

A SOCIETAL RELATED Project Report on
WhatsApp Automation

Submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

In

CSE (DATA SCIENCE)

By

B. Sai Vinay (22AG1A6709)

G. Abhiram (22AG1A6722)

S. Praveen (22AG1A6749)

Under the guidance of

K. Kiran Babu

Assistant Professor.



DEPARTMENT OF CSE (DATA SCIENCE)

ACE Engineering College

Ankushapur(V), Ghatkesar(M), Medchal Dist - 501 301

(An Autonomous Institution, Affiliated to JNTUH, Hyderabad)

www.aceec.ac.in

2024-2025



ACE Engineering College

UGC AUTONOMOUS INSTITUTION

(Sponsored by Yadala Satyanarayana Memorial Educational Society, Hyderabad)

Approved by AICTE & Affiliated to JNTUH

B.Tech Courses offered: CIVIL, CSE, IT, ECE, EEE & MECH, NBA Accredited Courses: CIVIL, CSE, ECE, EEE & MECH, Accorded NAAC A - Grade

DEPARTMENT OF CSE (DATA SCIENCE)

CERTIFICATE

This is to certify that the **Societal Related (Or) Real Time** project report entitled “WHATSAPP AUTOMATION” is a bonafide work done by **B. Sai Vinay, G. Abhiram & S. Praveen** bearing (22AG1A6709, 22AG1A6722, 22AG1A6749) in partial fulfillment for the award of Degree of BACHELOR OF TECHNOLOGY in CSE (Data Science) from JNTUH University, Hyderabad during the academic year 2023- 2024. This record of bonafide work carried out by them under our guidance and supervision.

The results embodied in this report have not been submitted by the student to any other University or Institution for the award of any degree or diploma.

(K. Kiran Babu)
Assistant Professor
Supervisor

(Dr. P. Chiranjeevi)
Associate Professor
H.O.D., CSE-DS.

ACKNOWLEDGEMENT

We would like to express my gratitude to all the people behind the screen who have helped me transform an idea into a real time application.

We would like to express my heart-felt gratitude to my parents without whom we would not have been privileged to achieve and fulfill my dreams.

A special thanks to our General Secretary, **Prof. Y. V. Gopala Krishna Murthy**, for having founded such an esteemed institution. Sincere thanks to our Joint Secretary **Mrs. M. Padmavathi**, for support in doing project work. I am also grateful to our beloved principal, **Dr. B. L. RAJU** for permitting us to carry out this project.

We profoundly thank **Dr. P. Chiranjeevi**, Associate Professor and Head of the Department of Computer Science and Engineering (Data Science), who has been an excellent guide and also a great source of inspiration to my work.

We extremely thank **Mrs. B. Saritha** and **Mr. M. Hari Krishna**, Assistant Professors, Project coordinators, who helped us in all the way in fulfilling of all aspects in completion of our Mini-Project.

We are very thankful to my internal guide **Mr. K. Kiran Babu** who has been an excellent and also given continuous support for the Completion of my project work.

The satisfaction and euphoria that accompany the successful completion of the task would be great, but incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crown all the efforts with success. In this context, I would like to thank all the other staff members, both teaching and non-teaching, who have extended their timely help and eased my task.

B Sai Vinay (22AG1A6709)

G Abhiram (22AG1A6722)

S Praveen (22AG1A6749)

WHATSAPP AUTOMATION

ABSTRACT

The **WhatsApp Automation Project** aims to optimize communication processes, streamline interactions, and elevate user experiences. Developing an automated management system integrating personal assistance, personalized messages and Event-triggered notifications in WhatsApp, addressing challenges such as customized message sending, manual errors, and time-consuming processes.

The WhatsApp Automation Project centers around automating various communication tasks within the WhatsApp ecosystem. By leveraging intelligent algorithms and predefined rules, businesses can enhance their responsiveness, reduce manual workload, and provide timely assistance to users. The project encompasses several essential features: Automated Replies, Personalized Messages, Event-Triggered Notifications, Enhanced user Experience, Chatbot Integration, Benefits and Future Scope.

In conclusion, the WhatsApp Automation Project empowers businesses to communicate effectively, improve customer relationships, and stay ahead in the digital age. By embracing automation, organizations can create seamless experiences that resonate with users worldwide.

CONTENT

S.NO	CHAPTER NAME	PAGE NO
1	INTRODUCTION	1
2	EXISTING SYSTEM & DRAWBACKS	2-3
3	LITERATURE SURVEY	4
4	PROPOSED MODEL / SYSTEM	5
5	REQUIREMENTS GATHERING	
	5.1 Software Requirements & Hardware Requirements	6
	5.2 Functional Requirements & Non-Functional Requirements	7
	5.3 Data Collection	8
6	SYSTEM ANALYSIS & DESIGN	
	6.1 Module Description	9
	6.2 System Architecture	9
	6.3 UML Diagrams	10-12
7	IMPLEMENTATION	
	7.1 Algorithm Used	13
	7.2 Sample Code	14-15
8	TESTING	16
9	RESULTS	17-18
10	CONCLUSION	19
	FUTURE WORK	20
	REFERENCES	20

LIST OF FIGURES

F.NO	FIGURE NAME	PAGE NO
1.	SERVER SIDE ARCHITECTURE	9
2.	CLIENT SIDE ARCHITECTURE	9
3.	DATA FLOW DIAGRAM	10
4.	CLASS DIAGRAM	10
5.	COMPONENT DIAGRAM	11
6.	USE CASE DIAGRAM	11
7.	SEQUENCE DIAGRAM	11
8.	DEPLOYMENT DIAGRAM	12
9.	E-R DIAGRAM	12
10.	OUTPUT DIAGRAM	17-18

1. INTRODUCTION

WhatsApp Automations are essentially the technology utilized to automate routine tasks. The automated processes are to simplify and enhance interactions between businesses and their customers. This includes automated replies, customer support, marketing and broadcast campaigns, and leads tracking. Automation enables companies to improve efficiency, reduce costs, and provide a better customer experience.

