LEAN TECHNICAL DOCUMENTATION

[LTJ Nhlapo]

[39013219]

Date: *2024-07-29*

Document Version: *V1.0*

Commented [A1]: Update

Commented [A2]: Update

TABLE OF CONTENTS

1 Introduction

The CMPG 323 course consists of five individual projects, which are aimed at building several technical solutions based on .NET, databases, and automation tools like UiPath. The project work in these projects is aimed at providing practical knowledge of project management and addressing real business concerns.

Each of the projects is designed to help develop a software engineering skill set in a specific area, for example, how to create a REST API, how to handle JWT tasked systems, how to connect to different databases and data sources, and how to perform business tasks with the help of UiPath. The solutions will be implemented step by step, which will help students improve their capabilities gradually during the semester.

The solved problems span the entire range from implementing basic CRUD operations in web applications to automating tedious repetitive tasks and business processes with the help of bots.

2 Solution Design

2.1 Detailed Solution Design

Projects within CMPG 323 utilize several core technologies combined to create the final product. The backend is developed in .NET with SQL Server or other databases available for data persistence. Regarding user authentication and attestation, JWT tokens are relied upon. Frontend, when there is one, makes cross-calls to these services via Apis, and part of the work is done using UiPath for process automation.

In the diagram below, a context diagram is presented that shows how the components in the solution interact:

- **Backend (ASP.NET):** Exposes APIs for Carrying out Create, Read, Update, and Delete Operations.
- **Database (SQL Server):** Fulfils the need to store different information including projects and clients.
- **UiPath**: A feature that allows you to work with a machine without a need for manual intervention extensively, works to perform repetitive tasks repeatedly.

- **GitHub**: Source control and collaboration.

