**LEAN TECHNICAL DOCUMENTATION**

**[SJ NKITSENG]**

[30776538]

**Date:** *2024-07-28*

**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

**Project 1**: Agile, Scrum & Source Control

Using GitHub, I combine strong source control procedures and agile approaches like Scrum to improve flexibility and efficiency while creating solutions that measure the value and advantages being realized in the client context. With this combination, I can maintain high-quality code, respond to changes rapidly, and provide customers with effective value.

**Project 2**: API Basics & Security. Cloud Basics

**Project 3**: Architecture & Design Patterns. Development Standards & Principles

**Project 4**: Testing & Automated Testing. Robotic Process Automation

**Project 5**: Data Analytics & Visualisation. AI in Business Intelligence

# Solution Design

## Detailed Solution Design

* Web API: A web API should be created which allows automations to store telemetry data linked to projects and clients.
* Web Application: A web application should be created to perform CRUD (create, read, update and delete) functionality on projects and clients stored within the system.
* Robotic Process Automation Testing: An RPA bot should be created to conduct testing which aims to ensure that the web application functions as it is intended to.
* Reporting: A report should be created to visualise the time and cost savings accumulated 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.

A screenshot of a computer screen

Description automatically generated

Figure 2‑2: Data Flow Diagram

1. Client Data, Project Data, and Telemetry Data are the raw inputs entering the system.
2. Project 2 (Web API) processes these inputs and makes them available through API endpoints.
3. Project 3 (Web App) consumes the raw inputs to create a user-facing application.
4. Project 4 (Testing RPA) uses the Web App's data for automated testing.
5. Project 5 (Report) generates reports using the data served by the Web API.

<Introduce the data design>

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.
* Assumes that the internet connectivity for both NWU Tech Trends and their clients is stable and reliable to support seamless data integration and real-time updates.
* Assumes that the data provided by clients and generated by the system is accurate, complete, and timely. This includes project data, client data, and time saved through automation.
* Assumes that the system is designed to handle increasing loads, both in terms of data volume and number of users, without performance degradation. This is essential for accommodating future growth and additional clients.
* Assumes that the solution complies with relevant security standards and regulations, such as GDPR for data protection, to ensure that all data is processed and stored securely.
* Assumes that any third-party APIs (e.g., for payment processing, data integration) used by the system have high availability and reliability to ensure consistent service delivery.

## 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.

* The accuracy and completeness of data provided by clients can vary, which may affect the reliability of the reports and insights generated by the system.
* Network performance can vary depending on client infrastructure, potentially impacting the real-time data processing capabilities of the system.
* The system needs to handle varying loads and potentially large volumes of data as NWU Tech Trends scales its operations.
* Clients may have unique requirements and workflows that necessitate significant customization of the solution.

## Wireframes

All prototypes for the reports can be found below:

# Errors & Exceptions

## Business Exceptions

The following business exceptions should be built into the solution:

| Exception Name | Step | Parameters | Action To Be Taken |
| --- | --- | --- | --- |
|  |  |  |  |

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 |
| --- | --- | --- | --- |
|  |  |  |  |

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 |  |
| Development Technologies Used |  |
| Deployment Technologies Used |  |
| Scalable |  |

Table 4‑1: Project Details