**LEAN TECHNICAL DOCUMENTATION**

**L SANDI**

38924846

**Date:** *2024-07-29*

**Document Version:** *V1.0*

Table of contents

[1 Introduction 1](#_Toc171508441)

[2 Solution Design 1](#_Toc171508442)

[2.1 Detailed Solution Design 1](#_Toc171508443)

[2.2 Data Design 1](#_Toc171508444)

[2.3 Technical Assumptions 1](#_Toc171508445)

[2.4 Technical Caveats 1](#_Toc171508446)

[2.5 Wireframes 1](#_Toc171508447)

[3 Errors & Exceptions 2](#_Toc171508448)

[3.1 Business Exceptions 2](#_Toc171508449)

[3.2 Application Errors 2](#_Toc171508450)

[4 Environment Details 2](#_Toc171508451)

# Introduction

NWU Tech Trends, a consulting firm specializing in Data Integration, Generative AI (GenAI), and Automation, faces the challenge of quantifying the benefits and value realized in client environments. This is crucial as benefits, especially qualitative ones, are often subjective and inconsistently measured. The goal of this project is to develop a comprehensive solution that addresses this challenge, enabling NWU Tech Trends to accurately measure and visualize the time and cost savings their automation solutions deliver to clients.

Projects Overview

To solve this problem, the solution will be divided into multiple interconnected projects:

Web API Development: To allow automations to store telemetry data linked to projects and clients.

Web Application Development: To enable CRUD operations for managing projects and clients.

Robotic Process Automation (RPA) Testing: To ensure the web application's functionality.

Reporting Component: To visualize time and cost savings for each project and client over time.

Problems addressed

Quantifying time and cost savings: The solution measures and combines time savings with costs grouped by project and customer.

Information management and visualization: The user interface stores and displays information related to each project and client.

Iterative value delivery: The solution is delivered and presented repeatedly.

Data security: Ensuring secure processing and storage of data.

Deployment flexibility: the solution supports both cloud services and local deployments.

Complies with industry standards: The solution is designed to be reusable and scalable.

# Solution Design

## Detailed Solution Design

The proposed solution will utilize several technologies, each playing a vital role in the overall system. The interaction between these technologies is illustrated in the context flow diagram below.

Technologies and Interactions

Web API: Built using a framework like ASP.NET Core or Node.js, the API will handle data submission from automation tools, storing telemetry data related to projects and clients.

Web Application: Developed with a front-end framework such as React or Angular, paired with a back-end like ASP.NET Core or Node.js, it will facilitate CRUD operations on project and client data.

RPA Testing: Utilizing tools like UiPath or Automation Anywhere, RPA bots will conduct tests to ensure the web application's functionality.

Reporting Component: Implemented with a tool like Power BI or Tableau, it will visualize the collected data, showing time and cost savings per project and client over time

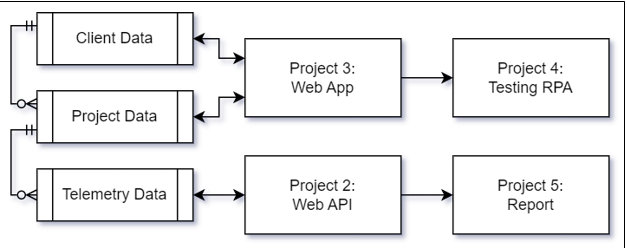


Figure 2‑1: Context Diagram

## Data Design

The context diagram is broken down into more detail to show how the different technologies will interact with one another within the developed solution. The transportation of data, across the solution, is detailed in the data flow diagram below.

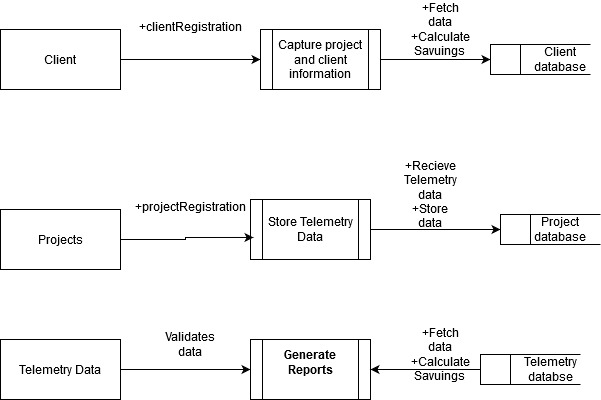


Figure 2‑2: Data Flow Diagram

**Capture Project and Client Information**:

Client Registration: Clients register their information, which is stored in the Client Database.

Project Registration: Projects are registered, linked to clients, and stored in the Project Database.

**Store Telemetry Data**:

Receive Telemetry Data: Automation tools submit telemetry data to the system.

Validate Data: The system validates the telemetry data.

Store Data: Validated data is stored in the Telemetry Database.

**Generate Reports**:

Fetch Data: The system fetches data from the Client, Project, and Telemetry Databases.

Calculate Savings: The system calculates time and cost savings based on the data.

Visualize Data: The system generates visual reports to display the calculated savings.