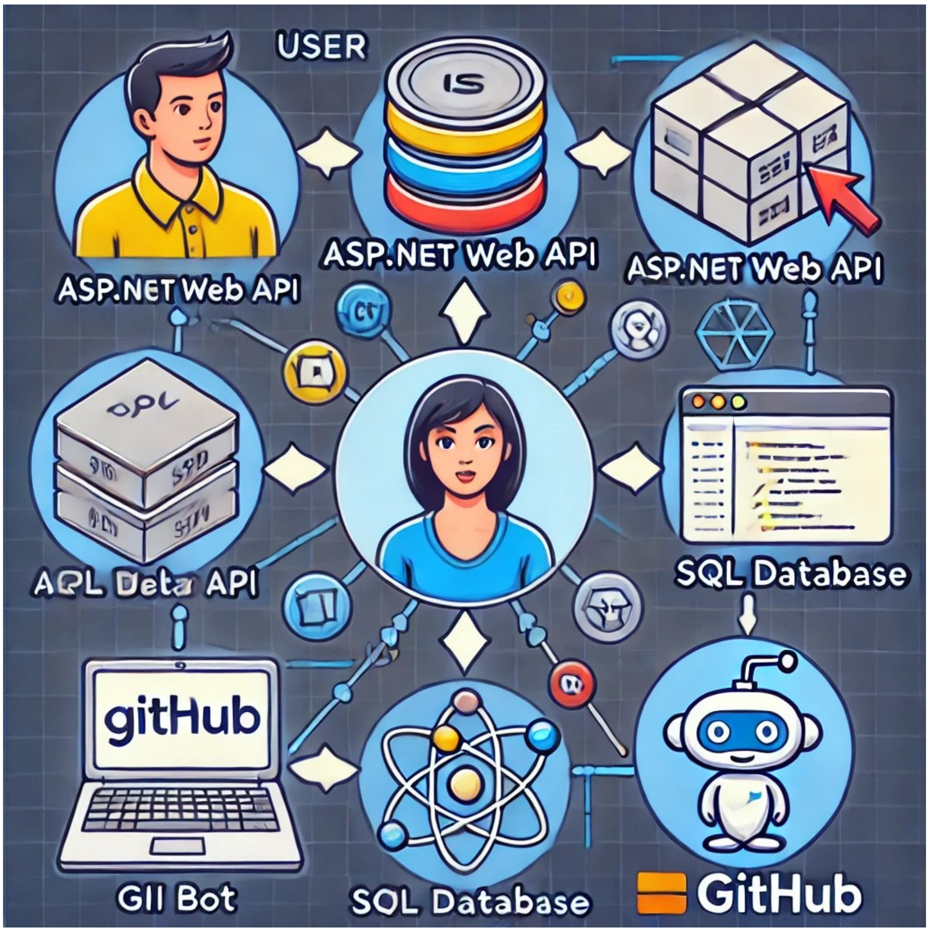


Figure 2-1: Context Diagram

Commented [A3]: Insert context diagram

2.2 Data Design

In most cases, APIs communicating to a central repository is a means by which data is moved from one project to another. The figure below depicts the transfer of information between components:

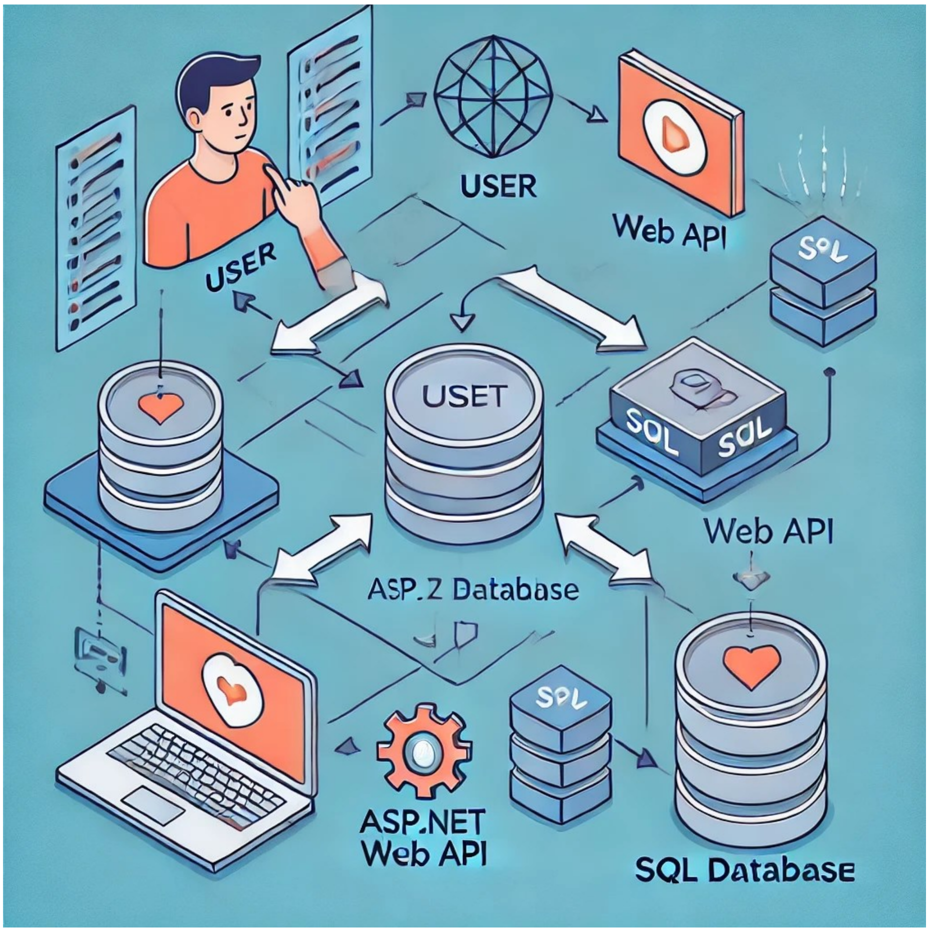


Figure 2-2: Data Flow Diagram

Commented [A4]: Insert the data flow diagram that shows how the same data is transported between projects.

MySQL workbench to design the database which includes various tables. In these tables, there are entities to handle managing projects, clients, and tasks. As shown below, how these tables relate to each other can be represented in Dolly's ERD. Such a design enables all the projects to utilize the same repository making it scalable.

