IARJSET



International Advanced Research Journal in Science, Engineering and Technology Impact Factor 8.066 ∺ Peer-reviewed & Refereed journal ∺ Vol. 12, Issue 4, April 2025 DOI: 10.17148/IARJSET.2025.12433

Automation for Customer Support Onboarding in RPA

Ms. P. Nava Bhanu¹, E. Sujitha Grace², K. Y. V. Kalyani³, G. Pravallika⁴, Ch. Kavya Sri⁵

MTech, (Ph.D.) Computer science & Engineering, Bapatla Women's Engineering College, Bapatla, AP, INDIA¹

BTech, Computer science & Engineering, Bapatla Women's Engineering College, Bapatla, AP, INDIA²⁻⁴

Abstract: Customer support onboarding is a critical phase that directly impacts client satisfaction and operational efficiency. Traditional onboarding methods often involve repetitive tasks and manual data handling, leading to delays, errors, and inconsistent service quality. This paper presents an intelligent automation framework designed to streamline the customer support onboarding process using Robotic Process Automation (RPA) techniques, developed with Python and deployed through a Streamlit based web interface. The architecture, system components, and automation logic are thoroughly discussed, emphasizing modularity, scalability, and user-centric design. Experimental evaluation confirms the effectiveness of the solution in reducing onboarding time and operational workload, demonstrating its suitability for broader enterprise integration and automation of customer service workflows.

Keywords: Robotic Process Automation, Customer Support, Onboarding, Streamlit, Python, Process Automation, Web Application.

I.INTRODUCTION

Robotic Process Automation (RPA) has emerged as a transformative solution for streamlining repetitive business operations across various domains. By mimicking human actions in interacting with digital systems, RPA enables the automation of rule-based tasks with higher speed and accuracy.

This paper proposes an RPA-based framework for automating customer support onboarding using Python as the core programming language and Streamlit for developing an interactive web application. The system introduces a dual-dashboard model—one for end users (customers) and another for administrators—allowing seamless registration, secure login, support ticket generation, and real-time tracking of service history.

The integration of RPA logic into a web-based platform offers several advantages, including reduced operational costs, minimal manual intervention, enhanced data accuracy, and faster response times. Moreover, by leveraging open-source technologies, the proposed solution ensures affordability and scalability, making it accessible for small to medium enterprises aiming to modernize their support infrastructure

II.LITERATURE SURVEY

The adoption of automation in customer service domains has gained significant traction in recent years, largely driven by advancements in Robotic Process Automation (RPA).

According to Singh et al., automating onboarding workflows not only accelerates the customer journey but also ensures consistent information processing, which is crucial for maintaining service standards. Recent works have integrated RPA with other technologies such as Natural Language Processing (NLP), machine learning, and web applications to build smarter customer support systems Several frameworks have been proposed to automate backend processes using Python-based automation tools and open-source RPA libraries like pyautogui, uiautomation, and TagUI. Yet, these implementations often lack an accessible user interface, limiting their adoption among non-technical users.

Its simplicity and rapid prototyping capabilities make it a suitable candidate for integrating with automation logic. Previous works, such as those by Kumar and Rao, have demonstrated how Streamlit can be effectively used to develop dashboards for internal enterprise applications.

IARJSET



International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 $\,\,st\,$ Peer-reviewed & Refereed journal $\,\,st\,$ Vol. 12, Issue 4, April 2025

DOI: 10.17148/IARJSET.2025.12433

III.SYSTEM ARCHITECTURE

A. The system comprises two main modules:

1. Customer Dashboard – for registration, login, ticket creation, and viewing service history.

2. Admin Dashboard – for managing customer queries, viewing tickets, responding, and tracking onboarding status.

Both modules interact with a backend automation layer and a database that handles task execution, data storage, and user management.

B. System Components Table

| Description |
|--|
| Lightweight Python framework for building interactive web UIs. |
| Handles backend automation for ticket creation, classification, and logging. |
| Validates user identity during registration using email. |
| Sends confirmations, and admin responses to customers. |
| Interface for staff to monitor, reply, and manage support tickets. |
| |

C. Technologies Used

| Technology | Purpose |
|---------------|---|
| Python | Core logic, automation scripts, and data handling |
| Streamlit | Frontend interface for both dashboards |
| smtplib/email | email notifications |
| Pandas | Tabular data display in dashboards |

RPA Libraries Task automation using tools like pyautogui, pywinauto or UiPath API (optional)

D. Functional Modules Description

1. Registration & Verification

- Users sign up using their email address.
- On verification, user details are stored in the dashboard and excel sheet.

2. Customer Dashboard

- Post-login, users can create tickets.
- A form captures issue title, description, and category.
- Users can view previous tickets and their status.

3. RPA Ticket Handling

- Tickets are auto-tagged and stored using Python scripts.
- RPA may interact with external systems (e.g., CRM) to update logs.

4. Admin Dashboard

- Displays ticket summaries.
- Admins can respond, update ticket status, and track customer interactions.

5. Notification Module

• Updates sent to customers for ticket creation, response, and closure.



