THOMPSON RIVERS UNIVERSITY

SENG 1210 – Programming for Engineers II

Project Design:
Natural Disaster Early Warning System

Luka Aitken (T00663672)

Toma Aitken (T00663680)

Braden Wielgoz (T00669054)

Date submitted: April 14, 2021

Table of Contents

1 Introduction 5
2 Design Problem 6
2.1 Problem Definition 6
2.2 Design Requirements 7
2.2.1 Functions 7
2.2.2 Objectives 8
2.2.3 Constraints 9
3 Solution 10
3.1 Solution 1 10
3.2 Solution 2 10
3.3 Final Solution 11
3.3.1 Components 14
3.3.2 Features 15
3.3.3 Environmental, Societal, Safety, and Economic Considerations 15
3.3.4 Limitations 16
4 Team Work 18
5 Project Management 22
6 Conclusion and Future Work 23
7 References 25
8 Appendix 27

List of Figures

Figure 1.1 - Aftermath of a Tornado. 5

Figure 2.1 - Aftermath of Cyclone Bhola. 6

Figure 2.2.1 - Function Tree. 7

Figure 2.2.2 - Objective Tree. 8

Figure 3.3.0 - Output for No disasters. 11

Figure 3.3.1 - Output for Forest Fire. 11

Figure 3.3.2 - Output for Blizzard. 12

Figure 3.3.3 - Output for Category 1 Hurricane (Outputs Category 1-5 Hurricanes). 12

Figure 3.3.4 - Output for F0 Tornado (Outputs F0-F5 Tornadoes). 12

Figure 3.3.5 - Block Diagram. 15

Figure 3.3.6 - Image of People Helping During a Major Flood. 16

Figure 3.3.7 - Aftermath of Earthquake and Tsunami in Japan 2011. 17

Figure 6.0 - People Helping Prepare for Natural Disasters. 23

Appendix A - Image of Ews.h and Ews.cpp File Code. 27

Appendix B - Image of Details.h File Code. 28

Appendix C - Image of Details.cpp File Code. 28

Appendix D - Image of Details.cpp File Code of a Function. 29

Appendix E - Image of main.cpp. 29

List of Tables

Table 2.2.1 - Importance Chart. 8

Table 2.2.2 - Requirement Table. 9

Table 3.3.0 - Decision Matrix. 13

Table 4.1 - Meeting 1. 18

Table 4.2 - Meeting 2. 18

Table 4.3 - Meeting 3. 18

Table 4.4 - Meeting 4. 19

Table 4.5 - Meeting 5. 19

Table 4.6 - Meeting 6. 20

Table 4.7 - Meeting 7. 20

Table 4.8 - Meeting 8. 21

Table 4.9 - Meeting 9. 21

Table 5.0 - Gantt Chart. 22

1 Introduction

Natural disasters are becoming more common and are increasing in frequency and severity worldwide [1]. Globally, natural disasters kill an average of over sixty thousand people per year. There are many different types of natural disasters, listing from earthquakes to tsunamis that affect people in certain ways. When a natural disaster hits, there is little to no time to prepare for the oncoming disaster. After the disaster hits, many people around the area are affected and require basic survival needs such as food and water. Without the basic needs to survive, the families who lost their homes from the disaster would become homeless and would need shelter. In addition, people affected by the disaster could have from minor injuries to life-threatening to even fatal injuries. These people would require immediate medical assistance as a result of the disaster [2]. Without knowing when the next disaster will hit, having some form of an early warning system would help reduce the number of possible injuries that may be sustained. That's where our early warning system for natural disasters comes in.

Figure 1.0 - Aftermath of a Tornado [3].

To begin with our early warning system (EWS) for natural disasters, we will come up with different ideas on how to create a reliable, easy-to-use program that can be used by a user, while following the project requirements and constraints. Our goal is to come up with a software application that can provide important information to the users that may be affected by the natural disaster. The application will provide information about the given natural disaster and how to take the necessary precautions to help the users prepare.

Through this report, we will discuss the problems we faced when designing and building an early warning system (EWS) program that helps determine if a natural disaster will occur. We will also discuss the other solutions that we didn't use, but took some inspiration from to come up with our final design. Additionally, the environmental, societal, safety, economic considerations as well as sustainability are some other key factors that will be discussed throughout this report.

2 Design Problem

2.1 Problem Definition

Every year, tens of thousands of people's lives are lost from natural disasters and even more are adversely affected by them. The extent of damage caused by natural disasters is not just related to its severity, but also the capacity of where people are living in disaster-prone areas all over the world [4]. From the National Center of Disaster Preparedness at Columbia University, found that 65 percent of Americans have no disaster plan, or their plans are not adequate. The Americans found in this category are people who live in poverty, have low income, and less education [5]. This leaves these Americans very vulnerable as there is no way for them to know if a natural disaster is coming which can result in lives lost. In 1970, one of the deadliest tropical cyclones known as Cyclone Bhola in Bangladesh killed between an estimated 300,000 to 500,000 lives. The lives that were lost were the result of a large storm that overwhelmed the low-lying islands and tidal flats along the shores of the Bay of Bengal. This was the result of people not being warned about the incoming disaster which brought about the WMO's Tropical Cyclone Programme which would help prevent and mitigate the effects of tropical cyclones around the world [6].

Figure 2.1: Aftermath of Cyclone Bhola [7].

As there is no way of preventing different kinds of natural disasters, with enough effort and proper preparation, it is possible to reduce and mitigate their effect by warning people. This can also lead to creating infrastructures that will help prevent disasters such as strengthening building codes, building shelters, and protecting environmental buffers [8]. That is why our group has decided to create an early warning system that is designed to provide people about to be affected by natural disasters with enough time and effective information to properly prepare.

2.2 Design Requirements

2.2.1 Functions

- Early warning systems (EWS) are an important information tool that may have a significant effect on people's everyday lives. Our design will have to meet all requirements for any kind of early warning system. One of the main aspects of an early warning system for natural disasters and the most important requirement is that it provides timely and meaningful warning information towards users that could be at-risk. This will help those at-risk to prepare and to act appropriately with enough time, in case of a natural disaster, to reduce any harm or loss. A successful early warning system should help save people's lives and jobs, land, and infrastructures and will support long-term sustainability for communities all around the world [9]. That is why our program for an early warning system for natural disasters needs to display important information in a consistent and meaningful manner that is easy to access and control for those at-risk.