**Data Stores**:

Client Database: Stores client information, accessible by processes related to client registration and report generation.

Project Database: Stores project information, accessible by processes related to project registration and report generation.

Telemetry Database: Stores telemetry data, accessible by processes related to telemetry data storage and report generation.

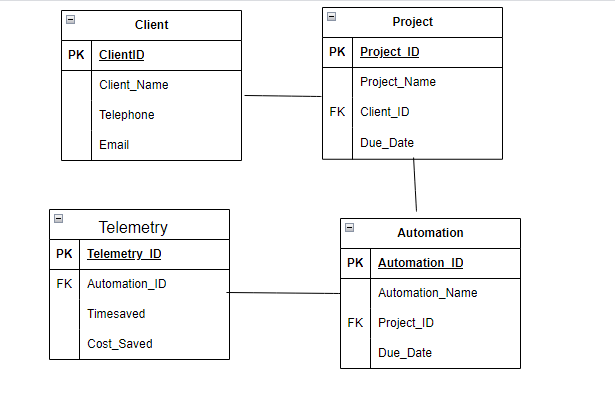


Figure 2‑3: Data Design

## Technical Assumptions

The following assumptions have been made while designing the solution:

* All license allocations will be done before development commences.
* The solution will have consistent internet access for cloud-hosted components.
* Clients will provide accurate and complete initial data sets.
* Security measures will be enforced according to industry standards.

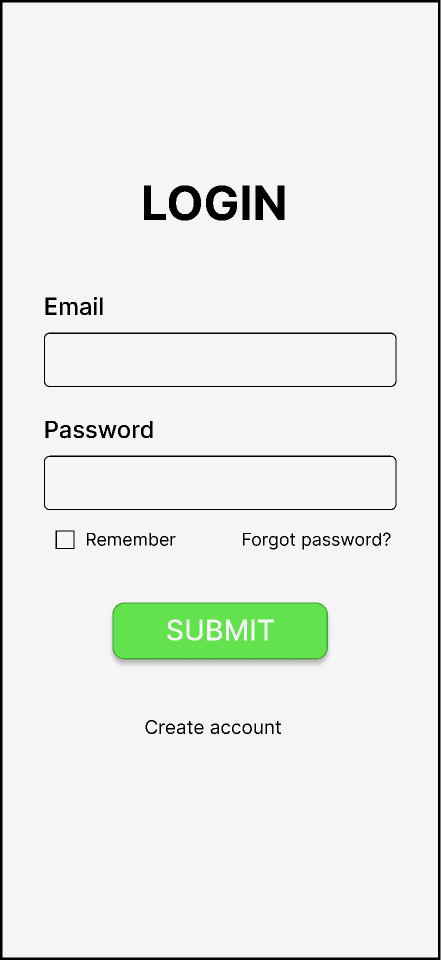
## Technical Caveats

The following caveats have been raised as part of the solution design. These caveats would need to be addressed and may have an impact on the design.

* Network latency may affect real-time data processing.
* Integration with legacy systems may require additional customization.
* Scalability might be limited by on-premise server capacities if not hosted on the cloud.

## Wireframes

All prototypes for the reports can be found below:

  
**LOGIN PAGE**

# Errors & Exceptions

## Business Exceptions

The following business exceptions should be built into the solution:

| Exception Name | Step | Parameters | Action To Be Taken |
| --- | --- | --- | --- |
| InvalidDataException | Data Submission | Invalid Data Format | Reject data and notify the user |

Table 1: Business Exceptions

## Application Errors

The following application (unknown) errors may occur as part of the solution:

| Exception Name | Step | Parameters | Action To Be Taken |
| --- | --- | --- | --- |
| DatabaseConnectionError | Data Processing | Connection Timeout | Retry connection or notify the admin |

Table 2: Business Exceptions

# Environment Details

The development of the solution would need to be executed as per the designated development strategy. The information below represents the solution and the appropriate environment(s) that will be used to implement the overall solution:

| Item | Description |
| --- | --- |
| Environment Type | Development  Testing  Production |
| Credentials Needed | Yes |
| Development Technologies Used | ASP.NET Core, React, Node.js, UiPath |
| Deployment Technologies Used | Docker, Kubernetes, Azure/AWS |
| Scalable | Yes |

Table 4‑1: Project Details

# References

OpenAI. (2024) Explanation of Data Flow Diagram for NWU Tech Trends Solution. Retrieved from ChatGPT model, OpenAI.

Liu, L., Bahrami, M., Park, J. and Chen, W.P., 2020. Web API search: discover Web API and Its endpoint with natural language queries. In *Web Services–ICWS 2020: 27th International Conference, Held as Part of the Services Conference Federation, SCF 2020, Honolulu, HI, USA, September 18–20, 2020, Proceedings 27* (pp. 96-113). Springer International Publish

Jiménez‐Ramírez, A., Chacón‐Montero, J., Wojdynsky, T. and González Enríquez, J., 2023. Automated testing in robotic process automation projects. *Journal of Software: Evolution and Process*, *35*(3), p.e2259.