International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 02 | Feb - 2025 SJIF Rating: 8.448 ISSN: 2582-3930

"Unified Multi-Platform Messaging System for Seamless Client Communication"

Prof. Amol Jadhav1, Durgesh Chavan2, Harshvardhan Deshmukh3, Aditya Gajghate4, Akanksha Dhanawade5

Information Technology / Zeal collage of engineering and research Pune / Savirtibai Phule Pune University/India

---

Abstract:
Businesses and people in the current period use a variety of communication platforms to interact with their clientele. An "Automated Multi-Platform Messaging System for Multiple Client Communication Tool" that combines Gmail, WhatsApp, and SMS services is proposed in this study in response to the increasing need for smooth, automated multi-channel communication. React.js is used for the front end, Node.js for the back end, MongoDB for database administration, and integrations with Nodemailer, Twilio, and Puppeteer for automation. The technology makes it easier to handle recipient data by enabling CSV/Excel file uploads, which enables effective, adaptable mass communications. This article describes the architecture, design, and possible advantages of the system.

***Key Words*:** *Automation, Bulk Messaging, Gmail., Puppeteer, WhatsApp Node-mailer, email, Twilio, SMS*

1. INTRODUCTION

a. Problem Statement :

Currently, a number of solutions enable bulk messaging on particular platforms. Examples include the Gmail email system, the Twilio SMS API, and the WhatsApp Business API. Nevertheless, these technologies have drawbacks such as platform-specific constraints and an ununified user interface. Zapier and other integration tools provide cross-platform functionality, but they might not be scalable or flexible enough for large-scale operations.

b. Proposed Solution :

Existing technologies don't offer a complete solution that combines scalable data processing features like CSV/Excel file uploads with different communication channels. Their applicability in situations involving dynamic communication is further restricted by inflexible systems for handling recipient information and message templates. The suggested solution fills this void by offering a scalable, adaptable, and unified cross-platform communications platform.

2. Literature Review :

a. *Existing Solution:*

Mass messaging across several platforms is presently possible with a number of tools and methods. For example, Gmail's native email system, the Twilio API for SMS, and the WhatsApp Business API are all often used for automated messaging. Nevertheless, platform-specific constraints frequently limit these solutions, and they might not be able to manage several platforms from a single interface or provide simple integration for managing CSV/Excel data. Integrations are possible with current technologies like Integromat and Zapier, but their scalability and customization are frequently constrained.

a. Gaps:

Current solutions don't provide a unified, adaptable platform that integrates many message channels into a single tool with an emphasis on scalable data management (e.g., uploading CSV/Excel files). Additionally, a lot of these solutions have inflexible mechanisms for handling data and message templates, which can be problematic for large, dynamic messaging. This gap may be filled by developing a flexible, scalable system that gives companies the freedom to handle data in a variety of ways and enables effective cross-platform communication.

Related Work :

1. Implementation of Selenium Automation & Report Generator using Selenium WebDriver & ATF

Methods of Research To manage web service testing, automated testing scripts were created with Selenium WebDriver and connected with the Automated Testing Framework (ATF). Important Results and Contributions produced thorough reports for improved result interpretation, enhanced test coverage, and successfully decreased manual testing efforts. Research Limitations and Gaps reliance on ongoing script updates to handle possible maintenance issues and real-time testing scenarios.[1]

2. Automation for Ease of Life: To Schedule Bulk Messaging and Email System using Python

Methodology of Research created an automated system with Python, SMTP protocols, and the Gmail API to schedule a large number of emails and messages. Principal Results and Inputs shown the capacity to manage several receivers at once and to schedule messages effectively. Limitations and Gaps in the Research lacks advanced AI-driven customisation and is restricted to simple email and messaging services.[2]

3. Applying Bulk SMS System to Enhance Educational Communications

created the B-SWU bulk SMS system for educational communication at Srinakharinwirot University. Important Findings and Contributions emphasised how useful SMS systems are for academic alerts, public announcements, and quizzes. Research Limitations and Gaps limited in terms of internet functionality and message size. [3]

4. Survey on Sending Bulk Messages using AI Tools

Investigate Using AI-driven automation, scheduling libraries, Python, and natural language processing were integrated for WhatsApp mass messaging. Important Results and Contributions NLP-enabled dynamic messaging and reliable scheduling were displayed for the best possible communication. Research Limitations and Gaps There was little discussion of scalability, spam possibilities, and WhatsApp compliance. [4]

5. Portal for Bulk SMS Communication

Methods of Research In order to manage operations like SMS scheduling and delivery, the paper creates a communication site utilising a PCP scheduling algorithm. Important Results and Contributions: Streamlined mass SMS management for businesses with customisable templates and delivery failure alerts Research Limitations and Gaps Dependency on pre-set user actions, lack of sophisticated automation, and user-centric features. [5]

