

ElectroNeek Studio Pro for LUMA Invoice Automation and Payment Processing

*Emanuel Carrasquillo Dávila
Master of Engineering in Computer Engineering
Advisor: Jeffrey Duffany, Ph.D.
Polytechnic University of Puerto Rico
Graduate Project EXPO, May 2025*

Abstract — *The increasing complexity and volume of recurring financial transactions have driven organizations toward implementing Robotic Process Automation (RPA) to streamline their operations and minimize human error. ElectroNeek Studio Pro, a leading low-code RPA platform, enables businesses to automate such repetitive tasks with precision and resilience. This project leverages the platform to automate the end-to-end process of accessing the LUMA Energy web portal, securely logging in using credentials stored in ElectroNeek's Credential Vault, retrieving the latest monthly invoice, and executing payment through a predefined card input sequence. The system is designed with fault-tolerance in mind; built-in retry loops, decision branches, and fallback handlers ensure that the bot recovers from common interruptions—such as login failures or file download delays—without user intervention. Critical milestones, such as successful payment execution or task failure beyond threshold attempts, are accompanied by real-time email notifications that close the feedback loop. By using ElectroNeek's orchestrated workflows, credential management, and browser automation capabilities, the system delivers a robust, scalable, and auditable automation solution that aligns with best practices in digital finance transformation [1] [2].*

Keywords — *ElectroNeek Studio Pro, Error Handling, Financial Automation, LUMA, RPA.*

INTRODUCTION

Modern utilities management demands high levels of accuracy, consistency, and operational speed—particularly in handling tasks such as secure login authentication, invoice retrieval, and online payments. Manual execution of these tasks is not only inefficient but prone to delays, human errors,

and inconsistent reporting. In response to these challenges, this project proposes an end-to-end automation solution built using ElectroNeek Studio Pro, a powerful low-code Robotic Process Automation (RPA) platform recognized for its scalability and integration features [1] [3].

The proposed solution automates the entire workflow of logging into the LUMA Energy portal [4], downloading the latest monthly invoice, and processing payment using stored card information. User credentials are securely managed using ElectroNeek's built-in Credential Vault, which ensures that sensitive data is both encrypted and retrievable during bot execution without exposing it on screen [1]. To ensure fault tolerance, the bot is programmed with retry mechanisms—specifically up to four retry loops in critical stages such as login and file download—to handle unexpected interruptions like network instability or server delays.

Once the invoice is downloaded, the system utilizes ElectroNeek's PDF extraction and Date activities to retrieve the total amount due and name the invoice file dynamically with the current month. This structure not only standardizes data storage but also facilitates future auditing. Following this, the bot initiates payment by navigating to the transaction page and securely inputting the stored card details. If payment is successful, a confirmation email is sent; in the event of persistent failure, a separate notification is dispatched alerting the user to review the issue manually.

By leveraging a modular architecture composed of task-specific automation blocks (Login.neek, Invoice_Download.neek, Payment.neek), this system adheres to best practices in automation design—ensuring maintainability and scalability [2] [5]. Through the strategic use of ElectroNeek's retry logic, conditional workflows, and notification

capabilities, the automation delivers enhanced reliability, auditability, and cost-efficiency in a traditionally manual utility management process.

BACKGROUND

Historically, utility invoice management in organizations has involved manual interactions with service provider portals—such as logging into the LUMA platform [4], navigating billing interfaces, downloading invoices, and executing payments using corporate credit cards. These procedures are time-intensive, error-prone, and susceptible to operational delays, particularly when dependent on individual user performance or subject to recurring system inconsistencies.

Manual financial processes not only introduce inefficiencies but can also lead to delayed bill settlements, inconsistent file storage, and increased exposure to compliance risks. If an error occurs—such as a login failure due to expired credentials or an unsuccessful invoice download—the user must typically restart the workflow manually, adding to the administrative burden. These operational inefficiencies are incompatible with modern digital finance standards, which increasingly demand automation, traceability, and auditability [5].

Recognizing these limitations, this project applies Robotic Process Automation (RPA) using ElectroNeek Studio Pro to modernize and streamline the LUMA billing workflow. By integrating automation elements such as credential vaults, structured retry logic, and email-based escalation channels, the solution addresses each failure point with robust recovery strategies. ElectroNeek Studio Pro's capacity to modularize automation tasks through low-code logic blocks makes it especially suitable for repetitive, rule-based finance tasks [1] [3].

Through this design, the automation replaces manual billing routines with an intelligent, repeatable sequence of interactions that not only improves task completion rates but also lays the groundwork for scalability and enterprise-wide adoption.