Problem Synopsis:

Developing an automated management system integrating personal assistance, personalized messages is the task to deal with. WhatsApp Automation helps in Event-triggered, Automated, Personalized notifications in WhatsApp, addressing challenges such as customized message sending, manual errors, and time-consuming processes.

Project Overview:

The WhatsApp Automation Project centers around automating various communication tasks within the WhatsApp ecosystem. By leveraging intelligent algorithms and predefined rules, businesses can enhance their responsiveness, reduce manual workload, and provide timely assistance to users.

Features and Functionalities:

The project encompasses several essential features:

- Automated Replies
- Personalized Messages
- Event-Triggered Notifications
- Enhanced user Experience.

Synopsis:

In conclusion, the WhatsApp Automation Project empowers businesses to communicate effectively, improve customer relationships, and stay ahead in the digital age. By embracing automation, organizations can create seamless experiences that resonate with users worldwide.

2. EXISTING SYSTEM & DRAWBACKS

EXISTING SYSTEM

Existing Features of WhatsApp:

WhatsApp has become an essential messaging tool in our daily lives, and if your business is not taking advantage of this, you may be losing out on certain benefits that can bring your brand to the next level.

WhatsApp Forwarding Messages:

- The WhatsApp provide a feature of Forwarding messages, which engages users in sharing information across other users.
- However, this WhatsApp feature restricts the user in forwarding messages to large number of users.

WhatsApp Business App:

- It allows businesses to send welcome messages, away messages, sales information and abandoned cart reminders to users.
- By automating common customer queries, businesses can reduce response time and keep their operations active 24/7, even during weekends and holidays.
- However WhatsApp Business fails in One to One Customer responsiveness and Personalized Messages to other customers or organizations.

EXISTING SYSTEM DRAWBACKS

- WhatsApp Messages Forwarding feature restricts the user in forwarding messages to large number of users.
- This Forwarding Message feature only sends a predefined common text for all the users at once.
- Timely Assistance and Work Output respective with time will be another drawback in existing system. It is difficult to maintain a person or even a job holder to work round the clock 24/7, even during weekends and holidays.
- However WhatsApp Business App Feature fails in One to One Customer responsiveness and Personalized Messages to other customers or organizations.
- Automation transforms the way businesses engage with their audience, making interactions more scalable, efficient, and personalized.
- Common Typing Errors is also a type of setback in Forwarding Messages, it leads in false information and misconceptions in sharing the information.
- Various Organizations need assistance of WhatsApp in its own way, The existing features in WhatsApp doesn't allow to work according to their specifications and usage.

3. LITERATURE SURVEY

WhatsApp automation, involving the automated execution of tasks within the messaging platform, has emerged as a focal point of research and practical implementation. Academic inquiries delve into technical methodologies like API utilization and chatbot development, offering insights into their effectiveness and limitations.

Practical guides and case studies demonstrate real-world applications across diverse sectors such as business, education, and healthcare, highlighting the transformative potential of automation. However, challenges persist, including concerns surrounding user privacy, data security, and adherence to regulatory frameworks. Looking ahead, future research directions may emphasize the refinement of chatbot capabilities through advancements in natural language processing and the integration of emerging technologies.

Additionally, comparative analyses with analogous platforms offer valuable perspectives for optimizing automation strategies.

4. PROPOSED MODEL/ SYSTEM

WhatsApp Bot simplifies tasks on WhatsApp, benefiting businesses and individuals.

- It aims to automate responses and manage tasks efficiently.
- WhatsApp Automation covers the drawbacks in existing system as maintaining work round the clock 24/7, even during weekends and holidays.
- Automated Responses, Task Scheduling, Personalization enhance user experience and productivity.
- WhatsApp Bot comprises an WhatsApp API Integration, and Database for user data storage.
- Developed using modern programming languages and frameworks, WhatsApp Bot ensures smooth operation and user satisfaction.
- WhatsApp Bot aids in Customer Support, Appointment Scheduling, and Task Management, optimizing workflows.
- Dynamic Messaging feature in the WhatsApp Automation helps in avoiding the Common typing Errors & helps in automating the forwarding messages task and Personalized messages task in a efficient way.

WhatsApp Automation offers a robust solution for automating tasks and improving communication on WhatsApp, with potential for further refinement and application.

5. REQUIREMENTS GATHERING

5.1 SOFTWARE REQUIREMENTS & HARDWARE REQUIREMENTS

Software Requirements:

1. **Programming Language:** Python
2. **Integrated Development Environment (IDE):** IDEs like Jupyter / Anaconda
3. **WhatsApp Web:** WhatsApp API (pywhatkit) to enable communication and automation capabilities on the platform.
4. **Libraries / Modules :** Selenium , Chrome WebDriver, SQLite
5. **Database Management System:** SQLite to store user data, chat logs, and task information.

Hardware Requirements:

1. **Processor:** CPU (Equal to Intel Core i3 2.6 GHz) with multi-core or high, 64-bit
2. **Memory (RAM):** 4 GB (Minimum) / 8GB (Recommended)
3. **Storage:**
 - **SSD:** For faster read/write operations.
 - **Capacity:** Starting from 100 GB to accommodate future growth
4. **Networking:** High bandwidth and low latency to manage real-time messaging effectively.
5. **Scalability:** Infrastructure that supports scaling, with load balancers and backup solutions for high availability and data protection.

