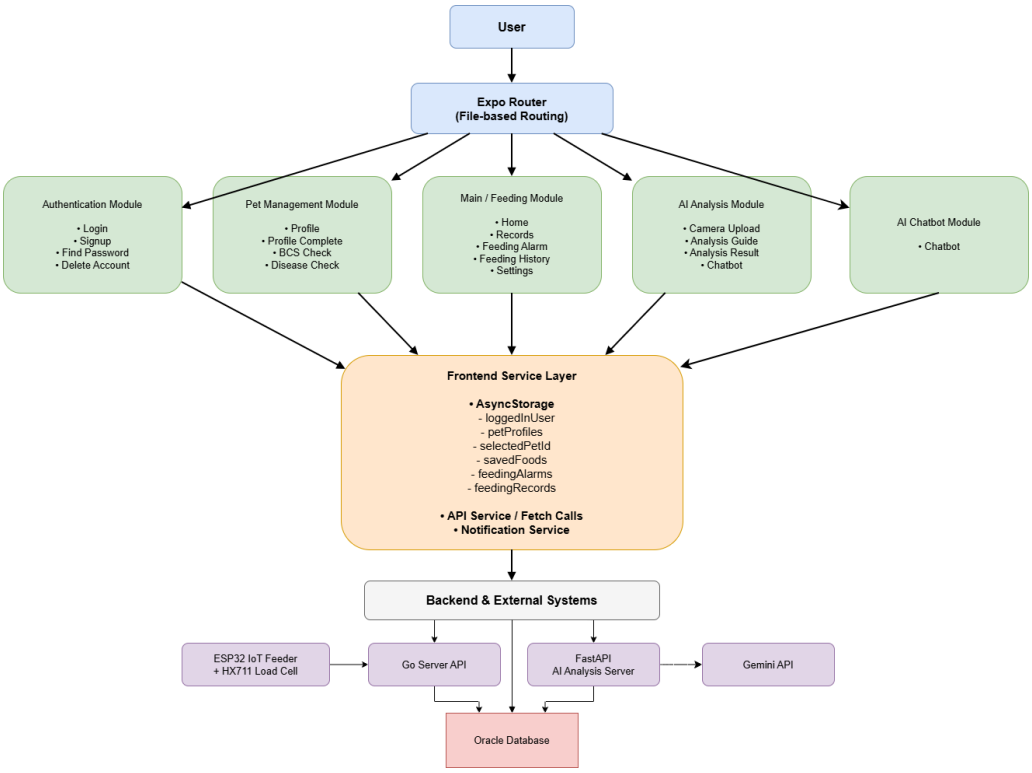


Figure 2. Frontend Architecture

3.4 Database Design

The database was designed to provide persistent and consistent management of the information generated throughout the pet feeding workflow. Oracle Database was used as the primary relational database management system.

The database contains entities associated with users, pets, feeding activities, AI analysis, chatbot interactions, and IoT measurements. The USER table stores account-related information, while the PET table stores individual pet profiles. Feeding-related tables maintain scheduled and completed feeding records, and the CHAT_HISTORY table

stores interactions associated with the AI chatbot. Separate tables are used to maintain weight measurements generated by the IoT smart feeder.

The database structure was designed according to relational database principles to reduce redundant data and maintain relationships among users, pets, and feeding activities. In particular, pet-specific information is associated with feeding records so that feeding history can be interpreted in the context of the corresponding pet.

IoT-generated measurements are stored separately from manually entered feeding records. This separation allows the system to preserve the original data source while providing an integration mechanism at the application or service layer. Consequently, measurements generated by the smart feeder can be compared with manually recorded feeding events and incorporated into the unified feeding history.

The database therefore functions not only as a storage layer but also as the central data foundation connecting the AI analysis, mobile application, feeding-management functions, and IoT subsystem.

4. System Implementation

4.1 User Authentication and Account Management

The user authentication module was implemented to provide secure and individualized access to the PetFeed Planner platform. The module supports user registration, login, password modification, and account deletion.

During registration, users create an account using an email address and password. The authentication information is validated through the backend API before the account is created and associated with the corresponding user profile. During login, the application submits the user's credentials to the backend server, which verifies the authentication information and returns the associated user data.

After successful authentication, user-specific information can be accessed throughout the application. This user-centered structure is particularly important because feeding records, pet profiles, feeding schedules, and AI analysis results need to be associated with the appropriate user and pet.

The account management module also supports password modification and account deletion. These functions allow users to maintain their account information and remove their stored information when necessary.

4.2 Pet Profile Management

The pet profile module provides the basic information required for personalized feeding management. Users can register individual pets and maintain information including species, age, body weight, Body Condition Score (BCS), disease history, and other relevant health-related attributes.

