CA400 Final Year Computer Applications Project

Functional Specification

‘Comparing the accuracy and affect a number of different gap-filling techniques has on meteorological datasets’

Mark Reid: 19414892

Kevin Boyle: 19731615

Submission Data: 18/11/2022

1. Introduction

1.1 Overview:

The presence of missing data is a challenging issue in processing real-world datasets. The aim of our fourth-year project is to compare the effects several techniques of gap-filling has on a number of different weather/meteorological datasets. Each of these datasets will contain *time-series* data, which is a collection of observations obtained through repeated measurements over time.

Time-series data is commonly used within weather stations to record weather information, however due to instrument failures or unfavorable conditions (e.g. clouds or vegetation thickness in case of remote sensing data), the time-series data may contain gaps. We intend on creating our own gaps inside each of the datasets using several scientific methods. These include: randomly distributed gaps, gaps that are spatially and temporally correlated or clustered.

These gaps often prove to be a significant problem for time-series forecasting as the majority of forecasting models cannot provide accurate predictions while there are gaps present in the data. Missing values are problematic as they can induce analytical bias which can cause the model to underestimate or overestimate the result. They may also inhibit statistical analysis.We believe that this is applicable today as gap filling is a necessary task whilst working with real-world datasets containing time-series data.

After we have developed each of our gap insertion techniques, we will then focus on analysing which of the techniques is most effective. We plan on computing an estimate of the unknown data points using techniques such as *interpolation* and comparing these values against the real data, testing the accuracy of each method. We will then measure its accuracy against other techniques.

1.2 Business Context:

As our target user is a climate analyst, we feel there is an opportunity for business with this system. Climate analysis is a booming industry given the current circumstances. As gapfilling is an undesired but necessary task in Earth System sciences, we believe that there is a potential for our findings to be used by employees/organisations working in Earth System science industry. Likewise, there also could be business context with public organisations such as Met Eireann here in Ireland, who are constantly working with massive meteorological datasets and have to deal with gaps in their data quite often.

1.3 Glossary:

1. ***Time-series data***: a collection of observations obtained through repeated measurements over time. If you were to plot time-series data, one axis will always be time and the other axis will be another value.
2. **Linear Interpolation**
3. **Forward Interpolation**
4. **Extrapolation** (Backward Filling Interpolation)
5. **Mean Average Method**
6. **Singular Spectrum Analysis**

2. General Description

This project aims to analyse the efficiency of several gap-filling techniques on a meteorological dataset. The gaps will be inserted using several scientific methods to imitate real world events where measuring-instruments have failed or unfavorable conditions occur cause time-series fragments in our dataset.

2.1 System Functions:

- **Choose & Analyse Dataset:**

Using a simple graphical interface featuring a drop down menu, users will have the ability to select a dataset from a choice of ten meteorological datasets. They then will be able to view the dataset before it has been fragmented with gaps. The ten datasets will be divided by the location of the weather station the information was gather from.

- **Pick Gap Insertion Technique:**

After the user has finished analysing the dataset before any gaps have been inserted, they can then choose from a list of scientific gap-insertion techniques. Ideally, a user would select the most appropriate technique in relation to the gaps present in their own dataset. After the user selects a technique from the following list: *randomly distributed gaps, gaps that are spatially and temporally correlated or clustered, scenario where time series completely missing and a mix of these gap structures*. They will then be presented with the same dataset with gaps throughout.

- **Pick Gap-filling Technique**

After choosing which gap-insertion method to be used, we will then provide users with a simple and effective way of gap-filling their datasets. Our system will incorporate five different methods of gap filling. They are listed below with a general description of each:

1. Linear Interpolation
2. Foward Interpolation
3. Extrapolation (Backward Filling Interpolation)
4. Mean Average Method
5. Singular Spectrum Analysis

- **Present results and comparison between the other gap-filling techniques:**

We will then focus on analysing which of the techniques is most effective. We plan on computing an estimate of the unknown data points using techniques such as *interpolation* and comparing these values against the real data, testing the accuracy of each method. We will then measure its accuracy against other techniques *The following is a list of the metrics in which we will use to test the accuracy of each of the gap-filling techniques*:

- *RMSE (Root Mean Squared)* - This metric tells us how accurate our predictions are and, what is the amount of deviation from the actual values.
- *Coefficient of Determination* - Used to explain how much variability of one factor can be caused by its relationship to another factor.

