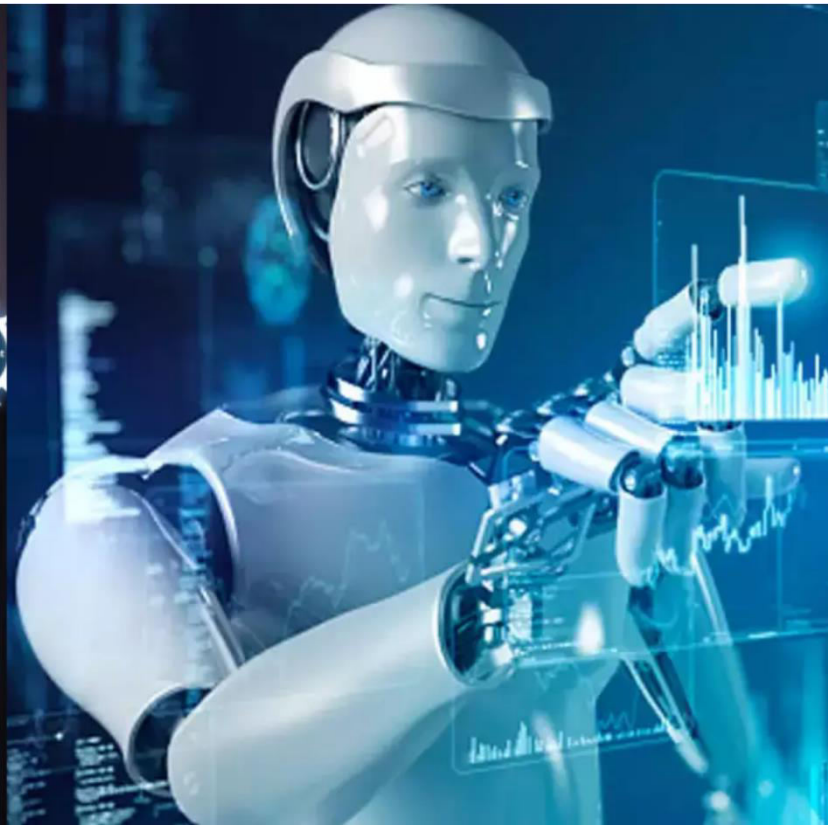
IJIRSET
International Journal of Innovative Research in Science | Engineering | Technology

ISSN(O): 2319-8753
ISSN(P): 2347-6710

International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.699

Volume 14, Issue 5, May 2025

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 5, May 2025

|DOI: 10.15680/IJIRSET.2025.1405333|

Real Time Disaster Information Aggregation Software

Prof. Shwetha G R, Karthik B A, Lavakumar N, Mithun M

Assistant Professor, Department of Information Science and Engineering, S J C Institute of Technology, Chickaballapura, Karnataka, India

U.G. Student, Department of Information Science and Engineering, S J C Institute of Technology, Chickaballapura, Karnataka, India

U.G. Student, Department of Information Science and Engineering, S J C Institute of Technology, Chickaballapura, Karnataka, India

U.G. Student, Department of Information Science and Engineering, S J C Institute of Technology, Chickaballapura, Karnataka, India

ABSTRACT: In the face of increasing natural and man-made disasters, timely access to accurate information is critical for effective response and decision-making. This project presents the development of a Real-Time Disaster Information Aggregation Software that collects, processes, and visualizes disaster-related data from multiple sources including social media, government alerts, news feeds, and sensor networks. The system employs natural language processing (NLP) and geospatial analysis to extract, verify, and localize disaster events in real-time. It features a user-friendly dashboard that provides live updates, severity assessments, and emergency response guidance. Designed to support first responders, government agencies, and the general public, the platform enhances situational awareness and enables quicker, more coordinated responses during crisis events. The software's modular architecture ensures scalability and integration with external systems, making it a robust solution for real-time disaster monitoring and management.