The registered pet profile serves as a central information source for subsequent feeding-management functions. In particular, pet-specific characteristics are used when generating feeding recommendations based on nutritional requirements. Users can also update profile information when the pet's condition changes, allowing subsequent feeding-management processes to use the most recent available information.

The use of an individual pet profile also enables the system to distinguish feeding records when multiple pets are associated with the same user account. Consequently, feeding

history and recommendations can be managed at the pet level rather than solely at the user level.

4.3 AI-Based Pet Food Analysis

The AI-based food analysis module enables users to obtain information about pet food products from images. Instead of requiring users to manually enter product information, the application sends a captured or selected food-product image to the AI analysis service.

The AI analysis service is implemented as a separate Python FastAPI server. When an image is submitted, the server communicates with the Gemini API to perform image-based analysis and generate information related to the pet food product. The analysis results are subsequently returned to the application through the service architecture described in Section 3.

The purpose of this function is to reduce the effort required to interpret complex product information and provide users with accessible information that can be considered during feeding management. The analysis module is therefore connected to the broader personalized feeding workflow rather than operating as an independent image-recognition function.

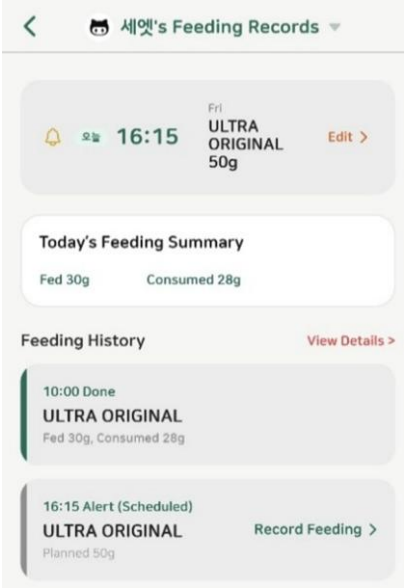


Figure 3. Feeding Record Screen

4.4 Personalized Feeding Recommendation

The personalized feeding recommendation module provides feeding guidance based on the characteristics of an individual pet. The system considers information stored in the pet profile, including species, age, body weight, body condition, and relevant health-related information.

Established nutritional concepts such as Resting Energy Requirement (RER) and Daily Energy Requirement (DER) provide the underlying basis for estimating energy requirements. These values are considered together with individual pet characteristics to generate feeding recommendations.

The recommendation function is designed to support users in determining feeding quantities more systematically than relying solely on generalized product guidelines or

personal judgment. Because pet profiles can be updated, the recommendation process can be repeated when important characteristics such as body weight or body condition change.

The recommendation results are also connected to the feeding-management functions, allowing recommended quantities to be considered when users configure feeding schedules and evaluate subsequent feeding records.

4.5 Feeding Record and Alarm Management

The feeding record module allows users to record and manage feeding activities. Records can be generated either manually by users or automatically through the IoT smart feeder. Integrating these two sources provides a more comprehensive representation of actual feeding activity.

Because the same feeding event may potentially be recorded by both the user and the IoT device, the system applies validation and record-integration logic to identify potential duplicate records. Feeding amounts obtained from sensor measurements are incorporated into the unified feeding history so that actual consumption can be distinguished from manually planned or recorded feeding events.

The feeding alarm module provides scheduled reminders to support consistent feeding routines. Users can specify feeding times, recurrence patterns, and feeding amounts according to their pets' feeding plans. The system supports both one-time and recurring schedules, and notifications are delivered through the mobile application.

By linking feeding schedules with feeding records, the system allows planned feeding activities and subsequent feeding results to be managed within the same platform.

4.6 Feeding History and Statistical Analysis

The feeding history module provides a longitudinal view of feeding activities. Users can review feeding dates, feeding amounts, consumption rates, and historical trends through the mobile application.

The system aggregates accumulated feeding records to provide monthly statistical information. These statistics allow users to examine feeding patterns over time and identify potential deviations from recommended feeding quantities. The visualization of historical records is intended to make feeding information easier to interpret and to support more systematic monitoring.

An important characteristic of this module is that the historical data can include both manually entered records and measurements generated by the IoT smart feeder. Therefore, the module serves as the monitoring layer connecting individual feeding events with longer-term feeding behavior.

4.7 IoT Smart Feeder Implementation

The IoT smart feeder was implemented using an ESP32 microcontroller, an HX711 load-cell amplifier, and a weight sensor. The primary purpose of the device is to measure actual feeding amounts and transmit the measurements to the backend system.

