

Faculty Assistant Bot-automation of administrative activities using robotic process automation

Mamidyala Durga Prasad, Balusu Nandini

Department of Computer Science and Engineering, Telangana University, Telangana State, India

Article Info

Article history:

Received May 5, 2024

Revised Jul 14, 2024

Accepted Aug 6, 2024

Keywords:

Artificial intelligence

Bot

Faculty assistant

Robotic process automation

UiPath

ABSTRACT

In this paper, a process workflow for Bot is created using robotic process automation (RPA), associated with artificial intelligence that is used to stream line the administrative tasks and alleviate stress levels of faculty in handling administrative tasks while teaching in higher education. These activities are must for National Academic Audit Council (NAAC) accreditation and All India Survey on Higher Education (AISHE) surveys, which strive to bring quality in teaching higher education by shaping educational policies. Therefore, ensuring the accuracy of this data is paramount to avoid misleading decisions. The Bot automatically gathers student results from the website and saves them into individual files, eliminating the need for human intervention. It is trained to find the related file of student and update his results of upcoming semesters or backlogs. The Bot efficiently manages folders during file saving to enhance retrieval. Additionally, it maintains pertinent student details such as community, caste, and religion, beneficial for educational policy surveys aiming for improved quality. Moreover, it generates and updates reports post each process execution, ensuring data integrity, and can be trained for statistical analysis to predict student outcomes. The UiPath tool is used in the design and testing of the developed Bot.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Mamidyala Durga Prasad

Department of Computer Science and Engineering, Telangana University

Dichpally, Telangana State, India

Email: durgamamidyala@gmail.com

1. INTRODUCTION

Higher education institutions significantly enhance the quality of education and contribute to a country's overall development [1], [2]. Making higher education more adaptable is really important so that it can up hold with how the economy is changing and becoming more connected to the rest of the world [3]. Educational policymakers in India collaborate with institutions like National Academic Audit Council (NAAC), University Grants Commission (UGC), and All-India Council for Technical Education (AICTE) to ensure quality assurance, accreditation, and regulation within the higher education sector [4]. These organizations set standards, conduct assessments, and provide guidelines for educational institutions. Initiatives like the All India Survey for Higher Education (AISHE) facilitate data collection on various aspects of higher education, informing policy decisions and progress monitoring [5].

The analysis and prediction of academic outcomes in higher education are incredibly important, benefiting various stakeholders such as individuals, government bodies, and educational institutions [6]. This process plays a crucial role in national development and is essential for students, educators, policymakers, administrative bodies, and the broader educational community. However, it comes with challenges, notably due to the extensive volume of unstructured data inherent in higher education systems. However, HEIs have

been severely impacted by coronavirus disease 2019 (COVID-19) [7], leading to the adoption of non-traditional alternatives like online learning to maintain educational continuity [8]. After COVID-19, many universities reintroduced face-to-face learning. However, future situations may arise where faculty must be equipped with their own material and technological knowledge for online learning. In COVID-19, many faculty members have relied on pre-existing teaching materials due to insufficient time and burden, i.e., the faculty have taught materials made by other [9]. Online platforms offer flexibility [10] but may pose challenges for faculty members [11] in adapting to new technology while also being involved in administrative tasks such as collecting student information, category, results, and alumni information for NAAC data collection and AISHE data submission.

These challenges for faculty members highlight the need for adequate support and training in managing both instructional and administrative responsibilities effectively [12]. The emergence of artificial intelligence (AI) has revolutionized the way businesses and organizations operate, replacing human involvement with machines. This transformation has been primarily facilitated by robotic process automation (RPA) [13]. Such technological advancements are poised to enhance organizational capabilities significantly. For instance, in the banking sector, the process of onboarding new customers involves numerous checks and data gathering, which can be standardized into a series of tasks executed by a robotic system across relevant platforms [14]. This technology offers solutions for automating routine office tasks and distinguishes itself by its ability to operate continuously, 7 days a week, 24 hours a day, without human involvement. The efficacy of this technology hinges on identifying repetitive processes and establishing standardized procedures during Bot development. Subsequently, these Bots can be scheduled to execute tasks autonomously, minimizing the need for employee intervention [13].

The proposed Bot is developed using UiPath [15], UiPath is RPA tool [16] used for large-scale end-to-end automation. This bot is equipped with features that streamline the management of students administrative details, making it a stress free task. NAAC related activities involve extracting results and performing administrative tasks. In an unattended mode, the Bot operates on the organization's system directly from the orchestrator, enabling automated processes without human oversight. The Bot will automatically log into Mail inbox via secure methods and utilize filtering mechanisms to determine which emails are useful and which are not. This automated capability streamlines the workflow by effectively managing the influx of incoming messages, allowing efficient handling of administrative tasks. After declaration of results, User initiates the process by sending mail to designated email address. Thereafter, the Bot will download email attachments automatically. It uploads data extracted from files into orchestrator queues for managing flow of work items. Each item in a queue is individual student id. The Bot will get details of each queue (i.e., student id) and runs user interface (UI) automation activities. It opens the browser, enters the id of the student, clicks submit button and scrapes results data of each and every student from website automatically. The Bot is designed to create or update data within the system. Specifically, it will generate new files for newly joined individuals or update existing files of students already in the system. The Bot is also trained for folder management. Folder management is implemented to organize student data based on different criteria such as colleges, courses, and student ID's, ensuring systematic storage and easy retrieval of information. It is also trained to generate reports and update them after every process execution for data integrity. A process refers to sequence of activities designed to accomplish series of tasks. It is also trained to run statistical data for predicting outcomes of students. The Bot is also trained to generate reports useful for NAAC and AISHE surveys automatically. The Bot can also generate different types of reports and protect the files against inappropriate access. The Bot is also trained to send emails to each and every faculty and students upon updating. The Bot can be utilized to save files in the cloud using APIs, expanding its capabilities to facilitate seamless integration with cloud storage platforms like Google cloud storage (GCS). The Bot has been tested on multiple test emails and attached documents, confirming its proper functionality. This kind of testing ensures that automation performs reliably across different scenarios, ultimately leading to more robust solution.

