Decision-Making Process | Automation Developer Task

Author: Denis Thapa

Date: 12 Aug 2025

▼ 1. Overview & Objective

The objective of this automation project was to design and implement a solution that:

1. Extracts records from a dataset in Microsoft Power BI.
2. Cleans and normalises phone numbers via a Custom Connector.
3. Sends the cleaned data to a REDCap data collection instrument via API.

The deliverables included a Power BI report, a Power Automate solution, a custom connector (Swagger JSON + C# code details in text file), a REDCap instrument export, and this documentation.

▼ 2. Key Decisions

▼ 2.1 Data Source & Preparation

- **Choice:** To use Power BI dataset query as the data source for Power Automate.
- **Reasoning:**
  - Power BI allows easy transformation and staging of relevant fields before integration.
  - DAX was used to create a custom table (qryWomenSupport_SCReport) containing only the required fields.

    - This reduced processing complexity in Power Automate and avoided sending unnecessary data.

▼ 2.2 Phone Number Cleaning Approach

- **Choice:** To build a Custom Connector pointing to a minimal ASP.NET Core API hosted on Render.
- **Reasoning:**
    - The task required using the Custom Connector in the flow.
    - A separate micro service isolates phone-cleaning logic, making it reusable in other integrations.
    - Using Render with Docker deployment provided quick, free hosting with public HTTPS.

▼ 2.3 Flow Design in Power Automate

Structure:

1. **Manually trigger a flow**: to allow manual testing without scheduling.
2. **Run a query against a dataset**: to fetch the staged table from Power BI.
3. **Parse JSON**: to ensure structured access to each row.
4. **Apply to each**: to repeat over each record.
5. **Custom Connector:** “Clean phone number” to call the Render API.
6. **Service_Type Short Logic**: to guard substring to avoid errors.
7. **Append to Array Variable**: to collect transformed records into a batch.
8. **HTTP to REDCap:** to send the batch as a JSON array.

Design Choices:

- Batched HTTP submission reduces API calls and speeds up processing.
- Guarded substring expression prevents “out of range” errors in Service_Type shortening.
- Inclusion of record_id to meet REDCap’s required field validation and avoid “record_id field is missing” errors.

▼ 2.4 Handling Errors & Data Integrity

- **Duplicate Phone Numbers:**
    - Verified they do not block import unless the phone field is set as unique in REDCap.
- **One-row Failures:**
    - Documented that a single bad row marks the flow run as “Test failed” in Power Automate, however accepted as per requirements.
- **JSON Formatting:**
    - Ensured the HTTP body sends data format to match REDCap’s array requirement.

▼ 2.5 REDCap Instrument Setup

- **Choice:** Export the instrument as provided by REDCap (ZIP with instrument.csv and originID.txt).
- **Reasoning:**
    - This format preserves all field labels, variable names, field types, branching logic, and validation rules for re-import.
    - Chosen over full project export to meet the deliverable’s scope (instrument-only).
    - Minimal alteration in use for smooth process requirement in validation rules or branching logic.

▼ 3. Tools & Technologies

- **Microsoft Power BI**: Data preparation, DAX transformations.
- **Microsoft Power Automate**: Workflow automation and API coordination.
- **Custom Connector**: Secure integration of external service into Power Automate.
- **ASP.NET Core Minimal API**: Phone cleaning micro-service (C#, Stored in GitHub, hosted on Render).
- **REDCap**: Data collection instrument and API endpoint for data import.

- **Terminal:** As an alternative code testing option.
- **Notion:** Presentable Documentation, ideas, wording, easy access.

▼ 4. Security Considerations

- API token for REDCap has been stored in Power Automate as a secure input, and not hard-coded.
- Render API endpoint is public, but logic is limited to phone normalisation, and has no sensitive operations.
- No sensitive data (names, IDs) have been committed to GitHub.

▼ 5. Testing & Validation

- Tested connector independently in Power Automate's "Test" tab, and copies of the main nearly-final Flow with other approaches.
- Verified cleaned phone output for various formats:
  - (07) 3456 7890 to 0734567890
  - +61 7 3456 7890 to 61734567890
- Confirmed successful REDCap imports with valid JSON array and record_id.
- Documented single-row failure scenario for transparency.

▼ 6. Lessons Learned

This project reinforced the importance of planning not only the technical approach but also the setup of the tools and accounts needed to deliver as task accordingly.

▼ 7. Challenges and Adaptations

One of my main challenges was simply getting all the resources ready at the start. My personal email account wouldn't let me begin the task, and my MacBook Pro couldn't run Power BI Desktop, which meant I had to rely on the web version at first. My email also wasn't approved for the Microsoft

Developer site immediately, which limited my access to certain features. Therefore, without spending too much time on what didn't work, I focused on what could work.

To work around this, I used my sister's TAFE email address for Automate access, and when I needed the full Power BI Desktop experience, I borrowed a friend's PC. While these resource hurdles slowed my start and made the process less smooth, they also gave me a chance to practise adaptability, finding alternative solutions to keep the work moving forward.

Ultimately, overcoming these challenges meant I could still deliver all five required outputs on time, with the same quality and completeness I aimed for from the beginning.

▼ 8. Proof of Work through Screenshots

Throughout the documentation, I included screenshots to visually demonstrate the key steps. From connector testing and flow configurations to REDCap imports. These screenshots are in place not only to act as proof of work but also to make the process easier to follow and verify for reviewers. I have also included a photo of myself presenting in front of the camera as an additional form of verification. This personal inclusion serves to illustrate my direct involvement, the time invested, and the commitment I maintained throughout the completion of this task, making it altogether an amazing experience.

▼ 9. Deliverables Summary

1. **Power BI PBIX ZIP**: qryWomenSupport_SCReport table creation, staging logic, and screenshots.
2. **Power Automate Solution ZIP**: Full flow and connector packaged together, and screenshots.
3. **Custom Connector Files**: Swagger JSON and C# code details in text file, and screenshots.
4. **REDCap Instrument ZIP**: Instrument definition with field settings, and screenshots.
5. **Documentation**: This file with detailed information.