PROBLEM

The manual process of paying LUMA invoices encompasses multiple stages, each vulnerable to errors and inefficiencies. These include failed authentication due to incorrect or outdated credentials, broken invoice download attempts caused by web interface lags or browser incompatibilities, and unreliable payment submissions that lack transaction feedback. For example, login failures due to session expiration or security timeouts can halt the billing cycle and delay compliance with payment deadlines. Moreover, invoice files—if downloaded manually—are frequently misnamed or saved inconsistently, which disrupts audit trails and increases data retrieval burdens for finance teams.

Another critical gap in this process is the absence of automated feedback and traceability. Without systematic logging and real-time alerts, users must manually verify each step—particularly payment status—leaving the organization exposed to unnoticed failures or delays. This dependency on user oversight contradicts modern RPA standards that advocate full-cycle automation, traceable logs, and exception handling for resilient operation [5]. In addressing these operational shortcomings, this project leverages ElectroNeek Studio Pro to create a fault-tolerant, intelligent automation system. The solution features:

- Credentials-based login secured through ElectroNeek's Credential Vault [1]
- Login and download retry logic (up to four times)
- Automated navigation to LUMA's billing portal [4] and document retrieval
- Invoice PDF parsing to extract total amount due
- File naming based on current month via ElectroNeek's Date activity [1]
- Autonomous entry of card data and payment submission
- Email-based confirmation and exceptions alert

This automation workflow embodies the best practices outlined in industry research on RPA

efficiency and resilience [1] [5], providing both accuracy and accountability in a task historically dominated by manual effort.

EQUIPMENT AND MATERIALS

To ensure a reliable and effective automation process, specific software, hardware, and modular workflow components are required. This section outlines all necessary tools and infrastructure used to build and operate the ElectroNeek-based solution for automating LUMA invoice processing and payment.

Software Tools and Applications

- **ElectroNeek Studio Pro:** A low-code RPA development environment used to build and manage workflow automation. It was utilized to create all core financial automation scripts for this project.
- **ElectroNeek Orchestrator:** A cloud-based orchestration platform responsible for scheduling workflows, managing credentials securely, and monitoring bot execution status in real time.
- **Adobe Acrobat Reader and PDF Parser Tools:** Used to access and extract data from downloaded invoices in PDF format, particularly for AAA and LUMA utility providers. Extraction was supported via OCR and pattern-based parsing.
- **Gmail SMTP Configuration:** Configured for use within ElectroNeek to enable automated dispatch of Excel-based reports and notifications to designated recipients.
- **Sungard Accounting System (Optional):** The backend system that received formatted feeder files from the automation process for ledger integration and financial reconciliation.

Hardware and Connectivity Requirements

- **Windows 10 Workstation:** Equipped with ElectroNeek Bot Runner for executing automation scripts. The machine requires moderate RAM (8GB+), at least 2GHz CPU, and SSD storage for optimal performance.

- **Secure Internet Connection:** Required for web-based automation involving ATH Movil and utility company portals. Firewall exceptions and SSL/TLS support were ensured.
- **Credential Vaults (Windows Credential Manager / ElectroNeek Vault):** Employed to securely store and retrieve user authentication data without exposing sensitive information in workflow code.
- **PDF-capable Printer or Virtual Printer (Optional):** Used during debugging phases for testing output formats and audit printouts.
- **Web Browser (Google Chrome):** Configured as the target browser for all web-based interactions due to its compatibility with ElectroNeek selectors and automation drivers.