KEYWORDS: Real-Time Data Aggregation ,Disaster Detection ,Emergency Alert System ,Machine Learning ,Natural Language Processing (NLP) ,Geolocation Services ,Disaster Response Coordination

I. INTRODUCTION

Disasters—both natural and human-induced—pose significant threats to life, property, and infrastructure. Earthquakes, floods, wildfires, hurricanes, industrial accidents, and other emergencies require rapid, coordinated responses to minimize their impact. However, a common challenge in disaster management is the lack of timely, accurate, and comprehensive information. Traditional communication channels often fail to deliver real-time updates, and fragmented data sources lead to delays in decision-making. To address this critical gap, this project proposes the development of a Real-Time Disaster Information Aggregation Software.

The system is designed to collect and consolidate data from various sources such as social media platforms, official government alerts, news APIs, and environmental sensors. By leveraging technologies like Natural Language Processing (NLP), geospatial analysis, and machine learning, the software can extract, verify, and present relevant disaster-related information in real-time. The core objective is to provide a centralized, user-friendly platform that enhances situational awareness and supports timely, informed decision-making for emergency responders, authorities, and the public. The software not only improves the speed and accuracy of disaster detection and reporting but also facilitates proactive risk mitigation and coordinated response efforts.

II. PROBLEM STATEMENT

In the face of increasing natural and man-made disasters, the lack of a centralized, real-time, and reliable information system often leads to delayed responses and inefficient disaster management. Existing systems are fragmented, with data scattered across multiple platforms such as news media, government alerts, social media, and sensor networks. This disjointed information flow causes confusion, misinformation, and poor coordination among emergency responders and the public. There is a significant challenge in verifying the credibility and accuracy of rapidly incoming

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 5, May 2025

|DOI: 10.15680/IJIRSET.2025.1405333|

data during crises. Moreover, the absence of predictive analytics and real-time visualizations limits proactive decision-making. Communities in disaster-prone or remote areas often lack timely alerts and updates. Communication gaps between authorities and civilians further worsen the impact of disasters. Therefore, there is a critical need for an intelligent, real-time disaster information aggregation software that consolidates data from multiple sources, filters it for relevance and accuracy, and delivers actionable insights to stakeholders.

III. LITERATURE REVIEW

[1] Abarquez and Murshed (2024) discussed traditional Disaster Management Systems (DMS), which relied heavily on manual reporting and centralized bulletins. These systems often lacked integration with real-time data sources, leading to delayed response and limited situational awareness during emergencies.

[2] Imran, Castillo, Diaz, and Vieweg (2020) explored the role of social media in disaster response. Their research showed that platforms like Twitter can be valuable for detecting events and gathering situational data. However, they highlighted challenges in filtering irrelevant content and verifying the authenticity of posts.

[3] Gao, Barbier, and Goolsby (2021) analyzed the use of crowdsourcing during disasters, citing the Ushahidi platform used in the 2010 Haiti earthquake. While effective for mapping incidents based on community input, the lack of real-time processing and automated validation reduced its reliability for emergency responders.

[4] Zanella, Bui, Castellani, Vangelista, and Zorzi (2021) introduced IoT-based solutions for disaster detection and smart city integration. Their research emphasized the potential of environmental sensors for real-time monitoring, though full integration into larger disaster platforms was still underdeveloped.

[5] Nguyen, Al-Mannai, Joty, Sajjad, Imran, and Mitra (2020) presented deep learning models capable of classifying social media content during crises. Their models achieved high accuracy in identifying disaster-related posts but required significant labeled data and robust model tuning for real-world deployment.

[6].Sharma, P., Wang, J.,Woods, B., Bausch, D., Chen, Z., Tiampo, K., Glasscoe, M., Schumann, G., Pierce, M., & Eguchi, R. (2020). DisasterAWARE – A Global Alerting Platform for Flood Events.

IV. DESIGN AND IMPLEMENTATION

The Real-Time Disaster Information Aggregation Software is designed as a web-based application that provides timely and structured information about disasters from various sources. The system integrates data aggregation, storage, analysis, and presentation modules to offer users real-time insights into ongoing and past disaster events.

