Introduction to
Robotic Process Automation

TED TALK VIDEO NOTIFICATION BOT

SAIKRISHNA H
220701238

Dr. DURAIMURUGAN
Associate professor - Computer Science & Engineering

HARDWORK DISCIPLINE REC
RAJALAKSHMI
ENGINEERING COLLEGE

Abstract

The **TED Talk Video Notification Bot** is an RPA solution developed using UiPath to automate the tracking and notification of newly uploaded TED Talk videos on YouTube. The bot periodically scrapes video details, including titles, descriptions, and URLs, compares them with previously stored data to identify new uploads, and sends personalized email notifications to a predefined list of subscribers. By leveraging web scraping, data organization, and email automation, the bot eliminates manual monitoring, ensures timely updates, and enhances user experience. With robust error-handling mechanisms, the bot is reliable and scalable, offering a practical solution for automating content tracking and user notifications.

Need for the Proposed System

The proposed **TED Talk Video Notification Bot** addresses the inefficiencies and challenges of manually tracking new TED Talk uploads on YouTube. For enthusiasts and educational communities, staying updated with the latest content requires regular monitoring of the TED Talk YouTube channel, which is time-consuming, prone to oversight, and unsustainable when managing multiple interests. Additionally, users may miss important updates due to irregular checks or lack of timely notifications. The proposed system automates this process, ensuring that subscribers are notified promptly whenever a new video is uploaded. By leveraging Robotic Process Automation (RPA), the bot eliminates the need for manual intervention, reduces the risk of missed updates, and provides a scalable solution for efficiently managing content tracking, thus enhancing user engagement and experience.

Advantages of the Proposed System

1. **Automation of Repetitive Tasks**: Saves time by automating the process of tracking new TED Talk uploads.
2. **Timely Notifications**: Ensures subscribers receive updates as soon as new videos are uploaded.
3. **Reduced Human Error**: Minimizes the risk of missing updates due to manual oversight.
4. **Scalability**: Can easily handle more subscribers or additional platforms.
5. **Personalization**: Notifications can be tailored to user-specific topics or speakers.
6. **Ease of Use**: Requires minimal user input for setup and operates autonomously.
7. **Improved Efficiency**: Quickly processes and compares video data to send notifications.
8. **Reliable Data Handling**: Maintains an organized record of previous uploads to avoid duplicates.
9. **Error Handling and Logging**: Manages network issues smoothly and logs processes for troubleshooting.
10. **Enhanced User Engagement**: Keeps users informed and connected with TED content efficiently.

Literature Survey

Paper 1:

Chien, Y., et al. (2015) – Automated Tracking of TV Show Episodes Using RPA
Advantages:

- Demonstrated how RPA can streamline the tracking of TV show episodes.
- Provided a reliable mechanism for collecting episode release dates from multiple sources.
- Improved user experience by automating notifications for upcoming episodes.

Disadvantages:

- Limited to TV shows, with no flexibility to adapt to other forms of content.
- Did not integrate a feedback mechanism for personalization based on user preferences.
- Lacked error-handling features for scenarios like dynamic webpage changes.

Literature Survey

Paper 2:

Zhu, J., & Zhang, R. (2020) – Personalized Content Delivery Systems Using Machine Learning and Automation

Advantages:

- Highlighted the importance of personalization in content tracking systems.
- Combined machine learning with automation to tailor notifications to user interests.
- Demonstrated scalability for large datasets and diverse content platforms.