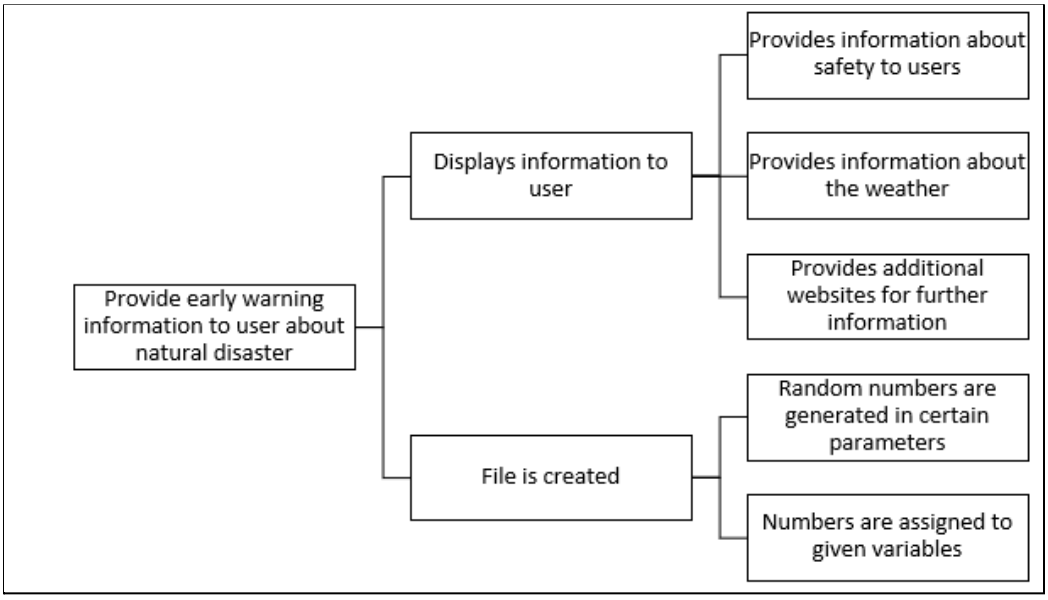

Figure 2.2.1: Function Tree.

Safe	8
Reliability	8
Performance	6
Simplicity and Maintenance	7
Easy to control	7
Easy to access	6
Low maintenance cost	7
Low manufacturing cost	7

Table 2.2.1: Importance Chart.

2.2.2 Objectives

- The objective of this project is to create and build a program that acts as an early warning system (EWS) for natural disasters. The program should display results of multiple kinds of natural disasters and provide relevant and correct data from different research articles in a timely response. This means that the program should be able to detect certain natural disasters and provide useful information about the natural disaster, but also provide ways on how to prepare and to be safe during the given natural disaster. The program should also be easy to access and control by displaying all the data in a simple meaningful manner for others to read easily. The system should also be cost-effective in that it is easy to manufacture and repair if there is something wrong with the system.

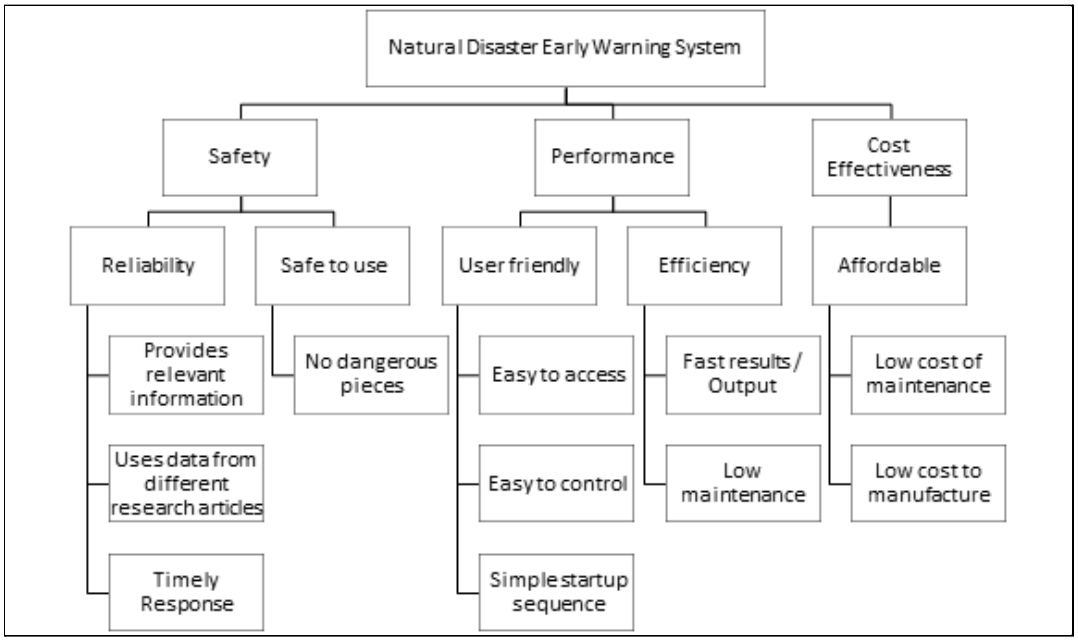

Figure 2.2.2: Objective Tree.

Requirement Table (Ranking)	
Rank	Requirement
1	Must be easily accessible to users before or during a natural disaster.
2	Provide ways for users to be safe.
3	Provide instructions to users to prepare for given natural disaster.
4	Use real data collected during different natural disasters.
5	Display information to users in a pleasing and formal way.
6	Provide reliable information
7	Must be easy to manufacture to create and be cost-effective

Table 2.2.2: Requirement Table.

2.2.3 Constraints

- Some of the constraints of this project are, how the program is designed and created. It must be created in project format with different header and implementation files. Our program will have to provide important information on health and safety for users during different kinds of natural disasters. The program will also provide information on different environmental factors and societal impacts of each natural disaster to users. Additionally, the program will be reliable to all users by providing quick responses with all of its information coming from trusted sources that focus on different kinds of natural disasters and weather. Another vital thing to consider is different regulatory compliances which are a set of guidelines that are found all over the world and to follow in the case of a natural disaster. With all this different information that must be included in our program, it has to be well organized and present the program to the user in a way that is aesthetically pleasing.

3 Solution

3.1 Solution 1

For our first solution, our approach was to make a simple early warning system program, where it would ask the user questions. The program would start by asking the user what natural disaster they would want to test, such as hurricanes, tornadoes, and more. After the user enters a disaster, the program would ask certain kinds of questions to the user about the environment, locations, speeds of wind or waves, etc. When the user is done answering the questions given by the program, the program would use the information to calculate and check if or if not, a natural disaster could occur. If the program says that the natural disaster entered by the user happens, the program will list off instructions to do during disasters, such as what to do and how to prepare for the oncoming disaster. When building the program, we would use many children classes that represent the different disasters and one parent class that asks what disaster they want to try. In each children class, it would ask certain questions on the disaster, which would then tally up the marks at the end of the questions. Once the program finishes asking questions, it would determine if or if not, that disaster would occur.

