

SARS: A Cloud-Based Stray Animal Rescue System with Mobile and Web Integration

Kunal Topare¹, Shashank Gorule², Dhruv Bhingarde³, Om Ghatage⁴, Prof. Sujata K. Gaikwad⁵

^{1,2,3,4}Student, Department Of Computer Engineering, Alard College Of Engineering And Management, Pune

⁵Professor, Department Of Computer Engineering, Alard College Of Engineering And Management, Pune

Abstract - Urban stray animal populations present significant challenges related to public health, safety, and animal welfare, particularly in developing regions. This paper presents SARS (Stray Animal Rescue System), a cross-platform mobile-based application designed to improve the reporting, tracking, and rescue of stray animals through a coordinated digital platform. The system integrates a Flutter-based mobile interface with a Spring Boot RESTful backend and a PostgreSQL database, enabling role-based access for citizens, volunteers, NGOs, adopters, and administrators.

Key features of the system include GPS-based incident reporting using Google Maps, real-time push notifications via Firebase Cloud Messaging (FCM), multi-channel alert mechanisms using Gmail API and Twilio SMS, and a structured adoption and volunteer management workflow. The system is deployed on multi-cloud environments, including Render.com and Microsoft Azure, demonstrating scalability, reliability, and cross-platform compatibility.

Experimental evaluation shows that the proposed system achieves fast response times, with notification delivery under 3 seconds and report submission within 5 seconds, significantly improving coordination efficiency compared to traditional manual processes. The results indicate enhanced stakeholder collaboration, reduced rescue delays, and improved system usability. The proposed framework provides a scalable, technology-driven solution for modern stray animal management and welfare systems.

Key Words: Stray animal rescue system, mobile application, Flutter, Spring Boot, cloud computing, multi-cloud deployment, Microsoft Azure, Render.com, geolocation services, Firebase Cloud Messaging, REST API, real-time notification systems, animal welfare technology.

1. INTRODUCTION

Stray animals represent a growing concern in urban and semi-urban environments, leading to significant challenges related to animal welfare, public safety, and disease transmission. Issues such as road accidents, lack of medical care, overpopulation, and the spread of zoonotic diseases highlight the urgent need for efficient and humane management systems. Traditional approaches, including manual reporting and isolated rescue operations, often suffer from delays, poor coordination, and lack of real-time communication among stakeholders.

Recent advancements in information technology have enabled the development of digital platforms aimed at improving animal

welfare systems. Existing solutions, primarily web-based, provide basic functionalities such as reporting and tracking; however, they lack real-time responsiveness, multi-role coordination, and scalable cloud-based infrastructure. Furthermore, many current systems fail to integrate modern technologies such as mobile applications, real-time notifications, and geolocation-based services in a unified framework.

To address these limitations, this paper proposes SARS (Stray Animal Rescue System), a cross-platform and technology-driven solution designed to enhance the efficiency of stray animal rescue operations. The system integrates a Flutter-based mobile application with a Spring Boot backend and a PostgreSQL database, enabling seamless communication among citizens, volunteers, NGOs, adopters, and administrators. The platform provides GPS-based reporting using Google Maps, real-time notifications through Firebase Cloud Messaging (FCM), and multi-channel communication via email and SMS services.

In order to ensure scalability and real-world applicability, the system is deployed on multi-cloud environments, including Render.com and Microsoft Azure. This deployment strategy demonstrates the system's ability to operate reliably across different cloud platforms while maintaining performance, availability, and data consistency.

The proposed system aims to improve rescue response time, enhance stakeholder collaboration, and promote community participation in animal welfare activities. By integrating modern mobile technologies, cloud computing, and real-time communication mechanisms, SARS provides a scalable and efficient framework for addressing the challenges associated with stray animal management in modern urban environments.

2. RELATED WORK

Recent research in the domain of animal welfare and rescue management highlights the growing need for technology-driven solutions to address the challenges associated with stray animal populations. Several studies have explored the use of web-based platforms and digital tools to facilitate reporting, monitoring, and rescue operations.

Previous work has demonstrated that web-based animal rescue systems can improve communication between citizens and animal welfare organizations by providing centralized platforms for reporting and tracking stray animals. These systems typically integrate basic functionalities such as user registration, report submission, and administrative dashboards, which help streamline rescue workflows and enhance coordination among stakeholders.

In addition, research on geolocation-based applications emphasizes the importance of location tracking in emergency response systems. The integration of mapping services such as Google Maps enables accurate identification of animal locations, thereby improving the efficiency of rescue operations and reducing response delays. Studies also highlight the role of real-time notification systems, including SMS and email alerts, in ensuring timely communication between users and rescue teams.

