**LEAN TECHNICAL DOCUMENTATION**

**[M MAFANEDZA]**

[35760338]

**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

This document outlines the structure and strategies for the planning of all projects that will be done in the IT Developments (CMPG 323) module this semester. Due to the content of CMPG 323, scheduling will be an important aspect of the module since it will allow us to devote enough time to the assignments that must be accomplished during the semester.

These are the five projects that will be done throughout the semester:

* Project 1: Agile and Scrum:

This project aims to enhance project management and delivery efficiency by implementing Agile methodologies and Scrum practices. It involves conducting training, and workshops, and establishing Scrum teams to facilitate iterative development and continuous improvement.

* Project 2: API Development:

This project focuses on creating and deploying APIs to facilitate smooth integration among various systems and applications. It involves the design, development, and testing of RESTful APIs to ensure they meet the specified business requirements.

High-Level Business Rules:

* + All APIs must adhere to RESTful principles.
  + APIs should be secured using OAuth2.
  + Versioning must be implemented for all APIs.
  + Documentation should be provided using Swagger.
  + APIs must handle requests and responses in JSON format.
* Project 3: Design & Architecture Standards:

The goal of this project is to provide a set of design and architecture guidelines that will ensure uniformity, scalability, and maintainability across all development initiatives. This will include developing standards, templates, and best practices for software design and architecture.

High-Level Business Rules:

* + All designs must follow established architectural patterns.
  + Use of microservices architecture for new developments.
  + Adherence to SOLID principles in software design.
  + Regular code reviews to ensure compliance with standards.
  + Use of UML diagrams for documenting architecture.
* Project 4: Testing and RPA:

This project aims to automate testing processes and implement Robotic Process Automation (RPA) to improve efficiency and minimize manual labour. It involves establishing automated testing frameworks and creating RPA bots to handle repetitive tasks.

High-Level Business Rules:

* Automation must be implemented for all test cases whenever possible.
* CI/CD pipelines should be utilized for automated testing.
* RPA bots must adhere to predetermined workflows and business rules.
* Regular monitoring and maintenance of RPA bots is required.
* Test results and bot activities should be logged and reported in detail.
* Project 5: Data Visualization and Monitoring:

The goal of this project is to create data visualization dashboards and monitoring tools that will offer immediate insights into business operations. This involves choosing the right visualization tools, designing effective dashboards, and establishing monitoring alerts.

High-Level Business Rules:

- Dashboards must be user-friendly and customizable.

- Real-time data should be used for monitoring the dashboards.

- Alerts and notifications need to be configurable.

- Data sources must be validated and cleaned before visualization.

- Regular updates and maintenance of the dashboards are required.

# Solution Design

## Detailed Solution Design

Technologies Used:

* Agile and Scrum: Kanban
* used for managing Agile projects, including task tracking, sprint planning, and documentation.
* API Development: .NET 8(C#)
* Design & Architecture Standards: .NET 8(C#)
* ensure consistent design and scalability in software development.
* Testing and RPA: UiPath
* for automated testing within CI/CD pipelines, and **Blue Prism** for RPA implementation.
* Data Visualization and Monitoring: Power BI
* for creating interactive dashboards, with Elasticsearch for efficient data retrieval and monitoring

Figure 2‑1: Context Diagram

Project 1: Agile & Scrum = Kanban

Project 2: API Development = C#

GitHub

Repository

Project 5: Data Visualization and Monitoring = Power BI

Project 3: Design & Architecture Standards = C#

Project 4: Testing & RPA = C#

## 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 license allocations will be done before development commences.
* <List all other assumptions, especially relating to business rules>

## 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.

## Integration complexities arising from the use of diverse technology stack.

## Potential performance issues regarding real-time data processing and visualization.

## Maintenance challenges associated with automated testing and RPA bots.

## Ensuring data consistency and accuracy across different systems.

## Dependence on third-party tools and libraries, which may have their limitations.

## 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