In the end, we did not choose this solution as our final solution because of all the questions the program asks. When someone uses the program, they could be irritated or overwhelmed by all the questions being asked. With all these questions being asked, it can be a waste of time for the user, as the program is wasting the user's time preparing for the disaster if the disaster does occur.

3.2 Solution 2

For our second solution, our approach was to use files to store all the data that was collected from the research of different natural disasters. For the program to be an early warning system, the program would begin to read all the files created in the program and use multiple if and else statements to print out and display the important messages of a given natural disaster. These messages that the program displays would provide information of what kind of natural disaster it is and the characteristics it has, such as temperature and wind speed. It will then have a list of things to do and inform the user of the incoming natural disaster so that the users can prepare and take the necessary precautions to be safe.

In the end, we decided not to use this solution as the final solution for an early warning system because the program itself would have used too many if and else statements. The program was not built in the project format which was one of the requirements and didn't use any classes or functions. If this way was done, it would have made the program more efficient to the users and easier to modify if needed. By not using the project format with classes and functions in the program, the time for the program to load would be much longer than if we did use classes and functions in the program.

3.3 Final Solution

For our final solution, we came up with an improved design that took different aspects from the previous solutions that we thought were necessary to create a reliable, easy-to-use early warning system program. The program was done in project format and used header and implementation files that would help organize all the different classes and functions in the program. The program uses data that was collected by our team and stores the information about different natural disaster aspects in one file that the program reads. The program will take the data from the file and use the data to figure out if a natural disaster is likely to happen. If a natural disaster happens, the program will output what disaster it is, what to expect, and instructions to the user about what to do in that scenario for the user to be safe. The natural disasters that were implemented into the program are Tornadoes, Hurricanes, Forest Fires, and Blizzards as shown in figures 3.3.0 - 3.3.4 .

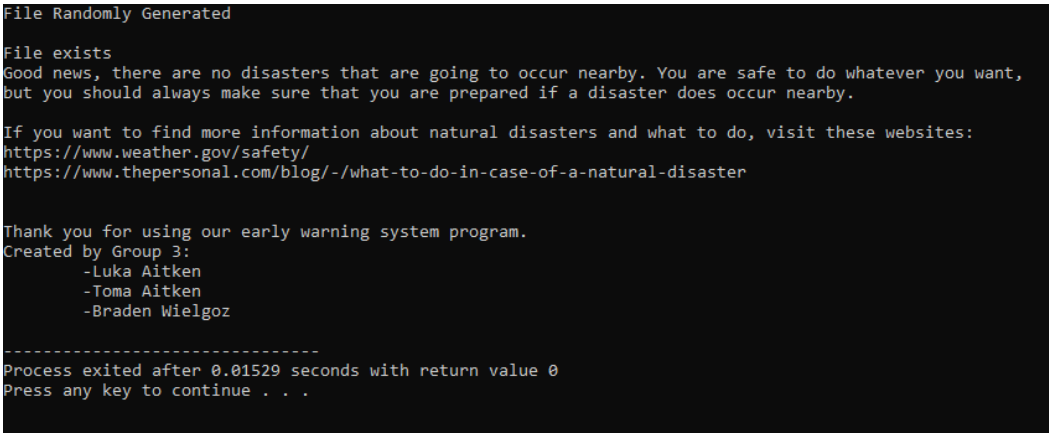

Figure 3.3.0: Output for No Natural Disaster.

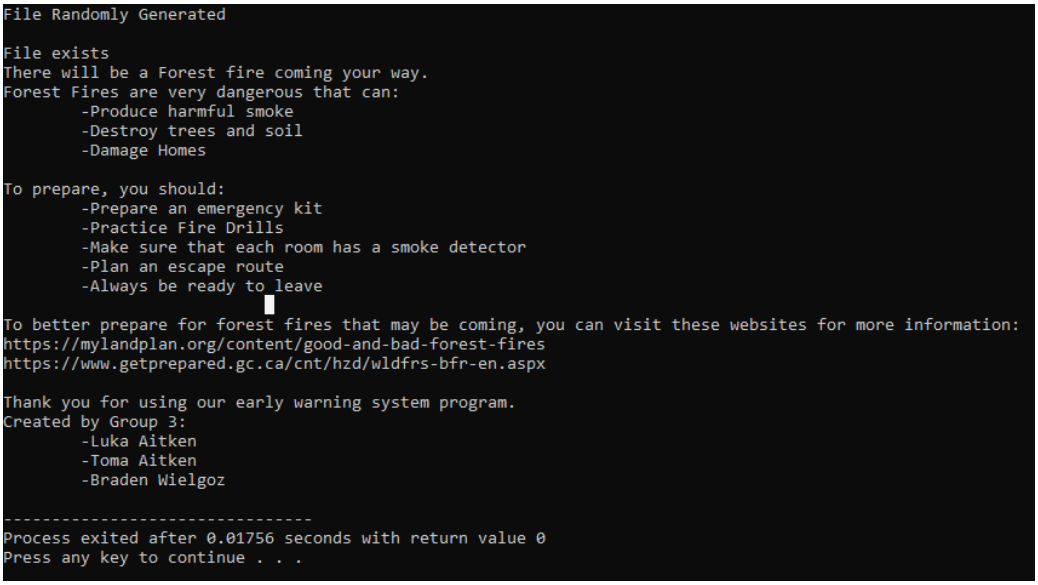

Figure 3.3.1: Output for Forest Fire.

```
File Randomly Generated

File exists
There will be a Blizzard coming your way.
Blizzards are very dangerous with the extreme temperature and high winds that can:
        -Cause Power Outages
        -Give Hypothermia and Frostbite
        -Cause car accidents
        -Last a few hours or days

To prepare, you should:
        -Stay warm with proper clothing
        -Stay in warm shelters
        -Not drive during blizzards
        -Prepare a winter emergency kit
        -Gather food and water

To better prepare for blizzards that may be coming, you can visit these websites for more information:
https://www.economical.com/en/blog/economical-blog/january-2020/what-to-do-before-during-and-after-a-snowstorm
https://www.ready.gov/winter-weather#:~:text=Winter%20storms%20create%20a%20higher,snow%2C%20ice%20and%20high%20winds.

Thank you for using our early warning system program.
Created by Group 3:
        -Luka Aitken
        -Toma Aitken
        -Braden Wielgoz

--------------------------------
Process exited after 0.01661 seconds with return value 0
Press any key to continue . . .
```

