UTM UNIVERSITI TEKNOLOGI MALAYSIA
INTERNATIONAL JOURNAL OF INNOVATIVE COMPUTING
ISSN 2180-4370
Journal Homepage : https://ijic.utm.my/

Design and Implementation of a Transparent and Modular Animal Welfare System Using React Native and Firebase

Nurulhanisa Mohd Ikhsan[1], Marina Md-Arshad[2] & Adlina Abdul-Samad[3*]
Faculty of Computing,
Universiti Teknologi Malaysia
81310 UTM Johor Bahru, Johor, Malaysia
Email: nurulhanisa@graduate.utm.my[1]; marinama@utm.my[2]; adlina6@graduate.utm.my[3]

Submitted: 10/8/2025. Revised edition: 27/11/2025. Accepted: 5/1/2026. Published online: 10/6/2026
DOI: https://doi.org/10.11113/ijic.v16n1.589

***Abstract*—This study presents Pethero KPH UUM, a centralized mobile application designed to streamline stray animal welfare management for Kelab Pencinta Haiwan Universiti Utara Malaysia (KPH UUM). The application addresses key organizational challenges, including inefficient communication, opaque donation processes, and disorganized volunteer coordination. Through interviews, surveys, and a comparative analysis of existing systems, critical limitations in current practices were identified. Developed using React Native, Firebase, and other modern tools, the application integrates key features such as donation tracking, real-time animal updates, volunteer management, and adoption services to enhance transparency and engagement. The Rapid Solo Software Development (RSSD) methodology was employed to ensure a structured, modular, and efficient development process. System testing and user evaluations confirm the platform's usability, effectiveness, and positive impact on operational transparency and community engagement. By digitizing KPH UUM's workflows, Pethero not only optimizes resource management but also cultivates a more proactive and compassionate approach to animal welfare within academic communities. The findings highlight the potential of tailored digital solutions in strengthening non-profit initiatives.**

***Keywords*—Mobile application, animal welfare**

I. INTRODUCTION

The welfare of stray animals, particularly cats, has emerged as a growing concern within higher education institutions [1], including Universiti Utara Malaysia (UUM). These animals are often malnourished, sick, or injured, and rely on the goodwill of students and staff to survive. Thus, Malaysia has enacted The Animal Welfare Act to protect the well-being of these animals while simultaneously promoting their welfare to the public [2].

While digital transformation has revolutionized many sectors, small-scale non-profit and student-led organizations often lack the resources and technical expertise to implement the tailored digital solutions. This gap between technological potential and on-ground implementation represents a critical challenge in maximizing social impact, particularly in welfare sectors [3]. The success of these organizations fundamentally depends on public trust and operational transparency—factors that are increasingly mediated through digital platforms. Research indicates that transparent donation mechanisms and visible impact tracking significantly influence donor participation and retention [4], yet these features remain underdeveloped in existing animal welfare applications.

Local organisations play a very crucial role in taking initiatives to ensure the well-being of these animals. However, they cannot stand solely on their own as they need and depend on tremendous support from the public to sustain their efforts [5]. Kelab Pencinta Haiwan UUM (KPH UUM) is a student-led organization working to improve the welfare of these strays, but their efforts are hindered by outdated methods, fragmented outreach, and a lack of digital infrastructure to manage donations, volunteers, and animal care effectively.

Presently, KPH UUM relies heavily on social media for communications, but important posts are often overlooked, donation methods lack transparency, and volunteer coordination is inefficient. These limitations reduce the organization's visibility and impact. To address this, the project proposes the development of Pethero KPH UUM, a centralized, transparent, and user-friendly mobile application

tailored to the needs of KPH UUM. The application will support progress-trackable donations, structured volunteer sign-ups, adoption processing, animal updates, and stray reporting. It aims to increase community engagement, build donor trust, and empower both members and non-members of the UUM community to contribute meaningfully.

The primary objectives of this study are threefold: (1) to analyze the existing operational framework of KPH UUM, (2) to design and develop a mobile application with specialized welfare modules, and (3) to evaluate the system through user testing. By consolidating all animal welfare activities into a single digital platform, Pethero not only improves operational efficiency but also fosters a more compassionate and proactive campus culture.

This research contributes to the broader discourse on digital solutions for non-profit organizations, demonstrating how technology can amplify the impact of student-led initiatives in animal welfare. It demonstrates how technology can amplify the impact of student-led initiatives in animal welfare, offering a replicable model for similar organizations facing operational and transparency challenges.