The load-cell sensor detects changes in the weight of the food container. The HX711 module amplifies and converts the sensor signal into a form that can be processed by the ESP32 microcontroller. The ESP32 then processes the measurement data and communicates with the backend server through a Wi-Fi network.

Measured information is displayed locally through an LCD module and is simultaneously transmitted to the backend server using HTTP POST requests. The server receives the transmitted measurements and stores them in the database. These data are subsequently made available to the mobile application for feeding-history and statistical analysis.

Rather than relying only on the nominal dispensing quantity, the proposed system uses changes in measured weight to estimate the actual feeding amount. This approach allows the system to incorporate physical measurements into the digital feeding records and reduces the discrepancy between scheduled feeding quantities and observed feeding activity.

The IoT subsystem therefore establishes a connection between the physical feeding environment and the software platform. Through this integration, the proposed system can continuously collect feeding-related measurements and incorporate them into the user's overall feeding-management workflow.

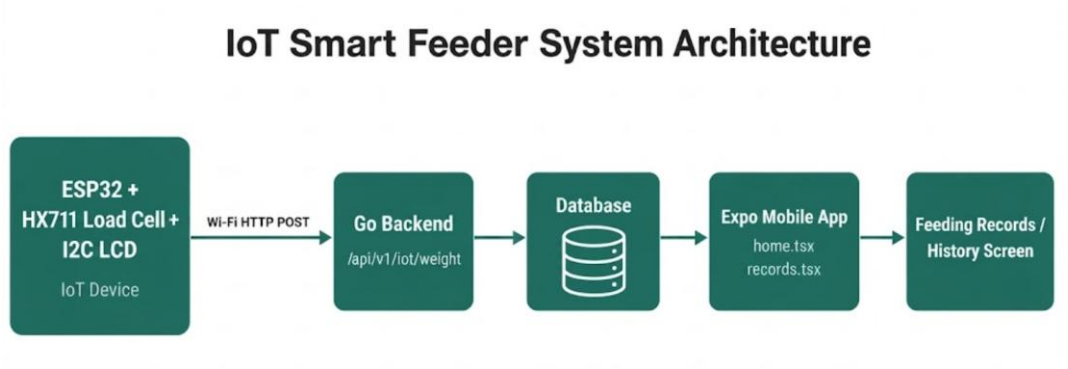


Figure 4. IoT Smart Feeder System Architecture

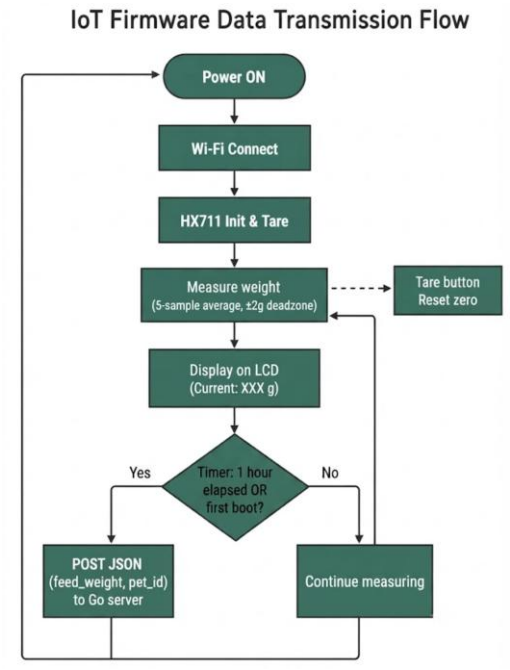


Figure 5. IoT Firmware Data Transmission Flow

5. Development Challenges and System Improvements

During the implementation of PetFeed Planner, several technical challenges emerged from the need to integrate mobile application functions, backend services, AI-based food analysis, and IoT-generated feeding data within a single system. In particular, local data management, exception handling, and the integration of manually entered and automatically measured feeding records required additional architectural improvements. This section describes the major development challenges and the approaches adopted to improve system consistency, reliability, and data integration.

5.1 Centralized Local Storage Management

The initial implementation of the mobile application used screen-specific keys for storing data in AsyncStorage. Although this approach was simple to implement, it resulted in duplicated storage keys, inconsistent naming conventions, and difficulties in tracking locally stored data as the number of functional screens increased.

To address these issues, the storage structure was reorganized using a centralized `storageKeys.ts` module. Storage keys were systematically defined and managed in a single location rather than being independently declared within individual screens. In addition, locally stored data were organized according to user and pet identifiers to reduce the possibility of data conflicts between different accounts or pets.

This improvement simplified local data management and enhanced consistency across application modules. It also established a more maintainable structure for managing persistent client-side data as the number of application functions increased.

