P_Fun Covid

Lucas Lordon – MID3
Vennes – B22
Xavier Carrel

Table des matières

1 Analyse préliminaire

1.1 Introduction

Le projet « P_Fun Covid » à été réaliser par Lucas Lordon dans le cadre du module de Programmation Fonctionnel I323. Ce projet à pour objectif d'afficher des graphiques à partir d'un jeu de données choisi. Les technologies utiliser sont libres du moment que l'on code en C# dans l'IDE Visual Studio 2024.

Personnellement j'ai créé un projet Windows Forms reposant sur le package nugget « ScottPlot Form ». Le jeu de donnée choisit vient de ce repo GitHub : https et concerne différente information collecter durant le covid par canton Suisse.

1.2 Objectifs

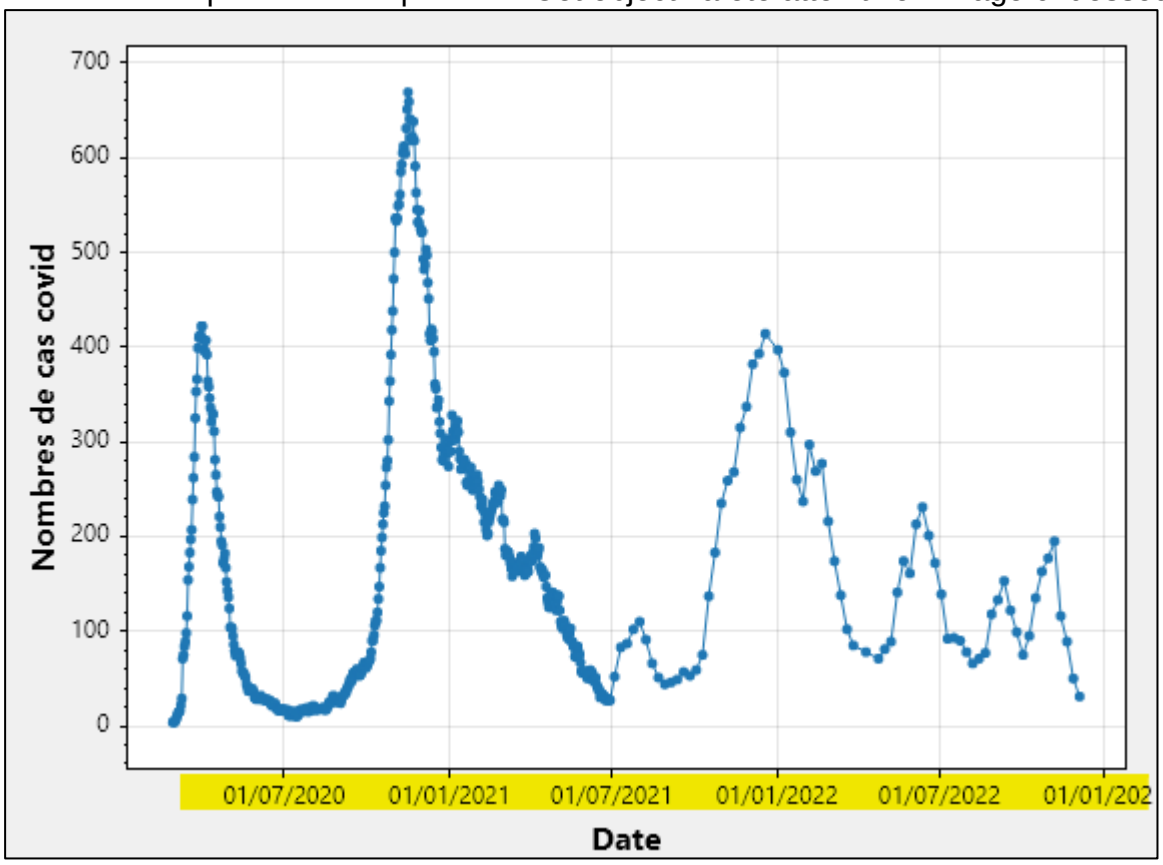
Filtre sur les dates :

Cet objectif a été atteint car je peux filtrer les dates grâce à deux élément « DateTimePicker » et à une méthode utilisant Linq.