II. LITERATURE REVIEW

This section presents a detailed background study of KPH UUM and evaluates existing solutions. The organization, formalized in 2023, has grown significantly but still struggles with managing activities manually. The literature review identifies several key issues through interviews and a survey:

- Low awareness of KPH UUM campaigns.
- Donor hesitancy stemming from a critical lack of transparency in fund allocation.
- Ineffective volunteer coordination,leading to resource mobilization inefficiencies.
- Overwhelming interest in a dedicated mobile application, indicating a clear readiness for digital intervention.

Four existing animal welfare platforms such as Stray Rescue Management System (SRMS), the PAW mobile application (UTM), SayangPet.my, and iPET were evaluated to identify strengths and gaps in functionality.

A. Stray Rescue Management System (SRMS)

The Stray Rescue Management System (SRMS) comprehensive web application designed to manage and report stray animals across Malaysia, facilitating their rescue and fostering while promoting responsible pet ownership [6]. A key strength of SRMS is its extensive functional scope, which integrates critical features such as donation processing, volunteer coordination, and adoption services. Furthermore, the system's operational scale across multiple states significantly broadens access to welfare services, enhancing its potential impact on a national level.

Despite its broad functionality, SRMS is limited by a lack of community engagement and transactional transparency. The platform operates primarily as an intermediary for individual actions rather than fostering a collaborative community environment. It does not provide users with post donation updates or progress reports on rescued animals, which can erode trust and diminish the perceived impact of their contributions. This absence of ongoing engagement and visible outcomes represents a significant weakness for building sustained supporter relationships.

B. The PAW Mobile Application (UTM)

The Mobile Application for Fundraising and Management of Stray Cats (PAW) addresses the critical challenge of funding shortages in managing stray cat populations, with its operations focused specifically within the Universiti Teknologi Malaysia, Skudai campus. A principal strength of the application is its dedicated and well-executed donation module, which provides comprehensive profiles for each rescued cat to inform donor decisions. This core functionality is supported by a simple, seamless user interface that promotes an intuitive and engaging user experience, available on both Android and iOS platforms.

However, the application's primary limitation is its functional narrowness. By concentrating exclusively on the donation process, PAW fails to maintain user engagement beyond the point of transaction. The absence of post-donation updates on the cats' conditions or any supplementary features such as volunteer coordination or adoption services results in low user retention. Consequently, the platform struggles to foster a sustained relationship with its user base, as interactions are typically brief and transactional in nature.

C. SayangPet.my

SayangPet.my is a multi-functional web application designed to support animal welfare activities for organizations like PAWS, NGOs, and the general public across Malaysia. A key strength of the platform is its versatility; it offers users a choice of platforms and integrates various features such as pet adoption, lost-and-found services, and donations. This multifaceted approach provides a comprehensive solution for a range of animal-related issues. Notably, its lost-and-found feature is highly accessible, serving not only stray animals but also helping pet owners locate their missing pets.

However, the application has significant weaknesses, primarily concerning transparency and user safety. The donation process lacks clarity, as it fails to provide users with updates on how their contributions are used, showing only pictures without substantive progress reports. Furthermore, the lost-and-found feature operates on a direct user-to-user contact model without any mediation. This poses a potential safety risk, as the platform does not offer safeguards or warnings against malicious actors, leaving users to manage interactions on their own.

D. iPET

The iPET web application demonstrates significant strengths rooted in its rigorous development and positive reception [7]. Constructed using the structured Waterfall methodology, the application achieved high scores across all ISO 25010 software quality standards, as evaluated by

Informatioon Technology experts. This confirms its robustness in terms of functional suitability, reliability, security, and usability. Furthermore, end-user assessments from the target community yielded highly acceptable ratings, particularly for its functionality and ease of use. This indicates that iPet successfully fulfills its core objective of providing a reliable and user-friendly platform to facilitate pet adoption and support, thereby directly addressing the issue of pet abandonment.

Despite its solid technical foundation, iPET exhibits key limitations that constrain its potential impact. A primary weakness is its exclusive existence as a web application, which limits accessibility and convenience for users who primarily rely on mobile devices. Additionally, the platform lacks integrated communication features, such as a chat system, preventing direct interaction between pet owners and potential adopters and hindering the community-building aspect of the adoption process. The application's scope is also currently narrow, having been tested only within a specific local community, which leaves its scalability and effectiveness in broader contexts unverified. These areas represent critical opportunities for future enhancement to increase the application's reach and utility.

