

*Process MeNtOR 3.0*

***Uni-SEP***

# Environment and Health Data Analyzer

## Design Document

Version:	1.0
Print Date:	15/03/2021
Release Date:	
Release State:	

Approval State:	
Approved by:	
Prepared by:	
Reviewed by:	
Path Name:	
File Name:	SDD-template-v2.00
Document No:	

## Document Change Control

Version	Date	Authors	Summary of Changes
1.0	15/03/2021	Haochen Hu, Xiao Ma, Taoye Na, Junshen Xv	completed section 1 - 4

## Document Sign-Off

Name (Position)	Signature	Date
Haochen Hu	HH	3/15
Haochen Hu	HH	3/20

## Contents

<b>1</b>	<b>INTRODUCTION</b>	<b>4</b>
1.1	Purpose	4
1.2	Overview	4
1.3	Resources - References	4
<b>2</b>	<b>MAJOR DESIGN DECISIONS</b>	<b>4</b>
<b>3</b>	<b>ARCHITECTURE</b>	<b>4</b>
<b>4</b>	<b>DETAILED CLASS DIAGRAMS</b>	<b>4</b>
4.1	UML Class Diagrams	4
<b>5</b>	<b>USE OF DESIGN PATTERNS</b>	<b>4</b>
<b>6</b>	<b>ACTIVITIES PLAN</b>	<b>4</b>
6.1	Project Backlog and Sprint Backlog	4
6.2	Group Meeting Logs	5
<b>7</b>	<b>TEST DRIVEN DEVELOPMENT</b>	<b>5</b>

# 1 Introduction

## 1.1 Purpose

*This document details the requirements of the system <Environment and Health Data Analyzer>. After the operator chooses the data type, country and time that he is interested, the system extracts the data from World Bank Database and then automatically calculates with Analysis Module which builds in the operating system and makes the data into the corresponding chart. Finally, the output will display the data to the UI for the operator to check directly. At the same time, the user can easily delete or add any data to the graph from the UI page.*

## 1.2 Overview

*We converted all the Major Decisions and the whole process into a Component Diagram that was easier to understand and more intuitive. In section 3, we listed all the interfaces that we need with each description during the whole system. For a Component Diagram it is not possible to show every detail of each class, so in the fourth section we present each important decision class with a UML diagram to show its individual details.*

## 1.3 Resources - References

<https://lucid.app>

<http://www.eclipse.org/downloads/index.php>

# 2 Major Design Decisions

*The system has been split to different modules to make sure the high cohesion and low coupling could be achieved. Modules communicate with other modules only when it needs data from other modules and each module has its own function.*

*The **front\_end module** encapsulates the user-facing portion of the system which includes the UI and screen elements as well as the login proxy.*

*The **selection module** encapsulates the functionality performed by the system at the users request (for example, user could select country, analysis, years, then the selection engine will process users selection by creating the selection object )*

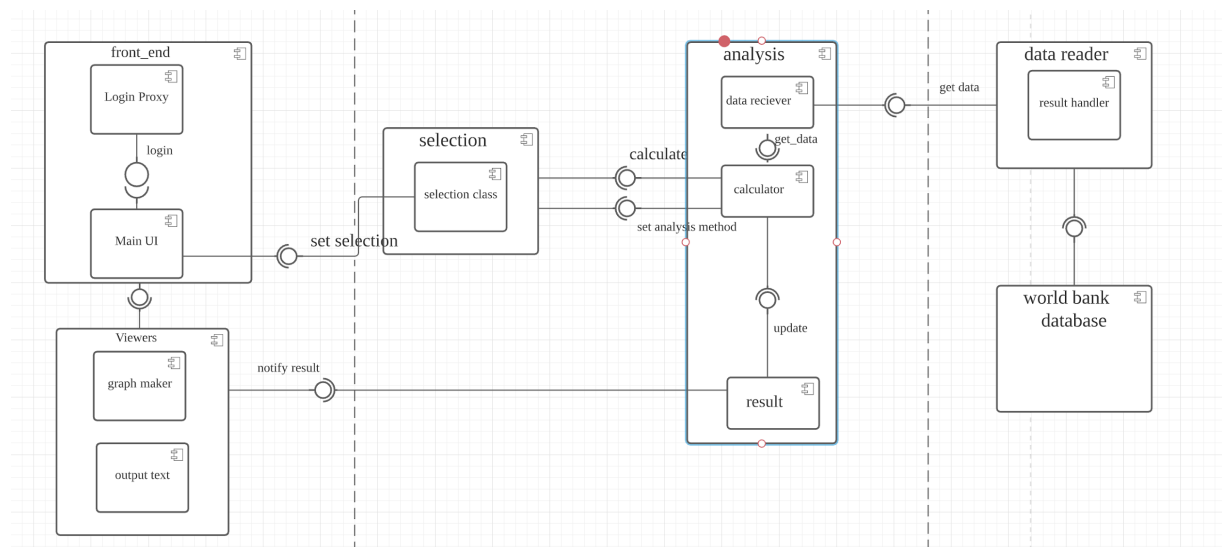
*The **Analysis Module** encapsulates the analysis portion of the system and dispatches the selected analysis as well as requests the data needed for the analysis. The data receiver requests the data from the data\_reader component for receiving the user needed data and passes it to the calculator server component to get the analysis performed. The third one was the result component which*