5.2 FUNCTIONAL REQUIREMENTS & NON-FUNCTIONAL REQUIREMENTS

Functional Requirements:

The system is designed to enhance communication efficiency by automating message sending through WhatsApp. Key functionalities include:

- **Automated Message Sending:** Supports text, images, videos, voice notes, and documents.
- **User Authentication:** Secure login using phone numbers and verification codes.
- **Contact Management:** Add, edit, and delete contacts, with the ability to sync from the user's phone.
- **Event-Triggered Responses:** Automates responses based on specific triggers.
- **Real-Time Notifications:** Alerts users of new messages and events.

Non-Functional Requirements:

The system's non-functional aspects focus on ensuring a robust and user-friendly experience:

- **Security and Privacy:** End-to-end encryption safeguards user data, with compliance to privacy regulations.
- **Scalability:** Designed to handle a growing number of users and messages, supporting horizontal scaling.
- **Compatibility:** Available on multiple platforms (iOS, Android, web), with adaptable screen sizes and orientations.
- **Performance:** Quick loading times and near-instantaneous message delivery ensure a seamless user experience.

5.3 DATA COLLECTION

Data collection in the WhatsApp Automation Project involves gathering necessary data to optimize and personalize interactions. This includes:

1. **User Data:** Collecting user data such as phone numbers, contact lists, and message logs. This data is essential for personalizing messages and managing contacts.
2. **Message Data:** Gathering data on the types of messages sent, their frequency, and response times. This data helps in understanding user behaviour and improving automated responses.
3. **Interaction Logs:** Maintaining logs of all interactions to monitor the performance of the automation system and make necessary adjustments.
4. **Database Management:** Using a database management system like SQLite to store and manage the collected data. This ensures that data is organized and easily accessible for processing and analysis .

6. SYSTEM ANALYSIS & DESIGN

6.1 MODULE DESCRIPTION

The system is divided into several key modules:

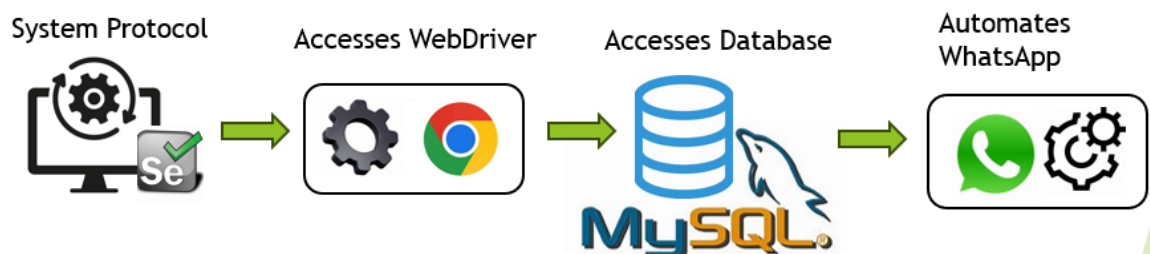
- Authentication Module: Manages user login and session management.
- Messaging Module: Handles the sending and receiving of various message types.
- Contact Management Module: Allows users to manage their contacts, including synchronization and manual entry.

Each module is designed to work seamlessly with the others, ensuring a cohesive and functional system.

6.2 SYSTEM ARCHITECTURE

SERVER SIDE ARCHITECTURE

- Backend services for message processing, authentication, and database management.
- Ensures secure and efficient handling of user data.



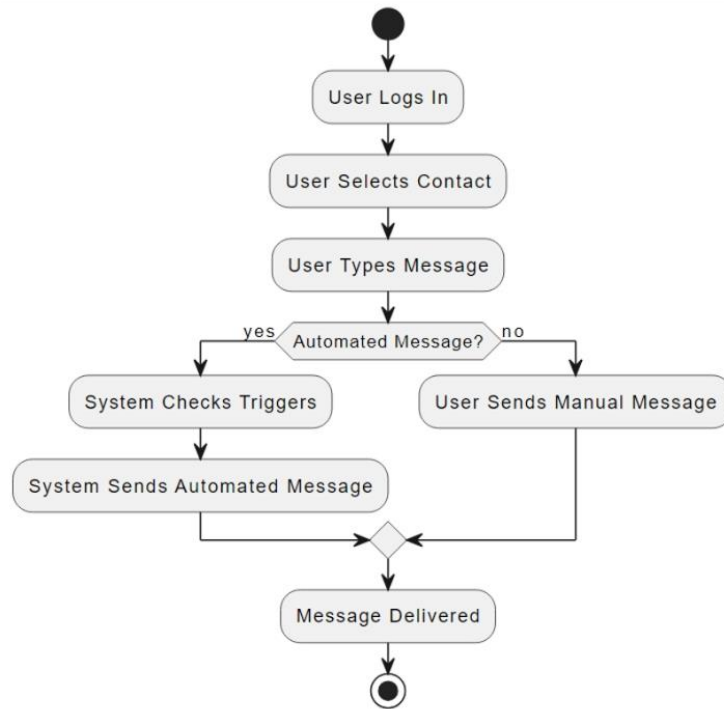
CLIENT SIDE ARCHITECTURE

- User interface accessible on web, Android, and iOS platforms.
- Implements UI components for messaging, contact management, and notifications.

