Anders Ingelsten 20095402

Higher Diploma in Science in Computer Science

KPI PROCESS IMprovement

Interim Report

Contents

[Summary 2](#_Toc125661815)

[Introduction 4](#_Toc125661816)

[Project Type 4](#_Toc125661817)

[The organisation 4](#_Toc125661818)

[System Background and Project Scope 4](#_Toc125661819)

[Smartsheet 5](#_Toc125661820)

[Fieldview 6](#_Toc125661821)

[Use Case 6](#_Toc125661822)

[Current Situation 7](#_Toc125661823)

[What is the problem. 7](#_Toc125661824)

[Potential Technologies, Tools, and Languages 8](#_Toc125661825)

[Potential Issues 8](#_Toc125661826)

[Objectives 8](#_Toc125661827)

[Methodology 9](#_Toc125661828)

[Dataflow 9](#_Toc125661829)

[Modelling 9](#_Toc125661830)

[Feedback Loop 9](#_Toc125661831)

[Technologies 10](#_Toc125661832)

[Sharepoint/Office365 10](#_Toc125661833)

[PowerBI 10](#_Toc125661834)

[SOAP API 10](#_Toc125661835)

[PowerShell 11](#_Toc125661836)

[Power Automate 11](#_Toc125661837)

[Potential Disadvantages 11](#_Toc125661838)

[Project Plan 11](#_Toc125661839)

[Project Planner 12](#_Toc125661840)

[Implementation 12](#_Toc125661841)

[Phase 1: Research & Training 12](#_Toc125661842)

[Phase 2: Smartsheet Data Connector setup 15](#_Toc125661843)

[Phase 3: FV API setup and Data Persistence 17](#_Toc125661844)

[Data persistence 19](#_Toc125661845)

[Phase 4: PowerBI Dashboard – Fieldview Vehicle-check 20](#_Toc125661846)

[Phase 5: PowerBI Dashboard – Fieldview Form Count 20](#_Toc125661847)

[Phase 6: PowerBI Dashboard – Fieldview Timesheet 20](#_Toc125661848)

[Phase 7: PowerBI Dashboard – SmartSheet Timesheet 20](#_Toc125661849)

[Project Evaluation 20](#_Toc125661850)

[Conclusion 20](#_Toc125661851)

[Reflection 20](#_Toc125661852)

[Challenges 20](#_Toc125661853)

[Future Work 20](#_Toc125661854)

[References 20](#_Toc125661855)

# Summary

Last Text to add

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# Introduction

## Project Type

This is a work-based project, where I will be learning new skills in relation to developing a KPI dashboard using the knowledge acquired from the course modules. The internal company mentor will be the Chief Financial Officer. This means that I will be able to spend time on this as part of my day-to-day work.

## The organisation

The company is an engineering solutions provider operating in Ireland, the UK and Scandinavia. It provides a wide range of services from the Design & Build of Sub-stations to construction of Airside Aviation Infrastructure to Turn-key Wind & Solar Energy Solutions. The company has a turnover of approx. 30million euro and employees approx. one hundred staff.

## System Background and Project Scope

Mainline Group has several systems that hold business information. Finance, Health and Safety and Operations staff query this data on a regular basis to produce business performance reports and to generate KPI (Key Performance Indicators) metrics for the different departments. The reports are generated on a weekly or monthly basis but there are also ad hoc reports.



*Figure 1: Overview of existing potential systems available to the project*

The current systems identified for the initial project scope are:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **System** | **Function** | **Location** | **Has API** | **Users** |
| **Fieldview** | Record QA/QC/Admin forms and Timesheet for site staff | Cloud | Yes - JXML | On-Site Staff / Operations |
| **Smartsheet** | Record timesheets for Salaried Staff | Cloud | Yes | All management & admin staff |
| **Office365** | SharePoint, Excel, word etc | MS Cloud | Yes | All -as per assigned rights |
|  |  |  |  |  |

Reworks and Expand a bit more in these

## Smartsheet

Smartsheet is an online hosted solution that allows organisations to plan, track, automate, and report on work. Mainline uses the online application to track timesheets for salaried staff.

