|  |  |
| --- | --- |
|  | **Athlone Institute of Technology**  **Faculty of Business and Hospitality**  **Higher Diploma in Science in Computing** |



Stewart Locke (A00277712)

Project Time Tracker

Athlone

2020

**ABSTRACT**

The Project Time Tracker is a Blazor Web Assembly based system designed to record and track time against projects within a company. This is achieved by storing a list of projects and recording the time each company member spends on each project. The aim of this project is to improve upon the existing software solution by adding to and improving the functionality of the existing system. This will be achieved by adding reporting and other ease of life improvements including: project templates; project dashboard; project closeouts; time reports and a closeout report. The deadline for this project is the 12th of April 2021.

**Keywords:** Blazor Web Assembly, Usability, Reporting, Dashboard, Project lifecycle

**ABREVIATIONS AND ACRONYMS**

**The System:** Project Time Tracker

**DB:** Database

**ERD:** Entity Relationship Diagram

**WASM:** WebAssembly

Table of Contents

[1 INTRODUCTION 5](#_Toc67899912)

[1.1 Blazor Web ASSEMBLY 5](#_Toc67899913)

[1.2 WebAssembly (Wasm) 5](#_Toc67899914)

[1.3 Asp.net core web APi 5](#_Toc67899915)

[1.4 .Net Core 5](#_Toc67899916)

[1.5 C# 5](#_Toc67899917)

[1.6 Entity Framework core 5](#_Toc67899918)

[1.7 SQL Server 6](#_Toc67899919)

[1.8 Stakeholders 6](#_Toc67899920)

[2 OVERVIEW 7](#_Toc67899921)

[2.1 PROBLEMATIZaTION 7](#_Toc67899922)

[2.2 JUSTIFICATION 7](#_Toc67899923)

[2.3 OBJECTIVES 7](#_Toc67899924)

[2.4 Risks 7](#_Toc67899925)

[3 Project Diagrams 8](#_Toc67899926)

[3.1 System Diagrams 8](#_Toc67899927)

[3.1.1 Before update 8](#_Toc67899928)

[3.1.2 After update 9](#_Toc67899929)

[3.2 Sequence Diagrams 10](#_Toc67899930)

[3.2.1 Project Closeout 10](#_Toc67899931)

[3.2.2 Report Generation 10](#_Toc67899932)

[3.3 Class Diagrams 11](#_Toc67899933)

[3.3.1 API’s 11](#_Toc67899934)

[3.3.2 Non-Report Entities 12](#_Toc67899935)

[3.3.3 Report Entities 12](#_Toc67899936)

[3.4 Entity Relationship DIAGRAM 13](#_Toc67899937)

[3.4.1 User Tables 13](#_Toc67899938)

[3.4.2 Project/Time Tables 13](#_Toc67899939)

[4 WIreframes 14](#_Toc67899940)

[4.1 create Project 14](#_Toc67899941)

[4.2 Project Dashboard 14](#_Toc67899942)

[4.3 project closeout 15](#_Toc67899943)

[4.4 Report Screens 15](#_Toc67899944)

[5 Requirements 16](#_Toc67899945)

[5.1 Functional 16](#_Toc67899946)

[5.2 Non-functional 18](#_Toc67899947)

[6 Testing 18](#_Toc67899948)

[7 Conculsion 22](#_Toc67899949)

# INTRODUCTION

This project is to create version 1.1 of the Project Time Tracker. In this project we will be improving upon the existing system that is used to track time against projects within an organisation. The improvements are aimed to improve the usability of the system and create meaningful reports out of the captured information. With these improvements the system becomes more useful and usable to the perspective clients. The System will be built on a Microsoft technology stack.

## Blazor Web ASSEMBLY

For the front-end code, I will be using Blazor Wasm, Blazor Wasm is a relatively new single-page app framework, developed by Microsoft and released in 2020. Blazor Wasm is designed for building interactive client-side web apps with .NET code in the browser with Wasm.

## WebAssembly (Wasm)

Blazor WebAssembly is compiled to run in the Wasm environment. Wasm is a new type of code that can be run in modern web browsers. It is a low-level assembly-like language with a compact binary format that runs with near-native performance. Wasm also allows access to JavaScript letting the two coding languages work alongside each other.

## Asp.net core web APi

To connect the front end to the back end I will be using ASP.Net Core, which is a Microsoft web framework that is built on top of .NET Core that supports creating web APIs using C#

## .Net Core

.NET is a free, open-source development platform for building many kinds of apps

## C#

For the back end code, I will be using C# 8.0, C# is a modern strongly typed object-oriented programming language developed by Microsoft around the year 2000, C# 8.0 came out in 2019.

## Entity Framework core

To connect the back-end code to the DB the system uses Entity Framework Core. Entity Framework Core is a modern object-database mapper for .NET Core that was initially released in 2008 with the latest release coming out in 2020.

## SQL Server

To store data between instances the system will be using Microsoft SQL Server. Microsoft SQL Server is a relational database management system developed by Microsoft. It is designed to store and retrieve data as requested by other software applications.

## Stakeholders

Project Manager: Stewart Locke

Customer: Company owner, Company employees

# OVERVIEW

## PROBLEMATIZaTION

Currently the project allows for the creation of projects and the ability to log time against those projects the problems with the existing system are twofold: adding a new project is timely and without a complete lifecycle; currently management has no way to review the information gathered.

## JUSTIFICATION

By improving the project creation process and completing the project lifecycle it greatly improves the usability of the system making the system a friendly to use solution. In the addition of reporting functionality to the system it makes the system a complete package usable by the management to track the time employees spend on each project.

## OBJECTIVES

I plan to improve the usability of the system by adding in project templates to the system, with the inclusion of project templates when a new project is created it will automatically start with a default set of tasks that can be modified by the “Setup” users.