- *Mean Absolute Error* - a measure of errors between paired observations expressing the same phenomenon, in our case our predicted gaps vs real values.
- *Mean Squared Error -* using the average squared difference between observed and predicted values.

2.2 User Characteristics & Objectives:

The community of users we expect to make use of the findings from our research project and system are, as we mentioned before, climate scientist's and organisations who collect/work with large amounts of real-world data such as *Met Eireann*. Climate scientists working at these organisations are constantly working with datasets containing time-series data which commonly contain null values/gaps within data points.

A user can expect to be provided with a system that allows them to view a dataset which contains gaps throughout. They can then analyse the dataset before any method of gap-filling has taken place. Once they are happy with the dataset, the user can select one of the gap filling techniques available to use. We will provide a detailed explanation of why one method of gap-filling is more effective than another in specific scenarios. Once they have selected a method, we will then return their dataset with gap-filled data.

As the user is a climate scientist, having a visually appealing GUI (graphical user interface) is not of great importance to them. They are more likely to focus on the effectiveness of our techniques, the clarity of our comparisons between each and the quality of the dataset returned after our system has delivered the gap-filled data.

When it comes to what a user will expect from our system, the key requirement we must satisfy is to successfully provide evidence and a clear explanation as to why one of the techniques available to them is more effective than another, and in what situations each is most effective. We are assuming that a user has prior knowledge of gap-filling techniques and has experience working with time-series data. However, we aim for our system to be easy to use and also to offer descriptions of the theory behind each technique, a clear explanation as to why one of the different techniques of gap-filling is more efficient than another and an overview of the theory behind the entire system.

2.3 : Operational Scenarios

- *Scenario 1:*

  Goal in context: *User Initiates System:*

  Description: The user opens the system and is met with our GUI displaying all available actions and a description of our system.
  1. User runs the application.
  2. The user is presented with the GUI.
  3. GUI displays the system menu, all available actions: Select Dataset, Select Gap-Insertion Method, Select Gap-Filling technique and view results/metrics.

- *Scenario 2:*

  Goal in context: *User Initiates Gap Insertion:*

  Description: Once data is uploaded, the user now initiates the gap insertion technique to the data, providing equally dispersed gaps throughout the entirety of the dataset. This ensures there is fair testing on any given dataset given the techniques/methods used for gap filling after.
  We will aim to give complete insight into how we inserted these gaps into the given dataset, the user will be able to get a full picture of the methodology of our gap insertion and the result of the gap-filled dataset.
  1. User initiates gap-insertion
  2. Receives insights on gap-insertion technique/gap-inserted dataset

- *Scenario 2:*

Goal in context: *User Selects Gap Filling Method*

Description: The user now will be presented with a selection of gap filling methods to use on the gap-inserted data. We aim to implement an uncomplicated GUI that will allow this user (the climate analyst) to review their options and carefully pick from the list of various different options on filling the gaps in these data.

1. User is presented an interface of options for gap-filling techniques
2. User selects their gap-filling technique and the method is initiated using our code.

- *Scenario 4*

Goal in context: *User Views results*
Description: The user will now be given complete insight into the operations of the system and in particular the outcome behind their chosen gap-filling method. We will provide scientific reasoning and reasoning behind the efficiency of the given method.

3. User had selected gap-filling method and receives back the given result
4. The user now is given a complete description of the effectiveness of the method and what occurred when fed to the code.
5. Given the insights, the user is now given the ability to try a different technique on their chosen dataset and compare it’s effectiveness to the original method.

2.4:Constraints

Time Constraint: Due to the fact that we are quite restricted on the development of our system due to the time, our main focus is to ensure the basic functionality of the system is implemented and works accordingly by conducting several tests. We hope to have enough time remaining to expand our application and incorporate more features and functionality.