6. Neither Snow Nor Rain Nor MITM: An Empirical Analysis of Email Delivery Security

Methods of Research Through empirical examination of Gmail's SMTP traffic and configuration data for Alexa Top Million domains, the study calculates the rates at which SMTP security extensions are being adopted globally. Important Results and Contributions: highlighted issues such as STARTTLS downgrades and highlighted advancements in TLS encryption spearheaded by major mail providers. Research Limitations and Gaps: vulnerability to DNS spoofing and downgrade attacks as a result of lax certificate checking. [6]

7. Automation of Message Sending Processes Using Specialized Software

Methods of Research created a bot with Python and Selenium to automate image-based messages and personalised holiday greetings on WhatsApp Web. Important Results and Contributions creative use of Beautiful Soup for web data parsing and automation of messaging operations without official API support. Research Limitations and Gaps Scalability is impacted by the lack of an official API, restricted scope for personal usage, and dependency on human input for certain settings. [7]

8. Automated E-Commerce and Web Automation using Puppeteer

As part of its research approach, the study uses Puppeteer to automate e-commerce processes including order placing and login. To incorporate speech input, Google's Speech API is utilised. Significant findings and accomplishments shown significant time savings by automating repetitive e-commerce operations and offered hands-free user experiences. Limitations and Gaps in the Research Its limited compatibility with specific browsers and dependence on the structure of webpages limit the automation system's flexibility. [8]

Summery :

The reviewed works demonstrate important developments in e-commerce, testing, and automated communication. They show the increasing demand for AI-driven, scalable, and secure solutions that can reduce human input while preserving platform compliance and reliable performance. To promote more complex and flexible automation systems, future research should concentrate on overcoming constraints like platform dependency, scalability issues, security flaws, and a lack of advanced customisation.

3. System Architecture :

3.1. Technology Stack :

- **Frontend:** React.js for building a responsive and interactive user interface.
- **Backend:** Node.js for handling HTTP requests, business logic, and server-side operations.
- **Database:** MongoDB for storing user data, message templates, and recipient information.
- **Automation Tools:**
  - **WhatsApp:** Puppeteer for WhatsApp web automation.
  - **Email:** Nodemailer library with Google SMTP for email messaging.

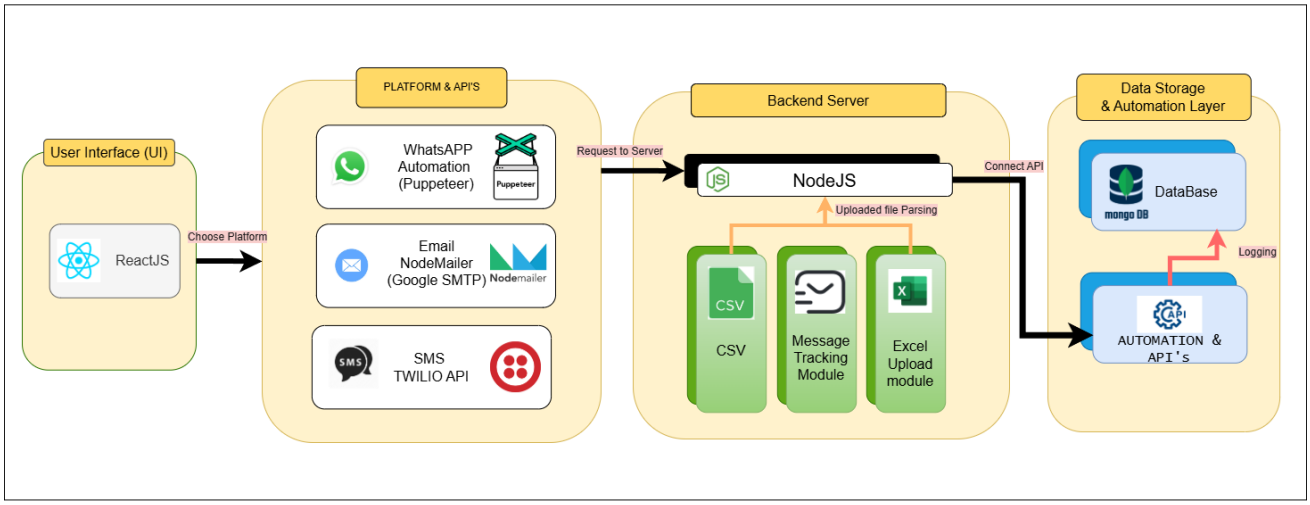

Fig 1: System Architecture

- **SMS:** Twilio API for SMS messaging.