I will complete the lifecycle of the systems projects by adding a historic backlog of comments to each project and having an ability to closeout projects within the system via a new screen that will be accessible via a new project dashboard screen.

I plan to add reports that retrieve the existing information about time spent against projects from the DB and display them in two ways, as a tabular format and in a graphical dashboard.

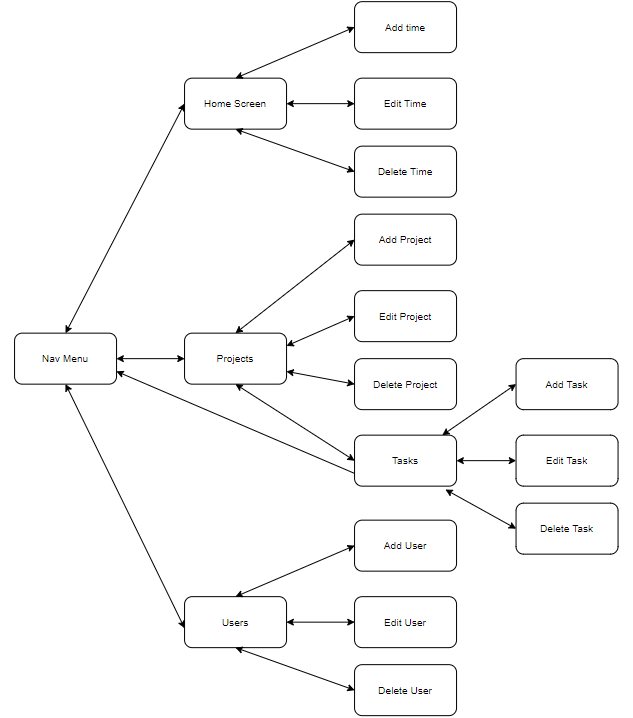
## Risks

The project needs to be complete by Monday the 12th of April

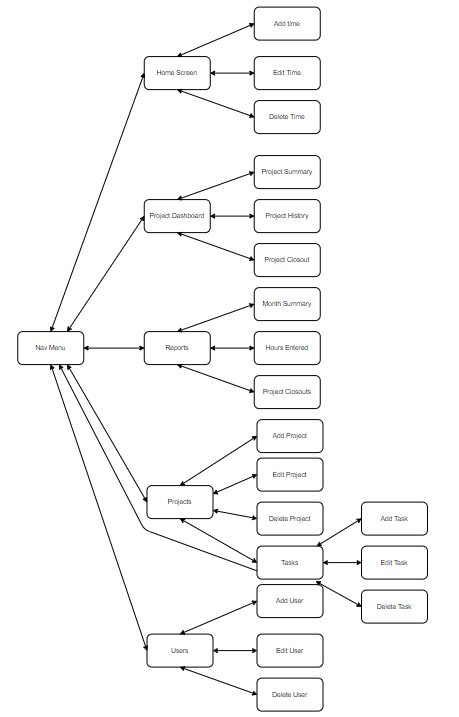
# Project Diagrams

## System Diagrams

### Before update



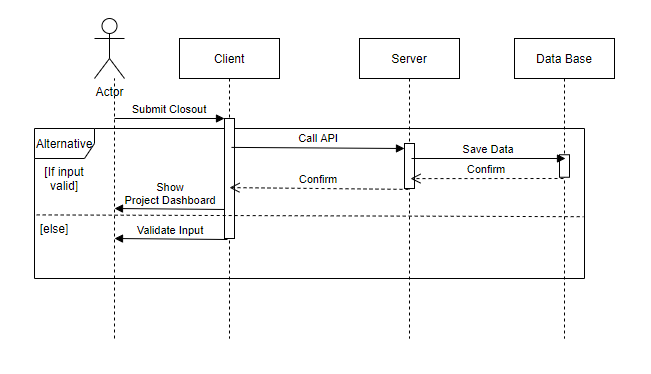
### After update



With the updated System we add the Project Dashboard and reporting screens. Via the Project dashboard we can maintain a comment log tracking the progress of the Project, this log can be viewed by the Project History Dialog. The Project Dashboard also links to a Project Summary Dialog that shows the time spent against a selected project and the Project Closeouts screen where a Setup user can end the lifecycle of a project. Via the reporting screens (Month Summary, Hours Entered and Project Closeouts) the users with the Reporting role can get information out of the system.

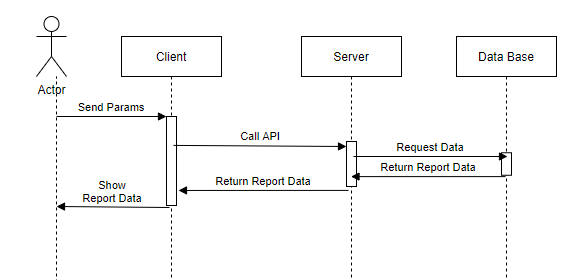
## Sequence Diagrams

### Project Closeout



When a Setup user wants to closeout a project from the Project Closeout screen, they need to fill in all input fields then click the submit button, if all fields are filled and valid an API call is made to the server which in turn updated the ProjectCloseouts table on the Data Base. Once the Data base is updated the API returns to the client which redirects the user back to the Project Dashboard screen.