Robotic process automation (RPA) represents a computer software technology. RPA is used for the creation, deployment, and management of software Bots. These Bots imitate human behavior when interfacing with digital systems and software. RPA is employed across different industries to automate repetitive tasks and streamline processes. Ribeiro *et al.* [17], examined the integration of RPA with AI techniques and algorithms is revolutionizing ERP-related processes, enhancing optimization, user interaction, and overall process intelligence in the era of Industry 4.0. Rafailidis and Manolopoulos [18] have concentrated on understanding user responses in various scenarios and have designed a conversational system. Bridging the technological gap between recommendation systems and virtual assistants in conversational platforms, alongside leveraging RPA, offers the potential to enhance user experience through collaborative efforts aimed at improving recommendation quality and overall efficiency. A comprehensive

mapping study on the existing status of research on RPA development and gaps in the scientific and industrial domain was published by Enriquez *et al.* [19].

RPA Bots can help teachers to deliver more effective instruction according to proposal of Khan *et al.* [20]. These Bots are trained to evaluate the quality of lessons delivered by teachers. Before selecting topics of widespread interest for teaching, the training requirements for teachers are assessed, along with potential challenges, if certain areas are taught by a robot. RPA robots have the potential to enhance a teacher's role as a personalized tutor or instructor. Furthermore, the implementation of RPA can be advantageous to various disciplines, like human resources, technical operations, finance, management, and administration. According to Gajra *et al.* [21], RPA is being used to integrate frequently occurring tasks within an ERP system and also used to automate many of the processes for managing a Student Management System. A web-based platform is designed to provide students, faculty and group with access to records and databases that works with both mobile and desktop devices. This platform enables faculty and administrators to circulate notifications, assign and post assignments, share schedules of class, announce the examination dates, and manage various other tasks for both individual students and groups. Additionally, the solution incorporates a ChatBot module that allows users to retrieve essential information through a user-friendly graphical interface. The technologies advancement made the business process into an automated systems using the software robots which can run with the help of predefined algorithms using artificial intelligence. Lacity and Willcocks [22] has proposed intelligent automation principles to guide leaders for their intelligent automation success. Intelligent automation systems play a crucial role in broader digital transformation initiatives, as organizations increasingly seek to automate process not only within their own boundaries but also across different departments and even with external partners.

Wewerka and Reichert [23] have provided an overview of relevant publications on RPA until June 2020, identifying seven thematic clusters addressing fundamental questions about RPA. It explores differences between RPA and related technologies like robotic desktop automation (RDA) and suggest methods for improving RPA implementation. Additionally, the paper reviews case studies and introduces a framework called ANCOPUR for analyzing and comparing RPA publications. While the research focus primarily centers on understanding RPA technology, trends suggest a growing interest in integrating AI with RPA and developing implementation methods, particularly in recent years. Additionally, the application of RPA in Bots offers numerous advantages for automating commercial and organizational processes related to the development of artificial intelligence tools, algorithms, and procedures that enable increased precision and accuracy.

Khare *et al.* [24] has introduced a Bot to manage emails. The Bot logs into mailbox securely, sorts between the not useful and useful emails, categorize the useful emails into labels, downloads the file attachments, creates directories, and downloaded files are kept in appropriate directories. The useless emails are moved into the trash. In addition, the Bot can also be trained to conveniently rename the associated files with the sender/applicant's name in case of a job application.

2. METHOD

Figure 1 presents the system design of the proposed Faculty Assistant Bot, describing the sequence in which the Bot performs different activities. The Bot is developed using the UiPath tool on Windows operating system. UiPath represents an innovative RPA platform created for extensive automation on a large scale, providing corporate solutions to simplify repetitive office tasks for rapid business transformation. Through this tool, users can convert heavy and boring tasks into automated processes using a variety of utilities [15]. The main components of this tool are i) UiPath robot, ii) UiPath orchestrator, iii) UiPath assistant, iv) UiPath API integrations, v) UiPath AI Fabric, vi) RPA algorithm, and vii) segregated mail router.

- a. UiPath robot: Executes automation workflows remotely on target machines.
- b. UiPath orchestrator: The orchestrator responsibility is to trigger a process upon receiving new mail based on certain conditions and schedule automation by assigning tasks to robots and monitoring their execution. UiPath Orchestrator is available in both on-premises and cloud-based versions. The cloud version of UiPath Orchestrator, known as UiPath automation cloud, provides the same functionality as the on-premises version but is hosted and managed by UiPath in the cloud.
- c. UiPath assistant: The role of UiPath assistant in running an unattended mode robot involves assisting users in managing and monitoring automation tasks on their desktop. While unattended mode robots typically operate without direct human intervention, UiPath assistant provides a user-friendly interface to trigger, manage, and monitor the execution of unattended automation processes. Users can initiate automation tasks, view task status and logs, and take necessary actions as needed, all from the UiPath assistant interface. Additionally, UiPath assistant may also provide notifications and alerts to users regarding the status of automation tasks, ensuring seamless oversight and control of unattended mode robot operations.