Au moins 3 courbes :

Cet objectif a été atteint car par défaut je peux afficher entre 1 et 26 courbes simultanément via les fichier CSV disponible sous le répertoire suivant : P_Fun\Code\Data

L'axe X correspond à la temporalité : Cet objectif a été atteint voir image ci-dessou :



1.3 Gestion de projet

La gestion de projet c'est effectuer via deux outils : IceScrum et un jounal de travail Excel. IceScrum pour la partie planification et validation du travaille effectuer. Le journal de travail pour avoir un suivit sur le travail effectuer.

Il est important de noter que la méthodologie utiliser est l'agilité (ce référer au Scrum guide) et que le chef de projet a été l'enseignant X.Carrel.

2 Analyse / Conception

2.1 Domaine

J'ai choisi le domaine des maladies et plus précisément du covid 19 en suisse. Les données sont un ensemble d'information tels que le nombre d'hospitalisation ou bien le nombre de mort journalier, ces données sont récoltées par chaque canton et enregistrer dans différents fichiers CSV.

Echelle de temps est très variable selon les maladies qui pourrait être étudier mais dans le cas du covid 19 les données commencent en 2019 et finissent en 2024 (Attention : selon le canton les date peuvent différée légèrement).

Ces données sont principalement destinées à des professionnelles du monde de la santé, et aux analystes. Mais l'on peut très bien imaginer d'autres cas d'utilisation comme des journaliste voulant appuies leur propos au moyen de graphique. L'on peut aussi imaginer que des professionnelles du domaine de la finance si intéresse dans le but d'expliquer la crise économique durant la période de pandémie.

2.2 Concept

Diagramme de classe :

L’image est dicponible sous : P_Fun\Doc\diagrameClasse.png

2.3 Analyse fonctionnelle

Find Data
(Auteur: Lucas Lordon)

As a developer I want to find data about swiss electricity consumption In order to start the development of the application
Tests d'acceptance:
Verifying the Data Format and Usability — Given that I have found a potential data source, When I access the data, Then the data should be in a usable format (e.g., CSV, JSON, or accessible via API), And the data should include relevant information such as date/time stamps and electricity consumption values.
Ensuring the Data Source Has a Free License — Given that I have identified a data source When I review the licensing terms or usage policy associated with the data, Then the data should be explicitly stated as being available under a free license, And the license should allow for unrestricted access and use of the data for the development,

create a basics wpf app using scottplot
(Auteur: Lucas Lordon)

As a developer, I want to create a WPF application using ScottPlot so that I can display interactive data visualizations.

Tests d'acceptance:	
Setting Up the WPF Project	Given that I have Visual Studio installed on my machine, When I create a new WPF project in Visual Studio, Then the project should be created successfully,
Installing ScottPlot via NuGet	Given that I want to use ScottPlot for data visualization, When I search for the ScottPlot package in the NuGet Package Manager, Then the ScottPlot library should be found and available for installation, And the installation should complete successfully without any errors.
Adding a ScottPlot Control to the Main Window	Given that I have installed ScottPlot in my WPF project, When I add a ScottPlot control to the MainWindow XAML file, Then the control should be visible in the designer view, And the application should compile and run without errors.
Plotting Basic Data on the ScottPlot Control	Given that the ScottPlot control is added to the WPF application, When I write code to plot basic data on the WpfPlot control, Then the data should be displayed correctly on the graph, And the graph should be interactive (e.g., allowing zoom).

import real data
(Auteur: Lucas Lordon)

As a developer, I want to import real data from a CSV file in order to stock them in a class.	
Tests d'acceptance:	
Import a valid CSV file with correct data	Given a CSV file named data.csv containing valid data in the required format (e.g., columns for Name, Age, and Email) When the developer runs the CSV import function Then the data from the CSV file should be successfully parsed And each row of the CSV should be stored as an instance of the class (DataClass) And each class instance should correctly store the respective data fields

spread real data on the graph
(Auteur: Lucas Lordon)

As a developer, I want to spread real data on a graph in order to see and manipulate them.	
Tests d'acceptance:	
Display real data on a graph	Given real data is available When the developer runs the graph plotting function Then the data should be correctly displayed on a graph And the axes should be properly labeled based on the data And the graph should scale dynamically based on the dataset size

see the data (X,Y) value where the cursor is on the graph
(Auteur: Lucas Lordon)