received the calculation result from the calculator server component and notified the viewer to update the result to the user interface.

The **data\_receiver** module handles the data requests from the *Analysis Engine* in the form of a get\_data request to the external service “WBD” which is the api we are going to use.

The **viewers module** waits for the completion of analysis and get updated once the result is produced by the result component.

### 3 Architecture



Interface	Actions	Description
Login	Login to the system	User input their username and password to login to the system
Set_analysis	choose the analysis type	<i>Allow user to choose the analysis type</i>
Set_country	<ol style="list-style-type: none"> <li>1.Choose the country from the list</li> <li>2.Check if the data of selected country could be fetched</li> <li>3.fetch the data for</li> </ol>	<i>Allow user to choose the country from the country list</i>

	<i>the selected country</i>	
<i>Set_years</i>	<ol style="list-style-type: none"> <li>1.Choose the years from the list</li> <li>2. Check the analysis for the years could be performed</li> </ol>	<i>Allow user to select years from the years list</i>
<i>graph_operations</i>	<ol style="list-style-type: none"> <li>1.adding and deleting graph for the selected analysis</li> <li>2.check if the graph could added or deleted</li> </ol>	<p><i>Allow users to add and delete viewers for the viewer list.</i></p> <p><i>Making sure viewers to be added are compatible with the analysis chosen and viewers to be removed are already in the list of viewers. In either case, if the criteria are not met, the user should receive a message.</i></p>
<i>Perform_Analysis</i>	<ol style="list-style-type: none"> <li>1.get needed data</li> <li>2.Perform analysis</li> </ol>	<i>The user could press the recalculate button to perform the analysis that they've selected and we will have the certain algorithms to process those data.</i>
<i>Display_result</i>	<i>display the result</i>	<p><i>The system will render the result to the selected viewers.</i></p> <p><i>The viewers should identify, based on the analysis, the</i></p>

		<i>number of series of data that need to be visualized.</i>
<i>Calculate</i>	<i>delegate the analysis</i>	<i>perform the actual analysis for the country and years selected.</i>

*Provide the component diagram of your system. If you need nested diagrams please use nesting levels. Include explanations on the functionality of each component. Provide the exposed interfaces of each component and list and briefly describe the functionality (one sentence) of the operations included in each such interface. Comment if you are using any specific architectural style or combinations of architectural styles.*