Across all four systems, critical limitations were observed, particularly in real-time animal updates, volunteer coordination, and donor engagement, highlighting the need for an integrated solution like Pethero KPH UUM. The proposed mobile application addresses these gaps by combining donation tracking with progress visibility, structured volunteer management, and community-focused features such as news updates and fostering support, as summarized in Table I.

TABLE I. COMPARISON OF FUNCTIONALITIES FOR ANIMAL WELFARE APPLICATIONS

Animal Welfare Application/ Functionalities	SRMS	PAW	SayangPet.my	iPet	Pethero KPH UUM
Application Type	Web Application	Mobile Application	Web Application & Mobile Application	Web application	Mobile Appliication
Donation	Yes	Yes	Yes	No	Yes
Updates on Rescued Animals	No	No	No	No	Yes
Volunteer	Yes	No	No	No	Yes
Report	Yes	No	No	No	Yes
News posting	No	No	No	No	Yes
Fostering	Yes	No	No	Yes	Yes
Adoption	Yes	No	Yes	Yes	Yes
Lost and Found	No	No	Yes	Yes	No

The Pethero app was conceptualized to overcome the shortcomings identified in existing systems by integrating a range of tailored features that directly address the needs of Kelab Pencinta Haiwan UUM. One of its core innovations is the implementation of a transparent donation system, which includes visual progress bars and itemized cost breakdowns for each rescued animal. This not only fosters donor trust but also improves financial accountability. Additionally, the app provides verified updates on the condition and treatment progress of animals after donations have been made, ensuring continued transparency and engagement. To enhance volunteer coordination, the system includes a structured scheduling feature with clear task descriptions, allowing volunteers to understand their responsibilities and commit more confidently. Furthermore, the application offers a dedicated space for post sharing and community engagement, helping to amplify awareness, promote events, and encourage broader participation in animal welfare initiatives.

The project utilized a carefully selected set of technologies to ensure flexibility, seamless integration, and developer efficiency. React Native was chosen for cross-platform mobile development, while Expo accelerated the development process and simplified testing. Firebase handled essential backend functions, including authentication, real-time database operations, and storage. For secure and traceable online payments, Stripe was integrated into the system. Additionally, SendGrid facilitated the delivery of OTPs and notifications via email. The entire development process was carried out in Visual Studio Code, a robust and efficient IDE. Together, these technologies provided a reliable and scalable foundation for the application.

III. METHODOLOGY

The project adopts the Rapid Solo Software Development (RSSD) methodology, an Agile-based approach that has been tailored to suit the workflow of solo developers. This methodology emphasizes three core values: efficiency, modularity, and revisit ability, and is structured through a sequence of well-defined phases that guide the entire development life cycle. Fig. 1 depicts the phases involved in Rapid Solo Software Development (RSSD).

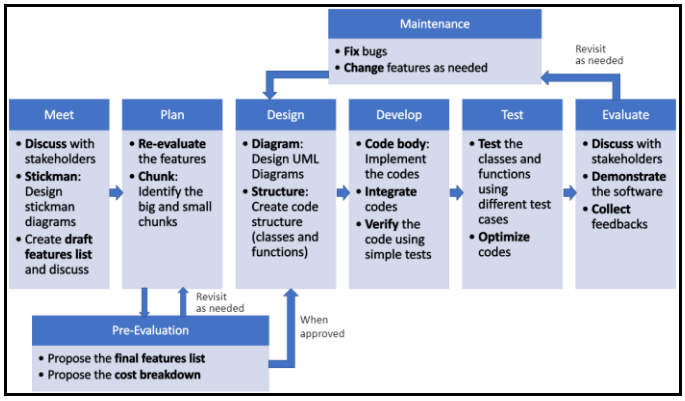

Fig. 1. Phases involved in Rapid Solo Software Development (RSSD)

The first phase, known as the Meet Phase, involved stakeholder engagement through meetings and interviews to gather detailed system requirements. This included discussions with the President of Kelab Pencinta Haiwan UUM and a supporting survey conducted among UUM students and staff. These sessions confirmed the specific needs of the organization and the community, validating the project's direction and focus.

A. Plan and Pre-evaluation Phase

Following this, the Plan and Pre-evaluation Phase was carried out to assess the complexity and feasibility of the requested features. Features were broken down into manageable components or "chunks," which helped in organizing development tasks efficiently. The feedback gathered earlier was also revisited and incorporated to refine the scope and roadmap of the application.