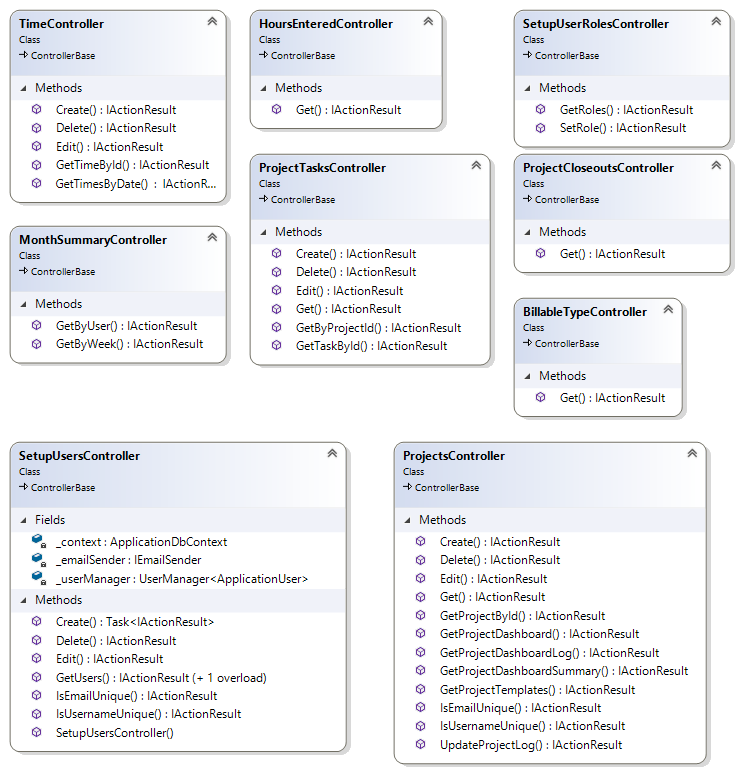
### Report Generation



Each of the Reports in the system work in a similar way when the page is called or the parameters of the report are updated am API POST request is called on the server which in turn goes to the data base to retrieve the report data. That report data is returned to the Client who then displays the report data in a table format.

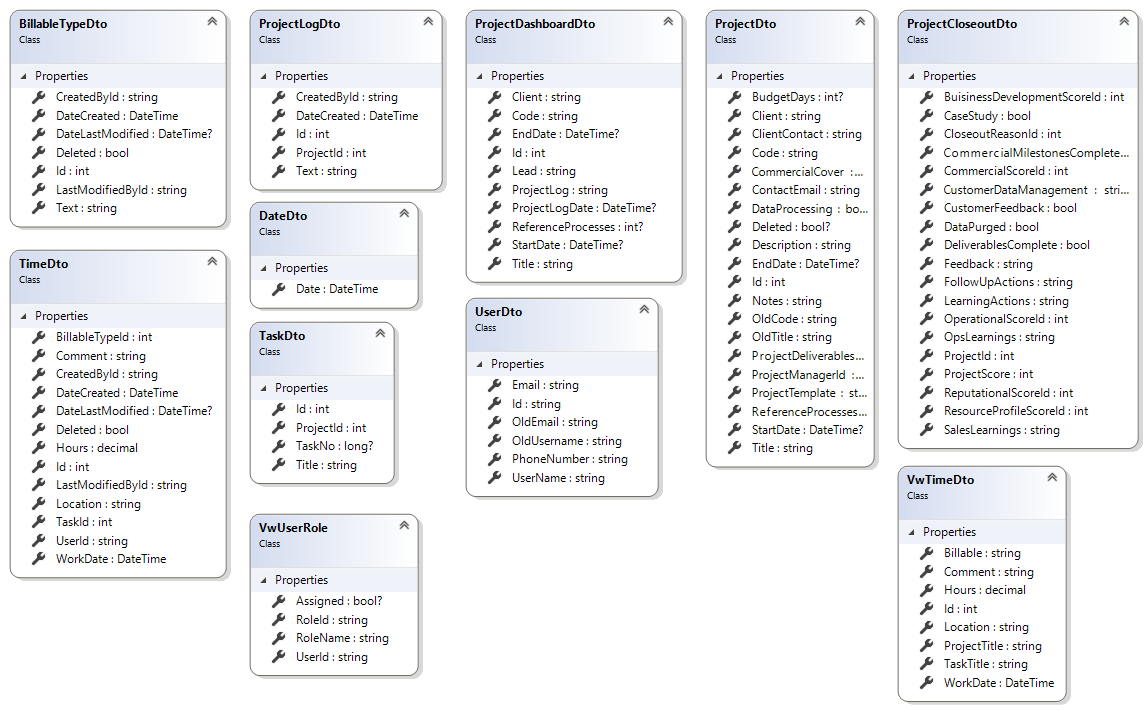
## Class Diagrams

### API’s



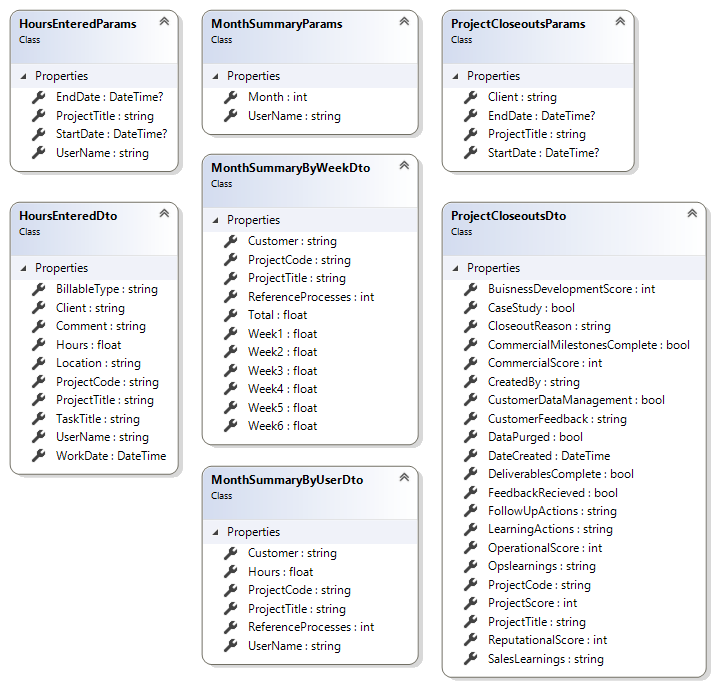
Every interaction an actor makes on the client side of the system that requires an action be made on the server side must go through an API call, the above are the classes that map to each API call