System Architecture Overview

The architecture follows a Modular MVC (Model-View-Controller) pattern, ensuring separation of concerns and scalability. The components include:

- Frontend (View): Built using HTML, CSS, and JavaScript, served via Flask templates.
- Backend (Controller): Developed in Python using the Flask framework, which handles routing, user management, and disaster information processing.
- Database (Model): SQLite database is used for lightweight storage of user data and disaster records.

Activity Diagram

The activity diagram for the Real-Time Disaster Information Aggregation Software starts with the user accessing the application and logging in or signing up. Upon authentication, the system assigns a role to the user (admin, responder, or general user). The system then begins aggregating real-time disaster data from various sources like APIs, social media, and user-generated reports. Once the data is collected, it is filtered and verified for accuracy using machine learning models. Verified disaster data is then displayed on an interactive map, showing affected areas, evacuation

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 5, May 2025

|DOI: 10.15680/IJIRSET.2025.1405333|

routes, and shelters. Relevant users receive real-time notifications based on their location and preferences. From the dashboard, the user can view real-time disaster information that is aggregated and presented in a structured format.

Users may also filter or search through disaster events based on parameters like type, location, or date. If the user has admin privileges, they have access to additional functionalities such as adding, editing, or deleting disaster records through an admin panel. At any point, the user can log out, which ends their session and returns them to the home page or login screen. This activity flow ensures both general users and administrators have a clear and functional path to interact with the system effectively. If the user has administrative privileges, additional options are available through an admin panel. From here, admins can manage the disaster database by creating new entries, updating existing records, or removing outdated or incorrect information. This ensures that the data shown to users is accurate and up to date.

Throughout their session, the user can navigate between different views and perform multiple actions. Once their tasks are complete, they can choose to log out, which ends the session and secures the system. This structured activity flow supports both public information access and secure administrative control, providing a comprehensive and interactive experience for different user roles within the system.

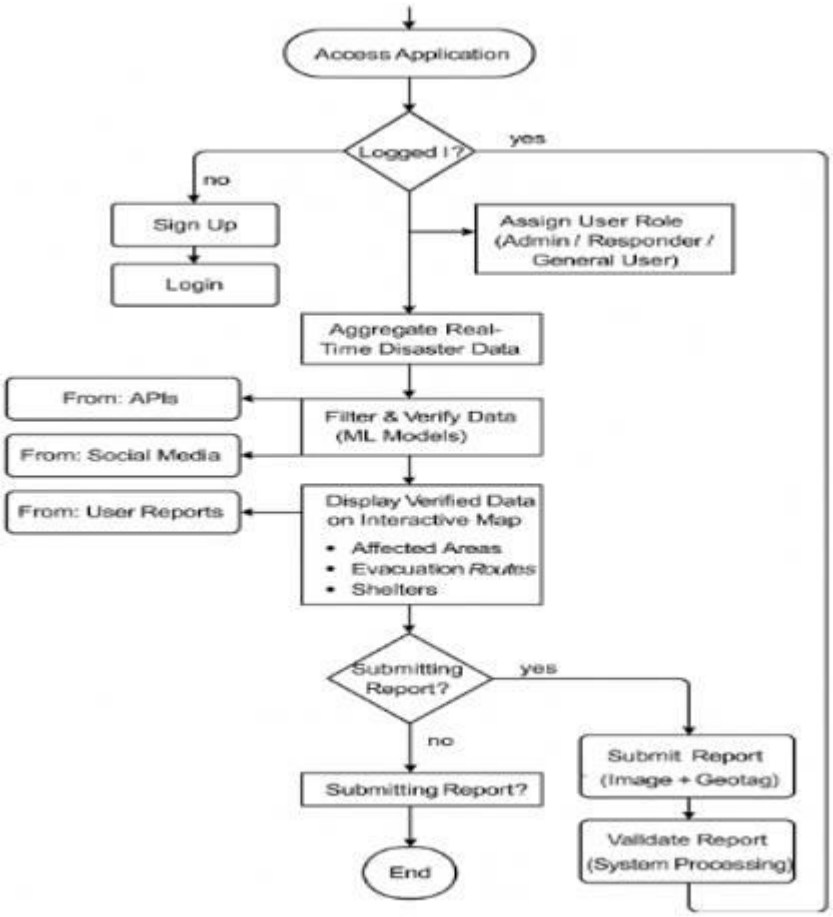


Fig. 1. Activity Diagram

Use Case Diagram

The use case diagram for the Real-Time Disaster Information Aggregation Software illustrates the interaction between different users and the system's core functionalities. There are three primary actors: Admin, Responder, and General User. The General User can sign up, log in, receive real-time alerts, and report disaster incidents by