B. Design Phase

In the Design Phase, various UML diagrams—including use case, sequence, and activity diagrams—were created to illustrate system functionality and workflow. The user interface (UI) mockups were developed using Figma to visualize the application's layout and user experience. Additionally, the system architecture and database structure were defined to ensure seamless integration between frontend and backend components.

C. Development Phase

This phase involved actual coding and system construction. The frontend of the application was built using React Native along with Expo to accelerate development and facilitate device testing. The backend relied on Firebase for database management, user authentication, and storage. Emphasis was placed on writing modular code with reusable components to simplify future updates and ensure scalability.

D. Testing Phase

During the Test Phase, multiple testing strategies were implemented. This included unit testing and black box testing to validate functionality and ensure the system met specified requirements. A User Acceptance Test (UAT) was also performed with representatives from KPH UUM to verify the system's usability and effectiveness. Code optimization followed, where performance bottlenecks were identified and improved.

E. Evaluation and Maintenance Phase

Finally, the Evaluate and Maintenance Phase marked the presentation of the completed application to stakeholders. Feedback was collected to assess satisfaction and identify any areas needing refinement. Although most of the feedback was positive, minor bugs and usability issues were logged for future updates and maintenance tasks, ensuring the application remains functional and adaptable after deployment.

IV. REQUIREMENT ANALYSIS AND DESIGN

This section presents the comprehensive requirement analysis and design of the Pethero mobile application, focusing on how the system functions are structured, visualized, and translated into a workable design blueprint. Using Unified Modeling Language (UML) models, the system's functionalities are broken down into understandable diagrams to represent user interactions, workflows, system logic, and database relationships. These tools help ensure that the system is designed logically and can be developed and maintained effectively.

A. Requirement Analysis

The requirement analysis begins with the construction of use case diagrams, which serve to illustrate the interaction between different types of users (such as general users and administrators) and the system itself. These diagrams encompass all critical modules of the application, including profile management, user authentication, donation processing, volunteer coordination, animal updates, adoption procedures, post creation, and reporting of stray animals. Each use case helps identify what functions each actor can perform and clarifies how the system will respond to their actions. One of the use case diagram for donation module and volunteer module based on Fig. 2.

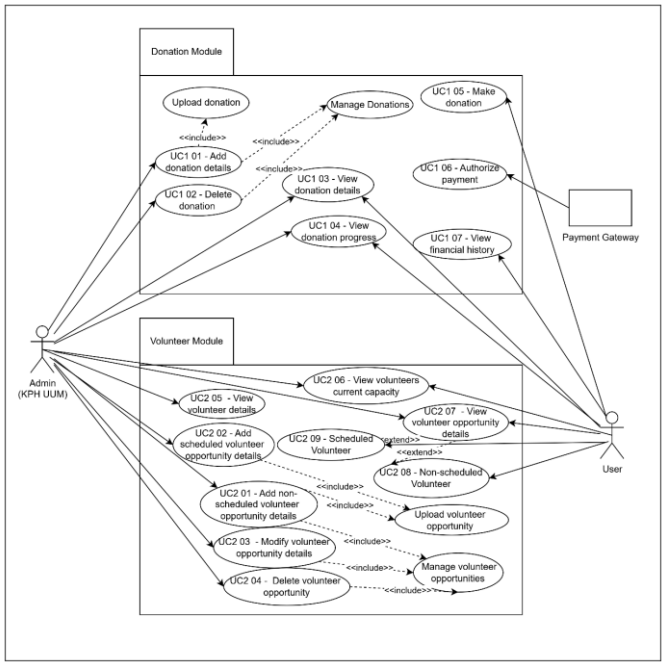

Fig. 2. Use case diagram for donation module and volunteer module

Following the use cases, sequence diagrams were created to depict the flow of operations for specific tasks such as making a donation, signing up, volunteering, and adopting an animal. These diagrams show how the system's components interact over time in response to user actions, helping developers understand the real-time behavior of the application and how data and requests move through the system.

In addition, activity diagrams were used to model the workflow logic within the application. These diagrams detail the sequences of operations and decision points for both users and administrators. For example, the activity diagram for volunteering includes steps such as viewing available opportunities, selecting a task, and confirming participation. These diagrams help ensure that the system's business

processes are clearly defined and aligned with user expectations.

A class diagram was then created to represent the object-oriented structure of the application. This diagram outlines the core entities in the system such as Users, Animals, Donations, Posts, and Volunteer Opportunities, and defines the attributes and methods associated with each. It also illustrates how these classes relate to one another through associations and inheritance, serving as the foundation for backend development and database design.