- d. UiPath API integrations: UiPath provides activities and integrations for interacting with email servers and clients directly.
- e. UiPath AI Fabric: Supports the deployment and management of AI models within automation workflows.
- f. RPA algorithm: It includes an email service protocol, UI automation, integrations, asset management, queue management.
- g. Segregated mail router: This module points the autoreply email addresses in right directions. This system segregates all incoming mails and download attachments into a folder and use this downloaded file to automatically run a process. This can also be used to generate actual data and predict outcome of the student and send mails to respective mails of students/faculty.

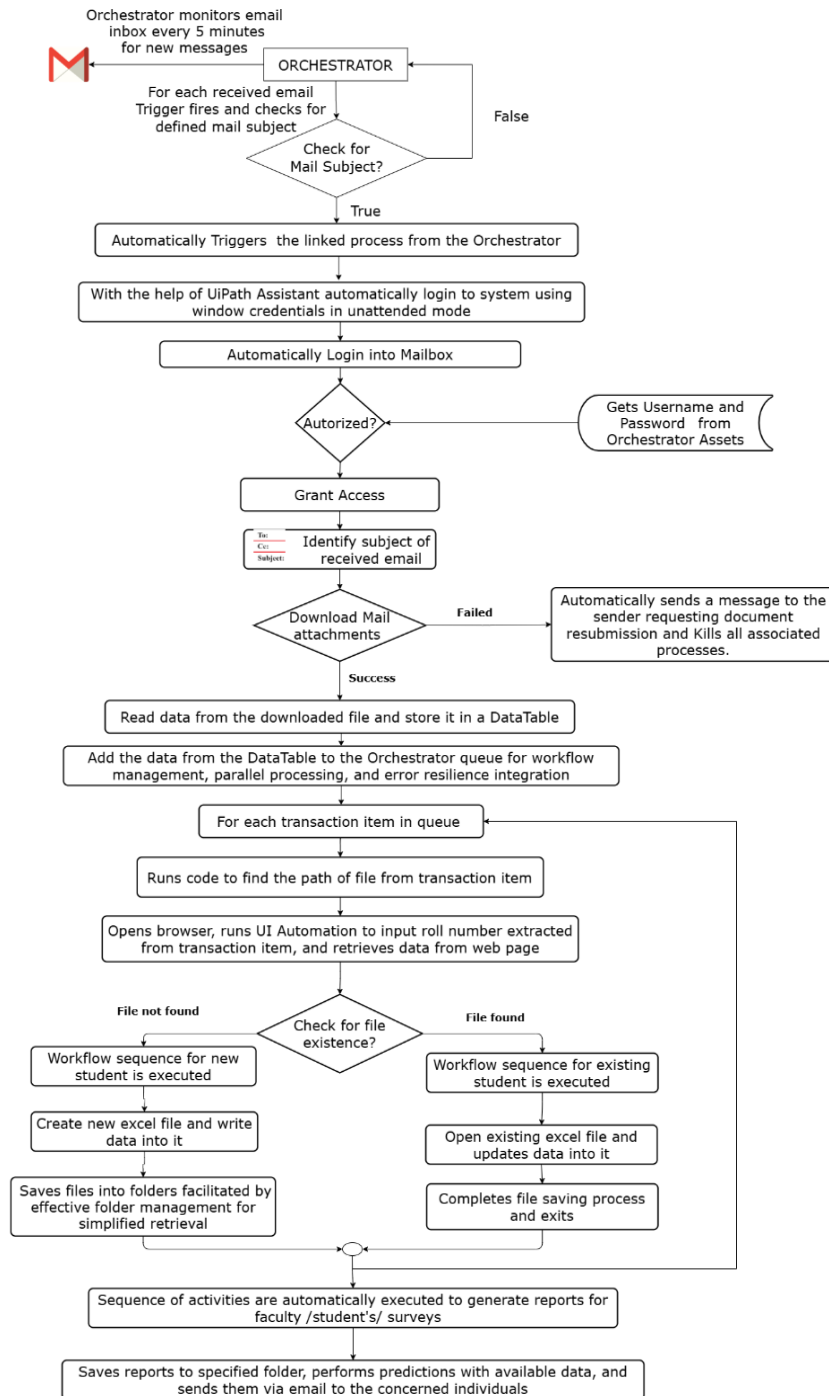


Figure 1. System design of proposed Faculty Assistant Bot

Once the desired subject mail appears orchestrator triggers a process. In UiPath, a process defines the automated workflow, executed by a robot within a designated machine environment. The main part of the Bot is to login to the machine of the user in unattended mode using Window credentials [25], but it is not a good practice to provide assets directly.