### Non-Report Entities



All the Entities that are used to transfer Data between the client and the server via API calls

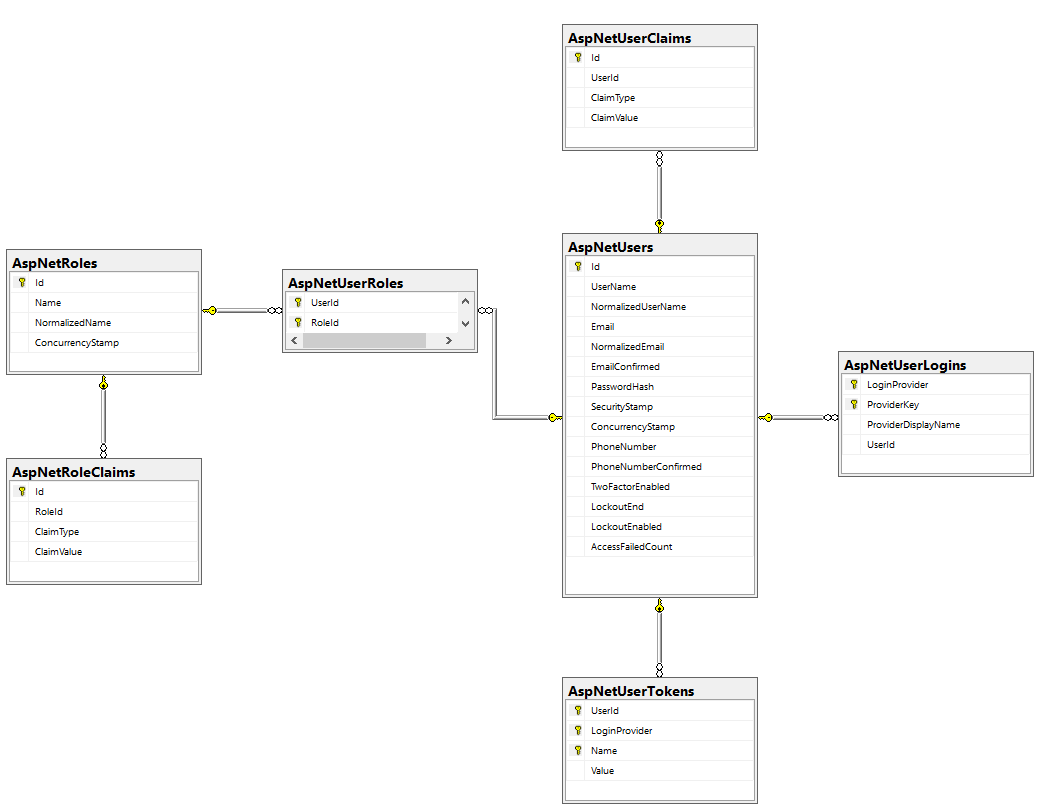
### Report Entities



All the Entities that are used to transfer Data between the client and the server via API calls for reporting purposes

