



OLLSCOIL NA GAILLIMHE

UNIVERSITY OF GALWAY

School of Computer Science

Project Definition Document

Name: Kacper Poniatowski

ID: 19461182

Academic Supervisor:

Michael Schukat

Contents

1. Introduction.....	3
1.1 Project Deliverables	3
2. Project Definition	3
2.1 Project Research.....	3
2.2 Tools and Software Used	4
2.2.1 C#	4
2.2.2 Git / GitHub	4
2.2.3 FactoryTalk Optix	5
2.2.4 Uniqo	5
2.2.5 Visual Studio 2022.....	5
2.2.6 StringTemplate 4	5
2.2.7 Selenium WebDriver	6
2.2.8 Specflow	6
2.3 Project Objectives.....	6
2.3.1 HMI Panels.....	6
2.3.2 Auto-Generation	8
2.3.3 Unit Testing	8
3. Project Timeline.....	10
4. Conclusion	10
5. References	11

Figures

Figure 1 - C# logo	4
Figure 2 - GitHub logo	4
Figure 3 - Uniqo logo	5
Figure 4 - Specflow logo	6

1. Introduction

The purpose of this Project Definition Document is to provide a brief overview of what my Final Year Project will entail and the direction in which I intend to take in the development phase.

My Final Year Project (FYP) will be a project containing a collection of human-machine interface (HMI) panels, with each panel displaying some different feature or functionality that “FactoryTalk Optix” from Rockwell Automation [1] has to offer. This software is to be released by the end of 2022 and rolled out within Medtronic upon release, replacing their current software. Therefore, it is necessary for the relevant personnel within Medtronic to get a grasp of the capability of this new software prior to the full-scale release.

1.1 Project Deliverables

Deliverable	Date
Project Definition Document	5 th November 2022
Project Report	1 st April 2023
Project Demonstration	3 rd – 6 th April 2023

2. Project Definition

2.1 Project Research

Seeing as I have no prior experience working with this software, I was required to conduct some extensive research to begin working. I began by following a YouTube tutorial series focusing on the basic features the software has to offer to become familiar with it. With this tutorial I learned about the functions of each of the buttons displayed to the user on the main screen, how objects interact with each other and how to create responsive designs based on user input, amongst other basic features.

Following on from this, I began to research the documentation provided by Uniqo. This documentation was written for Uniqo (earlier version to FactoryTalk Optix) but seeing as the software is largely the same the documentation is still useful. I read about some more advanced features that FactoryTalk Optix has to offer but practicing using these features is the best way to understand them therefore I moved onto the sample projects that are also provided with the documentation.

I studied a sample project that included a fully functional login and logout form which navigated panels based on user inputs (correct password entry navigated the user to the main panel and pressing the logout button in the bottom right of the screen brought the user back

to the login panel). From studying this project, I began to grasp how NetLogic files can be used to provide additional functionality to projects based on user inputs and how to include navigation between panels. Once I concluded studying this project, I decided to begin working on my first panel.

2.2 Tools and Software Used

2.2.1 C#

C# (pronounced C-sharp) is a multi-paradigm high-level programming language. It encompasses static typing, lexically scoped, strong typing, declarative, imperative, object-oriented, functional, generic and component-based programming disciplines. [2]

The language was designed in 2000 by Anders Hejlsberg from Microsoft and it was approved as an international standard by ECMA in 2002 and ISO/IEC in 2003. [2]



Figure 1- C# logo

FactoryTalk Optix allows the use of C# scripts to provide additional functionality to projects by allowing development of customised runtime and design time logics. These scripts can be used to create a responsive user interface [3], amongst other things.

2.2.2 Git / GitHub

Git is a free and open-source distributed version control system which keeps track of any changes to files in a project. It can handle all sizes of projects without compromising speed or efficiency. Git was authored by Linus Torvalds in 2005 and has been updated continuously ever since. [4]



Figure 2 - GitHub logo

Meanwhile, GitHub is (as of writing this) the most popular internet hosting service for version control and software development using Git. The company was founded in 2008 as 'Logical Awesome LLC'. It provides all the features that Git has to offer but also includes additional features for its users such as bug tracking, access control, software features requests, task management, wikis for every project and continuous integration. [5]

I will be using GitHub for version control for this project. I will have two repositories, one for the FactoryTalk Studio project and one for the automated tests. Regular commits to both repositories will ensure if any issues arise, I will be able to revert back to a previous working commit, narrow down on what's causing the issue and fix it.

2.2.3 FactoryTalk Optix

FactoryTalk Optix is a new product from Rockwell Automation. It is a cloud-enabled software that allows users to design, test and deploy human-machine interface applications from a desktop editor or web browser. As of writing this, FactoryTalk Optix is not yet available to the public. [6]

The bulk of my project will be conducted using FactoryTalk Optix. I will be using it to design all of the panels that will showcase the different functions and features FactoryTalk Optix has to offer.