Robotic enterprise framework (REFramework) is implemented in UiPath where sensitive information like passwords, including those for Gmail accounts, should not be directly given in the workflow. Instead, it is recommended to use secure methods for managing credentials, such as orchestrator assets or configuration files. Bot will get credentials directly from assets or configuration files, ensuring security. Then Bot will automatically login to the email by using API integration activities. There are different email activities which are available in UiPath like simple mail transfer protocol (SMTP), internet message access protocol (IMAP), post office protocol version 3 (POP3), outlook, and exchange. IMAP protocol is used in the developed Bot. In the workflow, initially to connect and download files the “*GetIMAPMailMessages*” mail message activity is configured with the port number, server name, mail folder name, email-id, and password. Here, the mail folder name is “Inbox”. The Bot is configured with a Gmail account, so port number 993 is used. The server’s name of a Gmail account should be “imap.gmail.com”. The output argument of this “*GetIMAPMailMessages*” activity is a *List < MailMessage >*. This output can be stored in a variable, “*out_MailMessage*”. The Save attachments activity is utilized to download attachments, given folder Path. This activity serves the purpose of obtaining the desired attachments. The Bot has been trained to go through each downloaded file one after the other and identifies the useful attachments among them (i.e., for ex: HT-NO as Column). If any attachment of downloaded files is not in a particular format or not downloadable it replies to sender with auto mail to resend document. According to the needs of user, the definition of useful attachments can be changed. In this paper, the following definition is used – If any document contains the following keywords: “Results-UG-college code,” it is classified to be useful.

Downloaded files contain information regarding student ID’s and announced results uniform resource locator (URL) links. The Bot reads information from downloaded file into *DataTable*. *DataTable* is a data type of UiPath which can hold information in rows and columns, just like an excel sheet. Each row of *DataTable* is a student record. The data row is uploaded to queue using *AddQueueItem* activity or *BulkAddQueueItems* activity. Then it gets each transaction item from queue using *gettransaction item* activity. For each item the Bot opens the browser of results link, and performs UI Automation activities.

Every basic activity needed to create automation projects are included in the UI Automation activity package. These activities are used by the robots to perform human interactions, like performing mouse and keyboard commands or typing and extracting text from web pages using basic UI automation activities. In order to properly identify and interact with UI elements with dynamic and complicated application interfaces, selectors and anchors are used in UI automation. Figure 2, shows how UiPath identifies target application and performs UI automation using targets and anchors.

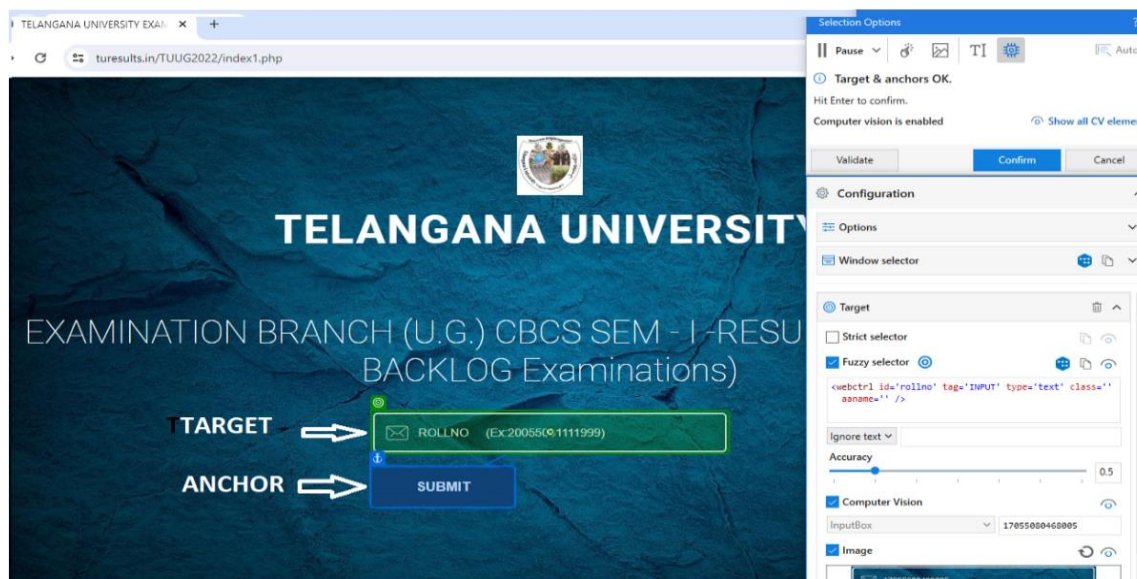


Figure 2. Indicating targets and anchor in identifying input elements of target application using unified targeting method

Target is the specific UI element being interacted with during automation. Selectors are expressions used to accurately identify UI elements within an application. Anchors are stable reference points used to locate dynamic or relative UI elements, enhancing automation reliability. The Bot enters student details in web page and then scraps the data of result page after clicking submit button. The data scraping is done through extract table data activity as shown in Figure 3.