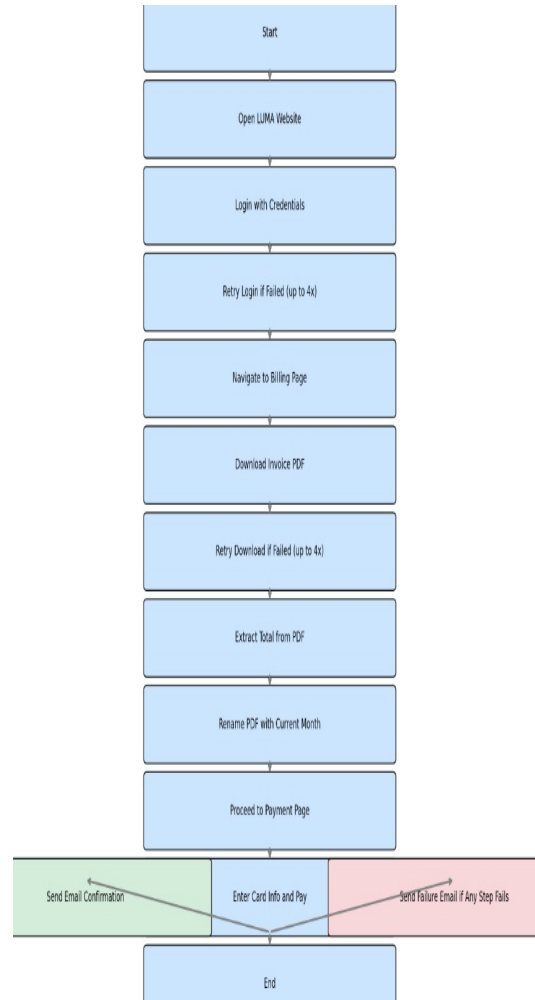


Figure 1
ElectroNeek Studio Pro Workflows Flowchart

Key Automation Components

- **Login.neek** – Automates login to the LUMA portal using stored credentials.
- **Extract_Invoice.neek** – Navigates to billing page, downloads PDF invoice, and handles retry logic.
- **Payment.neek** – Automates entry of card information and submission of payment.

METHODOLOGY

The automation process begins by launching a browser instance and navigating to the official LUMA customer portal [4]. At this point, the bot retrieves the user’s credentials securely from ElectroNeek’s Credential Vault, a component designed for encrypted credential storage and dynamic access during automation workflows [1]. The login form is filled using ElectroNeek’s form input and keystroke activities, ensuring a secure and efficient authentication process. In the event of login failure—whether due to expired credentials, network instability, or session timeouts—the bot reattempts the login process up to four times, following a structured retry mechanism as recommended in automation best practices [1] [5].



Figure 2
Studio Pro Showing All Three Workflows for the Bot

Once logged in, the bot navigates to the billing section of the portal using ElectroNeek’s browser control tools, including the Send Hotkey and Click on Element activities. The bot attempts to locate and initiate the download of the monthly invoice in PDF format. If this action fails (e.g., due to interface lag or file rendering issues), the bot activates a retry loop that returns to the billing page and reattempts the

download until it succeeds or reaches the maximum retry threshold.

Upon successful download, the PDF file is parsed using ElectroNeek’s built-in PDF Text Extractor tool. The bot identifies and extracts the total amount due and employs the Date activity to dynamically generate a filename labeled with the current billing month. This naming convention facilitates future referencing, auditing, and data retrieval.

Next, the bot transitions to the payment interface, where it auto-fills stored credit card information and confirms the transaction. Upon payment submission, an email is automatically dispatched to the user’s predefined address, indicating whether the payment was successfully processed. If any part of the process—login, download, PDF parsing, or payment—encounters persistent failure, a separate notification is issued alerting the user for manual review and escalation.

Each task—from login to notification—is encapsulated within its own workflow block (Login.neek, Invoice_Download.neek, Payment.neek), supporting streamlined maintenance, parallel testing, and seamless integration of future enhancements. The implementation follows industry-recognized RPA design patterns and fault-tolerant engineering principles to ensure operational continuity and traceability throughout the financial workflow [1] [3] [5].

RESULTS AND DISCUSSION

After executing multiple test cycles and real-time simulations, the automation system developed using ElectroNeek Studio Pro demonstrated a high degree of reliability, stability, and performance efficiency. The automated workflow consistently logged into the LUMA Energy portal, retrieved current invoices, extracted payment data, and submitted online payments using credentials and card details stored securely in ElectroNeek’s Credential Vault [1]. Invoices were downloaded and automatically renamed based on real-time date

capture using ElectroNeek’s Date activity, ensuring traceable, standardized file organization and simplifying downstream reconciliation processes [1].

Crucially, the built-in retry mechanisms helped manage routine points of failure. Whether dealing with browser delays, session timeouts, or temporary service outages, the bot leveraged conditional logic and fallback routines to reattempt critical steps such as login and file downloads up to four times before restarting the automation cycle [1][5]. This approach significantly reduced the risk of workflow abandonment and enhanced the system’s operational resilience, as emphasized in industry RPA frameworks [5].

Equally important, the system’s email-based alerting capability created a full feedback loop. The bot reliably generated notifications for both successful task completions and unresolved exceptions, enabling real-time user awareness and intervention when required. This reduced response lag in the event of disruption while improving confidence in the system’s autonomous execution. The project’s modular design allowed independent testing and rapid enhancement of task-specific workflows—such as Login.neek or Payment.neek—supporting maintainability and future scalability [3].