2.2.4 Uniqo



Figure 3 - Uniqo logo

Uniqo is the predecessor to FactoryTalk Optix which was developed by Asem Automation (owned by Rockwell Automation) [7].

I will be using this software alongside FactoryTalk Optix to conduct research. Both software are still quite similar in functionality, but all sample projects provided in the documentation are Uniqo projects which are not compatible with FactoryTalk Optix. To help my understanding of the features FactoryTalk Optix has to offer, I will be required to open the provided sample projects in Uniqo.

2.2.5 Visual Studio 2022

Visual Studio 2022 is a free, extensible and fully featured IDE that allows users to create applications for various platforms such as Windows, iOS and Android as well as cloud services and web applications [8]. It is the latest version of Visual Studio released by Microsoft.

Visual Studio 2022 is my IDE of choice to write and develop the unit tests for the project. I chose this IDE as I have experience using this IDE throughout my 3rd year placement, therefore I am familiar with it and the features it has to offer to its users.

2.2.6 StringTemplate 4

StringTemplate 4 is a java template engine which is used to generate source code, emails, web pages or any other sort of formatted text output. As stated, it is a java template engine, but it has ports to C#. [9]

I will be utilising StringTemplate 4 for auto-generation of the panels that I will be creating in FactoryTalk Optix. The panels are created using only yaml files, making them ideal to be generated using a template engine such as StringTemplate 4.

2.2.7 Selenium WebDriver

Selenium WebDriver is a web framework that allows users to create cross-browser tests. These tests are then used to verify that a web-based application performs expectedly. [10]

FactoryTalk Optix allows for the deployment of projects through a web browser, making it ideal for automated testing. I can open the project and deploy it to a web browser from which I can then run the tests and they will perform tests on my project automatically.

2.2.8 Specflow

Specflow allows users to manage, define and execute automated tests on applications as business readable specifications. It is designed to bridge the gap in understanding between the developers and business users. It is a part of the Cucumber ecosystem and is based on Gherkin [11].



Figure 4 - Specflow logo

I will be using Specflow to define my tests as 'scenarios', which are made up of a series of short and logical steps, which can be easily read and understood by anyone, whether they have a background in software development or not.

2.3 Project Objectives

The core objective of this project is to test the capabilities and functionality of FactoryTalk Optix. To do this, I will be creating a series of panels with each panel showcasing different features.

2.3.1 HMI Panels

Input Validation on Input Fields

Input validation is an absolutely necessary feature in any software that includes user inputs. Without it, user could knowingly (or unknowingly) cause errors in the software by inputting invalid characters.

To display this, I will insert an input field for the user to type text into and a submit button.

With the use of NetLogic files, I will implement dynamic input validation so that the input is verified with each change in the input string (if user types a new character and if user deletes characters). The fact that the input is checked dynamically will allow me to disable the submit button so that it can't be pressed until the input matches all the set criteria. The criteria in this case will be an Eircode regex pattern in the form of a string.

If the inputted string is incorrect, this will be communicated to the user with the use of visual cues such as changing the border colour of the input field, and an error message below the input field.

Button Patterns

This panel will display the possibilities of assigning different properties to objects depending on their state or the state of an object on which they are dependent on.

The objective is to insert several buttons each linked to a different object. For example, button one can be linked to a slider and when the user adjusts the value of the slider the state of the button will change.

This feature is paramount to the effectiveness of this new software because in a real-world example an operator may have the ability to adjust several parameters of a recipe in the manufacturing of a part. Therefore if they make a mistake, it is vital the software alerts the operator and stops progression by adjusting the state of objects until the mistake is resolved.

Responsive Design

This panel will display the capabilities of FactoryTalk Optix to handle resizing of the window screen and maintain usability by not breaking the design (E.g.: no overlapping objects once window is scaled down in size).

The planned objective is to insert a large amount of objects into the panel, each containerised accordingly to their position in the panel. The objects can be textboxes, sliders, buttons, gauges, amongst other object types. When this panel is resized (larger or smaller), it is expected the objects will scale accordingly and remain useable and remain true to the design.

Vector Graphics

This panel will display the possibility to insert vector graphics such as an SVG image into a panel and display them to the user.

The current plan is to create an SVG object inside the panel and adjust it programmatically in some sort. One plan I have for this is to adjust the sides of an object based on some user inputs (E.g.: User can select the colour of the sides of the object from a dropdown list or a colour wheel).

Stylesheets

I have not decided for certain if I will contain this in its own panel or add the stylesheet(s) to all of the other panels. Either way, the objective of this is to display the possibility to create a

stylesheet which can then be applied to the project or multiple stylesheets with each being applied to certain panels chosen by the user.

Functional 2D Barcode Scanner

The main objective of this panel is to display a fully functional 2D barcode scanner. This feature is imperative in real-world applications. In a manufacturing line in which this software will be used, each part that is manufactured needs to be scanned to log into the system whether it was successful or not.

I haven't finalised my decision on what to include in this panel other than the 2D barcode scanner, but my current idea is to replicate an actual HMI screen that includes a 2D barcode scanner and a database in which the scanned part can be logged with the necessary information on that part.

