WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Django-Based Emergency Alert System for Community Services Using WhatsApp and Email

Sejal Thakur[1], Prof. Ankush Dhamal[2]

[1] Prof. Ramkrishna More College(Autonomous) Pradhikaran Akurdi, Pune, India E-Mail: sejaltha1003@gmail@gmail.com
[2] Prof. Ramkrishna More College(Autonomous) Pradhikaran Akurdi, Pune, India E-Mail – ankushdhamal01@gmail.com

ABSTRACT :

The need for innovative communication solutions that can address emergencies and ensure swift assistance within communities is critical. This research project explores the development and implementation of a Django-based web application designed to facilitate emergency communication within communities. The application allows administrators to send WhatsApp and email alerts to registered users in real-time during emergencies or for requesting assistance in community services. The necessity of this project stems from past experiences where communities faced challenges in disseminating urgent information efficiently. By integrating modern communication technologies, this system aims to enhance the responsiveness and coordination of community members during critical situations.

Keywords: Emergency Communication, Django Web Application, WhatsApp API, Community Services, Real-time Alerts, Email Notifications, Twilio, User Authentication

Introduction :

In the digital age, swift and efficient communication during emergencies is critical for ensuring community safety and effective crisis management (3, 4). Traditional methods, such as phone calls or manual announcements, are often time-consuming and inefficient, particularly for large or dispersed populations. Studies highlight the advantages of digital platforms like WhatsApp and email for emergency communication, offering speed, reliability, and global reach (1, 2). This research develops a Django-based web application that leverages these tools for real-time alerts. WhatsApp's instant messaging capabilities, paired with email's reliability for detailed updates and traceability, create a multi-channel system that meets diverse communication needs (5, 6). The application allows administrators to send notifications quickly, with features like message templates, scheduling, and delivery tracking, enhancing emergency response coordination. Motivated by documented cases where delayed communication led to severe consequences, the platform is scalable for small communities and large organizations (7, 8).

Case Scenarios :

Scenario 1: Natural Disasters
Timely communication during disasters like earthquakes or floods is crucial for safety (3, 4). The system can quickly send alerts about evacuation routes, safe zones, and shelters, enabling real-time updates to reduce panic and save lives. For example, during floods, alerts can direct residents to higher ground or rescue points, coordinating efforts effectively (7).

Scenario 2: Public Health Emergencies
In health crises like pandemics, the system can disseminate accurate updates, safety protocols, vaccination schedules, and testing site locations (8, 9). For instance, during a pandemic, it can promote mask-wearing and social distancing while countering misinformation with verified resources, helping to control disease spread (1).

Scenario 3: Security Threats
During security threats, immediate alerts can safeguard lives by providing lockdown instructions and directing people to safe areas. For example, in an active shooter situation, the system can deliver real-time updates, helping individuals avoid danger zones until authorities secure the area (5, 10).

Scenario 4: Infrastructure Failures
In cases of infrastructure failures like power outages or water supply disruptions, the system can notify the community about repair timelines, safety precautions, or relief centers (11, 12). For instance, during power outages, it could provide blackout schedules and locations with temporary access to electricity and internet services (6).

Review of Literature :

The evolution of communication technologies has significantly enhanced emergency management, transitioning from traditional methods like sirens and public address systems to digital solutions (3, 4). While traditional tools are effective in limited contexts, they often lack scalability and immediacy for complex emergencies (7).

Digital platforms like WhatsApp have transformed emergency communication due to their real-time messaging, multimedia support, and high user adoption rates (1, 2). WhatsApp enables rapid alerts and two-way communication, improving engagement and real-time feedback from communities (7). Email, although less immediate, remains essential for formal communication, providing detailed updates, safety protocols, and resource documentation (8).

The integration of APIs, such as Twilio, automates and scales communication across channels, including SMS, WhatsApp, and email (5). This enhances operational efficiency and ensures timely delivery to large audiences, with customization options based on geographic or demographic factors (6). Research underscores the effectiveness of combining these tools to create a comprehensive communication framework for diverse user needs (9, 10).