As a developer, I want to see the data (X, Y) values where the cursor is on the graph in order to inspect specific points easily.	
Tests d'acceptance:	
Display (X, Y) values on hover	Given a graph displaying real data points When the developer hovers the cursor over a data point on the graph Then the exact (X, Y) values of the nearest data point should be displayed in a tooltip or an overlay

	next to the cursor And the displayed values should be accurate and update dynamically as the cursor moves across different points

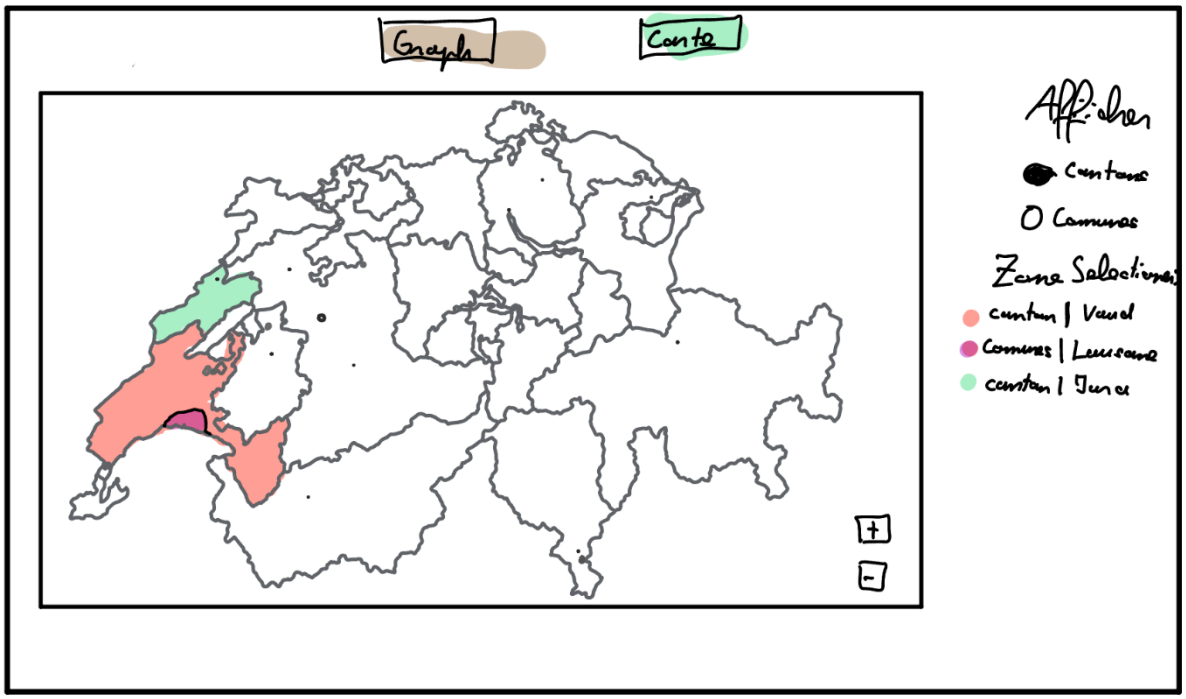
Add Custom Data File for COVID-19 Data Analysis
(Auteur: Lucas Lordon)