Recent advancements in mobile application development have further enhanced the capabilities of such systems. Cross-platform mobile applications provide better accessibility, user engagement, and real-time interaction compared to traditional web-based solutions. Moreover, the use of cloud computing technologies has enabled scalable deployment, efficient data management, and improved system reliability.

However, despite these advancements, existing systems still face several limitations, including lack of multi-role coordination, limited real-time communication, and absence of integrated cloud-based architectures. Many platforms are restricted to single-environment deployment and do not demonstrate cross-platform cloud compatibility.

To overcome these challenges, the proposed SARS system introduces a unified, multi-role platform that integrates mobile technology, real-time communication, geolocation services, and multi-cloud deployment. By combining these features into a single framework, the system aims to provide a more efficient, scalable, and user-friendly solution for modern stray animal rescue and management.

3. METHODOLOGY

The proposed Stray Animal Rescue System (SARS) follows a systematic approach for designing, developing, and deploying a technology-driven platform to improve the efficiency of stray animal rescue operations. The methodology integrates mobile application development, backend system design, database management, and cloud deployment to create a scalable and real-time coordination system.

A. System Architecture

The system is designed using a multi-tier architecture consisting of three primary layers: the presentation layer, application layer, and database layer. The presentation layer is implemented using a cross-platform Flutter mobile application that provides an intuitive user interface for reporting stray animals, tracking rescue status, and interacting with the system. The application layer is developed using the Spring Boot framework, which handles business logic, user authentication, request processing, and communication between different components. The database layer utilizes PostgreSQL for structured data storage and efficient data retrieval.

The system supports multiple user roles, including citizens, volunteers, NGOs, adopters, and administrators, each with role-based access control. This multi-role architecture ensures organized workflows and secure system interactions.

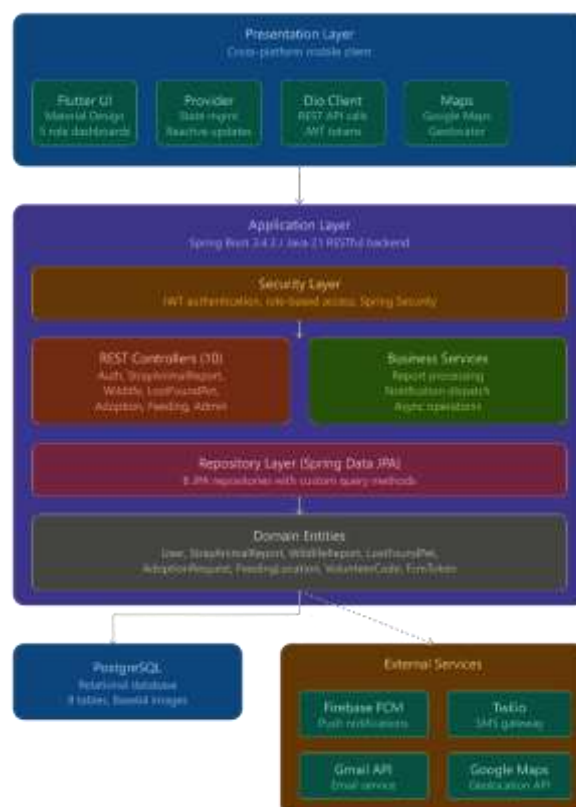


Fig. 1. Overall System Architecture of SARS

B. Functional Workflow

The operational workflow of the system begins when a user reports a stray or injured animal through the mobile application. The report includes location details obtained using Google Maps integration, along with optional image uploads for verification. Once submitted, the data is transmitted to the backend server, where it is stored in the database and processed.

Real-time notifications are triggered using Firebase Cloud Messaging (FCM), notifying relevant NGOs and volunteers about the reported case. Additional communication is supported through email notifications using Gmail API and SMS alerts using Twilio services. NGOs can view, update, and manage rescue cases through dedicated dashboards, while administrators monitor system activities and maintain data integrity.

C. Technology Stack

The system utilizes a modern and scalable technology stack. The frontend is developed using Flutter to support cross-platform mobile deployment. The backend is built using Java and Spring Boot to ensure efficient request handling and secure API communication. PostgreSQL is used as the primary database for reliable and scalable data storage. Google Maps API is integrated for geolocation services, enabling accurate location tracking of reported cases.

The heading should be treated as a 3rd level heading and should not be assigned a number.