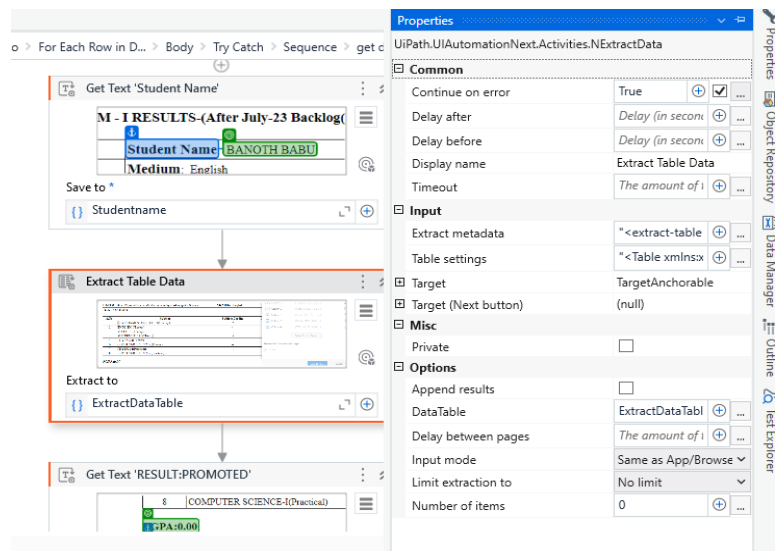


Figure 3. Extract table data activity with its property window

The extract table data activity output variable is a *DataTable*. The Bot scrapes the results into *DataTable* and will run excel automation activities and automatically updates details of particular students in desired excel file. The Bot will create new excel file for newly joined students or update existing excel file for already enrolled students. The Bot searches to determine if the file is present using activities like "File Exists". The excel files are stored in folders based on year of Admission, college code, course code. For Ex: A case study for a student of Telangana University where his roll number is 20055080445005 (where 2020 is Enrolled Year, 5080 is college code, 44 is course code), the UiPath uses string matching function to extract related string from roll number for better folder management. UiPath provides comprehensive excel management activities, allowing you to efficiently work with excel files and integrate excel-related tasks into your automation projects seamlessly. These functions are valuable for data processing, reporting, analysis, and other excel-dependent operations within various business processes. UiPath also uses exception handling to catch any errors or for any subsequent actions while processing. We also have retries mechanism in UiPath for any activity failure because sometimes due to slow network issues the activities take time to find the targets. Upon successful completion of the transaction UiPath updates queue item status in orchestrator to Success, it is important because this information shows whether an item is processed or not. Algorithm 1 outlines the steps to develop proposed Faculty Assistant Bot as discussed earlier.

Algorithm 1. Algorithm of the proposed Faculty Assistant Bot

Step 1: Configure the Bot for logging into windows by entering the username and passwords from orchestrator assets.

Step 2: Configure the Get IMAP mail message activity with all properties.

Step 3: Give permission to the Bot from the google account by "Allowing access of less secure apps"

Step 4: Configure the Bot to take email-ID and password from orchestrator assets or config file in *REFramework*.

Step 5: $Out_MailMessage := List < MailMessage >$

Step 6: For each mail in *out_MailMessage* repeat

if *is_important* (mail)

open the mail

Download attachments file.

Step 7: Open the downloaded file which contains unique ID's of student into *DataTable*.

Step 8: For each Student ID

perform UI automation activities and extract data from website.

Perform Folder management methods while creating excel file for new student.

Perform Excel management activities for inserting or updating particular records.

Step 9: After completing step8, Generate and update reports which are useful for NAAC, UGC, AISHE are updated periodically.

Step 10: Use UiPath AI Fabric provides capabilities for deploying and managing machine learning models, enabling users to integrate predictions from trained models directly into their automation processes.

Step 11: Send mail to all Faculties and students of their achievements and predictions. (optional)

End

3. RESULTS AND DISCUSSION

The UiPath version-2023.8.0 (Community edition) was installed on the Windows 10 operating system. The installation was carried out on a machine with 64-bit Intel (R) Core (TM) i5-8250U CPU @ 1.60 GHz and 8 GB RAM. The primary goal of this testing is to demonstrate that the Bot is working properly with all the designated functionalities. This verification aims to confirm that every designated feature of the Bot is working as expected.

Faculty were requested to send emails with attachments under the subject “Results-UG, college code”, while some were instructed to use a different subject as a means of testing correctness of our Bot. As soon as the orchestrator finds the particular subject it triggers the Bot to execute. The Bot logs into the email. It begins downloading attachments of the mails on the subject defined above. It ignores all other email subjects. Remember that Bot starts executing at the local machine of the user itself. The Bot runs UI automation, scrapes the data from web page and saves the gathered information in separate excel file, one per student. The files are stored in folders on the local machine based on year of admission, college code and course code. As shown in Figure 4, the Bot is scheduled to execute by performing REFramework.