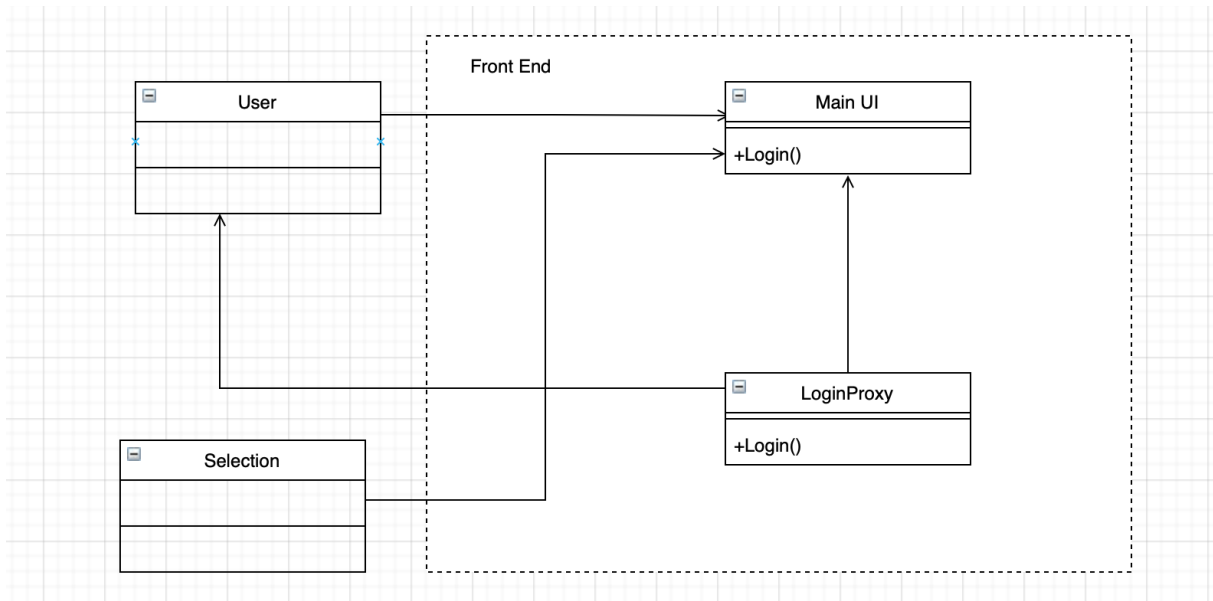
## 4 Detailed Class Diagrams

### 4.1 UML Class Diagrams

*Detailed class diagrams for each class you modify or write for the extensions. You can separate the class diagrams per module they appear. Tables should also be included listing the methods of each class with a short description of what each method does. Please indicate if a specific design pattern is used in your class diagrams.*

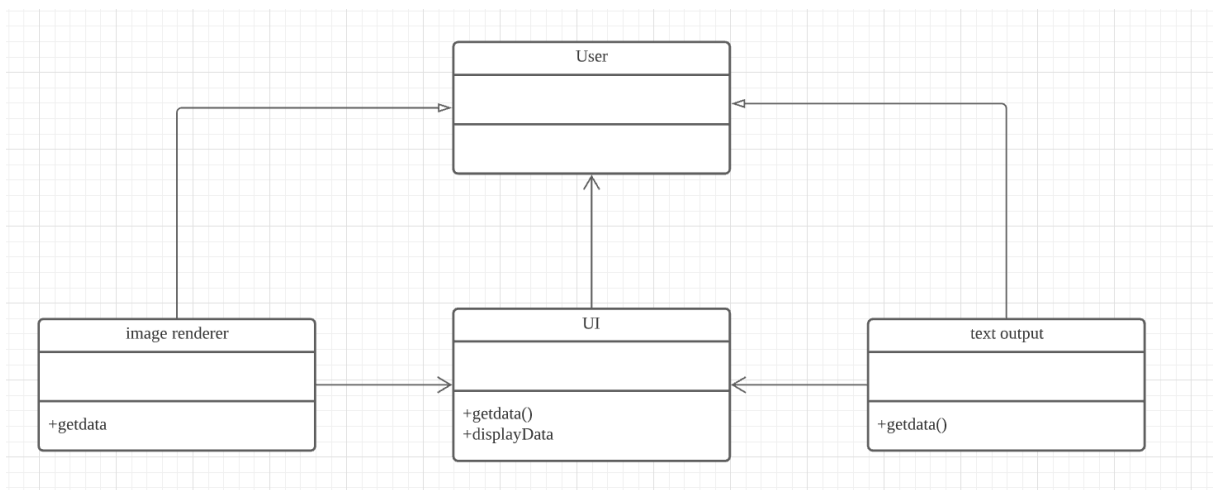
#### 4.1.1 Front End Module





Method	Description
<i>login</i>	<i>Login to the system in order to use the software</i>