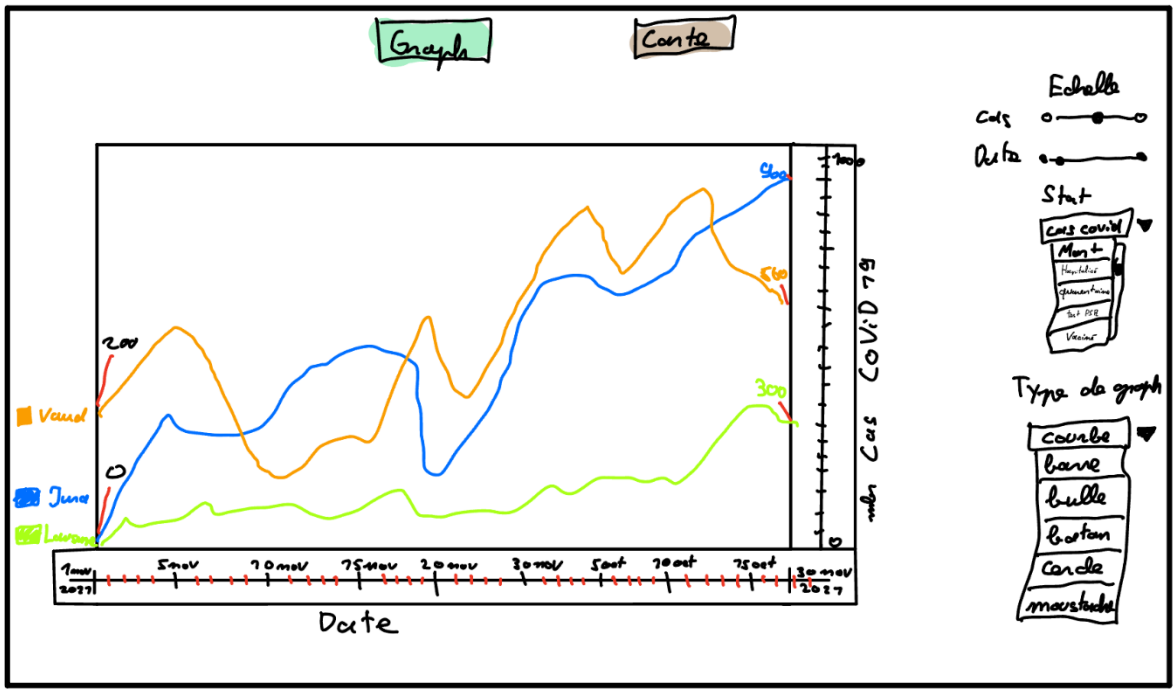
As a user, I want to add my own COVID-19 data file to the application and label it with a custom name in order to view and analyze my data alongside the existing dataset.	
Tests d'acceptance:	
The user can add a new data file by selecting it through a file dialog.	Given the user has selected a valid CSV file, When they confirm the file and enter a custom name, Then the file path should be added to the cantonPaths dictionary with the custom name as the key.
The user must provide a unique custom name for the new data file.	Given the user enters a custom name that does not already exist in cantonPaths, When they confirm the name, Then the application should accept and store the name.
The application verifies that the selected file has the required syntax before adding it.	Given the user selects a file to add, When the application checks the file syntax, Then the application should validate that the file has a similar structure to the existing data (same columns or format). If the file does not match, an error message should be shown to the user.
The system should prevent duplicate names in the dictionary.	Given the user enters a name that already exists in cantonPaths, When they confirm the name, Then the application should display a message prompting the user to enter a unique name.
If the file syntax is incorrect, the user should be notified and given the option to select another file.	Given the user selects a file with an incorrect syntax, When the system validates the file format, Then the application should display an error message explaining that the file syntax is incorrect and ask the user to select a different file.

Data Limitation Feature
(Auteur: Lucas Lordon)

As a user, I want to enable a data limitation feature with a start and end date selection, in order to view a specific time range of COVID data in the graph for better focus and analysis on that period.	
Tests d'acceptance:	
Data Limitation Checkbox Activation	Given the data limitation checkbox is unchecked, When I check the "Activate Data Limitation" checkbox, and i press show data Then the start and end date pickers should set a custom date range on the graph.
Deactivation of Data Limitation	Given the "Activate Data Limitation" checkbox is checked and a date range is selected, When I uncheck the "Activate Data Limitation" checkbox, and i press show data Then the date limitation should be disabled, and the graph should revert to displaying the full dataset without restriction.

Persistent Display on Valid Data Range Update	Given the data limitation feature is enabled with a previously set valid date range, When I modify either the start date or the end date within the allowed range, Then the graph should automatically refresh to display the updated range without requiring additional action.





Les maquettes sont disponibles sous : P_Fun\Doc\Maquettes

2.4 Stratégie de test

La stratégie de test repose sur plusieurs points essentiels pouvant être séparer en trois grand bloque : Planification, Code et Utilisateur.

Planification : durant cette partie des tests l'on crée des user story et l'on y implémente des tests d'acceptances. C'est une partie très importante où l'on confronte la fonctionnalité avant même de la coder. Cela permet de gagner beaucoup de temps sur le long therme en évitant de devoir foncer dans le code tête baisser pour se rendre compte que l'on code quelque chose qui ne nous a pas été demander.

La deuxième étape est la mise en place de test unitaire dans le code.