Graphical user interface

Description automatically generated

Figure 2: Sample view of Smartsheet form

## Fieldview

Fieldview is a third-party cloud-based and off-line mobile solution developed by Trimble. It is used in Mainline to replace paperwork on site. Users are equipped with a mobile device (phone and tablet), where the app has been installed. The users log in and use the application for snagging tasks, forms & permits, project delivery and handover. When the mobile device is synced – data is pushed to the cloud hosted database.

Graphical user interface, text, application

Description automatically generated

Figure 3:Sample view of Fieldview mobile form

# Use Case

The primary use case for developing a dashboard pulling data from sources is the time saving. For example, the Fieldview application is structed in such a way that when a user, for example a project manager, would like to access the online portal, and view and pull-down information it can ONLY be done on a project-by-project basis. The systems do not have functionality in the user interface for a user to pull down for example all vehicle checks across all projects.

This means a user must access one project, access the report view, pull the information down and then access the next project and repeat the process until completed. Currently a user can create widgets that displays summaries but again, this must be done for every user on an individual basis.

A simple calculation will show how a dashboard can provide efficiencies.

Example, a user spends 1.5 hour every week to access every project area and compile a report of the amount of vehicles report submitted by staff. 1.5 hours x 48 working weeks = 72Hours which equates to 9 working days per year.

Chart

Description automatically generated

Figure 4:Sample view of the Fieldview widget reports

## Current Situation

The company has over the summer of 2022, revised the KPI process and in September rolled out a new KPI process where departmental stakeholders fill in excel spreadsheet reports with various KPI data. For example, hours spent by employee per contract and other data like work site accident frequency etc.

The departmental stakeholders access several systems and from there pull-down pdf and excel reports, from where they extract data and then collate it into excel spreadsheets which is then presented to the leadership team. This process is repeated monthly.

The company is aware that this is a time-consuming process and when its well established is looking to improve the efficiencies in this process. The company has already identified that business intelligence tools like Power BI can be used to display almost real-time dashboard views of the business data. During 2022 there has been internal talks about identifying and bringing in business intelligence consultants/developers to do this work during 2023. Within the project, I will investigate business intelligence products for example the Microsoft Power BI platform. It’s worth noting that the company has fully migrated to Office365/SharePoint during 2022 and is using the SharePoint platform as Intranet and repository for company data.

## What is the problem.

The two main problems are:

1. Manual input of data by staff into excel spreadsheets is time consuming and this time can be better spent on other business processes.
2. The time difference between the live situation and the compiled report by staff leads to delays in business understanding, for example if work is profitable or not.

The CFO of Mainline Group is therefore looking for a solution that can display KPI information from systems on the internal company SharePoint internet website.

# Potential Technologies, Tools, and Languages

As this is a work-based project the preferred direction from the leadership team is to use known technologies that are currently in use by the company. The two main reasons for this are that it creates greater resilience, and any costs are known, for example the company already host a Windows 2019 server in the Azure cloud. List of potential technologies, tools and languages:

* Office365
* SharePoint
* Power BI Desktop
* Power BI Data Gateway
* Power Query Editor
* SOAP UI
* MS Planner
* Html
* PowerShell
* Sharepoint Online Management Shell
* M language and DAX

Potential Issues

As the project will have access to commercially sensitive and potentially personal data, measures will be put in place to minimise exposure of this data. This may limit the scope of which systems and data will be incorporated in the project. If it’s not feasible to limit the exposure, test data will be utilised.

# Objectives

This project has three main objectives:

* **Ease of access.** By connecting separate data sets, transforming, and cleaning the data into a data model and create charts or graphs to provide visuals of the data; it will assist users to find insights, within the organisation, of the operational data generated in Fieldview.
* **Real-time information.** To give users in the organisation, an updated almost real-time view of the situation in the company. This should provide the ability to solve problems and identify issues and opportunities.
* **Process improvement.** By streamlining publication and the distribution of the data into dashboards – users interpret published data whenever the underlying dataset is updated. This is instead of compiling reports through a time-consuming process and sharing the data in emails or a shared drive for stakeholders to then review.