B. System Design

The overall system architecture of Pethero based on Fig. 3 demonstrates the interaction between the mobile client application, the Firebase backend services, and third-party integrations such as Stripe and SendGrid. The client app handles the user interface and frontend logic, while Firebase supports database operations, user authentication, and cloud storage. Stripe is used to securely process donations, and SendGrid is integrated for delivering one-time passwords (OTPs) via email. This architecture ensures that the system is modular, scalable, and efficient, with each component handling a specific responsibility in the overall workflow.

The database design is presented through an Entity-Relationship Diagram (ERD), which maps out the key entities—Users, Animals, Donations, Volunteer Tasks, and Adoption Requests—and their interrelationships. Each entity includes attributes relevant to its function, such as donation amount, animal condition, user role, and volunteer schedule. The design ensures that the data structure is normalized, reducing redundancy and supporting accurate, reliable data handling throughout the application.

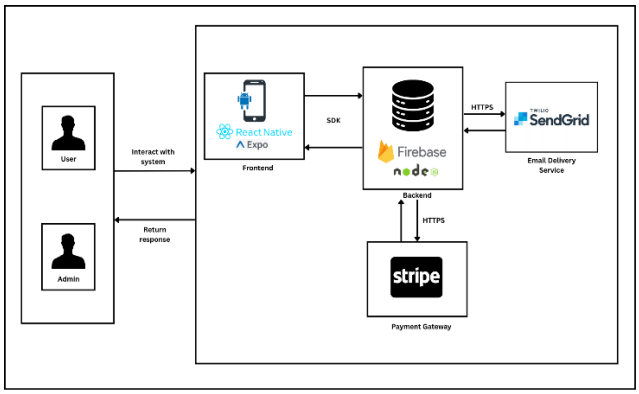

Fig. 3. Overall system architecture of Pethero KPH UUM Mobile Application System

C. Interface Design

Interface design were developed for both user and admin perspectives to ensure usability. These mockups provide a visual layout of the application's screens and how users will navigate through them. Based on Fig. 4 is for users, screens include donation pages, animal update listings, volunteer opportunities, and adoption forms. For administrators, dedicated interfaces allow them to manage and update system content, such as uploading new donation campaigns or modifying animal profiles showed in Fig. 5.

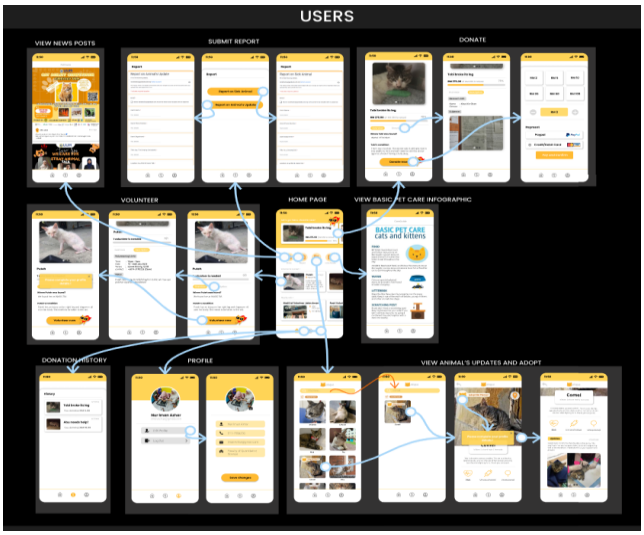

Fig. 4. Overall user's interface design

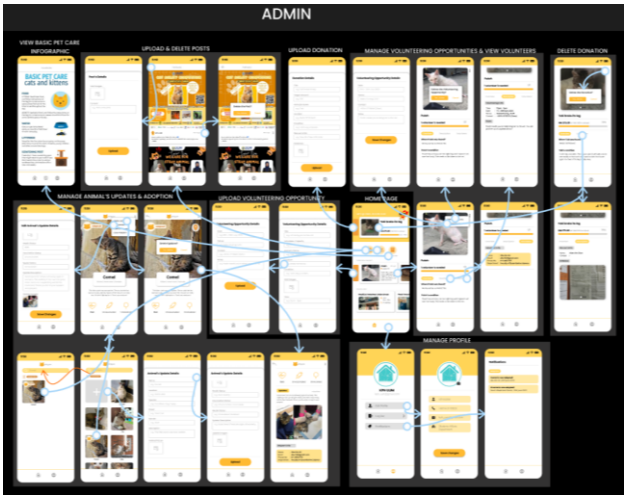

Fig. 5. Overall admin's interface design

The designs prioritize clarity and simplicity, with clean layouts, intuitive navigation, and responsive elements that cater to both tech-savvy users and those less familiar with mobile applications. Icons, buttons, and form elements are appropriately placed to minimize confusion and streamline task completion. The goal of the interface design is to deliver a seamless user experience while ensuring that all essential functions are accessible with minimal effort.