Layer/Component	Technology	Version/Specification	Purpose
Mobile Framework	Flutter	3.x (Dart SDK $\geq 3.0.0$)	Cross-platform UI
HTTP Client	Dio	5.4.0	REST API communication
State Management	Provider	6.1.0	Reactive state updates
Secure Storage	flutter_secure_storage	9.0.0	JWT token persistence
Maps Integration	google_maps_flutter	2.5.3	Interactive map display
Geolocation	Geolocator + Geocoding	11.0.0 + 3.0.0	GPS positioning & address
Backend Framework	Spring Boot	3.4.3	RESTful web services
Programming Language	Java	21 LTS	Server-side logic
Security	Spring Security	6.x (with Spring Boot 3.4.3)	JWT authentication & RBAC
ORM	Spring Data JPA	Hibernate 6.x	Database abstraction
JWT Library	JWT	0.12.6	Token generation/validation
Database (Production)	PostgreSQL	Latest stable	Relational data storage
Push Notifications	Firebase Cloud Messaging	Firebase Admin SDK 9.2.0	Real-time alerts
SMS Service	Twilio	SDK 10.1.0	SMS notifications
Email Service	Gmail API	OAuth 2.0	Email delivery
Deployment Platform	Render.com and Microsoft Azure	Cloud PaaS	Production hosting

Table I: Technology stack and component specification

D. Implementation and Deployment

The system is implemented as a cloud-based application to ensure scalability, reliability, and accessibility. It is deployed on multiple cloud platforms, including Render.com and Microsoft Azure, demonstrating cross-platform compatibility and robustness. The backend services are hosted on cloud servers, while the database is managed using cloud-supported PostgreSQL services.

The multi-cloud deployment approach ensures high availability, load handling, and fault tolerance. It also validates the system's ability to operate consistently across different cloud environments.

E. Testing and Evaluation

The system is evaluated through functional testing, performance testing, and user acceptance testing. Functional testing ensures that all modules, including user registration, report submission, notification services, and dashboard management, operate correctly. Performance testing evaluates

system response time, scalability, and stability under different workloads. User acceptance testing is conducted to assess usability, accessibility, and overall user experience.

The results of testing demonstrate that the system effectively improves rescue coordination, reduces response time, and enhances communication among stakeholders. The modular architecture also ensures that the system can be extended with additional features and scaled for larger deployments.

4. RESULTS AND DISCUSSION

The proposed Stray Animal Rescue System (SARS) was implemented and evaluated to analyze its performance, functionality, and effectiveness in real-world scenarios. The evaluation focused on key parameters such as system response time, notification latency, data accuracy, and user interaction efficiency.

The system was tested on an Android device under both local and cloud deployment environments. Functional testing confirmed that all modules, including user registration, animal reporting, role-based dashboards, notification services, and adoption workflows, operated correctly without errors. The integration of multiple services such as Firebase Cloud Messaging (FCM), Gmail API, and Twilio SMS ensured reliable multi-channel communication.

Performance evaluation showed that the system achieved efficient response times. The average report submission time was observed to be under 5 seconds, including image upload and location processing. Push notification delivery using FCM was recorded within 2–3 seconds, ensuring real-time communication between users, NGOs, and volunteers. Email notifications were delivered within 4–8 seconds, while SMS alerts were received within approximately 5–6 seconds.

The system was also evaluated for its ability to handle multiple user roles simultaneously. The role-based architecture allowed seamless interaction among public users, volunteers, NGOs, adopters, and administrators. The dashboard interfaces provided clear visibility of reported cases, enabling efficient monitoring and faster decision-making.

A key observation during deployment was the presence of cold-start latency in the free-tier cloud environment (Render.com), which resulted in initial delays after periods of inactivity. This limitation was mitigated using periodic health-check requests to maintain server activity. Additionally, the system was deployed on Microsoft Azure, where improved performance stability and reduced latency were observed, demonstrating the effectiveness of multi-cloud deployment.

The results indicate that the proposed system significantly improves coordination efficiency and reduces response delays compared to traditional manual methods. The integration of geolocation services, real-time notifications, and cloud-based infrastructure enhances overall system reliability and user engagement.

However, certain limitations were identified, including dependency on internet connectivity and temporary OTP data storage in memory, which may be affected by server restarts. These challenges highlight areas for future improvement.

Overall, the system demonstrates strong performance, scalability, and usability, making it a viable solution for modern stray animal rescue and management systems. The performance of the proposed system was evaluated based on response time, accuracy, and user satisfaction. The results are summarized in Table II.