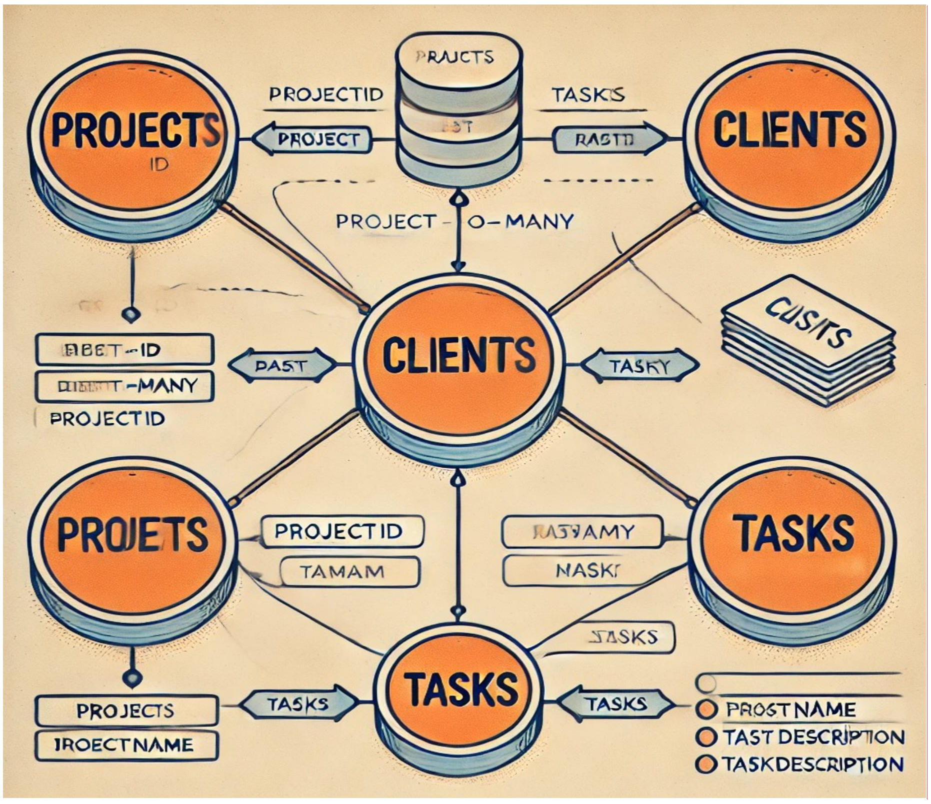


Figure 2-3: Data Design

2.3 Technical Assumptions

The design of this solution considered the following assumptions:

- All students have access to the licenses of the necessary development tools, such as Visual Studio, SQL Server, and UiPath.
- Data in the required database will be purged and sanitized.
- It is assumed that students will be provided with internet access for deployment on cloud infrastructure and sharing the work on GitHub.
- The automation scripts will work as intended on the operating systems designed to support them.
- The projects will be carried out in sequence and not at the same time.

Commented [A5]: Add an ERD-like structure that shows how the data is related.

Commented [A6]: Update technical assumptions with any assumptions that were made while you were designing the solutions you will be building for each project this semester.