Reusable Graphics

The purpose of this panel is to showcase the ability to design some sort of graphic (E.g.: an input field or box) to my specifications and use it throughout your project using multiple instances of this graphic.

One design I have in mind for this panel is to once again replicate a real HMI screen (similar idea to the functional 2D barcode scanner panel). Using this idea, I can create multiple different graphics and use them throughout the panel.

2.3.2 Auto-Generation

This section of the project relates to auto-generation of code with the use of a template engine. In FactoryTalk Optix, each screen is a yaml file which is then interpreted by the software and a graphical output is generated for the user. The use of yaml files makes this ideal to generate screens using a template engine. For this I will be using a template engine called StringTemplate 4.

2.3.3 Unit Testing

The final part of this project is related to the creation of unit tests. These tests will test the functionality of the panels that I created in FactoryTalk Optix. This ensures all features in each panel work as intended with no unexpected behaviours.

FactoryTalk Optix allows its users to emulate projects through a web browser instance, making it ideal for running unit tests against your project. I will be using Selenium to automate the browser instance of my project and to find objects in the webpage that will be required

for each test. Alongside Selenium I will be using Specflow to create a user-friendly, readable interface for each test. This will ensure people from all backgrounds (tech or not) will be able to read and understand the goal of my tests.

Another benefit of the web browser instance is the ability to use the web developer tools built into Chrome (my choice of browser for the unit testing). This will aid me in finding the specific path to find objects in the webpage which will be used by Selenium.

3. Project Timeline

Task ID	Task	Estimated Time in Hours	Dependencies	Expected Finish Date	Actual Finish Date
T1	Download necessary technologies	2h	None	29/09/2022	29/09/2022
T2	Research how to use FactoryTalk Optix	8h	None	01/10/2022	01/10/2022
T3	Panel 1: Display input field validation on input fields	4h	T1, T2	07/10/2022	10/10/2022
T4	Panel 2: Display button pattern capabilities	4h	T1, T2	21/10/2022	28/10/2022
T5	Panel 3: Display responsive design capabilities	6h	T1, T2	28/10/2022	---
T6	Panel 4: Display vector graphics capabilities	6h	T1, T2	04/11/2022	---
T7	Panel 5: Display vector graphics capabilities	8h	T1, T2	18/11/2022	---
T8	Panel 6: Implement functional 2D barcode scanner	4h	T1, T2	02/12/2022	---
T9	Panel 7: Display reusable graphics capabilities	4h	T1, T2	16/12/2022	---
T10	Auto-generation of panels using a template engine	16h	T3, T4, T5, T6, T7, T8, T9	24/01/2023	---
T11	Creating unit tests to test all above panels	20h	T3, T4, T5, T6, T7, T8, T9	24/02/2023	---
T12	Write FYP Report	24h	All	24/03/2023	---

4. Conclusion

With this project I aim to provide an in-depth analysis on the features and abilities of FactoryTalk Optix. This analysis can then be examined by the relevant personnel within Medtronic to get a feel of what can and can't be done on this new software. This project will also provide me with extensive knowledge of the inner workings of the software which may be useful to me later down the line.

5. References

- [1] Information on FactoryTalk Optix - [FactoryTalk Optix | FactoryTalk \(rockwellautomation.com\)](https://rockwellautomation.com/factorytalk-optix)
- [2] Information on C# - [C Sharp \(programming language\) - Wikipedia](https://en.cppreference.com/w/cpp)
- [3] Information on NetLogic - [NetLogic \(asem.it\)](https://www.asem.it/netlogic)
- [4] Information on Git - [Git \(git-scm.com\)](https://git-scm.com/)
- [5] Information on GitHub - [GitHub - Wikipedia](https://github.com)
- [6] Information on FactoryTalk Optix - [Rockwell Automation Speeds Time to Market for New Industrial Automation Design, Launches FactoryTalk Design Hub | Business Wire](https://www.businesswire.com/news/home/20220301005001/en/FactoryTalk-Optix-Speeds-Time-to-Market-for-New-Industrial-Automation-Design-Launches-FactoryTalk-Design-Hub)
- [7] Information on Uniqo - [UNIQO Application Software - ASEM S.r.l. con Socio Unico \(asemautomation.com\)](https://www.asem.com/uniqo)
- [8] Information on Visual Studio 2022 - [Visual Studio 2022 Community Edition – Download Latest Free Version \(microsoft.com\)](https://visualstudio.microsoft.com/free-downloads/#vs2022community)
- [9] Information on StringTemplate 4 - [StringTemplate](https://stringtemplate.org/)
- [10] Information on Selenium WebDriver - [Selenium Webdriver Tutorial with Examples | BrowserStack](https://www.selenium.dev/documentation/en/webdriver/)
- [11] Information on Specflow - [SpecFlowWiki/FAQ.md at master · SpecFlowOSS/SpecFlowWiki \(github.com\)](https://github.com/specflow/SpecFlowWiki)