5.2 Unified Exception Handling and Request Management

Another implementation challenge involved inconsistent exception handling across different application modules. In the initial implementation, API failures and validation errors were handled differently depending on the screen or service, which could result in inconsistent error messages, repeated API requests, incomplete error reporting, and an unstable user experience.

To improve this behavior, input validation and exception handling were systematically incorporated into the application workflow. Common error-processing logic was centralized where appropriate, while state flags were introduced to prevent unintended duplicate requests during asynchronous operations.

The improved approach provided more consistent responses to API and application errors and reduced unnecessary repeated requests. As a result, the application could maintain more predictable behavior across its major functional modules.

5.3 Integration of Manual and IoT-Generated Feeding Records

The integration of manually entered feeding records with records automatically generated by the IoT smart feeder presented another important challenge. Because the same feeding event could potentially be recorded manually by a user and automatically through the feeder, simply combining the two data sources could produce duplicate records and inaccurate feeding histories.

To address this issue, the system introduced a data-integration mechanism that identifies potentially duplicated feeding events and prioritizes valid IoT-generated measurements when appropriate. Rather than relying solely on the scheduled or nominal feeding amount, the system uses the weight measurements obtained from the smart feeder to determine the actual feeding amount.

This approach enables feeding records originating from different sources to be incorporated into a unified feeding history while reducing the possibility of duplicate entries. It also allows the feeding history and statistical functions to reflect measured feeding activity more accurately.

7. Conclusion and Future Work

This study presented PetFeed Planner, an AI- and IoT-integrated platform for personalized pet food analysis and feeding management. The system combines pet profile management, AI-based food analysis, personalized feeding recommendations, feeding records, alarms, feeding history, and IoT-based feeding measurement within a unified mobile application. It was implemented using React Native, Go, Python FastAPI, Oracle Database, and an ESP32-based smart feeder with HX711 load-cell sensors.

During development, centralized local-storage management, unified exception handling, and an integration mechanism for manual and IoT-generated feeding records were implemented to improve system consistency and reliability. Functional testing confirmed the operation of the major system functions without critical failures. In a user study with 36 participants, the overall mean score was 4.00 out of 5.00, with satisfaction and usefulness rated at 4.15 and 4.10, respectively.

Future work will focus on improving AI analysis latency and recognition accuracy, expanding supported species and food types, developing more dynamic nutritional recommendation algorithms, and enhancing system scalability through cloud-based infrastructure.

Acknowledgments

This research was supported by the MSIT(Ministry of Science ICT), Korea, under the National Program for Excellence in SW, supervised by the IITP(Institute of Information & Communications Technology Planning & Evaluation) in 2026 (No. 2024-0-00023).

REFERENCES

[1] S. Yin, C. Fu, S. Zhao, K. Li, X. Sun, T. Xu, and E. Chen, "A survey on multimodal large language models," National Science Review, vol. 11, no. 12, nwae403, 2024.
[2] National Research Council, Nutrient Requirements of Dogs and Cats. Washington, DC, USA: The National Academies Press, 2006.
[3] M. S. Hand, C. D. Thatcher, R. L. Remillard, P. Roudebush, and B. J. Novotny, Eds., Small Animal Clinical Nutrition, 5th ed. Topeka, KS, USA: Mark Morris Institute, 2010.
[4] P. Pico-Valencia and J. A. Holgado-Terriza, “The Internet of Things empowering the Internet of Pets—An outlook from the academic and scientific experience,” Applied Sciences, vol. 15, no. 4, art. 1722, 2025.
[5] A. V. Castillo Rangel, I. U. Ponce, and I. Soto, “Automatic pet feed system applying IoT,” Mathematical Modelling of Engineering Problems, vol. 11, no. 4, pp. 87–93, 2024.
[6] J. Lee and D. Kim, “Implementation of a Smart IoT System with an Automatic Pet Feeder,” Journal of Digital Contents Society, vol. 22, no. 2, pp. 209–214, Feb. 2021.
[7] M. Choi, C. Choi, S. Song, and J. Oh, “App-Based Pet Obesity Management Feeding System,” Journal of the Korea Institute of Electronic Communication Sciences, vol. 19, no. 1, pp. 275–282, Feb. 2024. [8] Meta Platforms, Inc., “React Native Documentation”, Meta Platforms, Inc., https://reactnative.dev/, Accessed 2026.06.
[9] Expo, “Expo Router Documentation”, Expo, https://docs.expo.dev/router/introduction/, Accessed 2026.06.
[10] React Native Community, “AsyncStorage Documentation”, https://react-native-async-storage.github.io/async-storage/, Accessed 2026.06.