Methodology :

The methodology for this project involves the design and development of a Django-based web application with specific functionalities tailored for emergency communication. The application includes user registration, authentication, and an admin dashboard for managing alerts. The integration of the Twilio API enables the sending of WhatsApp messages, while Django's email backend handles the delivery of email notifications.

The research will follow these steps:

1. **Requirement Analysis:** Understanding the needs of the community and the types of emergencies that require rapid communication.
2. **System Design:** Structuring the Django application to support user management, alert creation, and message dissemination.
3. **Development:** Implementing the application using Python, Django, and integrating Twilio for WhatsApp messaging.
4. **Testing:** Conducting functional and performance testing to ensure the reliability of the system in real-world scenarios.
5. **Deployment**: Launching the application and monitoring its performance in live environments.

Results and Discussion :

This project aims to develop a web application that enhances real-time emergency communication by integrating WhatsApp for instant alerts and email for detailed updates. The system will improve speed, reach, and reliability, with performance evaluated through metrics like response time, delivery rate, and user engagement.

Effectiveness:

Features like automated scheduling, real-time tracking, and simultaneous notifications optimize emergency response times.

Challenges:

Reliance on services like Twilio introduces risks such as interruptions and costs, addressed through fallback mechanisms like SMS and rigorous testing.

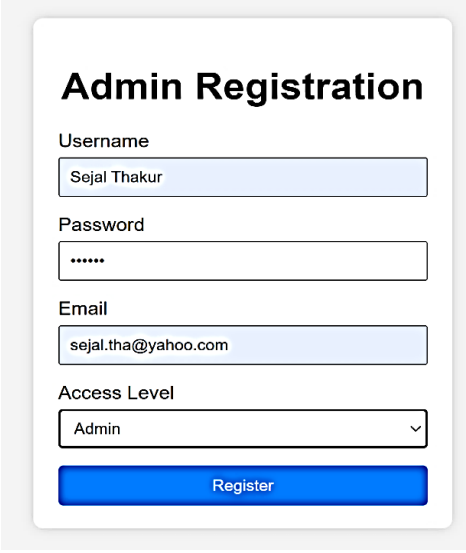
Scalability:

Designed to handle growing user bases, the system addresses server performance and user adoption challenges through optimization, user-friendly design, and awareness campaigns.

Future Prospects:

Planned enhancements include SMS notifications, multilingual support, and AI-driven analytics. The system aims to transform emergency communication by improving accessibility, reliability, and scalability.

Figure 1 : Admin Registration Form



This page in the website of community alert system is for the admins of specific areas to register at their own will and join the cause of this project. To volunteer to send alert messages to the people in the near by area about the situation which needs help to handle and where help is required. In this form the Admin has to input their Username, Password, Email and Access Level ex. Admin.

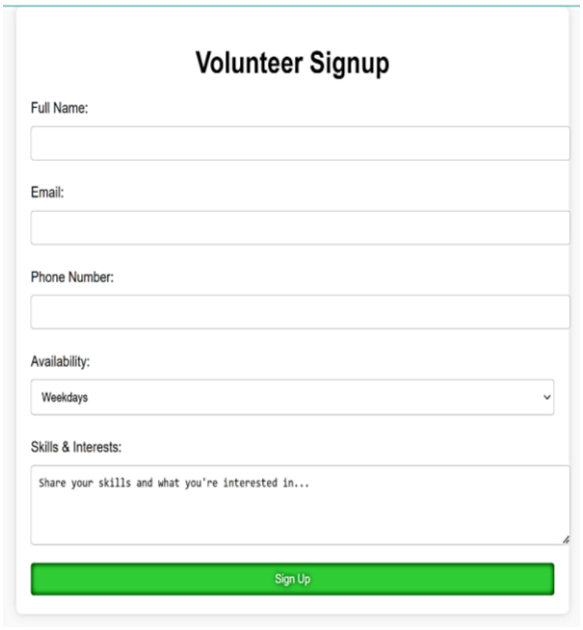


Figure 2 : Volunteer Signup Form

Figure 2 : Volunteer Signup Form

This page in the website of community alert system is for the Volunteers of specific areas to signup at their own will and join the cause of this project. To volunteer to consider helping in any way they can to the people in the near by area in the situation who needs help or assistance. In this form the Volunteer has to input their Full Name, Email, Phone number and Available time and Skills and Interests.