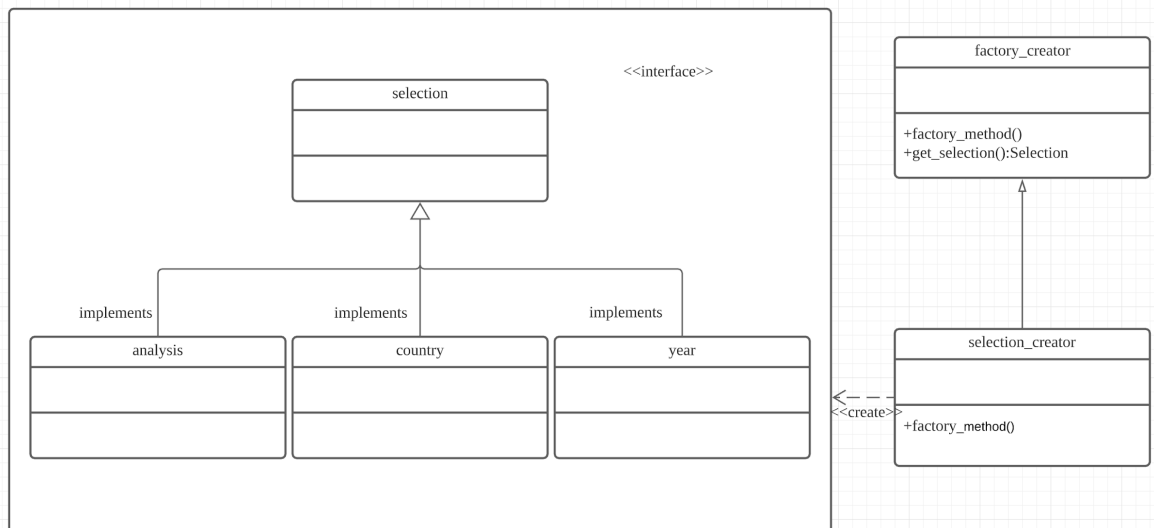
#### 4.1.3 viewer module



Method	Description
<i>displayData</i>	<i>show the user all data in main UI</i>
<i>get_selection():Selection</i>	<i>get the selection from the Main UI from what the user selected in so that it can create the factory.</i>

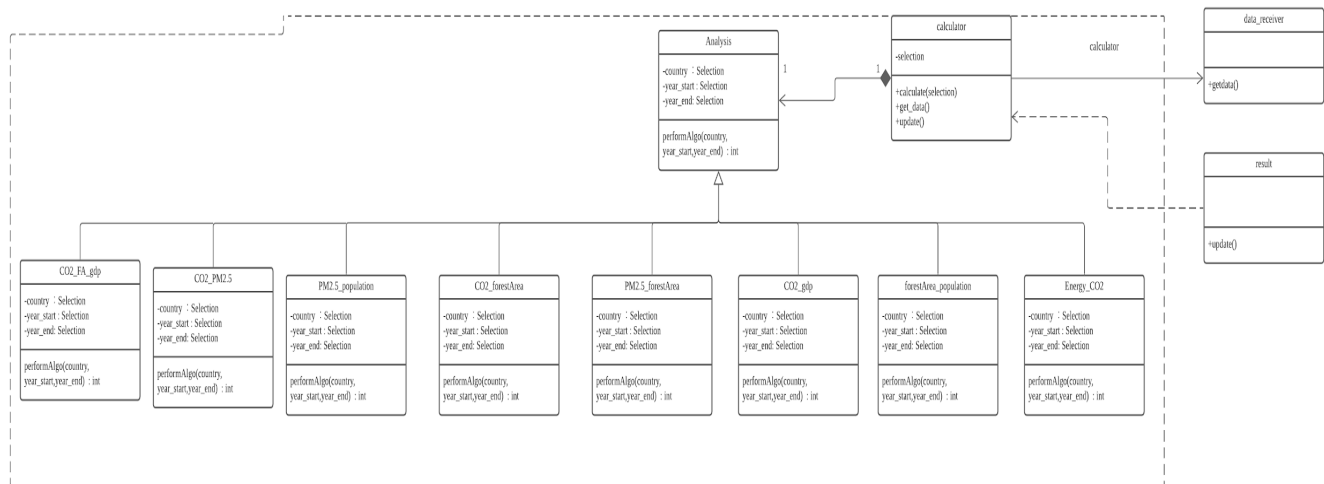
#### 4.1.3 selection module

Method	Description
<i>factory_method()</i>	<i>Creating the factory class to produce the selection object.</i>
<i>get_selection():Selection</i>	<i>get the selection from the Main UI from what the user selected in so that it can create the factory.</i>