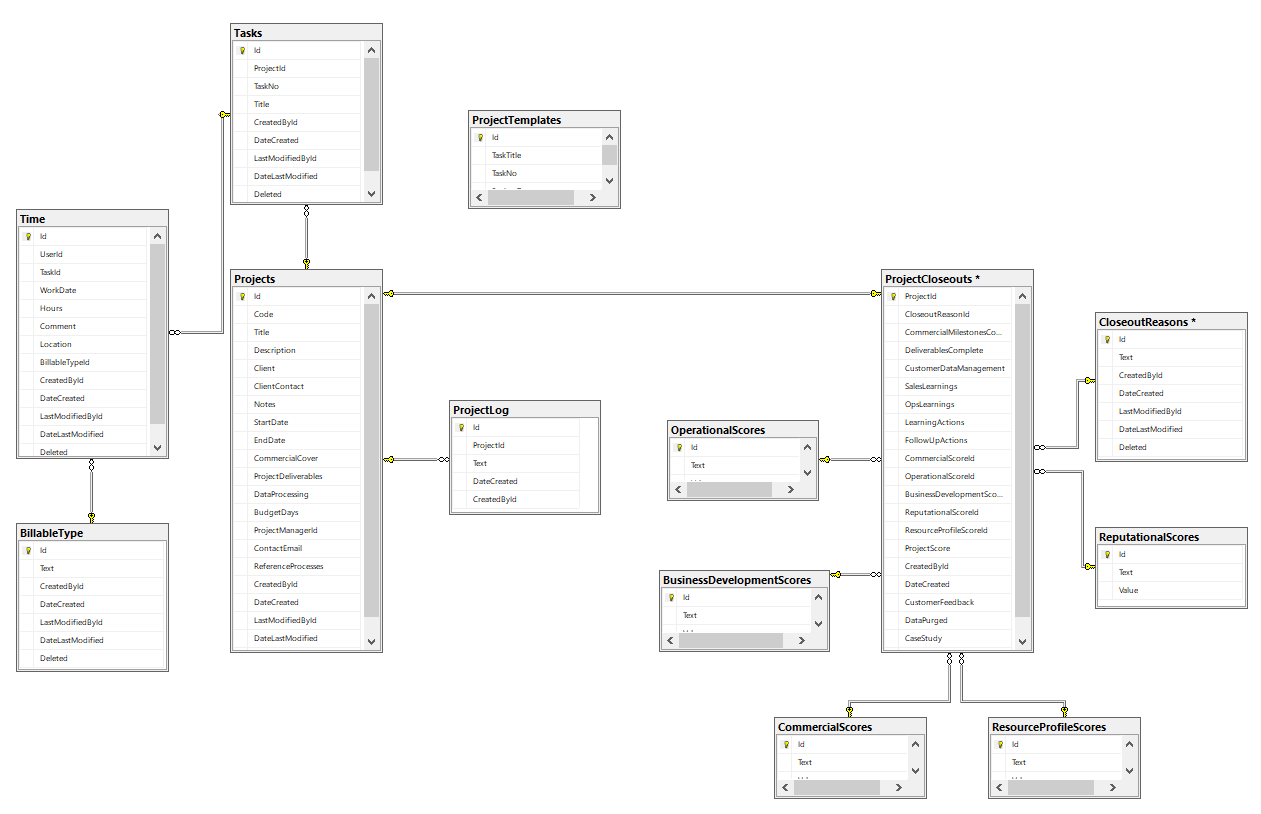
## Entity Relationship DIAGRAM

### User Tables



The Database tables used for Authorisation and Authentication

### Project/Time Tables



The Database tables used for tracking projects and time spent against each project