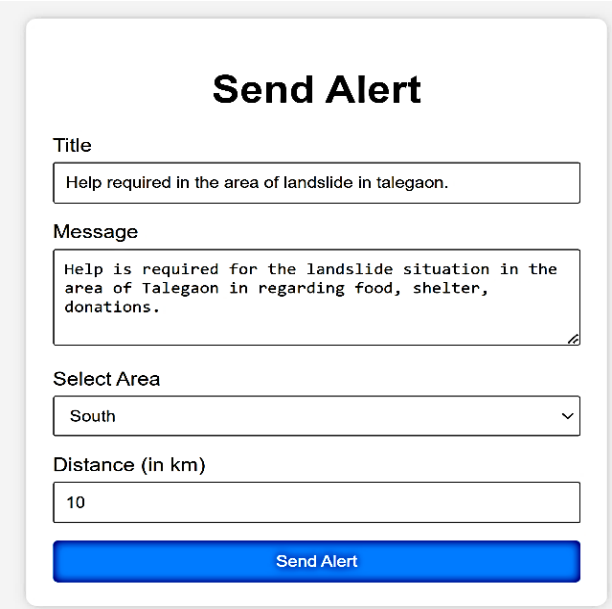


Figure 3: Send Alert Form

Figure 3 : Send Alert Page

This page in the website of community alert system is for the Admins to access to send the customized alert message to the nearby volunteers and inform them of the current situation and fulfil the cause of this project. In this form the Admin has to input the Title of the message, The body of the message, Select the Area and Distance (in Km).