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 5, May 2025

|DOI: 10.15680/IJIRSET.2025.1405333|

submitting descriptions, images, and location data. The Responder can access verified disaster data, view affected zones on maps, and coordinate emergency actions based on the alerts and reports received. The Admin has elevated privileges and is responsible for managing users, verifying reported data, configuring alert rules, integrating external data sources (such as government APIs and social media), and overseeing the system's performance. The system itself continuously aggregates data from various sources, filters and verifies the data using machine learning, and sends alerts to users based on location and severity.

The use case diagram for the Real-Time Disaster Information Aggregation Software outlines the interactions between the system and its primary actors: General Users and Administrators. A general user interacts with the system primarily to register, log in, and view disaster information. After registration and successful authentication, users gain access to the dashboard where they can explore real-time disaster data. They can also search and filter disaster events based on various criteria such as disaster type, location, and date. Additionally, users may view detailed information about specific disaster incidents, including descriptions, severity, and source data.

On the other hand, administrators have extended privileges. In addition to all the functionalities available to general users, admins can access the Admin Panel, where they can add, edit, or delete disaster records to maintain the accuracy and relevance of the data. This helps ensure that the information provided to users is current and trustworthy. Both user types can log out at any time to terminate their sessions securely.

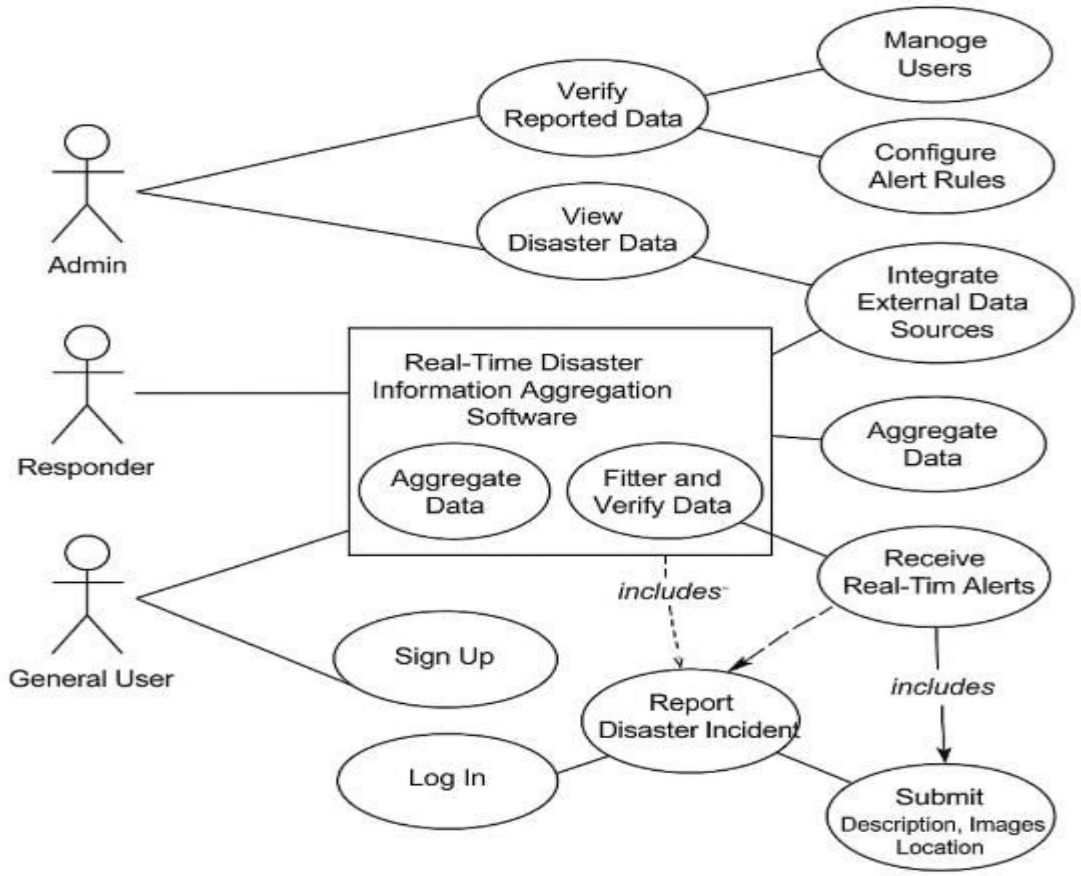


Fig. 2. Use Case

Implementation Details

- Backend is built using Python and Flask framework.
- Frontend uses HTML, CSS, Bootstrap, and some JavaScript.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 5, May 2025

|DOI: 10.15680/IJIRSET.2025.1405333|

- app.py serves as the main application file.
- Templates and UI files are located in the static/ directory.
- User authentication is implemented with registration, login, and session handling.
- Passwords are hashed securely using werkzeug.security.
- User and disaster data are stored in a SQLite database (users.db).
- Users can view real-time disaster data on the dashboard.
- Admins can add, edit, or delete disaster events via the admin panel.
- Each disaster record includes event name, type, location, time, description, and source.
- Search and filter options are available for disaster listings.