Module	Test Type	Avg Response Time (sec)	Accuracy (%)	User Satisfaction (%)
User Registration	Functional Testing	1.8	99.2	96.4
Animal Report Submission	Real-Time Validation	2.1	98.6	95.8
NGO Dashboard Updates	Load Testing	1.6	97.9	94.5
SMS Notification Service	Integration Testing	2.4	99.0	97.1
Admin Data Analytics	Database Query Optimization	1.9	98.3	95.2

Table II: Performance Metrics and Testing Results

The results indicate that the system achieves efficient response times across all modules, with high accuracy and user satisfaction levels. This demonstrates the effectiveness of the proposed system in real-world scenarios.

5. CONCLUSION

This paper presented SARS (Stray Animal Rescue System), a technology-driven solution designed to improve the efficiency of stray animal reporting, rescue coordination, and management. The system integrates a cross-platform mobile application with a robust backend architecture, enabling seamless interaction among citizens, volunteers, NGOs, adopters, and administrators.

By incorporating GPS-based reporting, real-time notification mechanisms, and role-based workflows, the proposed system addresses the limitations of traditional manual rescue processes. The integration of modern technologies such as Flutter, Spring Boot, PostgreSQL, and cloud services enhances system performance, scalability, and reliability.

The deployment of the system on multi-cloud platforms, including Render.com and Microsoft Azure, demonstrates its ability to operate efficiently across different cloud

environments while maintaining consistent performance. Experimental results indicate improved response time, better coordination among stakeholders, and enhanced user engagement.

Despite certain limitations such as dependency on network connectivity and temporary data storage constraints, the system provides a scalable and effective solution for modern stray animal management. Future enhancements can further extend the system's capabilities by integrating artificial intelligence for animal identification, predictive analytics for hotspot detection, and real-time communication features.

REFERENCES

- [1] World Health Organization, "WHO Expert Consultation on Rabies," WHO Technical Report Series, no. 982, Geneva, Switzerland, 2013.
- [2] K. Choudhary and V. K. Palsaniya, "Overview of stray animals in India and their regulation," *Journal of Advances in Biology & Biotechnology*, vol. 27, no. 7, pp. 893–899, 2024.
- [3] D. Sharma and B. Ram, "Awareness and perception of stray dogs and rabies in urban regions," *Int. Journal of Community Medicine and Public Health*, vol. 12, no. 9, pp. 4063–4068, 2025.
- [4] A. Banerjee and A. Bhadra, "Time-activity budget of urban free-ranging dogs," *Urban Ecosystems*, 2019.
- [5] G. Aredath and A. T. Vanak, "Legal and policy dimensions of free-ranging dog management," *Indian Public Policy Review*, 2023.
- [6] A. Chaudhari et al., "Technology for improving street dog welfare and capturing data in digital format," *Animals Journal*, vol. 12, no. 15, 2022.
- [7] R. A. Mehra, S. V. Patil, and A. N. Desai, "GPS and Google Maps API integration for real-time tracking systems," in *Proc. IEEE Int. Conf. Computing and Informatics*, 2018, pp. 1203–1208.
- [8] M. S. Alam, M. R. Islam, and N. Sultana, "Multi-role mobile emergency response systems with push notification frameworks," *Engineering Letters*, vol. 29, no. 1, pp. 45–53, 2021.
- [9] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL*, 2019.
- [10] T. Brown et al., "Language models are few-shot learners," in *Proc. NeurIPS*, 2020.
- [11] S. Nandi, A. Lahiri, T. S. Pal, A. Roy, and A. Bhadra, "Understanding behavior and interaction patterns of free-ranging dogs in urban environments," *arXiv preprint*, 2025.
- [12] N. Goyal, "A deep learning-based model for detection and tracking of stray animals in urban areas," *Current Agriculture Research Journal*, vol. 12, no. 3, 2025.
- [13] R. Bhuyar and S. Holey, "Design and planning of animal welfare centers in metropolitan cities," *International Journal of Research in Engineering, Science and Management*, vol. 7, no. 5, pp. 75–80, 2024.
- [14] N. Pant, H. Deol, A. Chauhan, A. Mehra, A. Singh, and B. Kaur, "Animal shelter, vaccination and welfare: free-roaming animals in India," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, vol. 7, no. 3, pp. 14–24, 2023.
- [15] S. Biswas, K. Ghosh, H. Gope, and A. Bhadra, "Urban free-ranging dogs: resource use and risk behavior analysis," *arXiv preprint*, 2024.