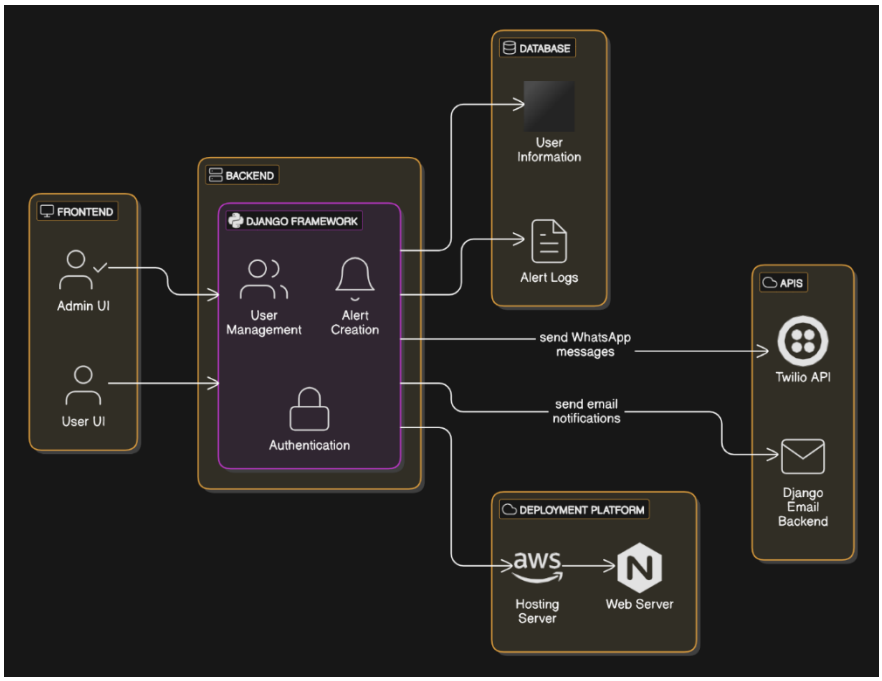


Figure 4 : Architecture of the Project

Figure 4 : Architecture of the Project

- Frontend: Admins manage users and alerts, while users receive notifications.
- Backend: Handles user management, alert creation, and secure authentication.
- Database: Stores user details and alert logs.
- APIs: Twilio sends WhatsApp messages; Django Email Backend handles email notifications.
- Deployment: Hosted on AWS with Nginx for reliability and scalability.

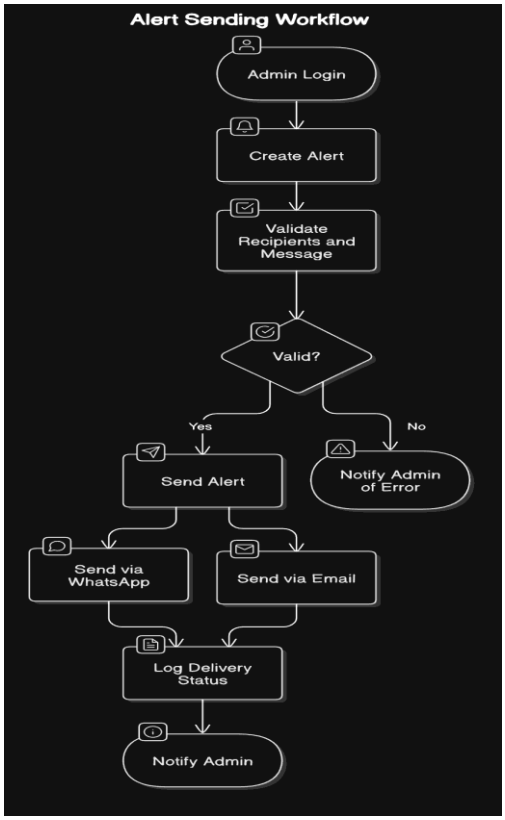
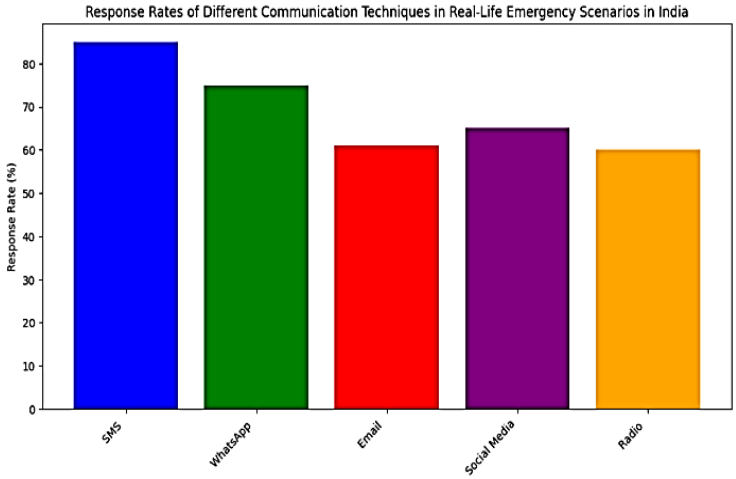


Figure 5: Layout of the system

Figure 5 : Layout of the system.

- Admin Login: The admin logs into the system.
- Create Alert: The admin drafts an alert message.
- Validation: Recipients and the message are validated for correctness.
  If valid, the process proceeds to send the alert.
  If invalid, the admin is notified of the error.
- Send Alert: The alert is sent via:
  WhatsApp for instant communication.
  Email for detailed updates.
- Log Delivery Status: Delivery details are logged for tracking.
- Notify Admin: The admin is informed of the alert's delivery status.