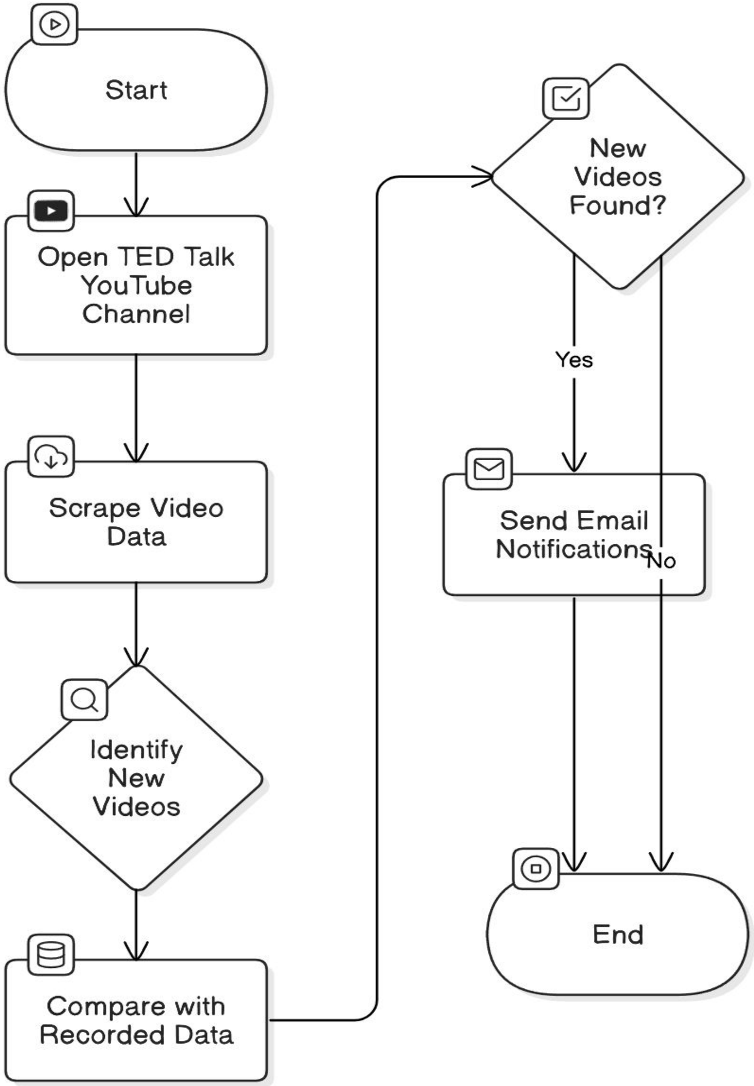
Disadvantages:

- Focused on theoretical implementation with limited real-world testing.
- Required complex configurations, making it less accessible for small-scale applications.
- Did not address real-time updates, which are critical for platforms like YouTube.

Main Objective

The main objective of the **TED Talk Video Notification Bot** is to automate the process of tracking newly uploaded TED Talk videos on YouTube and notifying a predefined set of subscribers in real time. By leveraging Robotic Process Automation (RPA) tools, the system eliminates the need for manual monitoring of the TED Talk YouTube channel, ensuring that users are promptly informed about new content. The bot achieves this by scraping video details, such as titles, descriptions, and URLs, comparing them with previously stored data to identify new uploads, and sending personalized email notifications. This automation not only saves time but also reduces the likelihood of missed updates, enhances user engagement with TED content, and provides a scalable and efficient solution for tracking digital media. The system demonstrates the potential of RPA to streamline repetitive tasks and improve the accessibility of knowledge-sharing platforms like TED.

Architecture



System Requirements

Hardware Requirements

1. **Processor**: Intel Core i3 or higher.
2. **RAM**: Minimum 4 GB (8 GB recommended).
3. **Storage**: At least 10 GB free disk space.
4. **Operating System**: Windows 7 or later (64-bit).
5. **Internet Connection**: Stable connection for web scraping and email services.

Software Requirements

1. **UiPath Studio**: For developing and executing the RPA workflow.
2. **Microsoft Excel**: To store and manage video data.
3. **Email Client**: Outlook or SMTP server for sending notifications.
4. **Web Browser**: Google Chrome or Mozilla Firefox (with UiPath extensions installed).
5. **Dependencies**: UiPath packages like WebAPI, Excel, Mail, and UIAutomation Activities.

Functional Description

Module 1: Data Scraping

Short Description:
This module extracts video details (titles, descriptions, URLs, upload dates) from the TED Talk YouTube channel using UiPath’s **Data Scraping** activity. It handles dynamic webpage elements and ensures accurate data retrieval.

DFD / Activity Diagram:

- **Input**: YouTube URL.
- **Process**: Open browser → Navigate to the "Videos" section → Scrape video details.
- **Output**: DataTable containing video metadata.