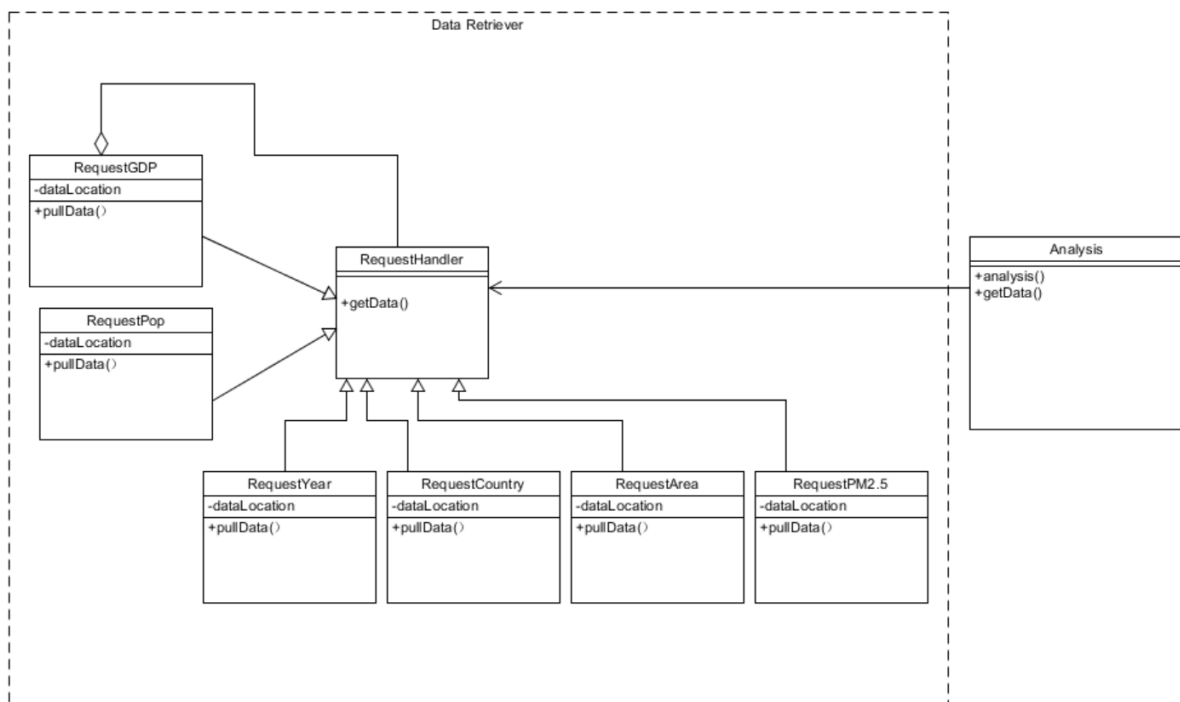


#### 4.1.4 Analysis Module

Method	Description
<i>calculate(selection)</i>	<i>The method takes the selection object and passes the selection type to the data receiver class and the Analysis class to perform calculation and data collecting.</i>
<i>get_data()</i>	<i>The method get the needed data for the selected analysis</i>
<i>update()</i>	<i>update the calculated result for rendering and notify the viewer.</i>
<i>perform_Algo(country,year_start,year_end)</i>	<i>perform the calculation and return the result.</i>



#### 4.1.5 Data Retriever Module



## 5. Use of Design Patterns

Description of the design patterns used along with their corresponding class diagrams.

*In the system, 6 design patterns will be used*

### 1. **Proxy design pattern & singleton design pattern**

Proxy design pattern (diagram 4.1.1) was used since a login functionality is required, which this design pattern allows for with the use of a protection proxy. The singleton design pattern allows the client for only using the get method instead of the setting the password or username.