Figure 3.3.2: Output for Blizzard.

```
File Randomly Generated

File exists
There will be a Category 1 Hurricane coming your way.
Category 1 Hurricanes are the lowest kinds of Hurricanes and have very dangerous winds that can:
        -Damage Roofs
        -Large Branches of Trees snapping
        -Cause Power Outages that can last a few days

To prepare, you should:
        -Gather emergency food, water and medicine supplies
        -Prepare an emergency kit
        -Stay inside
        -Stay away from windows
        -Always be ready to leave
To better prepare for hurricanes that may be coming, you can visit these websites for more information:
https://www.cdc.gov/disasters/hurricanes/before.html
https://www.nhc.noaa.gov/aboutsshws.php

Thank you for using our early warning system program.
Created by Group 3:
        -Luka Aitken
        -Toma Aitken
        -Braden Wielgoz

--------------------------------
Process exited after 0.01638 seconds with return value 0
Press any key to continue . . .
```

Figure 3.3.3: Output for Category 1 Hurricane (Outputs Category 1-5 Hurricanes).

```
File exists
There will be a F0 Tornado coming your way.
F0 Tornadoes, also know as Minor Tornadoes and are the lowest known kinds of Tornadoes and have dangerous winds that can:
        -Rip off some panelling off walls and roofs
        -Move automobiles
        -Break small trees

To prepare, you should:
        -Prepare an emergency kit
        -Stay inside, preferably in the basement
        -Stay away from windows
        -Drive to the nearest solid shelter
To better prepare for tornadoes that may be coming, you can visit these websites for more information:
https://www.getprepared.gc.ca/cnt/hzd/trnds-drng-en.aspx#:~:text=Go%20to%20the%20basement%20or,windows%2C%20outside%20walls%20and%20doors.
http://www.shorstmeyer.com/tornadoes/fpp.html#:~:text=TORNADO%20DESCRIPTION&text=%2D%20Wind%20speed%2073%20%2D%20112%20mph,moving%20automobiles%20pushed%20off%20roads
https://www.weather.gov/ffc/fujita

Thank you for using our early warning system program.
Created by Group 3:
        -Luka Aitken
        -Toma Aitken
        -Braden Wielgoz

--------------------------------
Process exited after 0.01669 seconds with return value 0
Press any key to continue . . .
```

Figure 3.3.4: Output for F0 Tornado (Outputs F0-F5 Tornadoes).

The reason why we chose this solution over the other solutions is that this solution offered a more reliable and faster program that can tell if a disaster is going to happen and provide the necessary instructions for the user to be safe. The overall design used many class functions to make the program run faster and smoother when compared to the previous solutions which had unnecessary features. This includes features such as solution 1, which has the program asking questions to the user so that the program can create different natural disaster scenarios. The final solution also follows the design constraints where the program must be designed and built-in project format which additionally helps boost the time the program runs. This will make it easier to maintain and modify compared to having the program built on a normal source file. Finally, this design does not waste as much time as the previous solutions for the user when converting the data into an answer, if there is going to be a natural disaster.

If our solution was actually implemented in the program instead of taking all the data out of a file it would be constantly collecting data from multiple different sources. The sources that the program would take data from would be a lot of different sensors hooked up near the area including anemometers to measure wind speed, thermometers to measure temperature, and hygrometers that measure humidity. These data collected from these devices would be combined and tested in our program and then if there was any serious chance of a natural disaster the program would send out a warning both to local news and to everyone's cell phones that people are both notified about the disaster and what to do to prepare for it. If someone in an area wanted to use other sensors or make the system compatible with other natural disasters the format is simple enough that they would probably be able to modify the program for what they need. An example of this would be if they wanted to check for Tsunamis they could add Tsunamis in with only a couple lines of code and the addition of a sensor that detects underwater earthquakes. This allows our system to be easily modified to best suit its area of use.

Decision Matrix Chart – Natural Disaster Early Warning System								
Design Criteria	Safe	Reliability	Performance	Simplicity and Maintenance	Easy to control and access	Low manufacturing and maintenance cost		Sum
Weight Factor	0.25	0.25	0.15	0.10	0.15	0.10		1.0
Solutions								
Solution 1	5 / 1.25	6 / 1.50	4 / 0.60	4 / 0.40	3 / 0.45	4 / 0.40		4.60
Solution 2	6 / 1.50	7 / 1.75	4 / 0.60	4 / 0.40	5 / 0.75	4 / 0.40		5.40
Final Solution	7 / 1.75	8 / 2.00	7 / 1.05	7 / 0.70	6 / 0.90	5 / 0.50		6.90

Table 3.3.0: Decision Matrix.

3.3.1 Components

For the components for our final design, we used a total of 5 different header and cpp files that are named main.cpp, Ews.h, Ews.cpp, Details.h, Details.cpp, and have their own functions within the program that have specific functions. The main header file called Ews.h is where the program sets up the units, a function that is creating the numbers, and a function that assigns those values to the variables. This cpp file is where the program will make a file called info.dat, and use a random number generator that stores and assigns the values from the generator to specific variables that will help determine the disaster. The file will then keep the data stored when the file closes so that later on in the program, the values can be extracted for other functions that help determine the disaster. Figures of the code for this component can be found in Appendix A on page 27. The inherited header file is where the program setups what disaster it could be. The inherited cpp file called Details.cpp, is where the program determines and displays information of the given disasters. The first function in this cpp file, the program will determine which disaster will occur by reading the values generated by the random number generator in the file, and check specific if-statements to see what disaster it is. Once it figures out what disaster it is, it will return to a specific function called from in the if-statements, and cout specific instructions to the users on how to prepare and to be safe, as well as website links for users to check if they for more information about the given natural disaster. Figures of the code for this component can be found in Appendix B-D on pages 28-29. If none of the values generated by a random number generator meet the requirements for any disaster, it will go to a function that will let the user know that no disaster will occur. It will still provide information to the user that they should prepare if one does occur. The main application called main.cpp, which acts as the int main of the program, is where the program starts and goes through the functions located in the main class and inherited class. After the program is finished going through the information about the natural disaster, it will output a thank you towards the user for using the program, followed by the names of group members who created the program. Figure of this component can be found in Appendix E. Our program contains 5 floats each of which have four bytes for a total of twenty bytes. We also have an int with another 4 bytes and a bool with 1 byte. adding these bytes causes the total bytes to go up to 25. The output size of the program is 1.848 MiB.

If we were to implement this design we would add in multiple sensors. The first sensor we would add would be multiple digital thermometers that would send information back about the temperature of the area. The second sensor we would add would be multiple hygrometers to measure the humidity of the area. The last sensor we would add would be multiple anemometers these would measure the wind speed of the area. The reason we would need multiple of each sensor would be to make sure that the information we are getting from these tools is not a false positive. These sensors would send back data to a computer so we would also need some form of computer to run the program and send the results.