Functional Description

Module 2: Data Comparison

Short Description:
Compares newly scraped video details with existing records stored in an Excel file to identify new uploads. It ensures only unique video entries are processed.

DFD / Activity Diagram:

- **Input**: DataTable (new data), Excel file (stored data).
- **Process**: Read Excel data → Compare with new data → Filter unique entries.
- **Output**: DataTable with new video details.

Functional Description

Module 3: Email Notification

Short Description:
Sends email notifications to subscribers containing details of newly identified TED Talk videos, such as title, description, and URL.

DFD / Activity Diagram:

- **Input**: Subscriber list, DataTable with new video details.
- **Process**: Format email content → Send emails to all subscribers.
- **Output**: Confirmation of email sent.

Functional Description

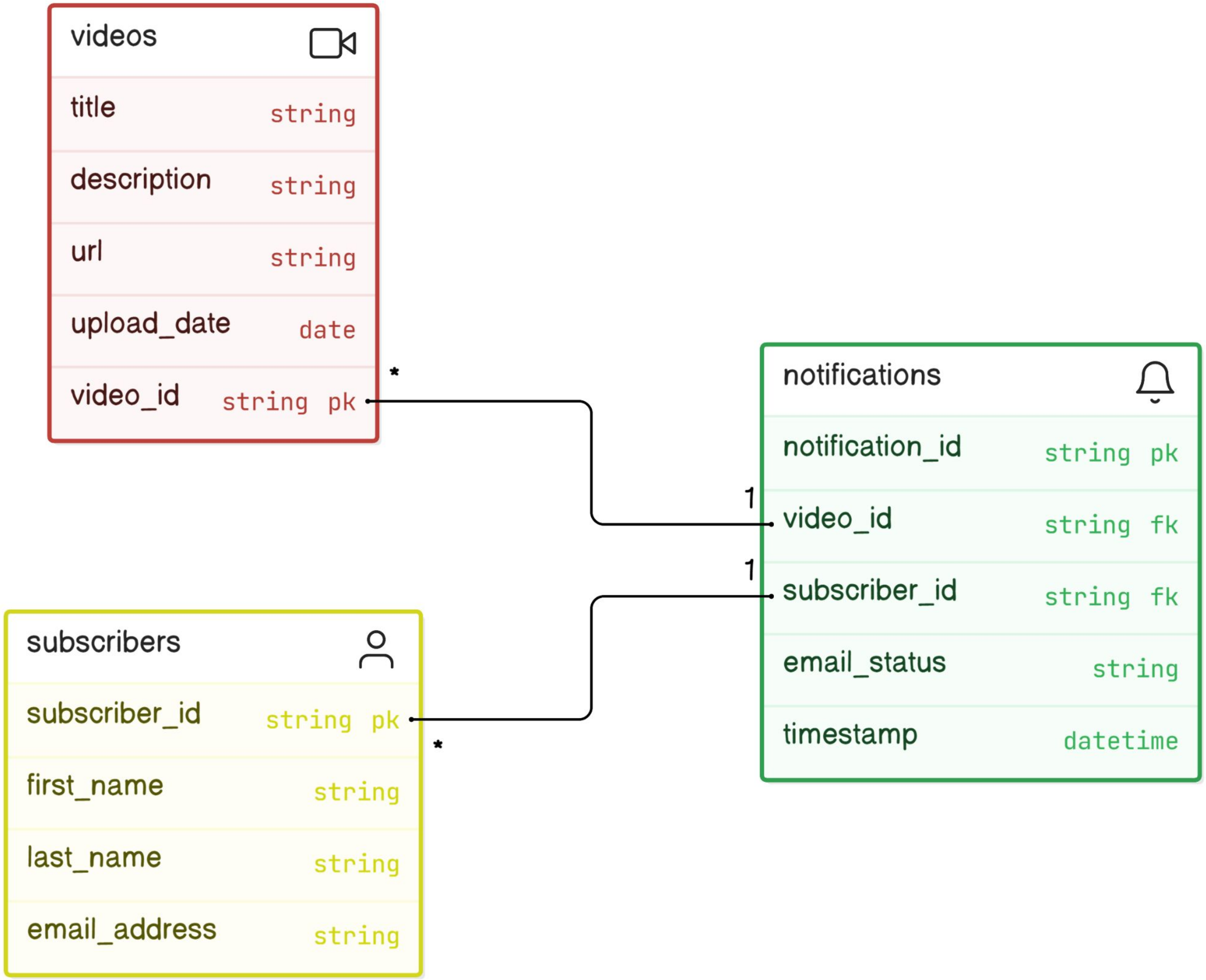
Module 4: Logging and Reporting

Short Description:
Maintains logs of each execution, including timestamps, videos processed, and email statuses. It provides a summary of the bot’s performance.

DFD / Activity Diagram:

- **Input**: Execution details, DataTable of processed videos.
- **Process**: Record logs → Save to file → Generate summary if required.
- **Output**: Log file with execution details.

ER TABLE



videos	
title	string
description	string
url	string
upload_date	date
video_id	string pk

notifications	
notification_id	string pk
video_id	string fk
subscriber_id	string fk
email_status	string
timestamp	datetime

subscribers	
subscriber_id	string pk
first_name	string
last_name	string
email_address	string

Process Design

Main Process Design

The **TED Talk Video Notification Bot** follows these steps:

1. **Initiate the Bot**: Load configurations like email credentials and file paths.
2. **Open YouTube Channel**: Navigate to the TED Talk YouTube channel using the **Open Browser** activity.
3. **Scrape Video Details**: Extract video titles, descriptions, URLs, and upload dates using **Data Scraping**.
4. **Compare Data**: Identify new videos by comparing scraped data with the existing Excel database.
5. **Store New Data**: Append new video details to the Excel file.
6. **Send Notifications**: Notify subscribers by email using the **Send Mail** activity.
7. **Log Process**: Record execution details and email statuses in a log file.
8. **End the Process**: Close the browser and terminate the bot.

Process Design

Subprocess Design

1. **Data Scraping**: Opens YouTube, scrapes video metadata, and handles dynamic web page elements.