# Methodology

## Dataflow

Text about data flow and why



Figure 5: The Flow of Data from Fieldview to PowerBI

## Modelling

Text about the data model



Figure 6:Sample Representation of the data model in PowerBI

## Feedback Loop

The senior leadership team and the supervisor are very keen on the KPI Process improvement, outputs, results of visualisation the data and the automation of the process. Due to the company is in a rapid growth spurt, and several contracts won during 2022/2023. The direction was to use a feedback loop during the phased development process. Envisage steps taken during the feedback loop:

1. Initial requirement meeting with suggestion by the supervisor of the visualisation of data.
2. Analyse, develop and display data as per initial requirement
3. Follow up meeting to determine result fit expected requirements
4. Push out any changes or improvements of any feedback received

Repeat step 3 and 4 until decision to move to next visualisation feature.

## Technologies

The technologies used in the project was dictated by several limitation set by several factors. For example, the type of API used by Fieldview, scripting language/framework that can very efficiently and quickly generate data to PowerBI and any other technologies linked, researched or discovered during the project development process. The key drivers for deciding the scripting language were the Fieldview API, SharePoint Online and how quickly and cost effective the solution could be put into productivity. There was no direction from the internal supervisor in relation to choose made of technology used.

### Sharepoint/Office365

During the spring of 2022, the organisation fully had migrated its whole IT environment to the Sharepoint Online cloud, i.e., the full file repository of the organisation approx. 2.5 TB of files is hosted online with 24/7 365 access.

SharePoint Online is cloud-based software as a service provided by Microsoft, where organisation and users store and share information and use it collaboratively. All data and software are hosted on Microsoft own servers. Access, storage, and use of software is done by using a subscription model.

### PowerBI

PowerBI will be the tool to display dashboards of KPI views to the stakeholders of the organisation. Developed by Microsoft, PowerBI is an interactive data visualization software product. It is a collection of software services, apps, and connectors that work together to turn unrelated sources of data into coherent, visually immersive, and interactive insights. Data can be inputted/connected by reading directly from a database, webpage, or structured files such as spreadsheets, CSV, XML, and JSON.

### SOAP API

Fieldview uses SOAP API. SOAP is an acronym for Simple Object Access Protocol, which is a messaging protocol specification for exchanging structured information. And allows for implementation of web services. Normally it uses XML Information Set for its message format, and relies on application layer protocols i.e., Hypertext Transfer Protocol (HTTP). It is worth noting Fieldview uses Hypertext Transfer Protocol Secure (HTTPS) for its API.

SOAP is over two decades old and allows users to pull or push data from a range of operating systems as well as numerous clients to run web services and receive responses over a range of script language and platforms.

### PowerShell

During the research process several scripting languages was considers for the project for example Node JS, PowerShell or Python.

PowerShell was selected as it was deemed it would cause leased amount of impact on any existing system or servers and could very easily be transferred between machines in organisations IT environment and with a potential of running future API requests inside the SharePoint Online Management Shell.

PowerShell is a command-line shell and scripting language. Which supports variables, functions, branching (if-then-else), loops (while do, for, and foreach) and structured error/exception handling and closures/lambda expression.

### Power Automate

Text here.

### Potential Disadvantages

Text here.

# Project Plan

The milestones to achieve are outlined below.

|  |  |  |
| --- | --- | --- |
| **Milestone** | **Description** | **Due Date** |
| Draft Proposal | Initial project proposal and concept | 6th of November |
| Final Proposal | Articulate project nature and concept | 4th of December 2022 |
| Interim Report | Substantial update progress of the project | 12th of February 2023 |
| Final Project | Final project submission | 2nd  of April 2023 |