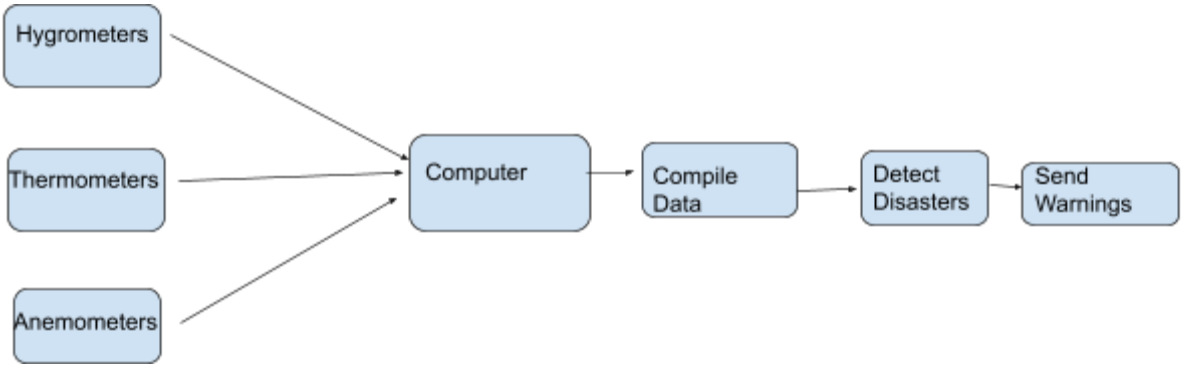

Figure 3.3.5: Block Diagram.

3.3.2 Features

The features for our final design is that it is reliable and can be used by anybody. With this design, the program will have multiple files with different information in them. The information in these files is data that our team found in our online research of different kinds of natural disasters. This makes the data collected reliable and realistic to the real world, rather than asking the user to input data themselves which takes time. When the program is running and uses the data in these files, it will then display if there is, or if there is not a natural disaster about to occur. If there is a natural disaster, it will display a warning and instructions for the user to do in an aesthetic way that is easy for the user to read and to understand. The program will also list links to different websites for more information on the given natural disaster in case the user wants to know more about the natural disaster and what to expect. Additionally, before the program runs as an early warning system, it will check to see if the files work.

3.3.3 Environmental, Societal, Safety, and Economic Considerations

When designing our early warning system for natural disasters, we took into consideration the program to make sure that it was environmentally friendly, safe to use, and inexpensive.

For environmental considerations, our program allows users to predict more easily different environmental factors for natural disasters such as wind speeds, dryness, humidity, and temperatures outside. The program will act as a safeguard for users to be safe, as well as indicate what is happening in the environment which will provide early warnings to users about the environment. For example, if the early warning system program indicated that there is going to be a forest fire, it tells the user that the forests are very dry and have low humidity which is enough for a forest to catch on fire.

For societal considerations, our program will help prepare users with warnings and instructions of the incoming natural disaster, but it will also provide communities, cities, or other places with people in the area a way to prepare for the natural disaster. That way the people who work or represent these places can be able to warn others of the potential danger of the incoming natural disaster and broadcast what people should do so that everyone can prepare and take the necessary precautions to be safe.

For safety considerations, we chose to use the most accurate and reliable information found on the internet about safety during a natural disaster. To find this information, we chose to use websites like the official website of the National Weather Service, where we got most of our information about how to be safe during different kinds of natural disasters. The National Weather Service provides all kinds of information such as where a disaster can hit, different characteristics of disasters, and how to tell when a disaster is about to come. We also considered having our program easy to use for users, as the user could have problems with other kinds of early warning systems for natural disasters as it requires them to go to different links. This could waste time for users and that an incoming natural disaster could be approaching. That is why our easy-to-use program, users will be able to get all their answers right away with warnings and instructions about the incoming natural disaster so that they can be safe [10].

Figure 3.3.6: Image of People Helping During a Major Flood [11].

For economic considerations, we chose to make an inexpensive and cost-effective program that gives all the basic information about if or if not, a natural disaster is going to hit and what the users should do. Most weather-warning apps and websites give the most information about natural disasters but are behind a paywall and require a subscription to access all the features. Our early warning system is cheap, cost-effective, and doesn't require a subscription to access all the features.

3.3.4 Limitations

The limitations for the final design is that there was no way to predict or to measure some of the well-known natural disasters all around the world. As an example, our group had no way of measuring or predicting earthquakes or tsunamis. To this day, scientists have never predicted a major earthquake or tsunamis, they can only predict and calculate the probability of where these disasters can take place [12]. With our program, we were unable to find detailed data of different units to measure for earthquakes and tsunamis, only the severity of the two disasters.

Figure 3.3.7: Aftermath of Earthquake and Tsunami in Japan 2011 [13].

This led our group to not include these two main natural disasters into our program, as there was no way for us to accurately predict if either of the two disasters could occur. Our program only has the capabilities of using a random number generator to store data that has some type of measurement in units, and that it can be predicted before the disaster hits such as wind speed and temperature. In the real world, early warning systems for natural disasters use sensors that can constantly measure seismic waves for earthquakes by using seismographs that can measure the slightest ground vibration in the Earth [14]. This is one of the only ways to see if an earthquake is about to happen and to send warnings to people just before it happens.

4 Team Work

4.1 Meeting 1

Time: February 18, 2021, 1:30 pm to 2:00 pm

Agenda: Discuss the Project and Choose Topic for an Early Warning System

Team Member	Previous Task	Completion State	Next Task
Luka Aitken	N/A	N/A	Researching/Assign Tasks
Braden Wielgoz	N/A	N/A	Researching/Assign Tasks
Toma Aitken	N/A	N/A	Researching/Assign Tasks

4.2 Meeting 2

Time: March 2, 2021, 6:00 pm to 6:30 pm

Agenda: Review and Talk about Research on Natural Disasters and Early Warning Systems

Team Member	Previous Task	Completion State	Next Task
Luka Aitken	Researching/Assign Tasks	90%	Writing Report: Problem Definition, Functions, Objectives
Braden Wielgoz	Researching/Assign Tasks	85%	Writing Report: Problem Definition
Toma Aitken	Researching/Assign Tasks	100%	Writing Report: Introduction, Constraints

4.3 Meeting 3

Time: March 9, 2021, 7:00 pm to 8:30 pm

Agenda: Begin Writing Part 1 of the Report.