Knowledge Constraint: As our target user is a climate scientist, we are assuming that they have a basic understanding of gap-filling techniques and experience working with time-series data. To provide insight on the methodologies being applied in the system, we will provide a description of each technique and a breakdown of the results after a gap-filling technique has been applied.

3. Functional Requirements

3.1 : Dataset Access

Description:

For this project it is crucial we can obtain access to multiple datasets to test on. Our goal is to test our algorithms on ten various different datasets. We plan on preprocessing this data and equally dispersing gaps throughout the data. It is crucial that we create the gaps in the data ourselves, this way we will have a way of testing the accuracy of our models on the gap filled data after. Creating the gaps in our data will allow us to both implement gap filling methods and test their accuracy on the original filled dataset after implementation. Otherwise we have no means of finding the accuracy of such gap filling methods. If there is an issue of obtaining a sufficient number of datasets, we plan on dividing up our current datasets by location or other major variables.

Criticality:

It is essential to the functionality of our project that we have access to an ample number of meteorological datasets. This will ensure we can get a full picture of the effectiveness of each of our models. If we can only get access to a small number of datasets it might result in a skewed conclusion towards the end of the project.

3.2 Gap insertion

Description:

Likewise, our gap insertion techniques are paramount to the functionality of our project. In order to fully test and understand our models further on in the project, we must initially take our filled datasets and insert gaps throughout, to simulate real world meteorological data. Without the proper insertion methods on our time-series data, we will struggle to imitate these real world gap-filled datasets.

Criticality:

As previously mentioned, inserting gaps in our data is paramount to testing our algorithms and providing various results/insights on the efficiency of our algorithms. Furthermore it is critical that our gaps are evenly distributed across the datasets to prevent skewed results in our results.

Technical Issues

It is possible that we will run into technical issues when attempting to insert gaps in these datasets ourselves. Issues such as the meteorological data containing gaps in the data already, in which case it will prove difficult or impossible to test the accuracy of our algorithm later on. There is also the possibility of our gap insertion not being uniform across the dataset in which case our results will once again be skewed.

Dependencies

This feature is entirely dependent on 3.1(Dataset Access) given that we need access to the datasets in the first place.

3.3 : Python Implementation

Description

Using various different python libraries (Pandas, SciKit-Learn, LightGBM), we plan on hosting the majority of our backend algorithm code in which we will implement methods on gap-filling our data. We will make use of python libraries such as lightgbm and sklearn in order to facilitate our code. LightGBM is a free and open source gradient boosting framework for machine learning, and will assist in accurately filling the gaps in our data. Similarly, SKlearn is also a free open source library that allows for regression testing and clustering algorithms. Our plan is to implement 5 individual different methods using these libraries and any other additional python code and libraries for this project. We will not only test these methods on our data, but test them against each other to provide insight on which method is more efficient on specific datasets.

Criticality:

Our python code being implemented properly will be crucial to the functionality of the project as a whole. Without a working backend code that provides a means to fill the gaps in our data, we can provide no insight on the effectiveness of each method. It is also crucial that our python code is not faulty and works properly, which we will test continuously throughout the course of the project.

Technical Issues

Some technical issues may arise in the python code given the extensiveness of our code. It is possible that we will come across some errors/bugs throughout the implementation of our code, that will need to be assessed straight away so our final results aren’t inaccurate.

Dependencies

This feature is also heavily dependent on 3.1 & 3.2 given we need access to the gap inserted data to test our python code on. Without access to this data and properly inserted gaps throughout our code will be useless.