The project will span over the following development activities with preliminary dates

|  |  |
| --- | --- |
| **Phase** | **Preliminary Sprint Due Dates** |
| Research & Training | 8th January 2023 |
| Data Connector setup – Smartsheet | 15th of January 2023 |
| FV API setup and Data Persistence | 29th of January 2023 |
| PowerBI Dashboard – Fieldview Vehicle-check | 12th of February 2023 |
| PowerBI Dashboard – Fieldview Form Count | 25th of February 2023 |
| PowerBI Dashboard – Fieldview Timesheet | 19th of March 2023 |
| PowerBI Dashboard – SmartSheet Timesheet | 26th of March 2023 |
| Project Wrap Up | 2nd April 2023 |

# Project Planner

The Microsoft Planner tool was used for planning and execution of the project. In planner the user can divide development phases also knowns as sprints into buckets. Inside the buckets the user would then add tasks into it.

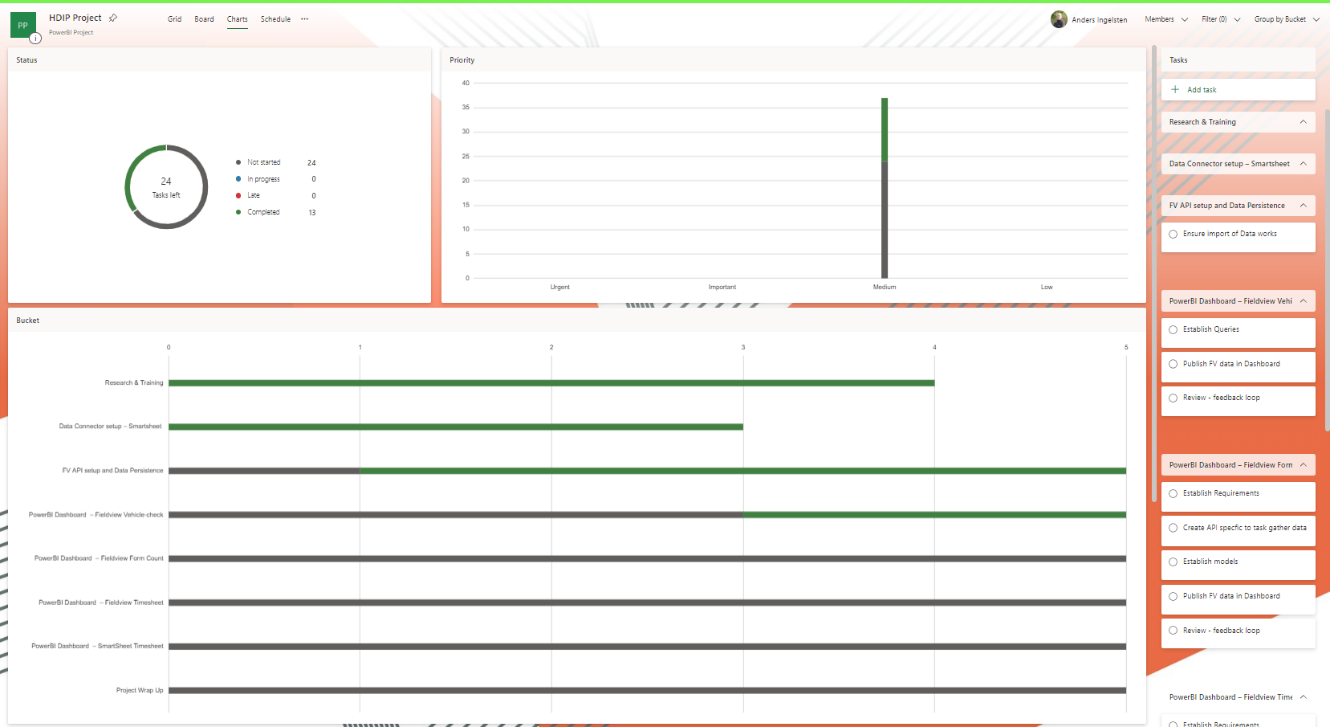


Figure 7: Microsoft Planner - Status of tasks and Buckets

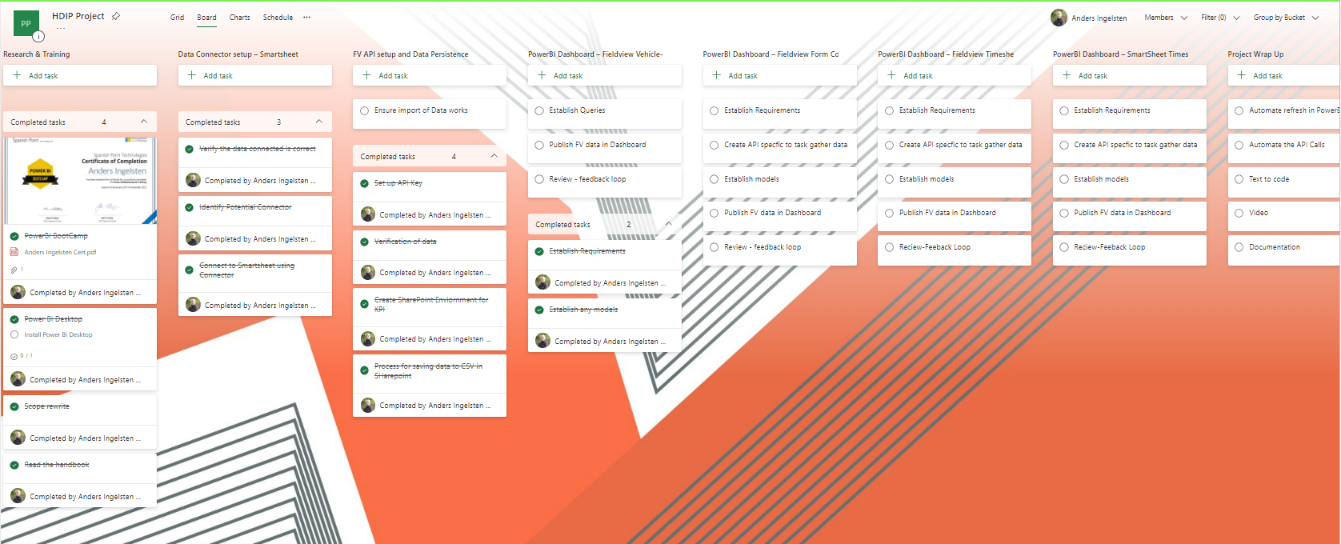


Figure 8: Microsoft Planner - Status of tasks and Buckets

# Implementation

# Phase 1: Research & Training

The phase of research and training covered three main areas.

**The Fieldview API.** The API Documentation of in the Viewpoint Help Section was investigated, as well as contact was made to the support desk of Trimble to get direction of how the API was structured. The research revealed the following.

Fieldview has 3 Data Centre API regions/URLs

1. UK
2. North America
3. Australia New Zeeland

The organisations data is stored in the UK region data centre.

Every Region has twelve APIs in two set of six different API. Grouped by XML or JSON, the APIs are:

* Configurations Services
* Forms Services
* Tasks Services
* Process Services
* Assets Services
* Project Services

2 APIs was explored for the purposes of the project

* Configurations Services – this API allowed me to identify and call project ids and associated information. For Example, the projectID is required to get Form Information
* Forms Services – this API allowed me to get form information, and individual answers.

Below follows 3 images of the formsservices API GetQuestionAnswer() command, first the parameters, a view of the SOAP API and then the returned answer information

Table

Description automatically generated

Figure 9: Sample of input Parameters

Graphical user interface, text, application, email

Description automatically generated

Figure 10: SOAP request and response

A picture containing table

Description automatically generated

Figure 11: Returned information

Time was spent understanding from which API’s information had to be pulled, stored, and passed on to other API’s. One important feature noted in the research period was the API Call quota. The call quota is the number of points an API token can spend within a minute. An API token in Fieldview has a maximum quota of 120 points allocated. This means you are limited to the number of calls that can be made per minute. For example, the GetProjectFormsList() has a point quota of 10, so this would means associated token to the API can only process 12 calls in one minute.