Et la dernière est le test de mon code par mes camarade (User Testing) cette étape me permet d'avoir des retours sur l'UI/UX et de découvrir de potentiel bug exemple : lien nom responsive vers mes fichiers de données.

3 Réalisation

Section ou les fonctionnements particulier de m'on code sont détailler

3.1 Points de design spécifiques

Dans chacun de ces point je vais expliquer quelque fonctionnalité importantes du code le titre est la référence où vous pouvez retrouver le code.

3.1.1 Fichier Graph.cs méthode InitializeCheckboxes et onShowDataButtonClick

Ce code génère dynamiquement des cases à cocher (checkboxes) et des boutons dans un panneau de contrôle, qui sont ensuite affichés dans un composant FlowLayoutPanel intégré dans cantonSelectionPanel. Voici une explication étape par étape du processus de génération des cases à cocher et de leur disposition dans différents panels :

1) Création du panneau de flux : La méthode commence par créer un FlowLayoutPanel pour disposer les contrôles de manière dynamique. Ce panneau utilise le DockStyle.Fill pour occuper tout l'espace disponible, et AutoScroll est activé pour permettre le défilement lorsque le contenu dépasse la taille du panneau.

2) Création des cases à cocher : Pour chaque canton répertorié dans le dictionnaire cantonPaths (les noms des cantons et leurs chemins

respectifs), la méthode crée une case à cocher (CheckBox). Chaque case a les propriétés suivantes :

Texte : défini par le nom du canton.
État coché : défini en true uniquement pour le canton "VD".

3) Ajout de boutons associés : Un bouton See more est créé pour chaque canton. Ce bouton est associé à un chemin spécifique pour chaque canton à l'aide de l'événement Click. Lorsque le bouton est cliqué, il exécute l'événement SeeDataButtonClick avec le chemin correspondant au canton sélectionné. Cela permet d'afficher des détails spécifiques au canton sélectionné.

4) Ajout des contrôles au panneau de flux : La méthode ajoute chaque case à cocher et chaque bouton au FlowLayoutPanel. Ce panneau assure l'agencement fluide des contrôles en les disposant les uns après les autres horizontalement ou verticalement, selon l'espace disponible.

5) Ajout au panneau principal : Enfin, le FlowLayoutPanel contenant les cases à cocher et les boutons est ajouté au cantonSelectionPanel principal, permettant une visualisation structurée de toutes les options de canton dans l'interface.

Exemple de structure de rendu
Pour chaque canton, l'utilisateur voit une case à cocher et un bouton côte à côte, ce qui rend l'interface plus intuitive pour sélectionner un canton ou afficher plus d'informations. Cette approche permet de générer et de mettre à jour l'interface de manière dynamique en fonction des cantons disponibles dans cantonPaths.

3.1.2 Fichier ImportData.cs classe ImportData et méthode ImportCsvData

Ce code lit un fichier CSV et importe les données dans une liste d'objets CovidData. Voici les détails de la structure et du fonctionnement de chaque partie du code :

Classe ImportData et méthode ImportCsvData : La classe ImportData contient une méthode ImportCsvData qui prend le chemin d'un fichier CSV (csvFilePath) comme paramètre et renvoie une liste de CovidData. Cette méthode utilise la bibliothèque CsvHelper pour faciliter l'importation et le traitement des données CSV.

2) Configuration et ouverture du fichier CSV :

Flux de lecture : La méthode commence par créer un StreamReader pour lire le fichier CSV spécifié.
Configuration de CsvHelper : Un objet CsvReader est créé avec une configuration CsvConfiguration pour adapter le format des données. Le séparateur est défini en tant que virgule (,), et les options MissingFieldFound et HeaderValidated sont définies sur null pour ignorer les erreurs de champ manquant et les erreurs de validation d'en-tête.

Lecture des enregistrements : Les enregistrements du fichier CSV sont lus et convertis directement en une liste d'objets CovidData à l'aide de csv.GetRecords<CovidData>(). La méthode GetRecords<T>() permet de mapper les colonnes du fichier CSV aux propriétés de la classe CovidData en fonction des noms des colonnes et des propriétés.

2) Gestion des exceptions : Si une erreur survient lors de la lecture du fichier (par exemple, si le fichier est introuvable ou mal formaté), elle est capturée et un message d'erreur est affiché dans la console pour aider à diagnostiquer le problème.

