**LEAN TECHNICAL DOCUMENTATION**

**NJ KHANYA**

34683259

**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

Over the semester, we will work on a series of projects designed to address various business problems faced by NWU Tech Trends. The organisation is struggling to quantify the benefits and value realized in client environments through its deployed solutions. The measurements of these benefits are often subjective and inconsistent. To solve this, we will develop a comprehensive solution that meets specific requirements, ensuring consistent and accurate measurement of time and cost savings, secure data handling and effective visualization of results. The projects are:

1. Agile (Scrum) and Source Control
2. API Basics and Security, Cloud Basics
3. Architecture & Design Patterns, Development Standards and Principles
4. Testing & Automated Testing, Robotics Process Automation
5. Data Analytics & Visualization

Project Breakdown:

1. Agile (Scrum) and Source control

Problem(s) to be solved:

* Inefficient project management and lack of team collaboration.
* Difficulty in tracking changes and maintaining the integrity of codebases.

Solutions to be developed:

* Implement Scrum methodologies to improve team collaboration, project planning, and iterative development. This will ensure regular cadence and iterative delivery of value.
* Utilize GitHub for source control to ensure efficient version control, collaboration, and tracking of code changes. This will maintain the integrity of the codebase and facilitate team collaboration.

1. API Basics and Security, Cloud Basics

Problem(s) to be solved:

* Lack of secure and efficient communication between software components
* Understanding the basics of cloud computing and its benefits.

Solutions to be developed:

* Design and implement secure APIs to facilitate communication between different parts of the application, ensuring data is processed and stored securely.
* Explore cloud computing fundamentals, including deployment, scalability, and cost-efficiency. Preferably, host the solution on the cloud, with an option for on-premise hosting based on client needs.

1. Architecture & Design Patterns, Development Standards and Principles

Problem(s) to be solved:

* Poor software design leading to unmaintainable and inefficient systems.
* Inconsistent coding practices and lack of adherence to development standards.

Solutions to be developed:

* Apply architectural patterns and design principles to create scalable and maintainable software systems.
* Establish and adhere to development standards and best practices to ensure code quality and consistency, making the solution reusable and scalable.

1. Testing & Automated Testing, Robotics Process Automation

Problem(s) to be solved:

* Inefficiencies and errors due to lack of proper testing.
* Repetitive manual processes that can be automated.

Solutions to be developed:

* Implement comprehensive testing strategies, including unit tests, integration tests, and automated testing frameworks, to ensure the solution is robust and reliable.
* Develop robotic process automation (RPA) solutions to automate repetitive tasks, improving efficiency and accuracy. This will help quantify the time saved by automations.

1. Data Analytics & Visualization

Problem(s) to be solved:

* Difficulty in interpreting and making decisions based on raw data.
* Lack of insight into business metrics and performance indicators.

Solutions to be developed:

* Apply data analytics techniques to extract meaningful insights from data.
* Create visualizations to represent data effectively, aiding in better decision-making and communication of insights. Develop a reporting component that visualizes the benefits realized (time and cost saved) per project and client over time.

# Solution Design

## Detailed Solution Design

To develop the comprehensive solution for NWU Tech Trends, we will use a variety of technologies and frameworks that interact seamlessly to meet the requirements. The solution includes components for API development, secure data handling, cloud deployment, front-end interface, data analytics, and visualization. Below is a detailed explanation of the technologies to be used and their interactions, followed by a context flow diagram.

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.