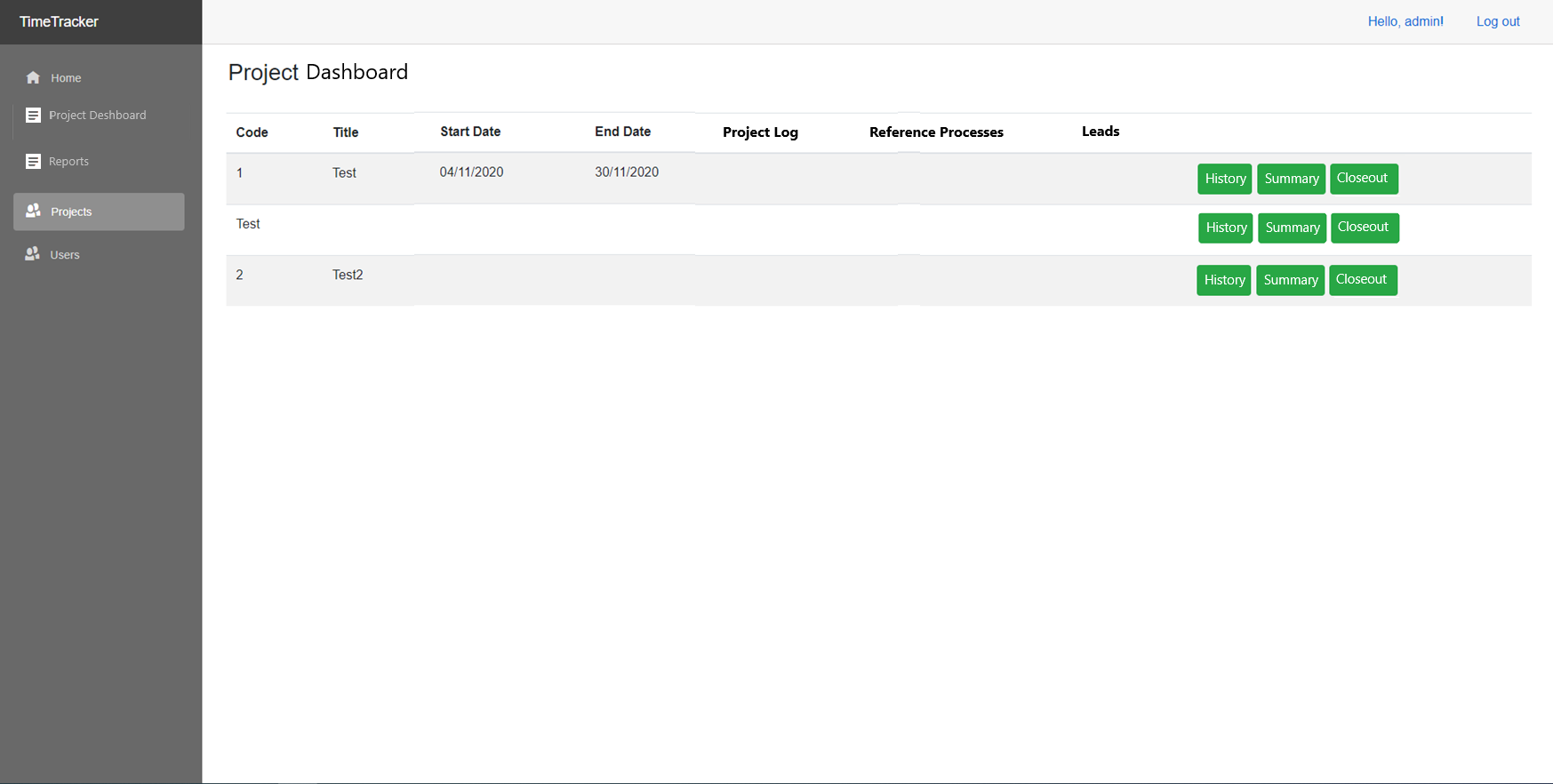
# WIreframes

## create Project



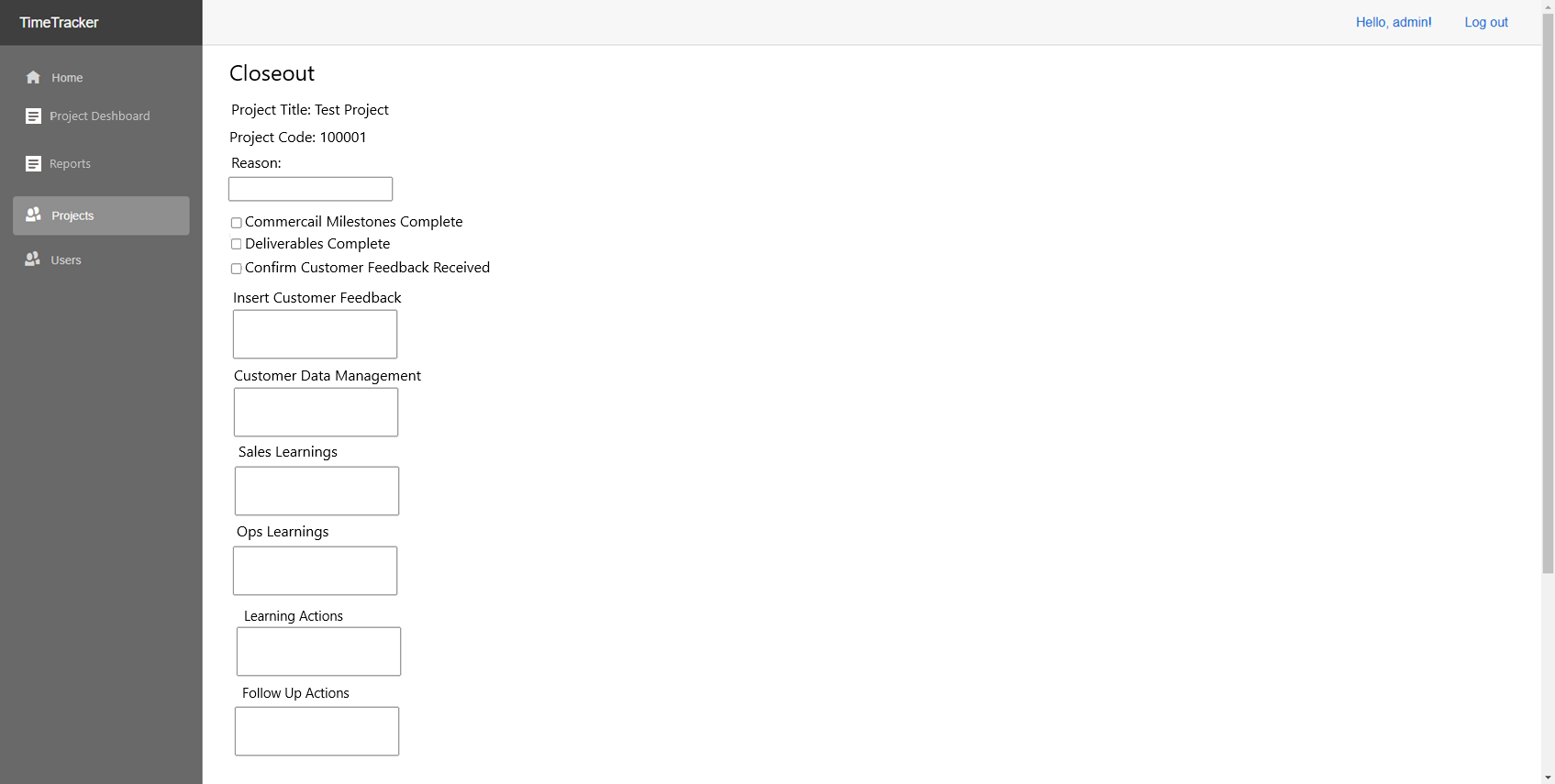
The top of the screen to create projects, this screen already exists in the system but we are adding the Project template section which allows tasks to be set according to templates

## Project Dashboard



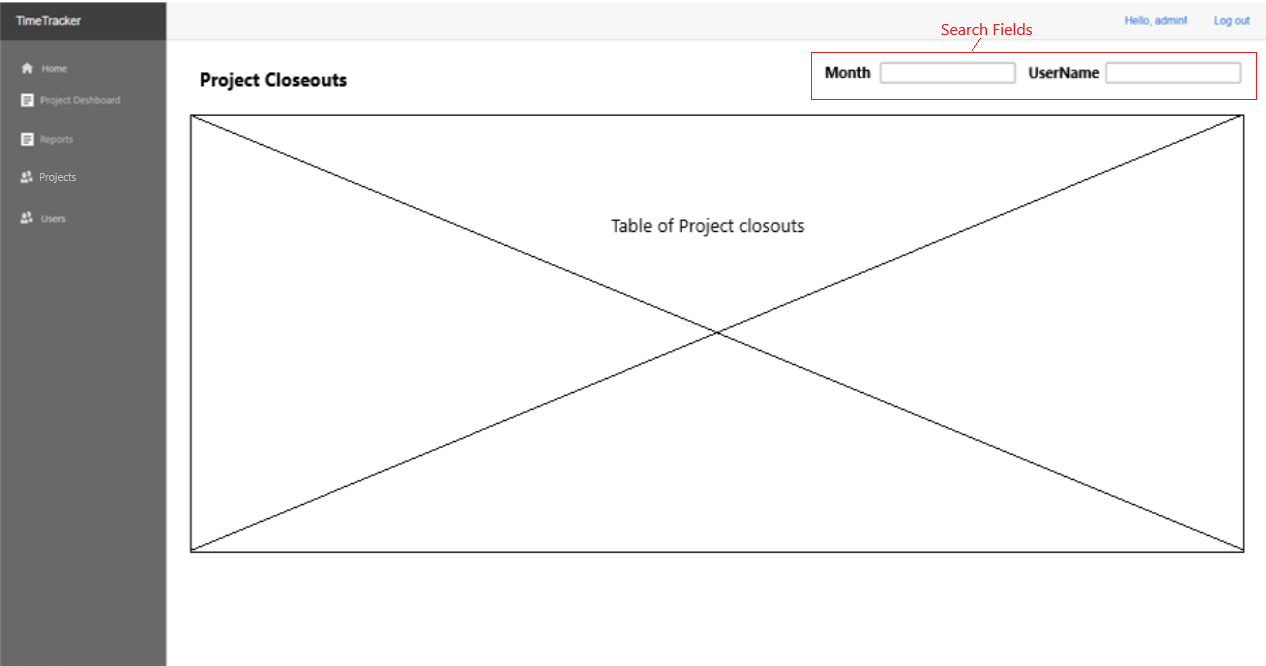
The Project Dashboard screen is a new screen that displays all information about current projects

## project closeout



The Project closeout screen is the final stage of a project’s lifecycle, where a project is set from current to closed. By filling out and submitting a Closeout report the project is considered closed and removed from the projects screen, time screen and the project dashboard.