International Advanced Research Journal in Science, Engineering and Technology Impact Factor 8.066 送 Peer-reviewed & Refereed journal ∺ Vol. 12, Issue 4, April 2025

IARJSET

DOI: 10.17148/IARJSET.2025.12433

E. Flowchart For Customer Support



IV.IMPLEMENTATION DETAILS

Customer support onboarding system, focusing on the core technologies, logic flow, and functional modules developed using Python and Streamlit.

A. Development Environment

| Tool/Library | Purpose | | | | | | | |
|-----------------|--|--|--|--|--|--|--|--|
| Python 3.10+ | Core programming language | | | | | | | |
| Streamlit | UI framework for dashboard development | | | | | | | |
| smtplib / email | Sending admin notifications | | | | | | | |
| time, datetime | Timestamping tickets and actions | | | | | | | |

B. Customer Dashboard Functionality

Developed using Streamlit, the customer-facing dashboard offers:

- Registration and Login
- Ticket Submission: Users can submit issues with a title, description, and optional category.
- Ticket History: Past tickets and their statuses are displayed using structured JSON parsing.

C. Backend Automation using RPA

The RPA module is implemented using native Python scripts that simulate traditional RPA behavior. A background script periodically checks the tickets.json file for new or updated entries:

- New tickets are tagged for review.
- The system may optionally apply basic text classification or prioritization.
- Logs are appended to a file or shown on the admin dashboard.

This setup replicates a simplified RPA workflow without external automation tools like UiPath.

D. Admin Dashboard Features

Admins access their dashboard via Streamlit, which reads from the same ticket storage file. Key features include:

- Viewing all open and resolved tickets.
- Filtering based on status or timestamp.



International Advanced Research Journal in Science, Engineering and Technology

IARJSET

Impact Factor 8.066 💥 Peer-reviewed & Refereed journal 😤 Vol. 12, Issue 4, April 2025

DOI: 10.17148/IARJSET.2025.12433

- Responding to tickets directly through a text input.
- Updating ticket status as "Responded" or "Closed".

E. Notifications and Status Updates

Upon admin response:

- An automated email is sent to the corresponding user using smtplib.
- The ticket's admin_response and status fields are updated.

This ensures users are kept informed without the need to manually check the dashboard frequently.

2 🖬 🔿 main - Streamlit 🛛 🗙 + - 0 × C 🕕 localhost:850 . Login O Custom Email Password Login New User? Register Here • P Hot Q Search 🧠 🖬 🦕 🐶 😆 👩 🧔 🖷 🖄 2 🖸 🔿 main - Streamlit × + - 0 × ब क्रे उ 🎓 -- 🧑 C () localhost8501 = Support Tickets Create New Ticket **Your Tickets** Ticket TKT-20250413220715 - Data Processing Services (Closed) Created: 2025-04-13 22:07:15 Description: what is the use for Data Processing Services and give me support Response: ok , we will send api docs. Responded by: admin at 2025-04-13 22:18:15 Logout 93°F Partly sunny 🧠 🖬 🐂 🧔 😆 💼 😒 🖷 🖳 🗣 🚳 ^ ENG ⊕ \$\$ \$\$ \$121 AM 4/14/2025 Q Search 2 🖸 🔿 main - Streamlit × 🔿 main - Streamlit × + - 0 × C (localhost850 🔂 G 🏚 -- 🧔 = **Admin Dashboard** All Tickets Ticket TKT-20250413220715 - eda all.com - Data Processing Services (Cli Ticket TKT-20250413223915 - komm **Customer Management** name email 0 edasujithagrace edasujithagrace@gmail.com 9,177,029,731 2000-04-02 Ims service 2025 1 K.Yamini kommineniyvkalyani17@gmail.com 9,652,021,831 2003-05-17 Hello Logout 93*F Q Search 🧆 🖬 📮 🧟 😸 📴 🦉 🏺 🕮 💱 🚇 ^ ENG ⊕ Φ) 128 AM 4/14/2025

V.EXPERIMENTAL RESULTS



International Advanced Research Journal in Science, Engineering and Technology Impact Factor 8.066 ∺ Peer-reviewed & Refereed journal ∺ Vol. 12, Issue 4, April 2025 DOI: 10.17148/IARJSET.2025.12433