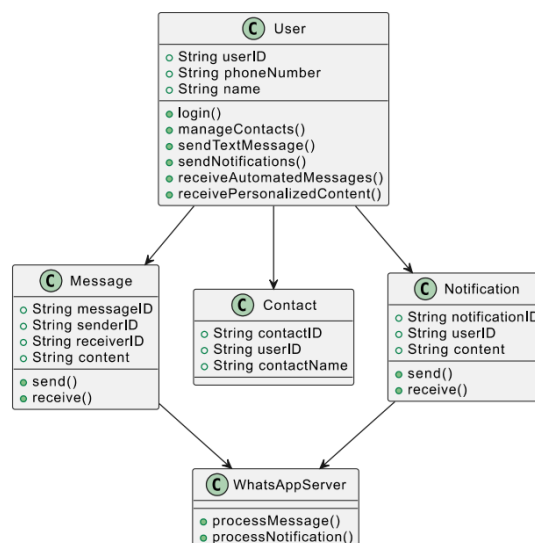


6.3 UML DIAGRAMS

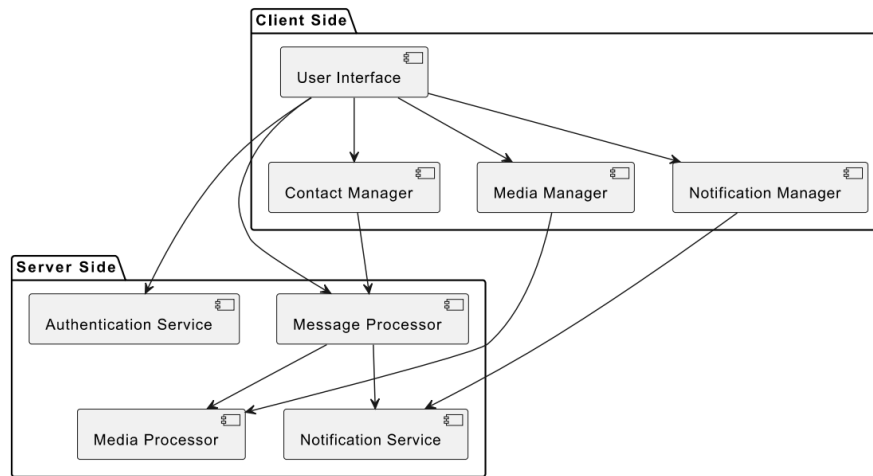
1. Dataflow Diagram



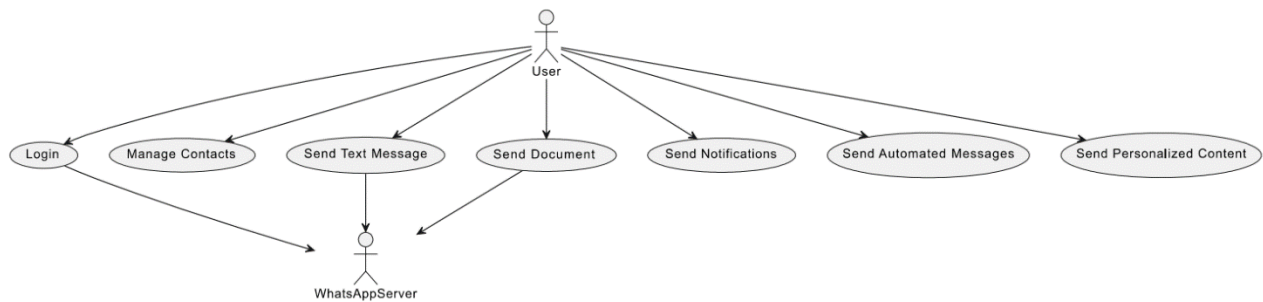
2. Class Diagram



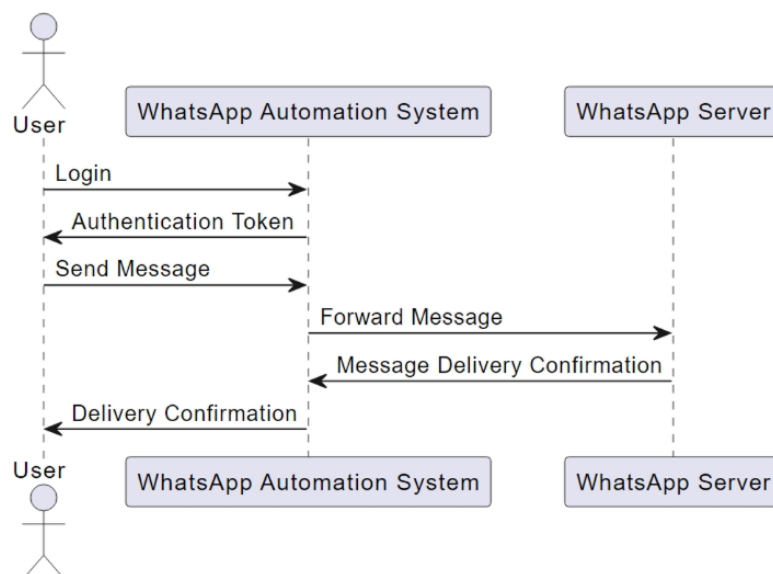
3. Component Diagram



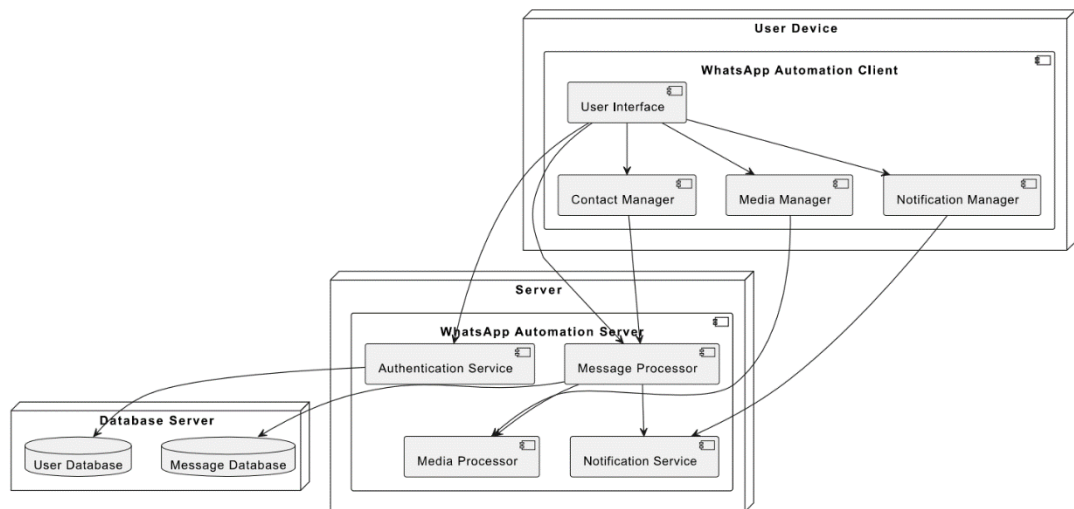
4. Use Case Diagram



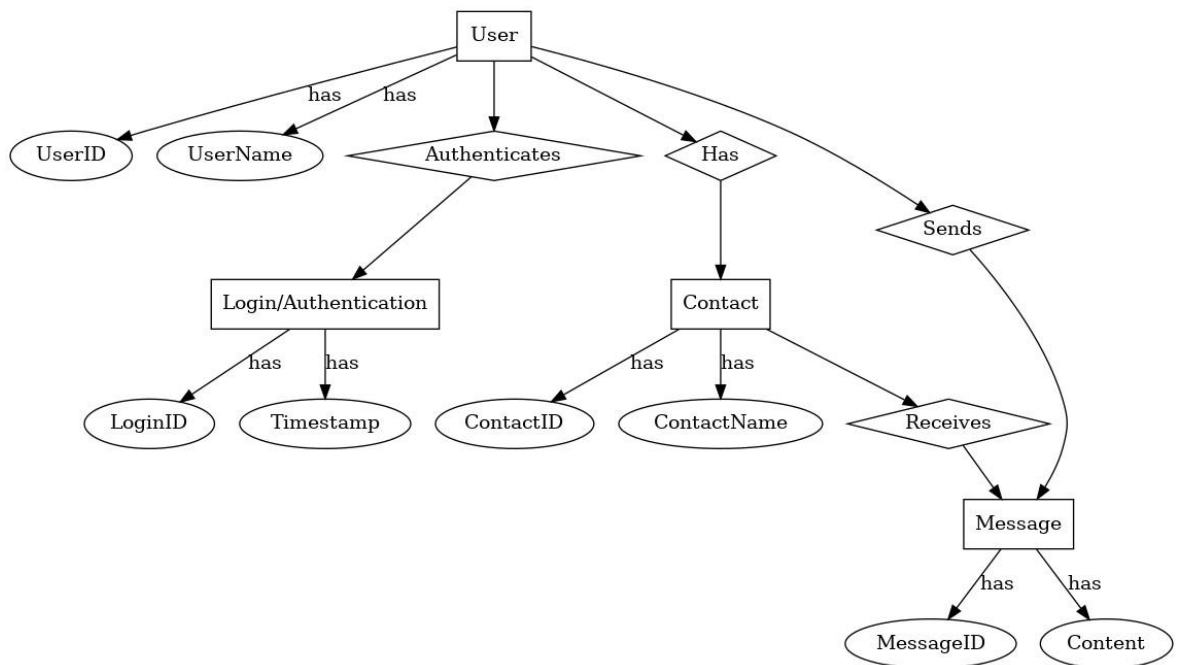
5. Sequence Diagram



6. Deployment Diagram



7. E-R DIAGRAM



7. IMPLEMENTATION

7.1 ALGORITHM USED

The implementation of the WhatsApp Automation system leverages several algorithms and libraries to achieve seamless communication and automation. Key components include:

Selenium WebDriver: This tool is used for automating web browser interactions, particularly for WhatsApp Web. Selenium allows the system to log into a user's WhatsApp account, navigate the interface, and interact with elements such as contact lists and chat windows. This is crucial for sending messages and checking message statuses.

Pywhatkit: This Python library facilitates sending messages to unsaved numbers, a feature not natively supported by WhatsApp. Pywhatkit enables the system to schedule messages and automate the sending process without manual intervention.

Data Processing and Storage: The system uses algorithms to manage and process collected data, such as contact details and message logs. Data is typically stored in a secure database, enabling efficient retrieval and management. The system may use SQLite databases depending on scalability needs and data complexity.