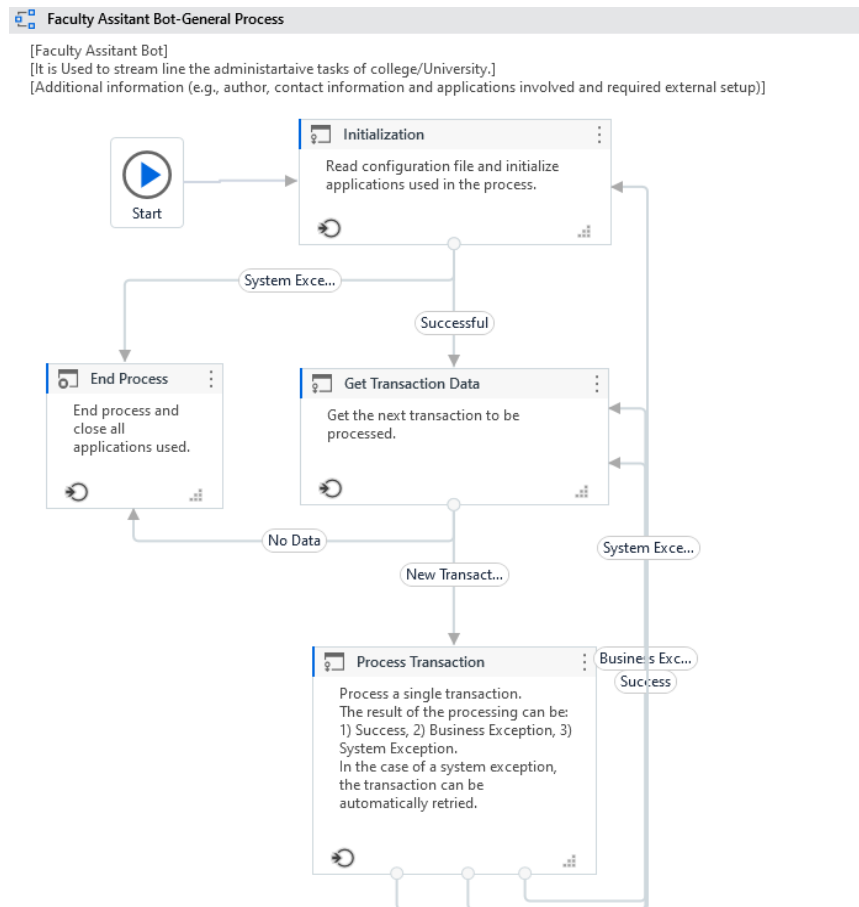


Figure 4. Faculty Assistant Bot general process using REFramework

The inbox page for the test email address gets email address from many contacts and of different colleges. Remember, the emails are not sorted automatically as useful or not. It is based on instructions given to the Bot to customize the useful ones. It downloads the attachments and then runs data scraping on website and stores the results into excel file one per student.

3.1. Performance evaluation

The effectiveness of the faculty assistant automation robot is evaluated through several quality-of-service parameters. i) Efficiency: the ratio of precious work that the Bot completes; ii) Time consumption: the Bot processing time and throughput; iii) Accuracy: a state of the Bot to be precise; iv) Precision: the Precision indicates how precise (accurate) the model of the Bot is. To be able to evaluate the effectiveness of the system, a Gmail account is used for reading incoming emails, and retrieve results of student from website using two cases.

- a. Case I – Performing without the Bot (i.e., manually): As a case study asked Faculty of Telangana University, India to manually collect data from mail and enter ID of student one by one in website and write results into Excel. They have taken 2 to 5 minutes per student.
- b. Case II – Performing with the Bot (i.e., automation): In Case II, the Bot does the process without human involvement. Once user sends file to intended email, Bot automatically downloads the attachments and performs UI automation, scrapes all the data and saves data into sheets accordingly. In comparison to the manual process, the automated approach operates significantly more efficiently, requiring minimal user intervention for task completion.

3.1.1. Efficiency

With reference to Case I, there are challenges inherent in human labor, such as errors, inconsistency, and emotional influences, which emphasize the limitations of relying solely on human effort. These difficulties lead to variations and unpredictability in outcomes. They also highlight the constraints of depending exclusively on human effort, suggesting that additional measures are necessary to improve efficiency. Conversely, in Case II, the efficiency demonstrated by Bots underscore the significant advantages of automation.

3.1.2. Time consumption

With reference to Case I, in the absence of automation observed that, manual work is more, leading to increased time consumption. Comparatively, Case II requires significantly less time when contrasted with Case I, which lacks the utilization of RPA. There are certain factors like the size of the attached files, number of student IDs, typing and retrieving content from website play a key role which determine how much time will be consumed in completing the task in both cases. In Figure 5, it is clear that Bot spent 2 minutes (120 seconds) to scrape and save the results of 11 students from a website into an Excel file, while also managing folders and generating all reports. In contrast, human took 2-5 minutes for each student, resulting in a total of nearly 55 minutes for all 11 students approximately. The figure also indicates that the Bot ability to work on different folders using Excel file.

3.1.3. Accuracy

The accuracy level is determined by how many results of student ID's are appropriately scraped from website in the designated excel file for each Student. In Case I, when the process is done by a human being, errors can occur during tasks such as reading file content and typing information into a website and saving data into an exact file because of their misunderstandings. In case II, when performed by Bot, the accuracy of reading content, typing, scraping, and saving data into an excel file while working on the RPA is significantly more effective. This makes the accuracy of the process with the reference to case I is less when compared to case II.

3.1.4. Precision

The Bot consistently performs tasks with minimal errors or deviations from the desired outcome. It suggests that the Bot algorithm, programming, or decision-making processes are well optimized. This fine-tuning makes them effective in achieving the intended objectives.

3.2. Computational complexity

In the context of UiPath RPA, computational complexity generally refers to the efficiency of the automation processes executed by the Bots. It encompasses factors like the resources and time required for the Bot to complete a task, the scalability of the automation solution, and how efficiently it handles bigger datasets. However, it is important information to note that traditional computational analysis methods like