Team Member	Previous Task	Completion State	Next Task
Luka Aitken	Writing Report: Problem Definition, Functions, Objectives	85%	Making Gantt Chart, function tree
Braden Wielgoz	Writing Report: Problem Definition	80%	Writing draft program
Toma Aitken	Writing Report: Introduction, Constraints	90%	Making Objective Tree, Requirements Table

4.4 Meeting 4

Time: March 20, 2021, 6:30 pm to 8:00 pm

Agenda: Editing Part 1 of Report and Discuss Part 2 of the Report

Team Member	Previous Task	Completion State	Next Task
Luka Aitken	Making Gantt Chart, function tree	85%	Writing Report: Considerations
Braden Wielgoz	Writing draft program	65%	Writing Report: Solution 2
Toma Aitken	Making Objective Tree, Requirements Table	85%	Writing Report: Solution 1, Considerations

4.5 Meeting 5

Time: March 27, 2021, 6:00 pm to 8:30 pm

Agenda: Discuss Draft Copy of Program, Continue Working on Report

Team Member	Previous Task	Completion State	Next Task
Luka Aitken	Writing Report: Considerations	90%	Begin writing Final Program, Writing Report: Limitations
Braden Wielgoz	Writing Report: Solution 2	90%	Begin writing Final Program, Writing Report: Final Solution
Toma Aitken	Writing Report: Solution 1, Considerations	95%	Begin writing Final Program, Writing Report: Features, Final Solution

4.6 Meeting 6

Time: March 28, 2021, 4:00 pm to 5:30 pm

Agenda: Finish writing part 2 of Report and Go over Final Program

Team Member	Previous Task	Completion State	Next Task
Luka Aitken	Begin writing Final Program, Writing Report: Limitations	90%	Writing Report: Components, Decision Matrix
Braden Wielgoz	Begin writing Final Program, Writing Report: Final Solution	90%	Writing Report: Block Diagram
Toma Aitken	Begin writing Final Program, Writing Report: Features, Final Solution	90%	Writing Final Program: Adding information about natural disasters

4.7 Meeting 7

Time: April 4, 2021, 5:00 pm to 6:30 pm

Agenda: Edit Part 2 of Report and Test Final Program.

Team Member	Previous Task	Completion State	Next Task
Luka Aitken	Writing Report: Components, Decision Matrix	95%	Begin writing PowerPoint, Writing Report: Conclusion
Braden Wielgoz	Writing Report: Block Diagram	100%	Begin writing PowerPoint
Toma Aitken	Writing Final Program: Adding information about natural disasters	85%	Continuing to add details to Program, Writing Report: Conclusion

4.8 Meeting 8

Time: April 12, 2021, 5:00 pm to 6:30 pm

Agenda: Finish Final Program and Begin PowerPoint Slides.

Team Member	Previous Task	Completion State	Next Task
Luka Aitken	Begin writing PowerPoint, Writing Report: Conclusion	90%	Working on PowerPoint Slides
Braden Wielgoz	Begin writing PowerPoint	80%	Working on PowerPoint Slides
Toma Aitken	Continuing to add details to Program, Writing Report: Conclusion	90%	Working on PowerPoint Slides, Editing Video Presentation

4.9 Meeting 9

Time: April 14, 2021, 5:00 pm to 6:30 pm

Agenda: Edit the Report and Finish making PowerPoint Presentation.

Team Member	Previous Task	Completion State	Next Task
Luka Aitken	Working on PowerPoint Slides	100%	Project Finished
Braden Wielgoz	Working on PowerPoint Slides	100%	Project Finished
Toma Aitken	Working on PowerPoint Slides, Editing Video Presentation	100%	Project Finished

5 Project Management

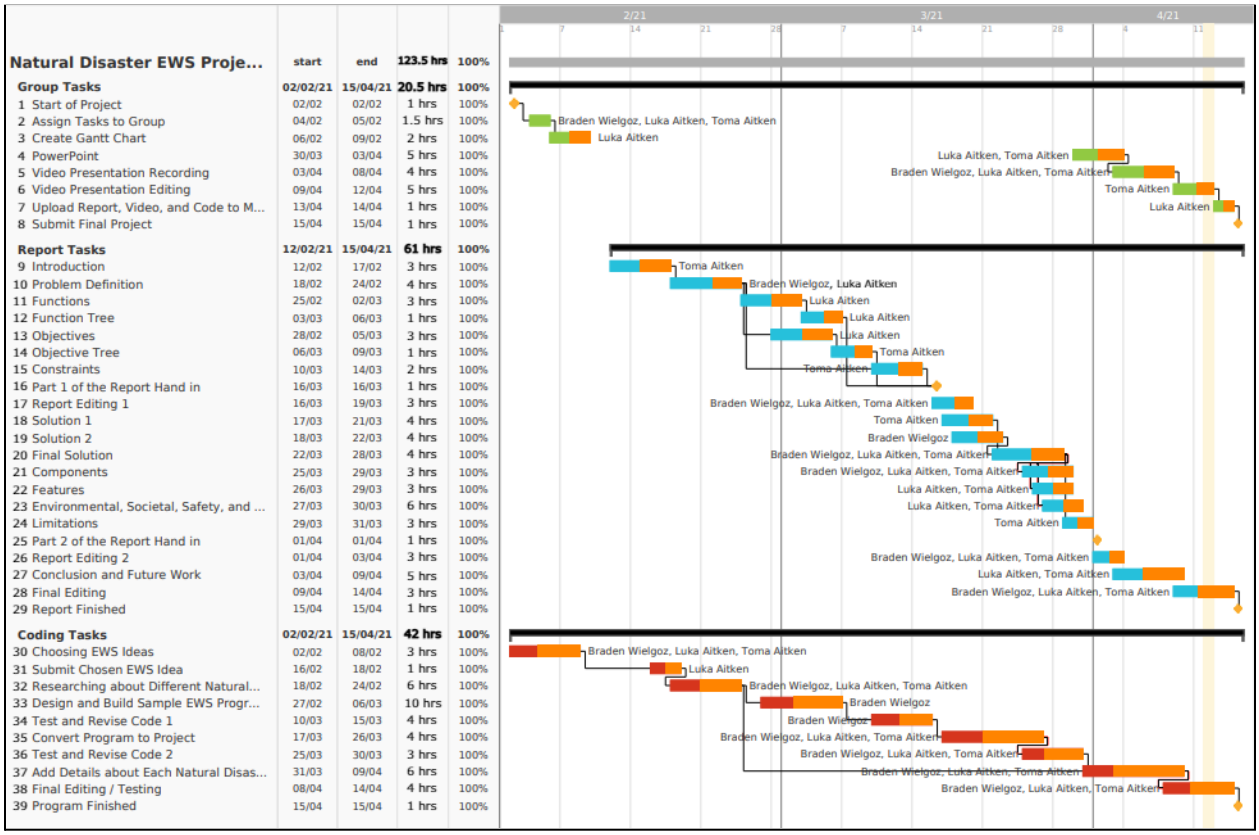

Table 5.0: Gantt Chart.