2.4 Technical Caveats

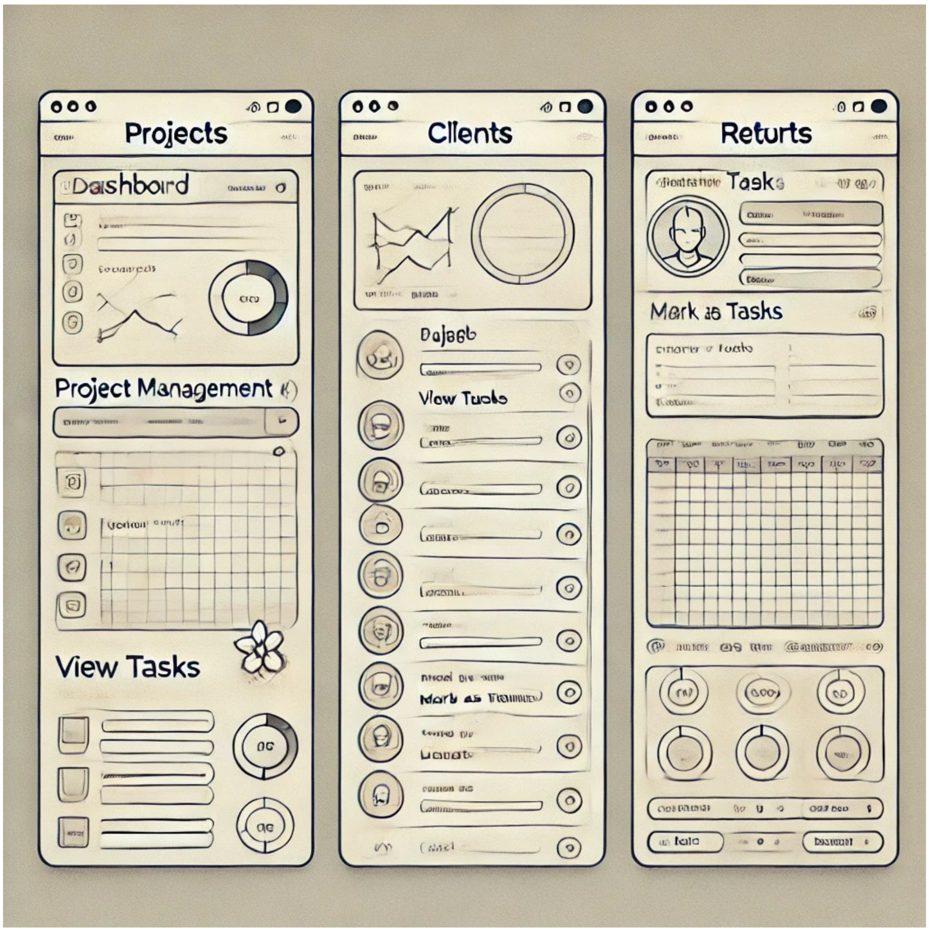
Commented [A7]: Any issues you have come across with using the proposed technology stack that will impact the implementation of your project should be listed here.

The following technical caveats could impact the implementation:

- Limited access to certain premium features in UiPath may constrain bot functionality.
- SQL Server versions across environments must be compatible to avoid deployment issues.
- GitHub's collaboration features may require additional setup for team-based projects.
- Authentication and authorization mechanisms may need adjustments for scaling to a larger number of users.

2.5 Wireframes

The wireframes for the various reports and interfaces to be developed across projects can be found below. These wireframes outline the expected layout and design of the user interface, ensuring usability and adherence to the project requirements.



Commented [A8]: Create some basic prototypes of the reports to be created and add the screenshots below. Please do not forget to add the figure captions below the images.

3 Errors & Exceptions

Commented [A9]: Add all exceptions and errors that may occur and that your solution will cater for

3.1 Business Exceptions

The following business exceptions should be built into the solution:

Exception Name	Step	Parameters	Action To Be Taken
Missing Data	Data Validation	Entity ID	Log error and prompt user for data input
Invalid Data Format	Data Parsing	Date, Currency	Trigger error and notify user
Timeout	Data Retrieval	SQL Connection Timeout	Retry operation and alert admin

Table 1: Business Exceptions

3.2 Application Errors

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

Exception Name	Step	Parameters	Action To Be Taken
Database Connection	API Call	Connection String	Retry connection or raise alert
Null Reference	API Request	Object Not Set	Validate inputs before operation
Unauthorized Access	Authentication	JWT Token	Redirect to login page and log event

Table 2: Business Exceptions

4 Environment Details

Commented [A10]: Complete the table below with all the relevant 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

Item	Description
Credentials Needed	GitHub SQL Server UiPath Credentials
Development Technologies Used	.NET Core SQL Server UiPath
Deployment Technologies Used	Azure GitHub Actions UiPath Orchestrator
Scalable	Yes (API & DB are scalable on Azure)

Table 4-1: Project Details