In summary, the requirement analysis and system design in this chapter lay the groundwork for the successful implementation of the Pethero application. By carefully analyzing user needs and translating them into a structured design, the system ensures clarity, scalability, and maintainability, all of which are critical for both current deployment and future enhancements.

V. RESULTS, TESTING, AND DISCUSSIOM

The Pethero KPH UUM mobile application was successfully developed using React Native (with Expo) and Firebase backend services. Key functionalities such as donations, animal updates, adoption applications, and volunteering management were fully implemented. Firebase Authentication, Cloud Firestore, Cloud Functions, Firebase Storage, and third-party services like Stripe and SendGrid were integrated to ensure performance, security, and real-time data handling.

A. Core Functional Results

One of the most critical features implemented was the donation module, which incorporates Stripe for real-time and secure payment processing. Users can make donations through a guided interface and view their transaction history, while the backend, built using Firebase Cloud Functions and secured with HTTPS, ensures safe handling of payment data and input validation. As shown in Fig. 6, the user interface for donation is intuitive and cleanly structured to guide users through the process efficiently.

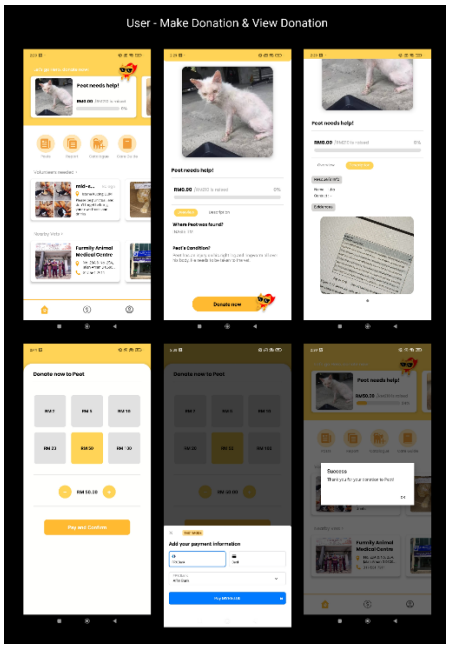

Fig. 6. Interfaces for Make Donation

To improve donor confidence and promote transparency, animal updates are regularly posted by administrators and made publicly visible. These updates include medical progress, and new rescue cases. Each update supports image uploads and real-time display using Firebase Storage and Firestore. The functionality and user view of this feature are depicted in Fig.7, demonstrating how media and textual updates are seamlessly combined in the interface.

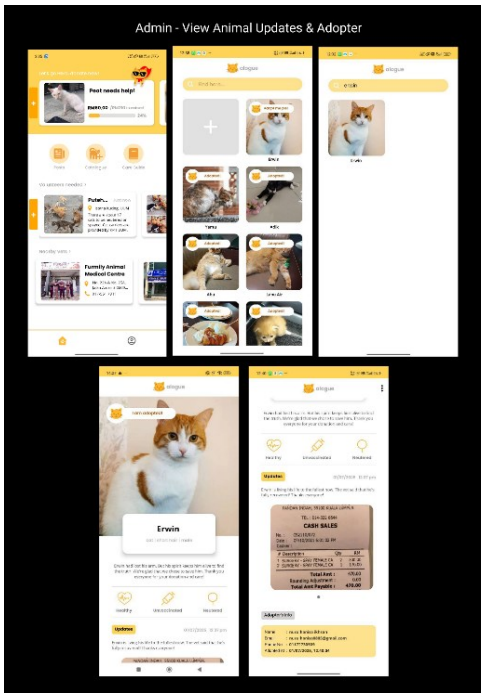

Fig. 7. Interfaces for Animal Updates & Adoption

The system also features a structured volunteering module, supporting both scheduled and flexible participation. Volunteers may register for specific dates with limited slots, which are then visible to admins in real-time. A calendar-based layout streamlines participation tracking and planning. The user interface and volunteer management tools are presented in Fig. 8, which shows the clarity and organization of this module.