3.4: User Interface

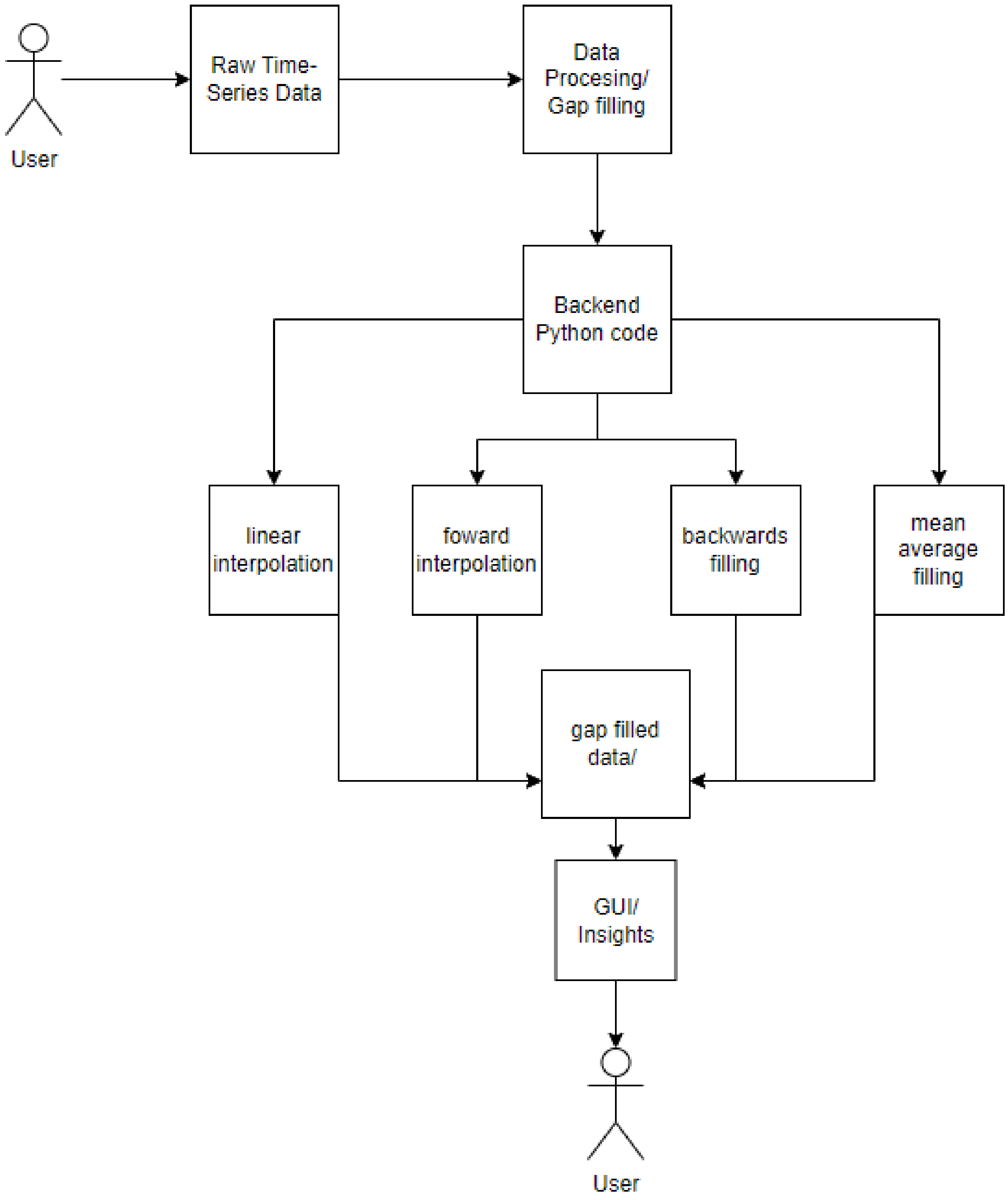
Description: UI

This system will involve a basic User Interface, in which the proposed user (climate analyst) will be able to upload the dataset to the system and be returned a full analysis on methodology effectiveness and the final processed dataset. Rather than creating a colourful and rich user interface, we want this project to highlight both the methods and the results of such inputted data in a simple GUI interface that is both simple and effective.

Criticality:

Although a GUI isn’t paramount to the functionality of the project, we believe it is important to have as a feature. Without a GUI the project will lack dimensionality.

4.1. System Architecture Diagram



The system architecture will involve a user feeding a series of raw time-series datasets to the system, in which we will then pre-process the given data and pass on to our backend python code to test various different methods on.