Classe CovidData : La classe CovidData, définie dans l'espace de noms P_Fun_Forms.Class, représente la structure des données COVID importées depuis le fichier CSV. Elle contient plusieurs propriétés correspondant aux colonnes du fichier :

Propriétés de données : Ces propriétés, comme date, abbreviation_canton_and_fl, et ncumul_conf, représentent les différents attributs de données COVID. Les types sont adaptés pour correspondre aux données attendues : DateTime pour la date, int? pour les valeurs numériques nullables, et string pour les champs texte.
Retour de la liste : Une fois les enregistrements chargés, la méthode retourne la liste records contenant tous les objets CovidData importés.

Exemple d'utilisation
Ce code permet de charger rapidement un fichier CSV contenant des données COVID, de les convertir en objets CovidData, puis d'effectuer des analyses ou des affichages en fonction de ces données.

3.2 Déroulement

Afin de réaliser mes user story je me suis aidé de Deepl et de ChatGPT

3.3 Mise en place de l’environnement de travail

La solution du projet ce trouve sous :
P_Fun\Code\App\P_Fun_Forms\P_Fun_Forms.sln

Pour démarrer l’application il suffit de cliquer sur démarrer en haut de l’écrant.

Les fichiers importants du code sont :

AddDataFile.cs : C’est une page forms qui permet d’ajouter des fichier de données

formSeeData.cs : C’est une page forms ou l’on voit dans un tableau les donnée de manière plus détailler

Graph.cs : C’est une page forms qui est cœur du projet c’est sur cette page que l’on affiche le graph et c’est courbe ainsi que le reste des fonctionnalité majeur tels que le filtrage par date.

CovidData : Class CovidData qui contient les header par default présent dans les fichier.

ImportData.cs : Class permettant d'importer les fichier csv via un Délimiter et le package CsvHelper

3.4 Description des tests effectués

4 Sprint 1

5 spread real data on the graph

Display real data on a graph	Given real data is available When the developer runs the graph plotting function Then the data should be correctly displayed on a graph And the axes should be properly labeled based on the data And the graph should scale dynamically based on the dataset size	OK 27 Sep

6 see the data (X,Y) value where the cursor is on the graph

Display (X, Y) values on hover	Given a graph displaying real data points When the developer hovers the cursor over a data point on the graph Then the exact (X, Y) values of the nearest data point should be displayed in a tooltip or an overlay next to the cursor And the displayed values should be accurate and update dynamically as the cursor moves across different points	OK 27 Sep

7 import real data

Import a valid CSV file with correct data	Given a CSV file named data.csv containing valid data in the required format (e.g., columns for Name, Age, and Email) When the developer runs the CSV import function Then the data from the CSV file should be successfully parsed And each row of the CSV should be stored as an instance of the class (DataClass) And each class instance should correctly store the respective data fields	OK 27 Sep

8 Find Data

Verifying the Data Format and Usability	Given that I have found a potential data source, When I access the data, Then the data should be in a usable format (e.g., CSV, JSON, or accessible via API), And the data should include relevant information such as date/time stamps and electricity consumption values.	OK 30 Aug
Ensuring the Data Source Has a Free License	Given that I have identified a data source When I review the licensing terms or usage policy associated with the data, Then the data should be explicitly stated as being available under a free license, And the license should allow for unrestricted access and use of the data for the development,	OK 30 Aug

9 create a basics wpf app using scottplot

Setting Up the WPF Project	Given that I have Visual Studio installed on my machine, When I create a new WPF project in Visual Studio, Then the project should be created successfully,	OK 27 Sep
Installing ScottPlot via NuGet	Given that I want to use ScottPlot for data visualization, When I search for the ScottPlot package in the NuGet Package Manager, Then the ScottPlot library should be found and	OK 27 Sep

	available for installation, And the installation should complete successfully without any errors.	
Adding a ScottPlot Control to the Main Window	Given that I have installed ScottPlot in my WPF project, When I add a ScottPlot control to the MainWindow XAML file, Then the control should be visible in the designer view, And the application should compile and run without errors.	OK 27 Sep
Plotting Basic Data on the ScottPlot Control	Given that the ScottPlot control is added to the WPF application, When I write code to plot basic data on the WpfPlot control, Then the data should be displayed correctly on the graph, And the graph should be interactive (e.g., allowing zoom).	OK 27 Sep