Graph 1: Response Rate Of Different Communication Techniques

Graph 1: Response Rate Of Different Communication Techniques

The bar chart illustrates the response rates of various communication methods used in emergency situations in India. SMS is the most effective, achieving a response rate of over 80%. WhatsApp follows closely with a response rate around 70%, indicating its strong effectiveness. Email has a slightly lower response rate, typically in the mid-60% range. Social Media and Radio have the lowest response rates, both falling in the 50-60% range, suggesting they are less reliable in emergencies.

Conclusion :

This research underscores the importance of integrating modern communication technologies into emergency management. The Django-based application developed here offers a scalable and user-friendly platform for fast, reliable dissemination of urgent information, enhancing community safety and preparedness. By streamlining communication channels, the system enables timely, life-saving responses during emergencies. As digital technologies evolve, such applications will become essential tools for crisis management, helping communities respond swiftly and reduce risks effectively.

REFERENCES :

1. How WhatsApp Stacks up Against SMS, Email, Messenger, and Voice Calls
By Adam Phillips, Technology Writer | ▷How WhatsApp Stacks up against more traditional Channels
2. Implementing an Emergency Alert System usingWhatsApp
November 24, 2023| by- Sid Bhutani
Implementing an Emergency Alert System usingWhatsApp - IDT Express
3. Alexander, D. (2002). Principles of Emergency Planning and Management. Oxford University Press.
4. Haddow, G., Bullock, J., & Coppola, D. P. (2017). Introduction to Emergency Management. Butterworth-Heinemann.
5. Twilio, Inc. (n.d.). Twilio API for WhatsApp. Retrieved from https://www.twilio.com/docs/whatsapp
6. Django Software Foundation. (n.d.). Django Documentation. Retrieved from https://docs.djangoproject.com/
7. Mani, N., & Mazumder, P. (2020). Leveraging WhatsApp for Emergency Communication: A Case Study. International Journal of Emergency Services.
8. Carreras-Coch, A., Navarro, J., Sans, C., & Zaballos, A. (2022). Communication Technologies in Emergency Situations. Electronics, 11(7), 1155. https://doi.org/10.3390/electronics11071155
9. Hussain, A. (2024). What's Emergency Communication? Understanding The Essentials. HSE Blog. Retrieved from https://www.hseblog.com/emergency-communications/
10. CISA. (2021). Emerging Communications Technologies: Emergency Communications by the Numbers. Retrieved from

https://www.cisa.gov/sites/default/files/publications/21_1116_necp_sns-emerging-tech_fact-sheet_508_c.pdf
11. Morell, A., Navarro, J., Sans, C., & Zaballos, A. (2022). Ubiquitous Sensor Networks for Emergency Communication. International Journal of Emergency Services.
12. National Emergency Communications Plan (NECP). (n.d.). Retrieved from https://www.cisa.gov/sites/default/files/publications/21_1116_necp_sns-emerging-tech_fact-sheet_508_c.pdf
13. European Union (EU). (n.d.). Emergency Communication Services in Different Countries. Retrieved from https://www.hseblog.com/emergency-communications/
14. United States and Canada. (n.d.). Emergency Communication Services in Different Countries. Retrieved from https://www.hseblog.com/emergency-communications/
15. Department of Emergency Communications. (n.d.). Key Communication Services. Retrieved from https://www.hseblog.com/emergency-communications/
16. Federal Communications Commission (FCC). (n.d.). Key Functions of the 911 System. Retrieved from https://www.hseblog.com/emergency-communications/
17. Fischer, H.W., III. (2008). Response to Disaster: Fact versus Fiction & Its Perpetuation. University Press of America.
18. Dynes, R.R. (1994). Community Emergency Planning: False Assumptions and Inappropriate Analogies. *International Journal of Mass Emergencies and Disasters*.
19. ITU (International Telecommunication Union). (2020). Emergency Communications Guidelines.
20 ISO Standards: Consider referencing ISO 22320:2018 for emergency management systems.
21. CDC (Centers for Disease Control and Prevention) or WHO for public health emergency communication strategies.