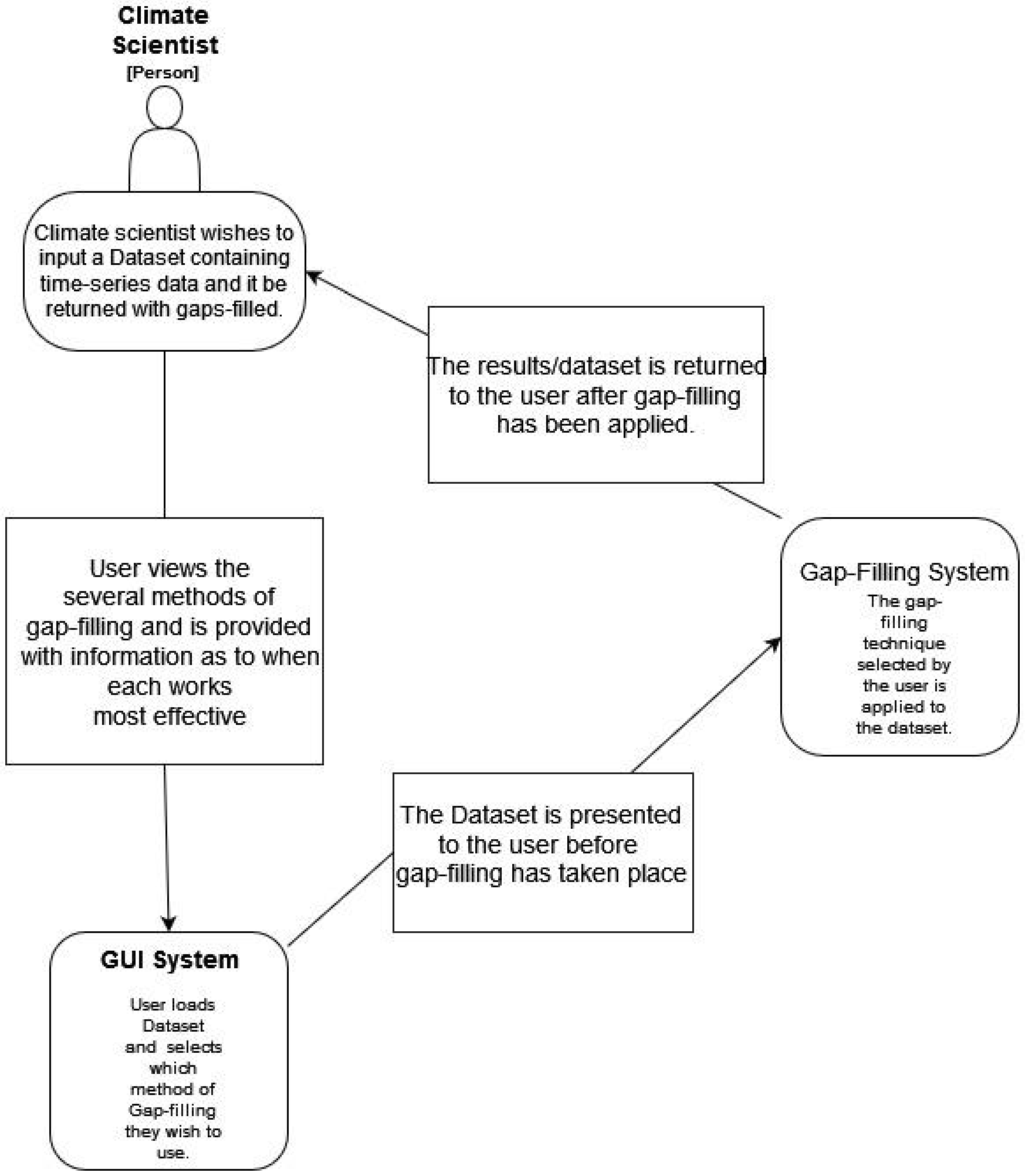
The user interface will be designed so the given user (a climate scientist for example) can input any desired meteorological datasets to our system and be returned not only the gap filled data via different methods but also explanations on what method yielded the most accurate results for that specific dataset. Our intent is that the user will be given a full detailed report on the performance of each and every method and dataset.

A straightforward and efficient GUI will be implemented on the front end for the proposed user, being a climate analyst. We will assume the user has some experience with both meteorological data and the utility of interpolation methods on gap-filled data.