## Report Screens



All report screens in the system will follow a similar layout with report title in the top left, report filters on the top right and the report data being displayed in a table in the centre of the screen.

# Requirements

## Functional

| Requirement ID | Screen | Description | Priority |
| --- | --- | --- | --- |
| FR 1 | Create Projects | The create projects screen must give the optional ability to select a project template type | Must |
| FR 1.1 | Create Projects | The system must on project creation create new tasks according to project template type | Must |
| FR 2 | Project Dashboard | The system must have a project dashboard | Must |
| FR 2.1 | Project Dashboard | The project dashboard must show all open projects (open projects are projects that don’t have a closeout assigned) | Must |
| FR 2.2 | Project Dashboard | The projects dashboard must have the ability to add a comment against a project | Must |
| FR 2.3 | Project Dashboard | The projects dashboard must have the ability to view the past comments on a project | Must |
| FR 2.4 | Project Dashboard | The project dashboard should have a project summary displaying time spent against the project | Should |
| FR 2.5 | Project Dashboard | The project dashboard must have a button linking to the project closeout screen | Must |
| FR 3 | Project Closeout | The system must have a project closeout screen | Must |
| FR 3.1 | Project Closeout | The project closeout screen must capture the following fields: Closeout Reason, Commercial Milestones Complete, Deliverables Complete, Customer Data Management, Sales Learnings, Ops Learnings, Learning Actions, Follow Up Actions, Commercial Score, Operational Score, Business Development Score, Reputational Score, Resource Profile Score, Project Score, Created By, Date Created, Insert Customer Feedback, Data Purged, Case Study, Customer Feedback Received | Must |
| FR 3.1.1 | Project Closeout | The project closeout screen must allow selection from prepopulated dropdowns for the following fields: Closeout Reason, Commercial Score, Operational Score, Business Development Score, Reputational Score, Resource Profile Score | Must |
| FR 3.1.2 | Project Closeout | The project closeout screen must automatically capture following fields: Created By, Date Created | Must |
| FR 3.1.3 | Project Closeout | The project closeout screen must calculate the Project Score as an addition of: Commercial Score, Operational Score, Business Development Score, Reputational Score, Resource Profile Score | Must |
| FR 4 | Hours Entered Report | The system must have Hours Entered Report | Must |
| FR 4.1 | Hours Entered Report | The Hours Entered Report must take the following parameters: Start Date, End Date, Username, Project Title | Must |
| FR 4.2 | Hours Entered Report | The Hours Entered Report must report on the following fields: Project Code, Project Title, Client, Task Title, Username, Work Date, Hours, Comment, Billable Type, Location | Must |
| FR 5 | Month Summary Report | The system must have a Month Summary Report | Must |
| FR 5.1 | Month Summary Report | The Month Summary Report must take parameters of: Month and Username | Must |
| FR 5.2 | Month Summary Report | The Month Summary Report must display a Summary by Week report | Must |
| FR 5.2.1 | Month Summary Report | The Summary by Week report must have the following fields: Customer, Project Code, Project Title, Reference Processes, Week 1, Week 2, Week 3, Week 4, Week 5, Total | Must |
| FR 5.3 | Month Summary Report | The Month Summary Report must display a Summary by User report | Must |
| FR 5.3.1 | Month Summary Report | The Summary by User report must have the following fields: Customer, Project Code, Project Title, Reference Processes, Users usernames, Total | Must |
| FR 6 | Project Closeouts | The System Must have Project Closeouts Report | Must |
| FR 6.1 | Project Closeouts | The Projects Closeouts Report must have the following parameters: Start Date, End Date, Client, Project Title | Must |
| FR 6.2 | Project Closeouts | The Project Closeouts Report must have the following fields: Project Code, Project Title, Closeout Reason, Commercial Milestones Complete, Deliverables Complete, Customer Data Management, Sales Learnings, Ops Learnings, Learning Actions, Follow up Actions, Commercial Score, Operational Score, Business Development Score, Reputational Score, Project Score, Created By, Date Created, Confirm Customer Feedback Received, Insert Customer Feedback, Data Purged, Case Study |  |

## Non-functional

|  |  |  |
| --- | --- | --- |
| Requirement ID | Description | Priority |
| NFR 1 | The system must not have exceptional wait times for any screen | Must |
| NFR 2 | The system must be web hosted | Must |
| NFR 3 | The system must target the Google Chrome web browser | Must |