10 Data Limitation Feature

Data Limitation Checkbox Activation	Given the data limitation checkbox is unchecked, When I check the "Activate Data Limitation" checkbox, and i press show data Then the start and end date pickers should set a custom date range on the graph.	OK 1 Nov
Deactivation of Data Limitation	Given the "Activate Data Limitation" checkbox is checked and a date range is selected, When I uncheck the "Activate Data Limitation" checkbox, and i press show data Then the date limitation should be disabled, and the graph should revert to displaying the full dataset without restriction.	OK 1 Nov
Persistent Display on Valid Data Range Update	Given the data limitation feature is enabled with a previously set valid date range, When I modify either the start date or the end date within the allowed range, Then the graph should automatically refresh to display the updated range without requiring additional action.	OK 1 Nov

11 Add Custom Data File for COVID-19 Data Analysis

The user can add a new data file by selecting it through a file dialog.	Given the user has selected a valid CSV file, When they confirm the file and enter a custom name, Then the file path should be added to the cantonPaths dictionary with the custom name as the key.	OK 30 Oct
The user must provide a unique custom name for the new data file.	Given the user enters a custom name that does not already exist in cantonPaths, When they confirm the name, Then the application should accept and store the name.	OK 30 Oct
The application verifies that the selected file has the required syntax before adding it.	Given the user selects a file to add, When the application checks the file syntax, Then the application should validate that the file has a similar structure to the existing data (same columns or format). If the file does not match, an error message should be shown to the user.	OK 30 Oct
The system should prevent duplicate names in the dictionary.	Given the user enters a name that already exists in cantonPaths, When they confirm the name, Then the application should display a message prompting the user to enter a unique name.	OK 30 Oct
If the file syntax is incorrect, the user should be notified and	Given the user selects a file with an incorrect syntax, When the system validates the file format, Then the application should display an error message explaining	OK 30 Oct

given the option to select another file.	that the file syntax is incorrect and ask the user to select a different file.	

12 spread multiple courb on the graph from different canton

Display multiple curves for each canton on a single graph	Given real data from multiple cantons When the developer runs the graph plotting function for multiple datasets Then each canton's data should be selectable in a checkbox that the data as a distinct curve on the same graph And each curve should have a unique color or style for clear differentiation And a legend should be displayed, indicating which curve belongs to which canton	OK 1 Nov

12.1 Erreurs restantes

Possible erreur de responsive selon la tailler de l'écran, exemple : bouton pas aligner avec les checks box des canton.

13 Conclusions

En conclusion je pense avoir acquis le projet selon les exigences demander mais selon mes propre exigence je ne pense pas l'avoir acquis car j'aurais voulu rajouter d'autre fonctionnalité tels que la maps durant les vacance mais étant pris par d'autres projets personnelles plus urgent je n'ai pas pu l'implémenter.

Les points positifs de ce projet ont été selon moi la grande liberté créative de ce projet, une évaluation des 80% claire et l'utilisation de GitHub approfondit via les normes de commit.

Selon moi il n'y pas de point négatif de pars les libertés données quant à notre façon de travailler. Bien sur je n'aime pas faire de la doc mais je comprends qu'il est nécessaire dans notre apprentissage de s'entrainer à réaliser de la documentation donc je ne peux pas le compter comme négatif. Le seul véritable défaut serait l'axe du projet qui je trouve après réalisation n'était pas idéal pour l'apprentissage de la programmation fonctionnelle.

Ma grosse difficulté à été ce bug Scott Plot qui a durée sur presque 4 séquences et qui ma donc fait perdre un temps phénoménal.

Mon projet pourrait avoir des suites presque illimitées allant de la petite retouche jusqu'à la refonte total en vue d'une commercialisation. Ce si dis voici quelques idées :

- Ajout de la Maps
- Refonte de l'UI et de l'UX
- Ajout du responsive
- Trie par catégorie
- Ne plus se limitée à une seule maladie
- Ne plus se limitée à un seul pays

- Version Web et mobile
- Importation de donnée facilitée
- Possibilité d’ouvrir plusieurs graphs en même temps
- Export du graph en PDF

14 Annexes

14.1 Journal de travail

Disponible sous : "C:\GitHub\P_Fun\Doc\jnltrav_P_FUN.xlsm"