- Basic CRUD operations are performed using SQL or SQLAlchemy.
- Input validation and sanitization protect against injection attacks.
- Flask's debug mode helps during development and troubleshooting.
- Logout feature clears sessions and ends the user session securely.
- Admin access is role-restricted and protected.

V. CONCLUSION

In conclusion, the Real-Time Disaster Information Aggregation Software offers a comprehensive and intelligent solution for enhancing disaster awareness, preparedness, and response. By integrating real-time data from multiple reliable sources and applying advanced machine learning and natural language processing techniques, the system is capable of detecting, classifying, and visualizing disaster events as they occur. The use of geospatial analysis and user-friendly dashboards ensures that critical information is delivered promptly and accurately to decision-makers and the public. Through robust backend processing, seamless API integration, and a responsive frontend, the software delivers high performance and scalability. Furthermore, with continuous monitoring, regular model updates, and user feedback integration, the system is designed to evolve and maintain effectiveness over time. This project not only demonstrates the practical application of data science and AI in crisis management but also highlights the potential of technology to save lives and reduce the impact of natural and man-made disasters.

REFERENCES

[1] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks.
[2] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. IEEE Conference on Computer Vision and Pattern
Recognition (CVPR).
[3] Radford, A., Kim, J. W., Hallacy, C., et al. (2021). Learning Transferable Visual Models From Natural Language Supervision (CLIP). OpenAI.
[4] Brown, T., Mann, B., Ryder, N., et al. (2020). Language Models are Few-Shot Learners (GPT-3). NeurIPS.
[5] Litjens, G., Kooi, T., Bejnordi, B. E., et al. (2017). A survey on deep learning in medical image analysis. Medical Image Analysis, 42, 60-88.
[6] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection (YOLO). CVPR.
[7] Jing, Y., Yang, Y., Feng, Z., Ye, J., Song, M., & Sebe, N. (2015). Visual search at Pinterest. KDD Conference on Knowledge Discovery and Data Mining.
[8] Morris, M. R., Koulichenko, E., Pruksachatkun, Y., et al. (2018). AI-powered accessibility tools: Seeing AI. ACM SIGACCESS Conference.
[9] Bengio, Y., Ducharme, R., Vincent, P., and Jauvin, C. (2003). A neural probabilistic language model. Journal of Machine Learning Research, 3(Feb):1137–1155.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi cross ref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details