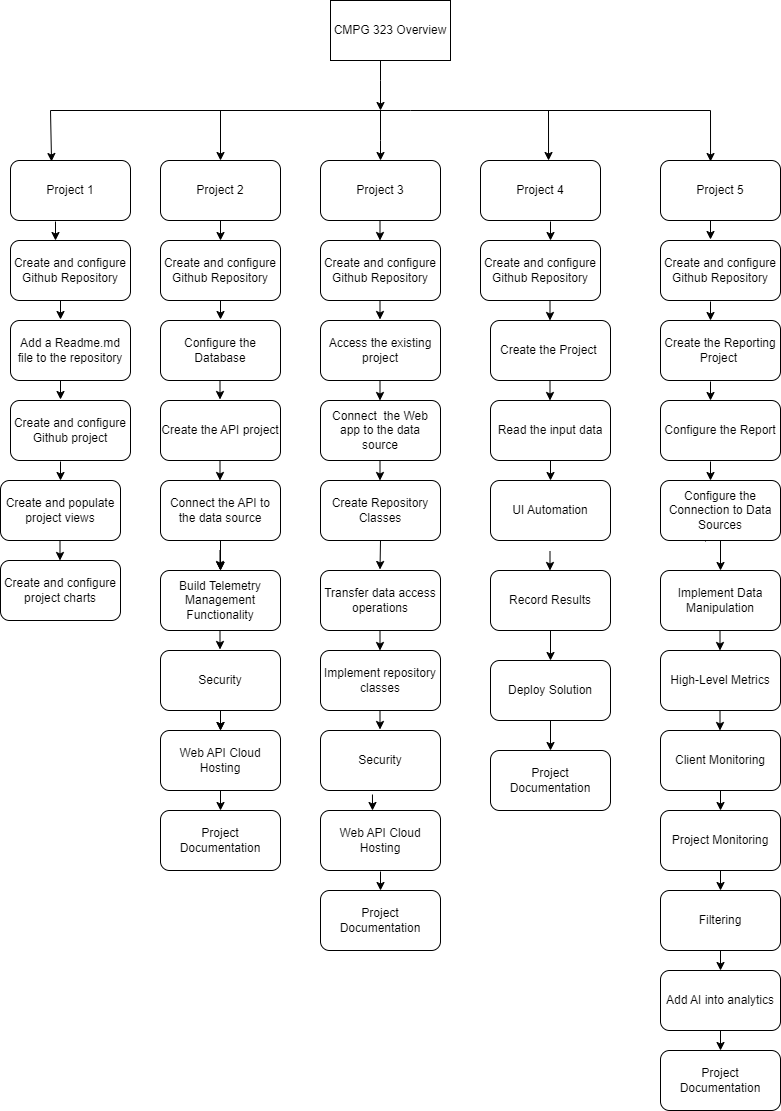


Figure 2‑2: Data Flow Diagram

<Data flow diagram explanation>

<Introduce the data design>

Figure 2‑3: Data Design

## Technical Assumptions

The following assumptions have been made while designing the solution:

* All necessary APIs and third-party services will be accessible and available for integration.
* Users will have basic knowledge of using web applications and reports.
* Sufficient infrastructure (servers, storage, networking) will be available for both cloud and on-premise hosting options.
* The solution will support standard web browsers (Chrome, Firefox, Edge).
* The client’s existing data will be clean, structured, and readily available for integration.

## Technical Caveats

* Internet connectivity is required for accessing cloud-hosted components and external APIs.
* The performance of the system might be impacted if the volume of telemetry data is very high.
* Potential for data privacy issues if data encryption is not properly managed.
* Cloud hosting costs need to be monitored to avoid budget overruns.

## 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 |
| --- | --- | --- | --- |
| Invalid Client Data | Create/Update | Client details | Notify user and request correct data input |
| Unauthorized Access | Access | User credentials | Deny access and log attempt |
|  |  |  |  |
| Data Overload | |  | | --- | | Data Processing |  |  | | --- | |  | | Telemetry data volume | Throttle processing and alert admin |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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 |
| --- | --- | --- | --- |
| Unknown Error | Any step | N/A | Log error and notify development team |
| API Connection Fail | Data Processing | API endpoint | Retry connection, log error, notify 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 | Admin, Developer, Tester |
| Development Technologies Used | JavaScript, HTML/CSS, Node.js, Express, MongoDB, Azure |
| Deployment Technologies Used | Docker, Kubernetes, Azure DevOps |
| Scalable | Yes |

Table 4‑1: Project Details

**Reference**

chatgpt (2024). *Technical caveats*. [online] Chatgpt. Available at: https://chatgpt.com/ [Accessed 27 Jul. 2024].