### 2. **Observer design patterns**

Observer design patterns (diagram 4.1.2) was used since it allows for other objects to be notified when there is a state change. For example: the display needs to be updated.

### 3. **The factory design pattern** (diagram 4.1.3) was used to create selection objects which avoid the client to know how the selection object was created.

### 4. **Strategy design pattern**

Strategy design pattern (diagram 4.1.4) was used since there are multiple analysis algorithms that could be used and this design pattern allows for them to be called independently and with a well designed interface

### 5. **Chain of Responsibility design pattern**

Chain of Responsibility design pattern (diagram 4.1.5) was used since data requests are made by the system, this pattern promote low coupling between the sender of a request and the objects that handle the request

## 6. Activities Plan

### 6.1 Project Backlog and Sprint Backlog

Backlog Item	Estimate
As a user I want to access the program	3
As a user I want to select the analysis type that I want	13
As a user I want to select the years of data that I want	8
As a user I want to select different type of graph type	8
As a user I want to add graph to the display screen	13
As a user I want to remove a selected graph from the display result	13
As a user I want the analysis result displayed in a window	13

## 6.2 Group Meeting Logs

In this Section you write minutes of each meeting, listing the attendance, what the topics of discussion in the meeting were, any decisions that were made, and which team members were assigned which tasks. These minutes must be submitted with the project report in each deliverable and will provide input to be used for the overall assessment of the project.

Present Group Members	Meeting Date	Issues Discussed / Resolved
4	3/15	Assigning parts for team member part 1,2 :Taoye Na part 3,4 :Xiao Ma part 5,6 :Haochen Hu part 6,7 :Junshen Xu
2	3/19	completed part 3 diagram
4	3/22	assigning part 4 to team member Part 4:Haochen Hu, Junshen Xu, Xiao Ma, Taoye Na

## 7 Test Driven Development

Test cases will be provided in the form of a table as follows:

<b>Test ID</b>	Test 1
<b>Category</b>	Testing the login system
<b>Requirements Coverage</b>	UC1 User-Login
<b>Initial Condition</b>	The program has been started and runs.
<b>Procedure</b>	1.The user selects login 2. The user provides a user name 3.The user provides a password

	4. The user logs-in into the system and is presented with the main UI window)
<b>Expected Outcome</b>	<i>the login form closes, and the user is presented with the main UI window</i>
<b>Notes</b>	if user user do actually input the correct username and password,they can enter the main UI

<b>Test ID</b>	Test_2
<b>Category</b>	Testing the login error notification system
<b>Requirements Coverage</b>	UC1 User-Login
<b>Initial Condition</b>	The program has been started and runs.
<b>Procedure</b>	<i>1The user selects login  2. The user provides a wrong user name or password  3.An error notification comes up and clear the input window</i>
<b>Expected Outcome</b>	<i>The login form remains, the input window has been cleared, user can enter the username and password again.</i>
<b>Note</b>	

<b>Test ID</b>	Test_3
<b>Category</b>	Testing the analysis type selection function
<b>Requirements Coverage</b>	UC2-Selecting Analysis Type Successfully-different
<b>Initial Condition</b>	User click on the Analysis Type list and select the type that needed
<b>Procedure</b>	<i>1. The User click on Analysis Type list  2. The system check if the selection can be processed  3. The program will check if previous analysis exists  4. If previous analysis exists, and if the previous analysis is different with the selected one, empty the viewer</i>
<b>Expected Outcome</b>	<i>There should be no false message returned.</i>

<b>Notes</b>	When only selection fail, a pop-up notice will tell the user the selection fail
--------------	---

<b>Test ID</b>	Test_4
<b>Category</b>	Testing the analysis type selection function
<b>Requirements Coverage</b>	UC2-Selecting Analysis Type Successfully-same
<b>Initial Condition</b>	User click on the Analysis Type list and select the type that needed
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The User click on Analysis Type list</li> <li>2. The system check if the selection can be processed</li> <li>3. The program will check if previous analysis exists</li> <li>4. If previous analysis exists, and if the previous analysis is same with the selected one, remain the viewer</li> <li>5. Analysis type selected successfully</li> </ol>
<b>Expected Outcome</b>	There should be no false message returned.
<b>Notes</b>	When only selection fail, a pop-up notice will tell the user the selection fail