6 Conclusion and Future Work

In conclusion, we achieved plenty of considerable designs on how to create an early warning system for natural disasters. First off, we came with a couple of solutions that could be used to make the best possible early warning system for natural disasters, while following all the constraints. Our objective was to code and create a prototype of an early warning system for natural disasters while making it cheap and accessible for users who believe that a natural disaster will occur.

From our research, we found that natural disasters have a negative impact that is significant on the growth and poverty in multiple communities [15]. With this, many people struggle trying to re-establish communities due to loss of homes, social and economic problems, injuries, and even the loss of people [16]. By having an early warning system for natural disasters, these communities could better prepare for the oncoming disaster, protect infrastructures and protect the people around the community who can't prepare when a disaster occurs.

Figure 6.0: People Helping Prepare for Natural Disaster [17].

From researching the importance and the need of an early warning system for natural disasters, we developed three different, yet similar programs with each one having its advantages and disadvantages. Our first program was more focused on asking certain questions about a specific disaster. The program would ask the user what disaster they want to check and the program would give questions based on the disaster. After receiving the answers from the user, the program would give the results to the user if or if not that disaster will happen. This program was very simple but had many problems. First off, the program would waste the user's time as the program requires the user to enter the information about the disaster. Secondly, the program would use lots of bytes as it used many if statements to determine if the disaster will happen. For our second solution, the program was more focused on using files to store all the data. The program would read the files created in the program and determine if or if not a disaster would occur and give important details about the disaster and what to do when preparing. This program is simple and small, but it had many problems when getting the information. The program used too many if and else statements to determine the disaster.

Additionally, the program did not have any classes and was not built in project format, which made the program load longer and didn't follow the required constraints.

In the final design, we achieved all of our functions, objectives and constraints that we listed in our design problem. Our final solution used multiple files to make the program fast and helpful for people who could be in danger. This design can check four different natural disasters, with all types of tornadoes and hurricanes by using random number generators as we cannot predict the temperature or wind speeds. With each disaster and the different types, we made sure to use reliable information to check if the disasters could occur, as well as giving reliable information to better prepare if the disasters does occur. Additionally, after the program outputs the answer and tells the user what to prepare, we added websites at the end of each disaster if the user wants to know more information about the disaster and how to prepare and be safe.

For future improvements to the final design, we would want to make more inherited files for all the specific disasters. This will make the program much faster and more reliable for the user to see if a disaster is coming. It will also help organize all the code within the program so that if needed, as it would be easier to fix the code or even add more details for each disaster. To further improve our early warning system, we would add more types of natural disasters that can be predicted before the disaster happens. Finally, we would use other c++ features that we learned over this course to help make the program even faster.

7 References

1. J. Horney, I. Karaye, A. Abusabara, S. Gearhart, S. Grabich, M. Perez-Patron. “The Impact of Natural Disasters on Suicide in the United States, 2003–2015.” 09-Oct-2020. [Online]. Available: https://econtent.hogrefe.com/doi/10.1027/0227-5910/a000723 (Accessed March 9, 2021)
2. H. Ritchie, M. Roser. “Natural Disasters.” Our World in Data, https://ourworldindata.org/natural-disasters#:~:text=Natural%20disasters%20kill%20on%20average%2060,000%20people%20per,before%20a%20large%20disaster%20event%20claims%20many%20lives (Accessed March 9, 2021)
3. S. Kilpatrick. “Tornado aftermath: A storm and its fallout, by the numbers.” OTTAWA CITIZEN. https://ottawacitizen.com/news/local-news/tornado-aftermath-a-storm-and-its-fallout-by-the-numbers (Accessed April 14 2021).
4. L. Pearson. “Early warning of disasters: Facts and figures.” SciDev.Net. https://www.scidev.net/global/features/early-warning-of-disasters-facts-and-figures-1/ (Accessed April 14 2021).
5. “Greater Impact: How Disasters Affect People of Low Socioeconomic Status.” SAMHSA. https://www.samhsa.gov/sites/default/files/dtac/srb-low-ses_2.pdf (Accessed April 14 2021)
6. “World’s deadliest tropical cyclone was 50 years ago.” World Meteorological Organization. https://public.wmo.int/en/media/news/world%E2%80%99s-deadliest-tropical-cyclone-was-50-years-ago#:~:text=Cyclone%20Bhola%2C%20which%20grew%20from,of%20the%20Bay%20of%20Bengal. (Accessed April 14 2021).
7. H. Koundajian. “From the archive, 18 November 1970: Fight for life as aid reaches stricken isles.” The Guardian. https://www.theguardian.com/world/2014/nov/18/bhola-cyclone-manpura-east-pakistan-bangladesh-1970 (Accessed April 14, 2021).
8. D. Rogers, V. Tsirkunov, “Global Assessment Report on Disaster Risk Reduction Costs and benefits of early warning systems.” THE WORLD BANK. http://documents1.worldbank.org/curated/en/609951468330279598/pdf/693580ESW0P1230aster0Risk0Reduction.pdf (Accessed April 14 2021)
9. “Early Warning Systems.” United Nations. https://www.un.org/en/climatechange/climate-solutions/early-warning-systems (Accessed April 14 2021)
10. “National Weather Service Safety Tips.” NATIONAL WEATHER SERVICE. https://www.weather.gov/safety/ (accessed April 14 2021)
11. “Occupational Safety In Natural Disasters.” EKU Online. https://safetymanagement.eku.edu/blog/occupational-safety-in-natural-disasters/ (Accessed April 14, 2021)

12. “Can you predict earthquakes?.” USGS. https://www.usgs.gov/faqs/can-you-predict-earthquakes?qt-news_science_products=0#qt-news_science_products (Accessed April 14 2021)
13. M. Dunham. “Japan earthquake and tsunami of 2011.” Britannica. https://www.britannica.com/event/Japan-earthquake-and-tsunami-of-2011/Relief-and-rebuilding-efforts (Accessed April 14, 2021)
14. C. Johnson, M. Affolter, P. Inkenbrandt, C .Mosher “Measuring Earthquakes.” GEOSCIENCES LibreTexts. https://geo.libretexts.org/Courses/Gettysburg_College/Book%3A_An_Introduction_to_Geology_(Johnson_Affolter_Inkenbrandt_and_Mosher)/03%3A_Crustal_Deformation_and_Earthquakes/3.07%3A_Measuring_Earthquakes#:~:text=To%20accurately%20study%20seismic%20waves,fixed%20solidly%20to%20the%20ground (Accessed April 14 2021)
15. N. Laframboise, B. Loko. “Natural Disasters : Mitigating Impact, Managing Risks.” 2012. [Online]. https://eds.a.ebscohost.com/eds/detail/detail?vid=2&sid=d9436db7-a571-44bc-9cca-d833836c8fff%40sdc-v-sessmgr01&bdata=JnNpdGU9ZWRzLWxpdmUmc2NvcGU9c2l0ZQ%3d%3d#AN=568176&db=nlebk (Accessed April 12, 2021)
16. E. Rodriguez-Oreggia, A. De La Fuente, R. De La Torre, H. Moreno. “Natural Disasters, Human Development and Poverty at the Municipal Level in Mexico.” Mar-2013. [Online]. https://eds.b.ebscohost.com/eds/detail/detail?vid=3&sid=8fe66185-69cd-436e-9ffe-b46546e1759f%40pdc-v-sessmgr03&bdata=JnNpdGU9ZWRzLWxpdmUmc2NvcGU9c2l0ZQ%3d%3d#AN=85318865&db=bth (Accessed April 12, 2021)
17. “BEING PREPARED FOR NATURAL DISASTERS: A SCOUT’S GUIDE.” SCOUTS. https://www.scout.org/node/380606 (Accessed April 14 2021)