IARJSET

| ਰ ਨਿੱ | C9 ~ | | | | | | | | us | ers - Excel | | | | | | | Sign in | œ | - 0 | × |
|--|--------------------------------|------------------------|--------------------------|----------------|-----------------------------------|-----------------------------|------------------------|----------------------|-----------------|-------------|-------------|-----------------------------|-----------------------------------|-----------------|---------------|---------|---|----------------------|-----------------|---------------------|
| File He | ome Ins | ert Page | | | | Review | | lelp 🖓 | | at you want | | | | | | | | | | |
| Paste 💞 | Calibri B I | ⊻ ~ ⊞ Font | • 11 • • <u>⊅</u> • | A A A ~ | | i v v e≣ v≣ Alignm | 환 Wrap Ti 편 Merge i | ext Bi Center i v | General \$ ~ | % * 5 | • • • | Conditional formatting ~ | Format as Table ~ Si Styles | Cell tyles ~ | Insert Delete | Format | ∑ - A Z V Sort 0 Filter Editio | k Find & Select ~ | Add-ins | _ |
| POSSIBL | E DATA LOS | SS <u>Some fea</u> | tures might | be lost if y | ou save this v | vorkbook in | the comma | -delimited (| .csv] format. | To preserve | these feat | ures, save it ir | an Excel file | format. | Don't show | r again | Save As | | | × |
| A1 \vee \times \checkmark \checkmark name \vee | | | | | | | | | | | | | | | | | | | | |
| | 8 | c | D | F | F | G | н | 1.1 | 1.1 | ĸ | 1.1 | м | N | 0 | P | 0 | R | s | т | |
| nm 2 edasujith 3 K.Yamini 4 5 6 7 7 8 9 10 11 12 13 14 15 16 17 18 | lemail edasujith kommine | password stujitha@C | 9.18E+09 | dob 4/2/200 | purpose 0 lms servi # Hello | registrati | on_date | | | | | | | | | у | N Image: Image of the state of the st | 3 | | |
| 19 20 | | | | | | | | | | | | | | | | | | | | |
| 21 | users | | | | | | | | | | | | | | | | | | | |
| Ready (4a) | ccessibility | Unavailable | | | | | | | | | | | | | | EEB | (m) (m) | | | - + 100% |
| 93*F Partly | sunny | | | | Q Se | arch | | • | - | 9 (| ۵ و | 0 | • | ļ. | | | ^ EI | NG 🜩 | 10 2 0 4 | 1:29 AM /14/2025 |

VI.CONCLUSION

The implementation demonstrates how essential onboarding workflows—such as user registration with OTP verification, ticket creation, admin response, and email-based notifications—can be streamlined through minimalistic yet powerful Python tools.

The system not only improves operational efficiency by reducing manual intervention but also enhances user experience through timely updates and structured communication. While the current version is designed without a database for simplicity, it remains modular and can be easily scaled or integrated with advanced RPA platforms or cloud-based services in the future.

VII.FUTURE SCOPE

While the current implementation achieves core automation objectives, several enhancements can be introduced to broaden the system's functionality, scalability, and intelligence.

1. Integration with Database Systems

Transitioning from file-based storage to robust database management systems (e.g., SQLite, PostgreSQL, Firebase) would allow more efficient data handling, especially for scaling to enterprise-level applications with a high volume of tickets and users.

2. Advanced RPA Tool Integration

Incorporating professional RPA platforms like UiPath, Automation Anywhere, or Blue Prism can extend automation beyond internal file handling to interact with third-party CRMs, ERPs, and email clients.

3. Natural Language Processing (NLP) Capabilities

Integrating NLP techniques for ticket classification and sentiment analysis can significantly enhance ticket triage, allowing the system to prioritize critical issues automatically.

4. Live Chat and Bot Assistance

Adding chatbot functionality for real-time user interaction and guided onboarding support could further reduce manual workloads and improve user experience.

5. **Analytics and Reporting Dashboard**

Building an analytics module for the admin panel could provide insights into ticket trends, resolution times, and user satisfaction, enabling continuous service improvement.

6. **Cloud Deployment and CI/CD Pipelines**

Deploying the application on cloud platforms (e.g., AWS, Heroku, GCP) with CI/CD integration would ensure reliability, scalability, and faster feature rollouts.

7. Multi-role Access Control and Audit Logging

Implementing secure role-based access controls, along with detailed activity logs, would improve security and compliance—particularly important for handling sensitive user information.

8. Multilingual and Accessibility Features

Expanding the interface to support multiple languages and accessibility standards (WCAG) can make the system more inclusive and globally usable.





International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 $\,\,st\,$ Peer-reviewed & Refereed journal $\,\,st\,$ Vol. 12, Issue 4, April 2025

DOI: 10.17148/IARJSET.2025.12433

REFERENCES

- [1] M. Lacity and L. Willcocks, "Robotic Process Automation: The Next Transformation Lever for Shared Services," *Journal of Information Technology Teaching Cases*, vol. 7, no. 2, pp. 67–74, 2017.
- [2] Streamlit. "Streamlit The fastest way to build data apps in Python"
- [3] Python Software Foundation, "The Python Programming Language,"
- [4] A. Agarwal, V. Kumar, and N. K. Rath, "Enhancing Customer Experience using Intelligent Automation," in *Proc. Int. Conf. on Computation, Automation and Knowledge Management (ICCAKM)*, 2021, pp. 98–103.
- [5] S. Choudhury and R. Dubey, "An Overview of Robotic Process Automation in Business Processes," *International Journal of Scientific and Research Publications*, vol. 10, no. 4, pp. 453–458, Apr. 2020.
- [6] A. Deshmukh, "A Review on Automation of Customer Support System using RPA," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 9, no. 6, pp. 5212–5216, June 2021.
- [7] M. Vukovic, "Customer Service Automation Using AI and Cloud Technologies," *IBM Journal of Research and Development*, vol. 62, no. 6, pp. 1–10, Nov.–Dec. 2018