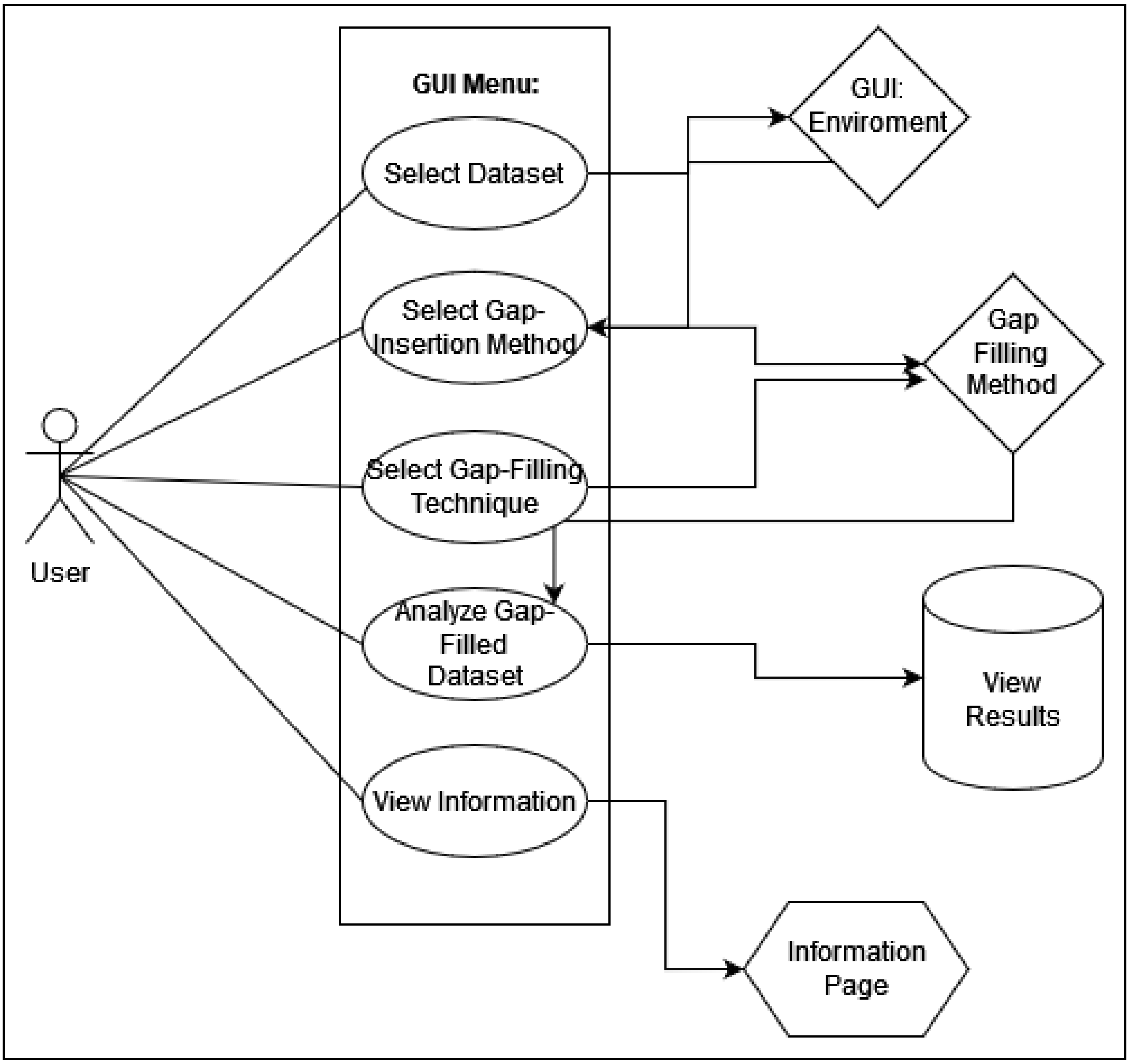
As we can see above, the proposed GUI will involve this user simply inputting their choice of dataset in which we will proceed to use gap insertion methods on the data and feed it to our backend python code. Our backend code will then feed the result of the tests back to the user, in which they will receive a synopsis of the different methods used and how efficient each was on the data.