8 Appendix

```h
#ifndef EWS_H
#define EWS_H
using namespace std;

class NaturalDisasterEWS
{
    public:
        float temp;
        float lhumidity;
        float hhumidity;
        float windspeed;
        float wwindspeed;
        int coast;

        NaturalDisasterEWS();
        void getData();
        void setData();


};
#endif

```

Ews.h Code

```cpp
#include "EWS.h"
#include <iostream>
#include <fstream>
#include <ctime>
#include <cstdlib>
using namespace std;

NaturalDisasterEWS :: NaturalDisasterEWS()
{
    temp = 0;
    lhumidity = 0;
    hhumidity = 0;
    windspeed = 0;


}

void NaturalDisasterEWS :: getData()
{
    ofstream outfile;   //creating randomized text file
    srand(time(0));
    outfile.open("info.txt");
    outfile<<(rand() % 100) - 50<<endl; //temperature
    outfile<<(rand() % 100)<<endl;  //high humidity
    outfile<<(rand() % 100)<<endl;  //low humidity
    outfile<<(rand() % 143)<<endl;  //windspeed
    outfile<<(rand() % 300)<<endl;  //Wwindspeed
    outfile<<(rand() & 1)<<endl; //near coast
    cout<<"File Randomly Generated"<<endl;
    cout<<endl;
    outfile.close();
}
```

Ews.cpp Code

```cpp
void NaturalDisasterEWS :: setData()
{
    ifstream infile;
    infile.open("info.txt");

    if(infile.fail())   //checks if file exists
        cout<<"File doesn't exist"<<endl;
    else
        cout<<"File exists"<<endl;

    if(!infile.is_open())
    {
        infile.open("info.txt");
    }
    infile >> temp >> hhumidity >> lhumidity >> windspeed >> wwindspeed >> coast;
    infile.close();
}

```

Ews.cpp Code (continued)

Appendix A: Image of Ews.h and Ews.cpp File Code.

```h
#ifndef DETAILS_H
#define DETAILS_H
#include "Ews.h"
using namespace std;

class Details : public NaturalDisasterEWS
{
    public:
        bool disaster;

        Details();
        void Results();

        void ForestFire();
        void Hurricane();
        void Tornado();
        void Blizzard();

};
#endif
```

Appendix B: Image of Details.h File Code.

```cpp
#include "Details.h"
#include <iostream>
using namespace std;

Details :: Details()
{

}

void Details :: Results()
{


    //Tornado, Hurricane, Forest Fire, Blizzards
       if(temp >= 25 && lhumidity <= 40)
       {
         disaster = true;
         return ForestFire();
       }

       if ((temp >= 0) && (temp <= 35) && (wwindspeed >= 119) && (wwindspeed <= 300) && (coast == 1))
       {
           disaster = true;
           return Hurricane();
       }

       if ((temp >= 0) && (temp <= 35) && (windspeed >= 32) && (windspeed <= 143) && (coast == 0))
       {
           disaster = true;
           return Tornado();
       }

       if(temp <= -20 && windspeed >= 16)
       {
           disaster = true;
           return Blizzard();
       }

       else
       {
           cout<<"Good news, there are no disasters that are going to occur nearby. You are safe to do whatever you want,"<<endl;
           cout<<"but you should always make sure that you are prepared if a disaster does occur nearby."<<endl;
           cout<<endl;
           cout<<"If you want to find more information about natural disasters and what to do, visit these websites: "<<endl;
           cout<<"https://www.weather.gov/safety/"<<endl;
           cout<<"https://www.thepersonal.com/blog/-/what-to-do-in-case-of-a-natural-disaster"<<endl;
           cout<<endl;
       }
}
```

Appendix C: Image of Details.cpp File Code.

```
52  void Details :: ForestFire()
53  {
54      cout<<"There will be a Forest fire coming your way."<<endl;
55      cout<<"Forest Fires are very dangerous that can: "<<endl;
56      cout<<"\t-Produce harmful smoke"<<endl;
57      cout<<"\t-Destroy trees and soil"<<endl;
58      cout<<"\t-Damage Homes"<<endl;
59      cout<<endl;
60      cout<<"To prepare, you should: "<<endl;
61      cout<<"\t-Prepare an emergency kit"<<endl;
62      cout<<"\t-Practice Fire Drills"<<endl;
63      cout<<"\t-Make sure that each room has a smoke detector"<<endl;
64      cout<<"\t-Plan an escape route"<<endl;
65      cout<<"\t-Always be ready to leave"<<endl;
66      cout<<endl;
67      cout<<"To better prepare for forest fires that may be coming, you can visit these websites for more information: "<<endl;
68      cout<<"https://mylandplan.org/content/good-and-bad-forest-fires"<<endl;
69      cout<<"https://www.getprepared.gc.ca/cnt/hzd/wldfrs-bfr-en.aspx"<<endl;
70
71  }
```

Appendix D: Image of Details.cpp File Code of a Function.

```
1   #include "EWS.h"
2   #include "Details.h"
3   #include <iostream>
4   #include <fstream>
5   #include <ctime>
6   #include <cstdlib>
7
8   int main(int argc, char** argv)
9   {
10      Details d1;
11
12      d1.getData();
13      d1.setData();
14      d1.Results();
15
16
17
18  cout<<endl;
19  cout<<"Thank you for using our early warning system program."<<endl;
20  cout<<"Created by Group 3: "<<endl;
21  cout<<"\t-Luka Aitken"<<endl;
22  cout<<"\t-Toma Aitken"<<endl;
23  cout<<"\t-Braden Wielgoz"<<endl;
24
25
26      return 0;
27  }
```

Appendix E: Image of main.cpp File Code.