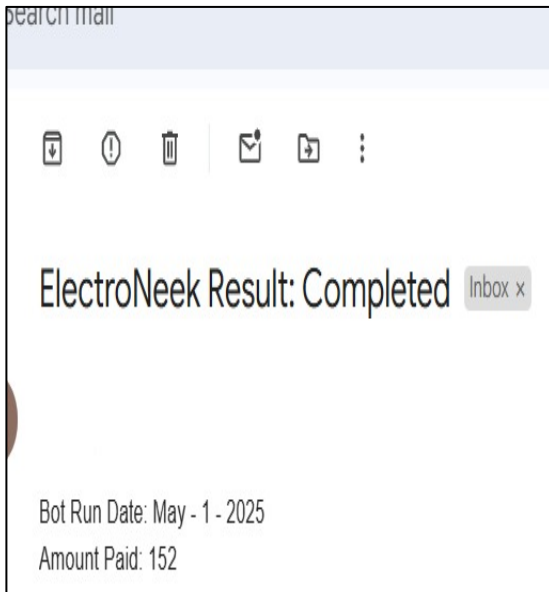


Figure 3
Results with the Payment Confirmed

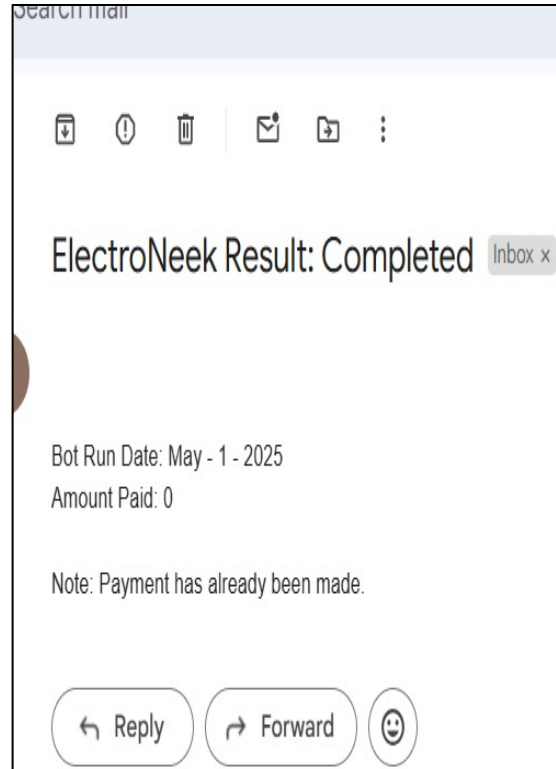


Figure 4
Results with No Payment to be Made, but Task Completed

CONCLUSION

This project successfully demonstrates how ElectroNeek Studio Pro can be used to fully automate the LUMA Energy invoice management process, from secure login to payment confirmation. By integrating credential vaults, intelligent retry logic, structured file naming, and email alerts, the system achieved significant improvements in consistency, accuracy, and operational efficiency.

The use of a modular architecture not only reduced development time and testing complexity but also made the system highly adaptable to future changes in the LUMA platform or billing structure. Most importantly, the automation proved capable of recovering from common error scenarios, making it a practical, scalable, and professional-grade RPA implementation.

FUTURE WORK

Future enhancements to the system will explore multiple dimensions of functionality and scalability,

with a focus on deepening integration, improving usability, and extending automation coverage across additional finance operations.

- **Integration with Microsoft Power BI for Dashboard Analytics:** By connecting automation logs and Excel outputs to Power BI, users will gain access to real-time dashboards for visualizing financial KPIs, payment summaries, and audit trails. This will enhance decision-making capabilities at the management level.
- **Notification Integration with Messaging Platforms:** Implementation of WhatsApp or Microsoft Teams notifications will provide real-time status updates, alerts for failed automations, or confirmation of critical tasks such as invoice parsing or Sungard uploads. This ensures proactive monitoring and faster response by stakeholders.
- **Advanced NLP-based Parsing of Scanned Documents:** Expanding the system's ability to process handwritten or scanned PDFs through Natural Language Processing (NLP) and AI-based OCR will allow automation of older or less structured invoice formats. This enhancement will reduce reliance on manual document classification.
- **Real-time Exception Reporting to IT Staff:** Automated exception handling will be expanded to include direct push notifications or logging to centralized IT monitoring systems. This will allow the IT team to respond instantly to errors, broken selectors, or data inconsistencies.

REFERENCE

- [1] ElectroNeek. (2024). *Studio Pro Documentation* [Online]. Available: <https://docs.electroneek.com/docs/electroneek-docs>. [Accessed: April 11, 2025].
- [2] ElectroNeek. (2024). *Community Forum: Tips on error Handling and Credential Management* [Online]. Available: <https://forum.electroneek.com/>. [Accessed: April 11, 2025].
- [3] G2. (2024). *RPA Platform Comparison* [Online]. Available: <https://www.g2.com/categories/robotic-process-automation-rpa>. [Accessed: April 11, 2025].
- [4] LUMA Energy. (2025). *Customer Login Portal* [Online]. Available: <https://miluma.lumapr.com/login>. [Accessed: April 11, 2025].
- [5] Deloitte. (2020). *The robots are waiting: Are you ready to reap the benefits of RPA?* [Online]. Available: <https://www2.deloitte.com>. [Accessed: April 11, 2025].