5. High level Design

5.1 System Context Diagram:

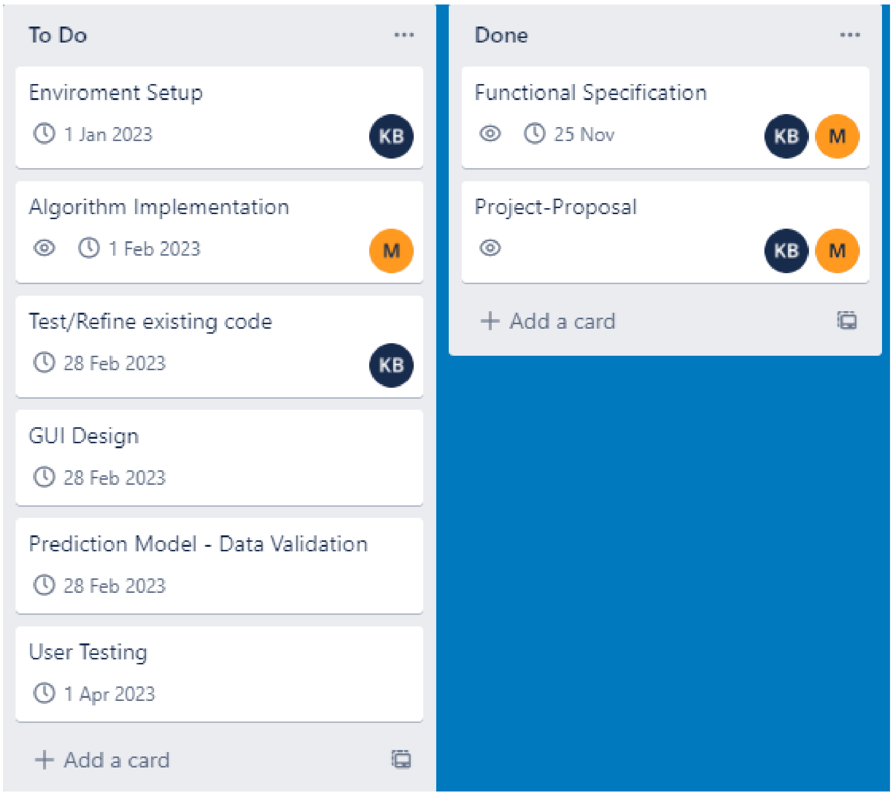

Climate Scientist
[Person]
Climate scientist wishes to input a Dataset containing time-series data and it be returned with gaps-filled.
The results/dataset is returned to the user after gap-filling has been applied.
User views the several methods of gap-filling and is provided with information as to when each works most effective
Gap-Filling System
The gap-filling technique selected by the user is applied to the dataset.
The Dataset is presented to the user before gap-filling has taken place
GUI System
User loads Dataset and selects which method of Gap-filling they wish to use.

5.2 Use Case Diagram:



6. Preliminary Schedule

Below is a brief breakdown of the workload required for this project and the approximate due dates of each task. We have used trello to organise our tasks at hand and will continue to keep it updated at all times throughout the course of this project.



To Do ···

Enviroment Setup
1 Jan 2023 KB

Algorithm Implementation
1 Feb 2023 M

Test/Refine existing code
28 Feb 2023 KB

GUI Design
28 Feb 2023

Prediction Model - Data Validation
28 Feb 2023

User Testing
1 Apr 2023

+ Add a card

Done ···

Functional Specification
25 Nov KB M

Project-Proposal
KB M

+ Add a card

7. Appendices

1. An extended approach for spatiotemporal gap filling: dealing with large and systematic gaps in geoscientific datasets, J.v. Buttlar, J. Zscheischler, and M. D. Mahecha - Max Planck Institute for Biogeochemistry: https://www.researchgate.net/publication/258615670_An_extended_approach_for_spatio-temporal_gap_filling_dealing_with_large_and_systematic_gaps_in_geoscientific_datasets#pf5
2. A Heuristic Gap Filling Method for Daily Precipitation Series, Jungjin Kim & Jae H. Ryu - https://rdcu.be/c0pXa