3.2. Workflow :

1. **Recipient Data Upload:** Users upload a CSV/Excel file containing recipient details.
2. **Data Processing:** The backend processes the data, allowing users to select a messaging platform (WhatsApp, Gmail, or SMS).
3. **Message Customization:** The system dynamically customizes the message with placeholders.
4. **Message Sending:** Messages are sent across the chosen platform using respective APIs (Puppeteer for WhatsApp, Nodemailer for email, Twilio for SMS).
5. **Data Tracking:** MongoDB tracks message delivery attempts and manages recipient data efficiently.
6. **Status Updates:** Users receive status updates on message delivery and success.

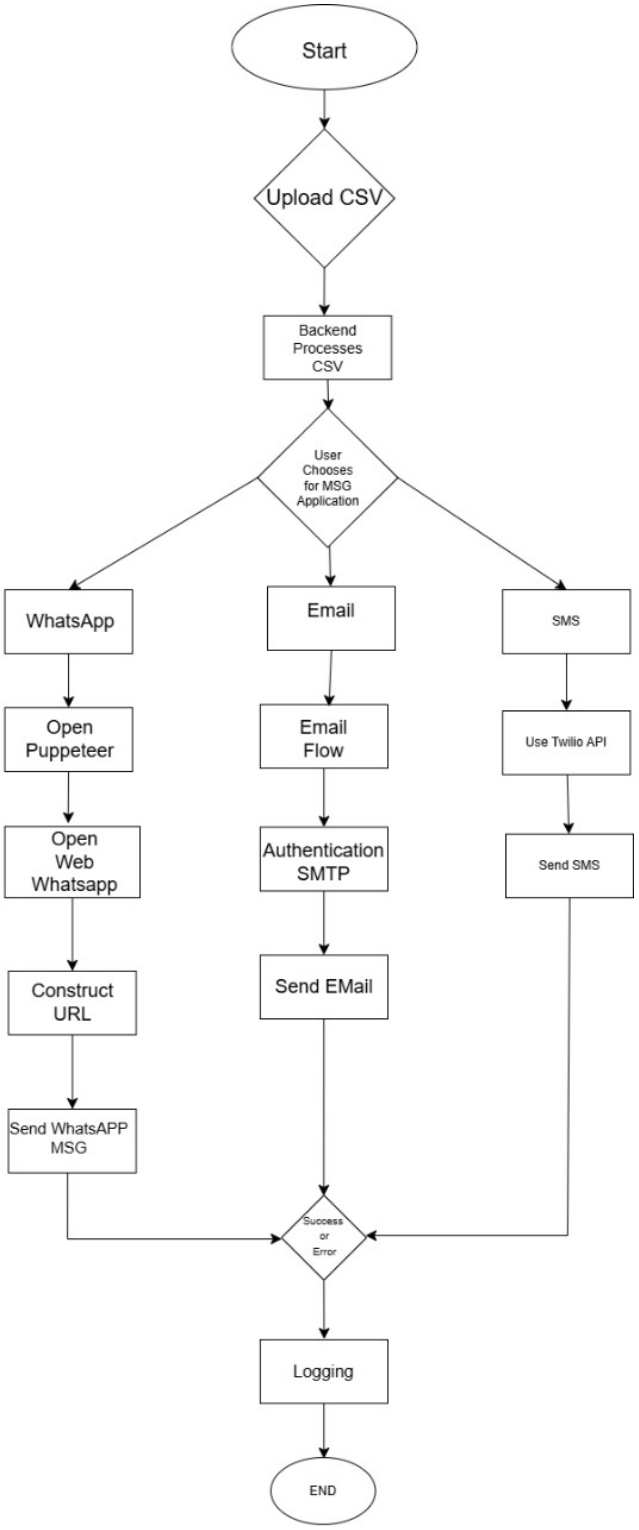

Fig 2: FlowChart

4. Advantages of the Proposed System:

1. **Cross-Platform Integration:** The system offers seamless integration between Gmail, WhatsApp, and SMS services.
2. **Efficiency:** Automation reduces the time spent on repetitive communication tasks.
3. **Data Handling:** The ability to upload CSV/Excel files ensures efficient recipient data management.
4. **Dynamic Message Customization:** Support for placeholders allows for personalized messages.
5. **Scalability:** The system can handle large volumes of messages efficiently.
6. **User-Friendly Interface:** A responsive frontend makes it easy to use even for non-technical users.

5. Challenges and Limitations :

1. **API Dependency:** The system's reliance on third-party APIs could affect performance if those APIs change.
2. **Compliance Issues:** Legal and compliance requirements may vary across platforms.
3. **Scalability Constraints:** Although scalable, managing extremely large datasets may require additional infrastructure.
4. **Delivery Uncertainty:** Messages may not always be delivered due to recipient platform restrictions.