<b>Test ID</b>	Test_5
<b>Category</b>	Testing the analysis type selection function
<b>Requirements Coverage</b>	UC2-Selecting Analysis Type Successfully-empty
<b>Initial Condition</b>	User click on the Analysis Type list and select the type that needed
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The User click on Analysis Type list</li> <li>2. The system check if the selection can be processed</li> <li>3. The program will check if previous analysis exists</li> <li>4. If no previous analysis exists, the page remain empty and analysis type selected successfully</li> </ol>
<b>Expected Outcome</b>	There should be no false message returned.
<b>Notes</b>	When only selection fail, a pop-up notice will tell the user the selection fail

<b>Test ID</b>	Test_6
<b>Category</b>	Testing the analysis type selection function
<b>Requirements Coverage</b>	UC2-Selecting Analysis Type Failed
<b>Initial Condition</b>	User click on the Analysis Type list and select the type that needed
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The User click on Analysis Type list</li> <li>2. The system check if the selection can be processed</li> </ol>

	<i>3. Selected successfully</i>
<b>Expected Outcome</b>	<i>There should be a false message returned.</i>
<b>Notes</b>	When only selection fail, a pop-up notice will tell the user the selection fail

<b>Test ID</b>	Test_7
<b>Category</b>	Testing the country selecting function
<b>Requirements Coverage</b>	<i>UC3-Selecting Country successfully</i>
<b>Initial Condition</b>	User click on the country selecting list and select country
<b>Procedure</b>	<i>1. The user selects country</i> <i>2. The system check whether country selected can be processed</i> <i>3. select successfully</i>
<b>Expected Outcome</b>	<i>Countries can be selected with no error notification.</i>
<b>Notes</b>	

<b>Test ID</b>	Test_8
<b>Category</b>	Testing the country error notification function
<b>Requirements Coverage</b>	<i>UC3-Selecting Country failed</i>
<b>Initial Condition</b>	User click on the country selecting list and select country
<b>Procedure</b>	<i>1. The user selects country</i> <i>2. The system check whether country selected can be processed</i> <i>3. select failed</i> <i>4. an error notification comes up</i>
<b>Expected Outcome</b>	<i>error notification comes up to the window</i>
<b>Notes</b>	

<b>Test ID</b>	Test_9
<b>Category</b>	Testing the year selection function
<b>Requirements Coverage</b>	<i>UC4-Selecting Year Successfully</i>
<b>Initial Condition</b>	User click on the year selection list and select the year that needed
<b>Procedure</b>	<i>1. The User click on year list</i> <i>2. The system check if the selection can be processed</i> <i>3. Selected successfully</i>



<b>Expected Outcome</b>	<i>There should be no false message returned.</i>
<b>Notes</b>	

<b>Test ID</b>	Test_10
<b>Category</b>	Testing the year selection function
<b>Requirements Coverage</b>	<i>UC4-Selecting Year Failed</i>
<b>Initial Condition</b>	User click on the year selection list and select the year that needed
<b>Procedure</b>	<ol style="list-style-type: none"> <li><i>1. The User click on year list</i></li> <li><i>2. The system check if the selection can be processed</i></li> <li><i>3. Selected successfully</i></li> </ol>
<b>Expected Outcome</b>	<i>There should be a false message returned.</i>
<b>Notes</b>	When only selection fail, a pop-up notice will tell the user the selection fail

<b>Test ID</b>	Test_11
<b>Category</b>	Testing the adding graph for computed / Obtained
<b>Requirements Coverage</b>	<i>UC5-adding-graph-Successful</i>
<b>Initial Condition</b>	User click on the add button
<b>Procedure</b>	<ol style="list-style-type: none"> <li><i>1. The user selects the add button</i></li> <li><i>2. The system check if the selection is compatible with The analysis</i></li> <li><i>3. Graph added</i></li> </ol>
<b>Expected Outcome</b>	<i>There should be no false message returned.</i>
<b>Notes</b>	