Big O notation cannot be directly applied to evaluate the computational complexity of UiPath RPA Bots. In Figure 5, it is clear that the Bot has spent 2 minutes (120 seconds) to scrape and save the results of 11 students from a website into a file, while managing folders and generating all reports. The Bot effectively performs following the requested functionality after being implemented and tested on multiple test emails. It consistently operates within the 120–140-second range thereby demonstrating Bot effective performance.

```
03/28/2024 07:51:30 => [Info] FacultyAssistantBot execution started
03/28/2024 07:52:15 => [Info] Processing New Queue item
03/28/2024 07:52:22 => [Info] Audit: Using Excel File: D:\20\5081\44\20055081445004.xlsx
03/28/2024 07:52:28 => [Info] Process Completed
03/28/2024 07:52:29 => [Info] Processing New Queue item
03/28/2024 07:52:36 => [Info] Audit: Using Excel File: D:\20\5081\44\20055081445005.xlsx
03/28/2024 07:52:36 => [Info] Process Completed
03/28/2024 07:52:37 => [Info] Processing New Queue item
03/28/2024 07:52:42 => [Info] Audit: Using Excel File: D:\20\5081\44\20055081445007.xlsx
03/28/2024 07:52:42 => [Info] Process Completed
03/28/2024 07:52:43 => [Info] Processing New Queue item
03/28/2024 07:52:48 => [Info] Audit: Using Excel File: D:\20\5081\44\20055081445008.xlsx
03/28/2024 07:52:48 => [Info] Process Completed
03/28/2024 07:52:49 => [Info] Processing New Queue item
03/28/2024 07:52:55 => [Info] Audit: Using Excel File: D:\20\5080\12\20055080129001.xlsx
03/28/2024 07:52:55 => [Info] Process Completed
03/28/2024 07:52:56 => [Info] Processing New Queue item
03/28/2024 07:53:00 => [Info] Audit: Using Excel File: D:\20\5080\12\20055080129003.xlsx
03/28/2024 07:53:01 => [Info] Process Completed
03/28/2024 07:53:01 => [Info] Processing New Queue item
03/28/2024 07:53:06 => [Info] Audit: Using Excel File: D:\20\5080\12\20055080129004.xlsx
03/28/2024 07:53:07 => [Info] Process Completed
03/28/2024 07:53:07 => [Info] Processing New Queue item
03/28/2024 07:53:12 => [Info] Audit: Using Excel File: D:\20\5080\44\20055080445001.xlsx
03/28/2024 07:53:12 => [Info] Process Completed
03/28/2024 07:53:13 => [Info] Processing New Queue item
03/28/2024 07:53:18 => [Info] Audit: Using Excel File: D:\20\5080\44\20055080445003.xlsx
03/28/2024 07:53:18 => [Info] Process Completed
03/28/2024 07:53:18 => [Info] Processing New Queue item
03/28/2024 07:53:23 => [Info] Audit: Using Excel File: D:\20\5080\44\20055080445004.xlsx
03/28/2024 07:53:23 => [Info] Process Completed
03/28/2024 07:53:24 => [Info] Processing New Queue item
03/28/2024 07:53:29 => [Info] Audit: Using Excel File: D:\20\5080\44\20055080445006.xlsx
03/28/2024 07:53:29 => [Info] Process Completed
03/28/2024 07:53:31 => [Info] FacultyAssistantBot execution ended in: 00:02:00
```

Figure 5. Exported log of Faculty Assistant Bot

4. CONCLUSION

This paper deals with the automation of administrative activities of higher education. The automation gives relief for faculty in maintaining student records and also helps them in identify weak learners by automating all activities from collection of results to analyzing and generating reports, which are helpful for NAAC and AISHE. The robot provides a great level of comfort to faculty with well-defined functionalities. The testing and execution of the workflow have been done on many test emails. It is found to be working as per the specified functionalities.

Indeed, utilizing machine learning approaches can significantly enhance the intelligence and robustness of this robot. By training the Bot with image recognition algorithms, it can be equipped to process and analyze images effectively, expanding its capabilities beyond just working with files. This advancement opens up various possibilities for the Bot to interact with and extract information from visual data, enabling it to perform tasks such as identifying objects, recognizing patterns, and even understanding handwritten text or symbols. With continuous learning and improvement, the Bot can become increasingly proficient in handling a wider range of tasks and adapting to diverse user needs in the future. Optimizing RPA Bots, particularly in addressing the challenge of frequent user interface changes, and leveraging machine learning techniques to extend the capabilities of platforms like UiPath represent key future directions in enhancing RPA systems.




REFERENCES

- [1] R. Barnett, *Improving higher education: total quality care*. SRHE & OU, 1993.
- [2] S. Sharma and P. Sharma, "Indian higher education system: challenges and suggestions," *Electronic Journal for Inclusive Education*, vol. 3, no. 4, pp. 1–4, 2015.
- [3] P. Agarwal, "Higher education in India: the need for change," *Indian Council for Research on International Economic Relations (ICRIER)*. New Delhi, 2006.
- [4] K. Ravikumar, S. Samanta, and A. K. Rath, "Impact of NAAC accreditation on quality improvement of higher education institutions in India: a case study in the state of Karnataka," *Purushartha*, vol. 14, no. 1, pp. 34–49, 2021.
- [5] V. S. Oberoi, "All India survey on higher education AISHE (2015-16)," Government of India, Ministry of HRD, Department of Higher Education, 2017.