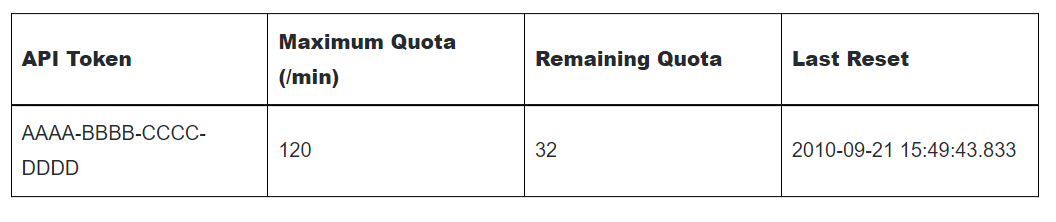


Figure 12: Call quota per API token

The research also showed that the Fieldview API would not easily return any data using the built-in data connectors in PowerBI and that the route of developing scripting of an API had to be taken. Which is in line with the purpose of the project i.e., to showcase skilled learned from the course.

It’s worth noting that there exist third party apps like ZappySys ODBC Power Pack do integrate SOAP API with PowerBI. However, this not a free software and annual subscription is approx. $650 per desktop install. Source [Calling SOAP API in Power BI (Read XML Web Service data) | ZappySys Blog](https://zappysys.com/blog/call-soap-api-power-bi-read-xml-web-service-data/)

**PowerBI.** To get an understanding of how PowerBI capabilities and its functionality I signed up for Spanish Point Technologies “Dashboard in a day” course. Spanish Point Technologies is a software company and a Gold Certified Microsoft partner. Spanish Point specialises in Azure, Microsoft 365, SharePoint, Dynamics 365, PowerApps & Power Automate, Power BI and SQL Server solutions. Mainline has over the years used Spanish Point for various software solutions.

Dashboard in a day is a free full day workshop covering the capabilities of PowerBI through an instructor led online course.

* The goal of the course is to better understand how to:
* Connect to, import, and transform data from a variety of sources
* Define business rules and KPIs
* Explore data with powerful visualization tools
* Build stunning reports
* Share dashboards with their team and business partners, and publish them to the web

Add source. [Spanish Point Technologies Ltd.](https://www.spanishpoint.ie/)

The power BI Bootcamp and training was completed on the 23rd of November 2022.

A picture containing text

Description automatically generated

Figure 13: Certificate of Completion

In addition to above training, I also conducted various research online to find out how to do certain various items, anything from PowerShell Scripting to SharePoint Online solutions and Power BI queries.

# Phase 2: Smartsheet Data Connector setup

Smartsheet provides a Data Connector, which is part of PowerBI built in connectors.

Following steps were completed in this Phase

1. Set up of user credentials in SmartSheet
2. Ensure user had access to relevant tables in SmartSheet
3. Connect Smartsheet to PowerBI using the built-in get Data feature in Power BI
4. Authenticate connection in PowerBI with Smartsheet user credentials.
5. Load and transform the relevant table from Smartsheet into PowerBI

Graphical user interface, application

Description automatically generated

Figure 14: Get Data view in PowerBI of Online Services

Graphical user interface, application, table

Description automatically generated

Figure 15: Preview of Connected Table ready to be loaded into PowerBI

# Phase 3: FV API setup and Data Persistence

The focus of this phase was to make sure to be able to reliably pull information from the Fieldview API to the organisation’s hosted Data repository i.e., SharePoint Online.

The first thing I did was to generate API keys inside Fieldview, and API key can be set on Group Organisation and Company level. A Company is a child of an Organisation. All keys were set up on Group Level.

Graphical user interface, text, application

Description automatically generated

Figure 16:View of Created API Tokens in Fieldview

The SOAP protocol does not have to have any readily available plugins and add-on and the SOAP API relies on discreet calls for every interaction with the host. These calls also need to be structured inside a SOAP API envelope. I needed to be able to use a scripting language compatible with SharePoint online and which has the capabilities to execute general programming tasks like for each loop and if statements.