7.2 SAMPLE CODE

```
import time
import sqlite3
import tkinter as tk
from tkinter import messagebox, scrolledtext
from tkinter import ttk
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.by import By
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from selenium.common.exceptions import WebDriverException, TimeoutException, ElementClickInterceptedException
from selenium.webdriver.common.action_chains import ActionChains
from webdriver_manager.chrome import ChromeDriverManager

# Function to send a whatsapp message to a saved contact
def send_message_to_contact(contact_name, message):
    try:
        log_message(f"Searching for contact: {contact_name}...")

        search_box = wait.until(EC.element_to_be_clickable((By.XPATH, '//div[@contenteditable="true"][@data-tab="3"]')))
        search_box.click()
        search_box.send_keys(Keys.CONTROL + "a")
        search_box.send_keys(Keys.BACKSPACE)
        time.sleep(1)

        actions.move_to_element(search_box).click().send_keys(contact_name).perform()
        time.sleep(2)

    except Exception as e:
        log_message(f"An error occurred while sending message to {contact_name}: {str(e)}")

def send_message_to_unsaved_number(target_number, message):
    try:
        url = f"https://web.whatsapp.com/send?phone={target_number}&text={message}"
        driver.get(url)

        input_box = wait.until(EC.element_to_be_clickable((By.XPATH, '//div[@contenteditable="true"][@data-tab="10"]')))
        input_box.send_keys(Keys.ENTER)
        time.sleep(2)

        log_message(f"Message sent to {target_number}")
    except Exception as e:
        log_message(f"An error occurred while sending message to {target_number}: {str(e)}")

def start_whatsapp_automation():
    global driver, wait, actions
    driver, wait, actions = initialize_driver()
    driver.get('https://web.whatsapp.com/')
    log_message("Please scan the QR code to log in to WhatsApp Web")
    time.sleep(15)

    conn = sqlite3.connect('whatsapp_contacts.db')
    c = conn.cursor()
    c.execute('SELECT name, message FROM contacts')
    rows = c.fetchall()

    for row in rows:
        contact_name, message = row
        send_message_to_contact(contact_name, message)

def save_to_database():
    conn = sqlite3.connect('whatsapp_contacts.db')
    c = conn.cursor()

    c.execute('DROP TABLE IF EXISTS contacts')
    c.execute('''
        CREATE TABLE contacts (
            id INTEGER PRIMARY KEY,
            name TEXT NOT NULL,
            message TEXT NOT NULL
        )
    ''')

    contacts = contacts_text.get("1.0", tk.END).strip().split('\n')
    messages = messages_text.get("1.0", tk.END).strip().split('\n')

    if len(contacts) != len(messages):
        messagebox.showerror("Error", "Number of contacts and messages must match.")
        return

    for contact, message in zip(contacts, messages):
        c.execute('INSERT INTO contacts (name, message) VALUES (?, ?)', (contact, message))

    c.execute('DROP TABLE IF EXISTS unsaved_contacts')
    c.execute('''
        CREATE TABLE unsaved_contacts (
            id INTEGER PRIMARY KEY,
            name TEXT NOT NULL,
            message TEXT NOT NULL
        )
    ''')
```

```

# Setting up the GUI
root = tk.Tk()
root.title("WhatsApp Automation")
root.geometry("700x800")
root.configure(bg="#d0f0c0")

style = ttk.Style()
style.configure("TLabel", background="#d0f0c0", font=('Helvetica', 12))
style.configure("TButton", font=('Helvetica', 12), padding=10)
style.configure("TFrame", background="#d0f0c0")

# Heading
heading = ttk.Label(root, text="WhatsApp Automation", font=('Helvetica', 16, 'bold'), background="#d0f0c0")
heading.pack(pady=10)

# Frame
frame = ttk.Frame(root, padding="20", style="TFrame")
frame.pack(fill=tk.BOTH, expand=True)

# Saved Contacts Section
contacts_label = ttk.Label(frame, text="Saved Contacts (one per line):")
contacts_label.grid(row=0, column=0, sticky=tk.W, pady=(0, 10))
contacts_text = tk.Text(frame, height=5, width=60)
contacts_text.grid(row=0, column=1, pady=(0, 10))

messages_label = ttk.Label(frame, text="Messages for Contacts (one per line):")
messages_label.grid(row=1, column=0, sticky=tk.W, pady=(0, 10))
messages_text = tk.Text(frame, height=5, width=60)
messages_text.grid(row=1, column=1, pady=(0, 10))

```

```

# Unsaved Contacts Section
unsaved_numbers_label = ttk.Label(frame, text="Unsaved Numbers (one per line):")
unsaved_numbers_label.grid(row=2, column=0, sticky=tk.W, pady=(10, 10))
unsaved_numbers_text = tk.Text(frame, height=5, width=60)
unsaved_numbers_text.grid(row=2, column=1, pady=(10, 10))

unsaved_messages_label = ttk.Label(frame, text="Messages for Unsaved Contacts (one per line):")
unsaved_messages_label.grid(row=3, column=0, sticky=tk.W, pady=(0, 10))
unsaved_messages_text = tk.Text(frame, height=5, width=60)
unsaved_messages_text.grid(row=3, column=1, pady=(0, 10))

# Buttons
save_button = ttk.Button(frame, text="Save Contacts and Messages", command=save_to_database)
save_button.grid(row=4, column=0, pady=(20, 10))

start_button = ttk.Button(frame, text="Start WhatsApp Automation", command=start_whatsapp_automation)
start_button.grid(row=4, column=1, pady=(20, 10))

clear_button = ttk.Button(frame, text="Clear Database", command=clear_database)
clear_button.grid(row=5, column=0, pady=(10, 10))

show_button = ttk.Button(frame, text="Show Database Info", command=show_database_info)
show_button.grid(row=5, column=1, pady=(10, 10))

# Log Area
log_area = scrolledtext.ScrolledText(root, height=10, width=100)
log_area.pack(pady=20)

root.mainloop()

```

8. TESTING

i) Unit Testing

Unit testing in the WhatsApp Automation system focuses on individual components to ensure each function works correctly. Key areas tested include:

- **Message Sending Functions:** Tests ensure that the system correctly sends various message types, such as text, images, videos, and documents, to both saved and unsaved numbers.
- **Contact Management:** Tests for adding, editing, and deleting contacts, ensuring the system accurately handles contact data.
- **User Authentication:** Verifies the security and functionality of the login and registration processes, ensuring secure access to the system.