2. **Data Comparison**: Compares scraped data with stored data to filter out duplicates.
3. **Email Notification**: Sends formatted video details to subscribers via email.
4. **Error Handling**: Logs and retries failed tasks during scraping or email sending.
5. **Logging and Reporting**: Updates log files with execution and error details for analysis.

Conclusions

In conclusion, the **TED Talk Video Notification Bot** offers an efficient, automated solution for tracking new TED Talk videos and notifying subscribers in real time. By leveraging Robotic Process Automation (RPA), the bot streamlines the process of monitoring new uploads, ensuring users stay informed without the need for manual intervention. With the potential for future enhancements such as multi-platform integration, real-time notifications, and personalized recommendations, the bot can evolve into a more versatile and user-centric tool. Ultimately, it demonstrates the power of automation in enhancing user engagement and improving content discovery.

Future Enhancement

Future enhancements for the **TED Talk Video Notification Bot** can focus on expanding the platform's reach and personalizing the user experience. One potential enhancement is integrating the bot with **multiple video platforms** beyond YouTube, such as TED's official website or other social media channels, ensuring users receive updates for new TED content no matter where it is uploaded. Additionally, incorporating **real-time notifications** through mobile push services, like Firebase Cloud Messaging (FCM), would allow users to receive instant alerts on their smartphones, making the system even more efficient and accessible.

Another enhancement could be the introduction of **machine learning algorithms** to personalize notifications further. By analyzing user preferences and viewing patterns, the bot could recommend TED Talks based on the topics, speakers, or genres that users engage with most. This would make notifications more relevant and tailored to individual users, improving overall engagement with the content and enhancing the user experience. These improvements would increase the bot's adaptability and ensure it meets the needs of a wider audience.

References

1. **Research Paper: Chien, Y., et al. (2015)**
"Automated Tracking of TV Show Episodes Using RPA," which highlights the use of robotic process automation for managing video content and notifications.

2. **Research Paper: Zhu, J., & Zhang, R. (2020)**
"Personalized Content Delivery Systems Using Machine Learning and Automation," emphasizing the importance of user-centric notification systems.

Thank You