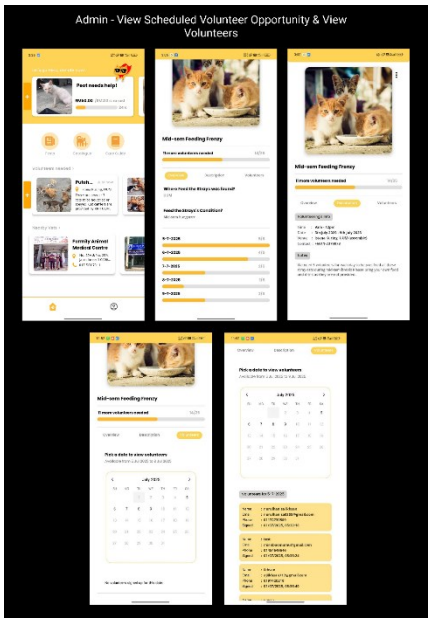

Fig. 8. Interfaces for Volunteer Sign-Up

User access throughout the system is controlled through role-based authentication. Admins have additional privileges to manage adoptions, updates, and volunteers, while regular users access their personalized donation and participation interfaces. OTP-based login was integrated using SendGrid to enhance account security without requiring traditional passwords. Sensitive configurations, such as API secrets, are stored securely in Firebase's environment configuration, adhering to security best practices.

To optimize performance, image upload utilities compress and resize files prior to storage, ensuring faster load times across devices. Additionally, React.memo() was applied to key components to reduce unnecessary re-renders, significantly enhancing interface fluidity on lower-end Android devices.

B. Testing Results

1) Black box testing

Use-case testing validated each feature from a user perspective. Donation input handling passed boundary value analysis. All core functions, donations updates, adoption, and volunteering to produced expected outputs are shown in Table II.

TABLE II. DONATION TEST CASES

Test Case ID	Test Case Description	Role	Pre-condition	Test Input	Expected &Actual Result	Status (Pass/ Failed)
TC1-01-01	Upload donation post (valid)	Admin	Admin creates a new donation post with all valid details and images.	Valid text details, image files.	Post is created successfully and appears in the donation section.	Pass
TC1-01-02	Upload Donation Post (Invalid)	Admin	Admin attempts to create a post with missing required text fields.	Blank text fields, one image file.	System correctly displays an error message and prevents upload.	Pass
TC1-02-01	Delete Donation Post (Confirm)	Admin	Admin selects to delete a post and confirms the action.	User confirms "Delete".	Donation post is successfully removed; success message appears.	Pass
TC1-02-02	Delete Donation Post (Cancel)	Admin	Admin selects to delete a post but cancels the action.	User selects "Cancel".	Confirmation dialog closes; the donation post remains unchanged.	Pass
TC1-03	View Donation Details	Admin, Users	Any logged-in user views the full details of a donation post.	User clicks on a donation post.	System displays all post details (description, target amount, progress, files).	Pass
TC1-05-01	Make a Donation (Valid)	Users	A user initiates a donation with a valid amount.	A numerical amount less than the target.	System successfully proceeds to the payment input options.	Pass

2) White box testing

Internal logic was tested using Jest. Each function's code paths were verified via branch testing, ensuring coverage and correctness. Based on Table III shows the white box testing for test case which is make donation by role user.

TABLE III. WHITE BOX TESTING FOR TEST CASE (MAKE DONATION BY ROLE USER)

Test Case ID	Brach/ condition	Test Purpose	Test Input	Expected Result	Status (Pass /Failed)
WB-TC1-05-01	No user logged in	Check rejection when no user is logged in	User = null	Triggers alert("Error", "User not logged in"	Pass
WB-TC1-05-02	User doc not found	Handle missing user profile	getUserDoc()returns userSnap exists() = false	Triggers alert("Error", "User profile not found"	Pass
WB-TC1-05-03	Successful donation flow	Confirm donation proceeds when all valid	Valid form, addDonorDoc and updateDonation Doc succeed	Both Firestore functions called once, no alert shown	Pass
WB-TC1-05-04	DB error on donor add	Stimulate failure in donor add step	addDonorDoc mocked to throw error	Triggers alert("Database Error", "Failed to add donor")	Pass
WB-TC1-05-05	DB error on donation update	Stimulate failure in donation doc update	updateDonation Doc mocked to throw error	Triggers alert("Database Error", "Update failed")	Pass

3) User acceptance testing (UAT)

The user acceptance test was conducted right after Pethero KPH UUM has completed its black box testing to ensure the user requirements are fulfilled and to gain feedback for further improvements. The user acceptance test tested by stakeholder, role admin for posts function in Table IV.