6. CONCLUSIONS:

For both people and enterprises, the Automated Multi-Platform Messaging System offers a scalable and effective way to handle bulk communications across multiple platforms. The tool provides an easy-to-use user interface and backend processing by utilizing contemporary web automation tools, APIs, and a versatile database architecture. This cohesive strategy closes the gaps left by previous solutions, allowing for smooth, automated messaging on the SMS, Gmail, and WhatsApp platforms.

REFERENCES:

[1] Implementation of selenium automation & report generation using Selenium Web Driver & ATF. (n.d.). Retrieved from https://ieeexplore.ieee.org/document/9392455/

[2] Kapoor, N. a. S. S. a. S. (2020). Automated E-Commerce and Web Automation using Puppeteer. International Journal for

Modern Trends in Science and Technology, 6(12), 277–281. https://doi.org/10.46501/ijmtst061251

[3] N. S. Chandra, N. U. Raj, N. R. Sonkar, N. P. Srivastava, N. V. P. Singh, and N. M. A. A. Niazi, "Automation for Ease of Life : To Schedule Bulk Messaging and Email System using Python," International Journal of Scientific Research in Computer Science Engineering and Information Technology, pp. 363–365, Nov. 2022, doi: 10.32628/cseit228649.

[4] Sukaphat, S. (2007). Applying of bulk SMS system to enhance educational communications. ResearchGate. Retrieved from https://www.researchgate.net/publication/228680297_Applying_of_bulk_SMS_system_to_enhance_educational_communications

[5] Survey on Sending Bulk Messages using AI Tools 1Mr. Nilesh Vispute, 2Chirag Patil, 3Dhruv Surve, 4Pranay Kamble. Retrieved from https://ijrpr.com/uploads/V4ISSUE10/IJRPR18359.pdf

[6] G, N., & S, V. (2020). Bulk SMS Communication Portal. In Advances in parallel computing. https://doi.org/10.3233/apc200171

[7] Neither Snow Nor Rain Nor MITM...: An Empirical Analysis of Email Delivery Security . https://dl.acm.org/doi/10.1145/2815675.2815695

[8] Automation of message sending processes using specialized software. (n.d.). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/document/9396564/

[9] Puppeteer Documentation [Online]. Available: https://pptr.dev/

[10] Nodemailer Documentation [Online]. Available: https://nodemailer.com/about/

[11] Twilio API Documentation [Online]. Available: https://www.twilio.com/docs

[12] MongoDB Official Documentation [Online]. Available: https://www.mongodb.com/docs/

[13] React.js Documentation [Online]. Available: https://react.dev/

[14] Node.js Documentation [Online]. Available: https://nodejs.org/en/docs/

[15] Vorontsov, M., & Radmir, S. I. (2021). Automation of Message Sending Processes Using Specialized Software. 2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus), 746–748. https://doi.org/10.1109/elconrus51938.2021.9396564

[16] Varun M., Aman Simon Santhosh, & Shah, K. R. (2023). Power of Automation Testing. https://doi.org/10.2139/ssrn.4464453

[17] Sawant, K., Tiwari, R., Vyas, S., Sharma, P., Anand, A., & Soni, S. (2021). Implementation of Selenium Automation & Report Generation Using Selenium Web Driver & ATF. 2021 International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT), 1–6. https://doi.org/10.1109/icaect49130.2021.9392455

[18] Zhang, Y., Li, X., & Wang, J. (2023). Multi-message Multi-receiver Signcryption Scheme Based on Blockchain. Mathematical Biosciences and Engineering, 20(10), 806. https://doi.org/10.3934/mbe.2023806

[19] Chandra, S., Raj, U., Sonkar, R., Srivastava, P., Singh, V. P., & Niazi, M. A. A. (2022). Automation for Ease of Life: To Schedule Bulk Messaging and Email System using Python. International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 8(6), 363–365. https://doi.org/10.32628/CSEIT228649

[20] Boldyreva, A., & Kumar, V. (2007). Multi-recipient Encryption Schemes: How to Save on Bandwidth and Computation Without Sacrificing Security. IEEE Transactions on Information Theory, 53(11), 3927–3943. https://doi.org/10.1109/TIT.2007.907471

[21] García, B., Ricca, F., del Álamo, J. M., & Leotta, M. (2023). Enhancing Web Applications Observability through Instrumented Automated Browsers. Journal of Systems and Software, 203, 111723. https://doi.org/10.1016/j.jss.2023.111723

[22] Raharjana, I. K., Arifin, M. F., & Nur, M. (2023). Conversion of User Story Scenarios to Python-Based Selenium Source Code for Automated Testing. TEM Journal, 12(1), 309–315. https://doi.org/10.18421/TEM121-40

[23] Lukasczyk, S., Kroiß, F., & Fraser, G. (2023). An Empirical Study of Automated Unit Test Generation for Python. Empirical Software Engineering, 28(2), 36. https://doi.org/10.1007/s10664-022-10248-w