By utilising PowerShell and its module New-WebServiceProxy I managed to achieve this. The PowerShell New-WebServiceProxy will download the API’s WSDL and use it to generate types for the proxy’s interface, data contracts and headers. PowerShell also allows you to import variables from a file and export variables to csv files. Tasks can then be scheduled to run at certain intervals, but this will be brought up in a later phase. Source [New-WebServiceProxy (Microsoft.PowerShell.Management) - PowerShell | Microsoft Learn](https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.management/new-webserviceproxy?view=powershell-5.1)

Text

Description automatically generated

Figure 17: Code Snippet of the New-WebServiceProxy used to pull data from Fieldview

Because the New-WebServiceProxy sets up a proxy object, it allowed me to interact with the Fieldview SOAP API and by utilising PowerShell capabilities of running foreach loops, if statements, I could connect store all the project ids, and then loop them back into the next call which would in this phase retrieve form information. Due every API Token has a call quota per minute I generated a few API keys and added a delay in the loop, not to exceed the call quota.

Text

Description automatically generated

Figure 18: Code snippet of foreach loop with an if else statement

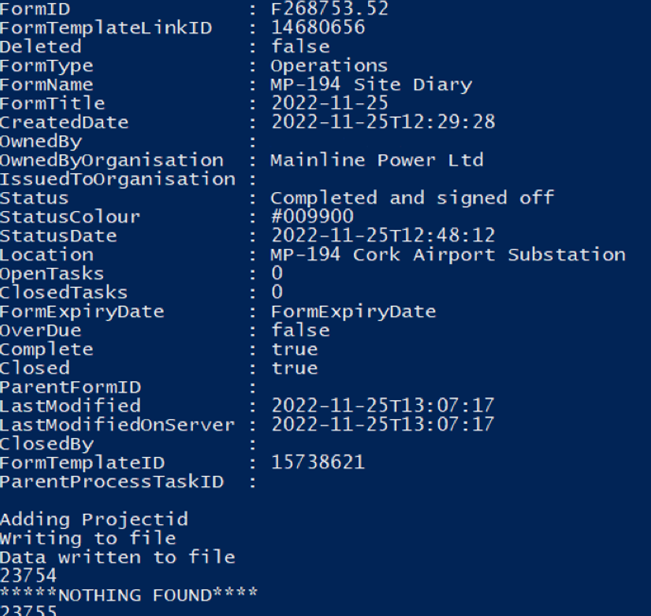


Figure 20: Screenshot of Sample Data from the GetProjectFormsList( ) call, returning ProjectFormsListInformation.

### Data persistence

The main factor that decided which type of method for data persistence was going to be used in the project was availability and cost. The organisation has access to hosted servers but none of these are operational 24 hours a day, 7 days a week.

Increasing availability would lead to increased cost and for the data to be persistent it must write to non-volatile storage. Data persistence was achieved by saving the data to a file, after duplications was removed, it was then exported to Sharepoint Online. By saving a file with the same name to the same location, SharePoint online just saves a new version of the file. A CSV file stored in SharePoint Online allows very easy connection to PowerBI by its built in Web connector.

Source: [How to Connect to SharePoint Online using PnP PowerShell? - SharePoint Diary](https://www.sharepointdiary.com/2018/03/connect-to-sharepoint-online-using-pnp-powershell.html)

Text

Description automatically generated

Figure 21: Sample Code of writing a file to Sharepoint Online

Text

Description automatically generated

Figure 22: Screenshot of successfully writing the data to Sharepoint Online in PowerShell

# 

# Phase 4: PowerBI Dashboard – Fieldview Vehicle-check

Text here

# Phase 5: PowerBI Dashboard – Fieldview Form Count

Text here

# Phase 6: PowerBI Dashboard – Fieldview Timesheet

Text here

# Phase 7: PowerBI Dashboard – SmartSheet Timesheet

Text here

# Project Evaluation

### Conclusion

### Reflection

Table

Description automatically generated

### Challenges

### Future Work

SQL?

Chainage control per project

PDF files and images

# References

Mainline

[Mainline Group - Projects that Matter](https://mainline.ie/)

Fieldview

[Field View (priority1.uk.net)](https://www.priority1.uk.net/FieldView/default.aspx)