TABLE IV. USER ACCEPTANCE TEST FOR POSTS FUNCTION IN THE APPLICATION

Test Case ID: UAT -TC6-02-01				
Test Case Description: Add news post details – valid details				
Pre-condition: 1. Admin has logged into the system 2. Admin already in "Post" section				
Test Steps	**Test Input**	**Expected Result**	**Actual Result**	**Status (Pass/ Fail)**
Admin clicks add button in post section	Add button	Posts form appears	Posts form appears	Pass
Admin fills out the post details	Content: Hello Everyone!	Text fields are filled in	Text fields are filled in	Pass
Admin adds an image	Image file	Image is attached	Image is attached	Pass
Admin clicks upload	Upload button	Successful pop-up window appears. New post can be seen immediately in the news posts section	Successful pop-up window appears. New post can be seen immediately in the news posts section	Pass

VI. CONCLUSION

This research concludes the development process and outcomes of the Pethero application. The project has successfully fulfilled all of its initial objectives, which included analyzing the existing challenges within Kelab Pencinta Haiwan UUM's workflow, designing and developing a centralized mobile application tailored to their needs, and rigorously testing the system to ensure its functionality and usability. Through this process, the application has

demonstrated its effectiveness in addressing critical problems such as low visibility due to fragmented communication channels, manual and inefficient methods of donation tracking, and the lack of structure in volunteer coordination and post-donation accountability.

The modular architecture of Pethero establishes a robust foundation for scalability beyond its initial deployment at UUM. However, scaling the platform to a multi-institutional level presents significant long term challenges. These include the need for organizational diversification, sustainable maintenance models, and coherent multi-tenant data governance. Successfully addressing these challenges requires ensuring architectural flexibility for diverse workflows, instituting a sustainable operational and financial support framework, and formulating clear data portability and privacy policies. This progression is imperative for transitioning Pethero from a localized solution into a widely adopted platform. Such a transformation would substantially amplify its positive impact on animal welfare management across a broader ecosystem of communities and institutions.

In summary, Pethero KPH UUM successfully delivers a practical and impactful solution to the operational difficulties faced by the organization. Beyond solving immediate challenges, it sets a strong foundation for wider implementation across other institutions dealing with similar issues in animal welfare management.

ACKNOWLEDGMENT

I would like to express my sincere gratitude to my supervisor, Ts.Marina Md Arshad, for her invaluable guidance, support, and expertise throughout this project. I also extend my appreciation to KPH UUM for their contributions and collaboration, which were essential to the success of this work.

CONFLICT OF INTEREST

The authors declare that there is no conflict of interest regarding the publication of this paper.

REFERENCES

[1] S. M. I. Munir, M. I. Mokhtar, and A. F. Arham. (2023). Public perspectives on strays and companion animal management in Malaysia. *BMC Public Health*, 23(1). https://doi.org/10.1186/s12889-023-16276-5.

[2] Uptech. (2024). What is the Animal Welfare Act 2015 Malaysia All About? Persatuan Haiwan Terbiar Malaysia. https://www.animal.org.my/animal-welfare-act-2015/.

[3] Vojtkovská, V., Voslářová, E., & Večerek, V. (2020). Methods of assessment of the welfare of shelter cats: A review. *Animals*,*10*(9), 1527.

[4] Aruna, S., Sivaprakash, S., Bharathidasan, K., & Harishkumar, N. (2024). A new mobile application design to empower local communities through seamless resources sharing. In *2024 International Conference on Cognitive Robotics and Intelligent Systems* (ICC-ROBINS) (pp. 916-920). IEEE.

[5] S. Lee. (2022). Animal shelters in dire need of funds from the public. The Star. https://www.thestar.com.my/news/nation/2022/02/28/animal-shelters-in-dire-need-of-funds-from-the-public.

[6] Mammi, H. K. (2025). Stray rescue management system (SRMS). *International Journal of Innovative Computing*, *15*(1), 139–144.

[7] Gatmaitan, J. E., Azurin, D. V. P., Saclolo, J. P. T., Yango, J. A. G., Lorenzo, K. P., Olipas, C. N. P., & Alegado, R. T. (2024). iPET: The design, development, and assessment of a web-based application for pet adoption. *Formosa Journal of Computer and Information Science, 3*(1), 15–26. https://doi.org/10.55927/fjcis.v3i1.6031.