Unit testing isolates these components to identify and fix bugs early in the development process.

ii) Integration Testing

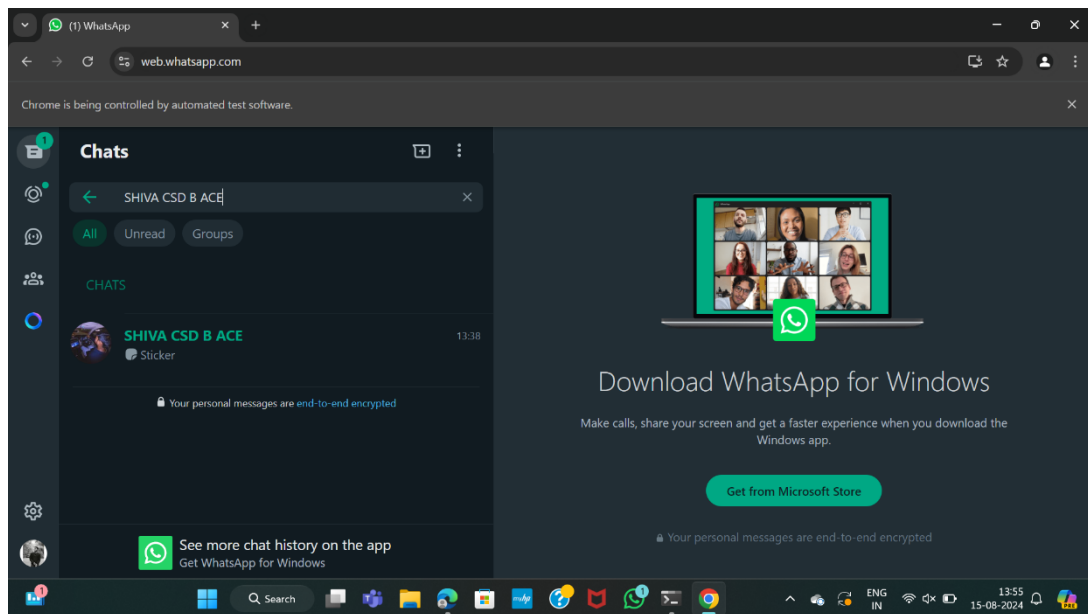
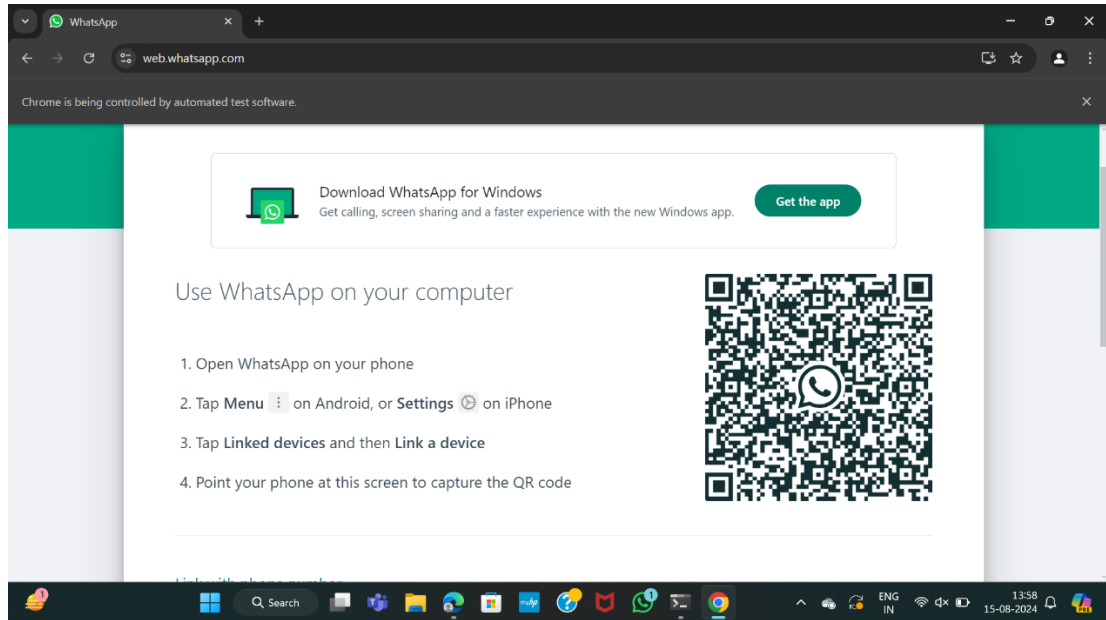
Integration testing checks the interactions between different modules to ensure they work together seamlessly. This includes:

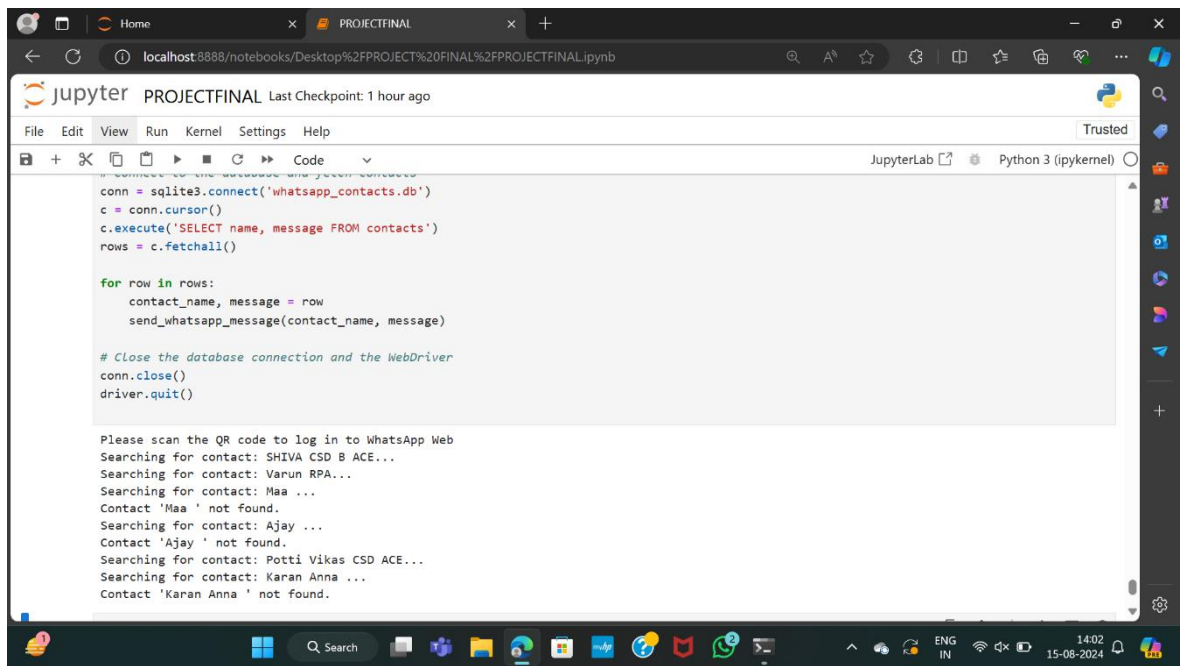
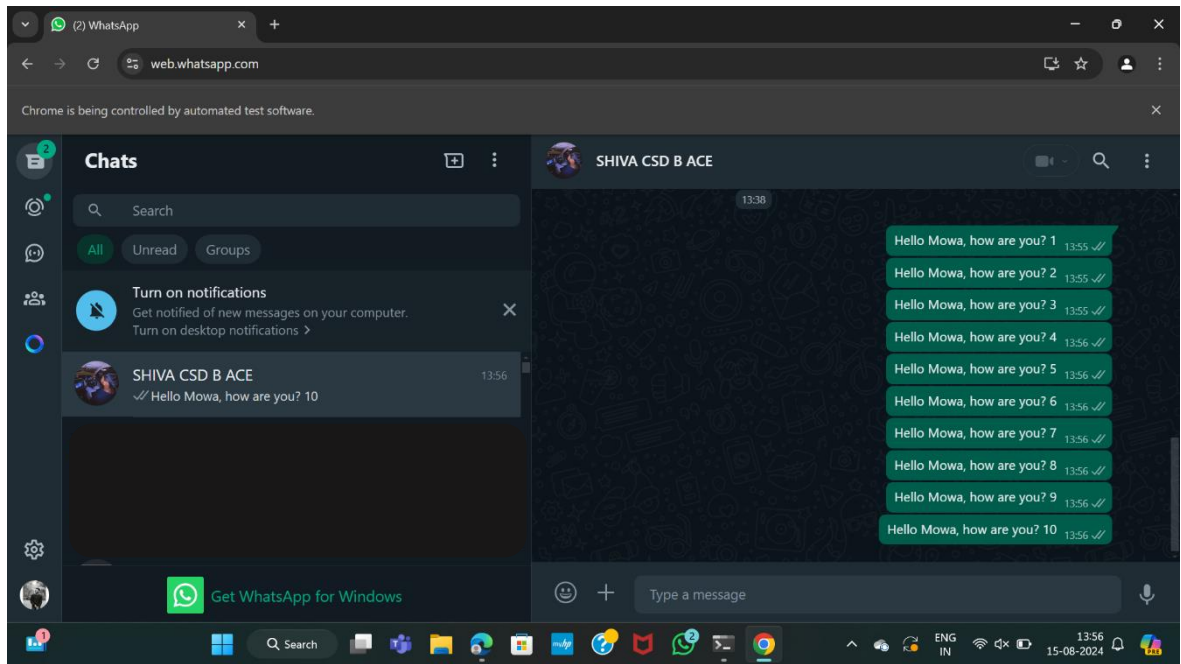
- **Client-Server Communication:** Tests to verify that data, such as messages and contact information, is correctly transmitted between the frontend and backend.
- **External Libraries:** Ensures proper integration with third-party libraries like Selenium for web automation and Pywhatkit for scheduling messages to unsaved numbers.
- **Data Flow:** Validates that data, including message logs and user preferences, moves smoothly and accurately between different system components.

Integration testing ensures that the system's components work together as expected, providing a seamless user experience.

9. RESULTS

OUTPUT OF THE CODE





10. CONCLUSION

The WhatsApp Automation project successfully addresses the need for efficient and automated communication solutions, particularly for businesses and individuals managing large volumes of messages. Through the use of technologies like Selenium and Pywhatkit, the system enables automated sending of text messages, multimedia content, and scheduled messages to both saved and unsaved contacts. This automation significantly reduces the manual effort involved in communication tasks, allowing users to focus on more strategic activities. In addition to functional achievements, the project demonstrates a strong commitment to user experience. Comprehensive testing, including unit, integration, system, and user acceptance testing, has been integral to delivering a robust and reliable system.

In conclusion, the WhatsApp Automation project not only meets the current communication needs of its users but also provides a foundation for future enhancements. Potential future work includes expanding the system's capabilities to support other messaging platforms, integrating advanced AI-driven features for more personalized communication, and optimizing performance for even greater efficiency. The project's success thus far lays the groundwork for continued innovation and development, offering a versatile tool for modern communication challenges.

FUTURE WORK :

Future work could involve expanding the system's capabilities, such as:

- **Integration with Other Platforms:** Extending support to other messaging services.
- **Enhanced AI Features:** Implementing AI-driven responses for more personalized interactions.
- **Performance Optimization:** Further improving scalability and responsiveness.

These enhancements would broaden the system's utility and improve user experience.

REFERENCES :

1. <https://chatgpt.com/>
2. <https://youtu.be/Zf7Wlijvc-0?si=gNYacwPFNmdDL7TZ>
3. <https://youtu.be/KqWUC-xWYpA?si=MUybtQnzaagRUcqW>
4. https://github.com/Python-World/python_miniprojects/tree/master/projects/whatsapp_Bot