- [6] E. Alyahyan and D. Düşteğör, "Predicting academic success in higher education: literature review and best practices," *International Journal of Educational Technology in Higher Education*, vol. 17, no. 1, 2020, doi: 10.1186/s41239-020-0177-7.
- [7] A. Aristovnik, D. Keržič, D. Ravšelj, N. Tomaževič, and L. Umek, "Impacts of the COVID-19 pandemic on life of higher education students: a global perspective," *Sustainability (Switzerland)*, vol. 12, no. 20, pp. 1–34, 2020, doi: 10.3390/su12208438.
- [8] M. Ghosh, J. Jansz, and A. Ghosh, "Effect of COVID-19 pandemic on traditional teaching," *International Journal on Studies in Education*, vol. 4, no. 2, pp. 107–129, 2022, doi: 10.46328/ijonse.63.
- [9] R. L. Garza Mitchell, W. DeCamp, B. S. Horvitz, M. G. Kowalske, and C. Singleton, "'I'm not teaching them per se': designing and delivering asynchronous undergraduate online STEM courses," *Innovative Higher Education*, vol. 49, no. 1, pp. 91–111, 2024, doi: 10.1007/s10755-023-09670-9.
- [10] V. Singh and A. Thurman, "How many ways can we define online learning? a systematic literature review of definitions of online learning (1988-2018)," *American Journal of Distance Education*, vol. 33, no. 4, pp. 289–306, 2019, doi: 10.1080/08923647.2019.1663082.
- [11] C. H. Karjo, W. Andreani, A. Herawati, Y. Ying, A. P. Yasyfin, and K. Marie, "Teachers' challenges and needs in e-learning environment," in *Proceedings - 2021 International Seminar on Application for Technology of Information and Communication: IT Opportunities and Creativities for Digital Innovation and Communication within Global Pandemic, iSemantic 2021*, 2021, pp. 405–409, doi: 10.1109/iSemantic52711.2021.9573242.
- [12] Jenna Gillett-Swan, "The challenges of online learning: supporting and engaging the isolated learner," *Journal of Learning Design*, vol. 10, no. 1, pp. 20–30, 2017.
- [13] N. Jha, D. Prashar, and A. Nagpal, "Combining artificial intelligence with robotic process automation-an intelligent automation approach," *Studies in Computational Intelligence*, vol. 945, pp. 245–264, 2021, doi: 10.1007/978-3-030-65661-4_12.
- [14] M. Ahmet Unal and O. Bolukbas, "The acquirements of digitalization with RPA (robotic process automation) technology in the Vakif Participation Bank," in *ACM International Conference Proceeding Series*, 2021, pp. 68–73, doi: 10.1145/3459955.3460602.
- [15] UiPath, "Robotic process automation," *UiPath*, <https://www.uipath.com> (accessed Mar. 22, 2023).
- [16] A. M. Tripathi, *Learning robotic process automation: Create Software robots and automate business processes with the leading RPA tool – UiPath*. Packt Publishing Ltd, 2018.
- [17] J. Ribeiro, R. Lima, T. Eckhardt, and S. Paiva, "Robotic process automation and artificial intelligence in Industry 4.0—a literature review," *Procedia Computer Science*, vol. 181, pp. 51–58, 2021, doi: 10.1016/j.procs.2021.01.104.
- [18] D. Rafailidis and Y. Manolopoulos, "The technological gap between virtual assistants and recommendation systems," *arXiv preprint arXiv:1901.00431*, 2018.
- [19] J. G. Enriquez, A. Jimenez-Ramirez, F. J. Dominguez-Mayo, and J. A. Garcia-Garcia, "Robotic process automation: a scientific and industrial systematic mapping study," *IEEE Access*, vol. 8, pp. 39113–39129, 2020, doi: 10.1109/ACCESS.2020.2974934.
- [20] S. Khan, R. K. Tailor, R. Pareek, R. Gujrati, and H. Uyyun, "Application of robotic Process Automation in education sector," *Journal of Information and Optimization Sciences*, vol. 43, no. 7, pp. 1815–1834, 2022, doi: 10.1080/02522667.2022.2128534.
- [21] V. Gajra, K. Lakdawala, R. Bhanushali, and S. Patil, "Automating student management system using ChatBot and RPA technology," *SSRN Electronic Journal*, 2020, doi: 10.2139/ssrn.3565321.
- [22] M. Lacity and L. Willcocks, "Becoming strategic with intelligent automation," *MIS Quarterly Executive*, vol. 20, no. 2, pp. 169–182, 2021, doi: 10.17705/2msqe.00047.
- [23] J. Wewerka and M. Reichert, "Robotic process automation-a systematic literature review and assessment framework," *arXiv preprint arXiv:2012.11951*, 2020.
- [24] A. Khare, S. Singh, R. Mishra, S. Prakash, and P. Dixit, "E-mail assistant – automation of e-mail handling and management using robotic process automation," in *2022 International Conference on Decision Aid Sciences and Applications (DASA)*, Mar. 2022, pp. 511–516, doi: 10.1109/DASA54658.2022.9765017.
- [25] D. Rountree, "System security," in *Security for microsoft windows system administrators*, USA: Elsevier, 2011.

BIOGRAPHIES OF AUTHORS



Mamidyala Durga Prasad    is Pursuing Ph.D. in computer science and engineering from Department of Computer Science and Engineering, Telangana University, Telangana State. He completed his B.Tech. in computer science and information technology and M.Tech. in computer science and engineering from Jawaharlal Nehru Technological University, Hyderabad. His research work mainly focuses on machine learning and artificial intelligence. He can be contacted at email: durgamamidyala@gmail.com.



Balusu Nandini    has completed her Ph.D. from Jawaharlal Nehru Technological University, Hyderabad. Her current research focuses on artificial intelligence, block chain technologies, machine learning, deep learning and computer networks. Currently, she is working as associate professor in the Department of Computer Science and Engineering, Telangana University, Telangana State. She has Published papers in SCI, Scopus, UGC Care listed national, international journals and conferences. She can be contacted at email: cnuvnk@gmail.com.