# Testing

| Requirement ID | Preconditions | Input | Expected Output | Result |
| --- | --- | --- | --- | --- |
| FR 1 | Be on the Create Projects screen | None | Ability to select a project template type | Pass |
| FR 1.1 | Be on the Create Projects screen | Project Code: 1  Project Template: Software | Project Created with Template tasks assigned | Pass |
| FR 2 | Be logged in to the system | Click Project Dashboard in the menu | Successfully navigate to the Project Dashboard | Pass |
| FR 2.1 | Be on the Project Dashboard | None | Able to see all open projects | Pass |
| FR 2.2 | Be on the Project Dashboard | Click on a project comment | Ability to edit the comment | Pass |
| FR 2.3 | Be on the Project Dashboard | Click on a project’s history button | A history of all comments is correctly shown | Pass |
| FR 2.4 | Be on the Project Dashboard | Click on a Projects Summary button | A summary of all time spent on a project is correctly shown | Pass |
| FR 2.5 | Be on the Project Dashboard, A project has the comment of closeout | Click on the projects Closeout button | Successfully navigates to the Project Closeout screen | Pass |
| FR 3 | Be on the Project Dashboard, A project has the comment of closeout | Click on the projects Closeout button | Successfully navigates to the Project Closeout screen | Pass |
| FR 3.1 | Be on the Project Closeout screen | Enter a: Closeout Reason, Commercial Milestones Complete, Deliverables Complete, Customer Data Management, Sales Learnings, Ops Learnings, Learning Actions, Follow Up Actions, Commercial Score, Operational Score, Business Development Score, Reputational Score, Resource Profile Score, Insert Customer Feedback, Data Purged, Case Study, Customer Feedback Received | The project closeout report must correctly show the entered data for: Closeout Reason, Commercial Milestones Complete, Deliverables Complete, Customer Data Management, Sales Learnings, Ops Learnings, Learning Actions, Follow Up Actions, Commercial Score, Operational Score, Business Development Score, Reputational Score, Resource Profile Score, Project Score, Created By, Date Created, Insert Customer Feedback, Data Purged, Case Study, Customer Feedback Received | Pass |
| FR 3.1.1 | Be on the Project Closeout screen | None | Populated dropdowns appear for: Closeout Reason, Commercial Score, Operational Score, Business Development Score, Reputational Score, Resource Profile Score | Pass |
| FR 3.1.2 | Be on the Project Closeout screen | Enter closeouts required fields | The project closeout report must correctly show the Created By, Date Created fields | Pass |
| FR 3.1.3 | Be on the Project Closeout screen | Enter closeouts required fields | The project closeout report must correctly show the Project Score | Pass |
| FR 4 | Be logged in to the system with the reporting role | Click on the Hours Entered Report in the navigation menu | Successfully navigate to the Hours Entered Report | Pass |
| FR 4.1 Test 1 | Be on the Hours Entered Report | Enter: Start Date, End Date  Click Submit | Report correctly appears | Pass |
| FR 4.1 Test 2 | Be on the Hours Entered Report | Enter: Start Date, End Date, Username, Project Title  Click Submit | Report correctly appears | Pass |
| FR 4.2 | Be on the Hours Entered Report | Enter: Start Date, End Date  Click Submit | The Hours Entered Report shows the following fields: Project Code, Project Title, Client, Task Title, Username, Work Date, Hours, Comment, Billable Type, Location | Pass |
| FR 5 | Be logged in to the system with the reporting role | Click on the Month Summary Report in the navigation menu | Successfully navigate to the Month Summary Report | Pass |
| FR 5.1 Test 1 | Be on the Month Summary Report | Enter: Month  Click Submit | The Month Summary Report correctly displays | Pass |
| FR 5.1 Test 2 | Be on the Month Summary Report | Enter Month and Username | The Month Summary Report correctly displays | Pass |
| FR 5.2 | Be on the Month Summary Report | Enter: Month  Click Submit | A Summary by Week report displays | Pass |
| FR 5.2.1 | Be on the Month Summary Report | Enter: Month  Click Submit | The Summary by Week report displays the following fields: Customer, Project Code, Project Title, Reference Processes, Week 1, Week 2, Week 3, Week 4, Week 5, Total | Pass |
| FR 5.3 | Be on the Month Summary Report | Enter: Month  Click Submit | A Summary by User Report displays | Pass |
| FR 5.3.1 | Be on the Month Summary Report | Enter: Month  Click Submit | The Summary by User report displays the following fields: Customer, Project Code, Project Title, Reference Processes, Users usernames, Total | Pass |
| FR 6 | Be logged in to the system with the reporting role | Click on the Project Closeout Report in the navigation menu | Successfully navigate to the Project Closeout Report | Pass |
| FR 6.1 Test 1 | Be on the Projects Closeouts Report | Enter a: Start Date, End Date  Click Submit | The Projects Closeouts Report displays | Pass |
| FR 6.1 Test 2 | Be on the Projects Closeouts  Report | Enter a: Start Date, End Date, Client, Project Title | The Projects Closeouts Report displays | Pass |
| FR 6.2 | Be on the Projects Closeouts Report | Enter a: Start Date, End Date  Click Submit | The Project Closeouts Report displays the following fields: Project Code, Project Title, Closeout Reason, Commercial Milestones Complete, Deliverables Complete, Customer Data Management, Sales Learnings, Ops Learnings, Learning Actions, Follow up Actions, Commercial Score, Operational Score, Business Development Score, Reputational Score, Project Score, Created By, Date Created, Confirm Customer Feedback Received, Insert Customer Feedback, Data Purged, Case Study | Pass |
| NFR 1 | Have the site open | Navigate between pages | All sections of the website load within 5s | Pass |
| NFR 2 | None | None | The system is accessed via a web browser | Pass |
| NFR 3 | Have Google Chrome | Open the site in Google Chrome | The system is fully accessible in Google Chrome | Pass |

# Conculsion

In completing the software project, I reinforced and gained many skills required for developing in Blazor and .Net Core. Alongside the skills required for implementing the solution I also improved my ability to carry out analysis and design for any future projects.

On finalisation of this round of development I have successfully rounded out and completed the Project Time Tracker software solution by adding in: project templates to the system; a historic backlog of comments per project; an ability to closeout projects and reports that retrieve the existing information. Each of these objectives were accomplished by either adding to an existing page or the addition of a new page in the system.

Development went smoothly with few minor difficulties that were all solved in a timely manner. Some of the issues I faced include the default dropdown only accepting string values, calling stored procedures and views from the DB and how to make use of project templates.

Final testing of the system went smoothly, with all aspects of the solution passing validating that the system is ready for deployment.

I consider the system complete with no further add-ons or development required as all required functionality has been implemented, in the future new requirements or changes may arise but at this time the system is finalized and ready for deployment.