<b>Test ID</b>	Test_12
<b>Category</b>	Testing the adding graph for computed / Obtained
<b>Requirements Coverage</b>	<i>UC5-adding-graph-Failed</i>
<b>Initial Condition</b>	User click on the add button
<b>Procedure</b>	<ol style="list-style-type: none"> <li><i>1. The user selects the add button</i></li> <li><i>2. The system check if the selection is compatible with The analysis</i></li> <li><i>3. Graph could not be added</i></li> </ol>
<b>Expected Outcome</b>	<i>There should be a false message returned.</i>

<b>Notes</b>	Add fails when the selection is not compatible with the analysis or the selection is already in the list
--------------	--

<b>Test ID</b>	Test_13
<b>Category</b>	Testing the removing graph for computed / Obtained
<b>Requirements Coverage</b>	<i>UC6-removing-graph-Successful</i>
<b>Initial Condition</b>	User click on the remove button
<b>Procedure</b>	<ol style="list-style-type: none"> <li><i>1. The user selects the remove button</i></li> <li><i>2. The system check if the selection is selection in the list</i></li> <li><i>3. Graph removed</i></li> </ol>
<b>Expected Outcome</b>	<i>There should be no false message returned.</i>
<b>Notes</b>	

<b>Test ID</b>	Test_14
<b>Category</b>	Testing the adding graph for computed / Obtained
<b>Requirements Coverage</b>	<i>UC6-removing-graph-Failed</i>
<b>Initial Condition</b>	User click on the remove button
<b>Procedure</b>	<ol style="list-style-type: none"> <li><i>1. The user selects the remove button</i></li> <li><i>2. The system check if the selection is selection in the list</i></li> <li><i>3. Graph could not be removed</i></li> </ol>
<b>Expected Outcome</b>	<i>There should be a false message returned.</i>
<b>Notes</b>	Graph removed fails when the selected item is not in the list

<b>Test ID</b>	Test_15
<b>Category</b>	Testing the analysis function will return correct value
<b>Requirements Coverage</b>	<i>UC7-Successful-performing-analysis</i>
<b>Initial Condition</b>	User click on Recalculate
<b>Procedure</b>	<ol style="list-style-type: none"> <li><i>1. The User click on Recalculate</i></li> <li><i>2. The system check if the required data can be collected</i></li> <li><i>3. The System perform analysis with given data</i></li> <li><i>4. The System check if the analysis succeeded</i></li> </ol>

	<i>5.Perform if succeeded, system return analysed data</i>
<b>Expected Outcome</b>	<i>There should be no false message returned.</i>
<b>Notes</b>	When only missing few years from the required years,the analysis will success

<b>Test ID</b>	Test 16
<b>Category</b>	Test if the analysis function will display error message
<b>Requirements Coverage</b>	<i>UC7-Failed-performing-analysis</i>
<b>Initial Condition</b>	User click on Recalculate
<b>Procedure</b>	<i>1. The User click on Recalculate</i> <i>2. The system check if the required data can be collected</i> <i>3. The System perform analysis with given data</i> <i>4. The System check if the analysis succeeded</i> <i>5. Part of data missing, system return error message</i>
<b>Expected Outcome</b>	<i>There should be a false message returned.</i>
<b>Notes</b>	The performance will fail when values are missing.

<b>Test ID</b>	Test 17
<b>Category</b>	Test if the display function will return the required graph with required data
<b>Requirements Coverage</b>	<i>UC8-Successful-display-result</i>
<b>Initial Condition</b>	Once data has been fetched and all calculation is returned
<b>Procedure</b>	<i>1. The system will collect all needed data</i> <i>2. The system determine the number of series of data needed</i> <i>3. screen is updated with new graph</i>
<b>Expected Outcome</b>	<i>The screen is updated with graphs that user selected</i>
<b>Notes</b>	Years may be missing due to lack of data

<b>Test ID</b>	Test 18
<b>Category</b>	Test if the display function will return the required graph with required data
<b>Requirements Coverage</b>	<i>UC8-Failed-display-result</i>
<b>Initial Condition</b>	Once all data has been fetched and all calculation is returned
<b>Procedure</b>	<i>1. The system will collect all needed data</i>

	2. <i>The system determine the number of series of data needed</i> 3. <i>The system displays an error message due to incomplete analysis, missing data.</i>
<b>Expected Outcome</b>	<i>The screen will display an error message</i>
<b>Notes</b>	