THOMPSON RIVERS UNIVERSITY

SENG 1210 – Programming for Engineers II

Early Warning System

Andrei Vivar (T00668428)

Raiden Yamaoka (T00661480)

Jack Ukitetu (T00647975)

Date submitted

Table of Contents

List of Figures

List of Tables

1 Introduction

Many places around the world face the consequences of disastrous weather conditions; this is something we can not control. However, it is something we can warn others about to prepare in advance. On top of these very real struggles that everyone faces, some have even more detrimental weather conditions. Places like Japan and coastal countries have it worse in that they have typhoons, hurricanes, tsunamis and earthquakes and this happens to be the most common in these coastal countries with risk rates as high as 36.28%. Globally approximately 3 billion people live in or near coastal cities and countries and have to live with that risk. When choosing somewhere to live, a nice location is always considered. In most cases, a coastal home will have a greater cost being in that nicer location. With this early warning system, you could have additional time to prepare for these disasters making the net loss of belongings and lifeless.

We would like to give these people a helping hand to hopefully get them to a place where they can feel safe in their own homes and have peace of mind to let them live freely. It helps to have something reassuring you of your safety, especially when you just move to a new place and even if you've lived and dealt with these disasters it can let you be at rest knowing something will tell you in advance. In the case of coastal cities and countries and those who live there given the large population, some people may not feel comfortable with one source of warning such as the news or government warning system. Our early warning system seeks to provide a viable product that will give them the extra closure that they desire. While also providing safety and peace of mind to let them have the time to collect all of their important belongings. This will all be achieved while delivering it in a cheap or free, simple to use, and long-lasting format because we know everyone deserves to be warned about something out of anyone's control. Our main goal is to relieve stress on anyone that lives in disastrous weather conditions.

2 Design Problem

2.1 Problem Definition

For this project, we were tasked to design an early warning system application for an area of our choice. For this project, we decided to choose an early warning system app for extreme weather conditions. This will help in places that are subjected to extreme weather conditions by making an application that will alert the user on the weather condition.

2.2 Design Requirements

2.2.1 Functions

- The function of this early warning system app is to display an early warning for those in near high-risk weather conditions. So it alerts a user on the weather condition, earthquakes and other natural disasters. Therefore, prevent the user from harm caused by these natural disasters.
- **Goal of the function:** The main goal is to design an application code that serves as an early warning system for extreme weather conditions.
    1. **Action:** The user should be able to tell the weather situation of a given day.
    2. **Functions:** The main functions of the device are:

        i. It detects extreme weather beforehand and alerts the user.

        ii. It also helps to display the weather situation for a given day.

        iii. It also detects the weather condition of other places
- **Behaviours:** A date is entered by the user and then it displays the weather condition of the day and any extreme weather condition coming soon.
- **Structure:** The application is made up of: various files which will contain a simulation of weather at different places, a main code for the input, a processing code that will process all data entered and an output which will display the result.

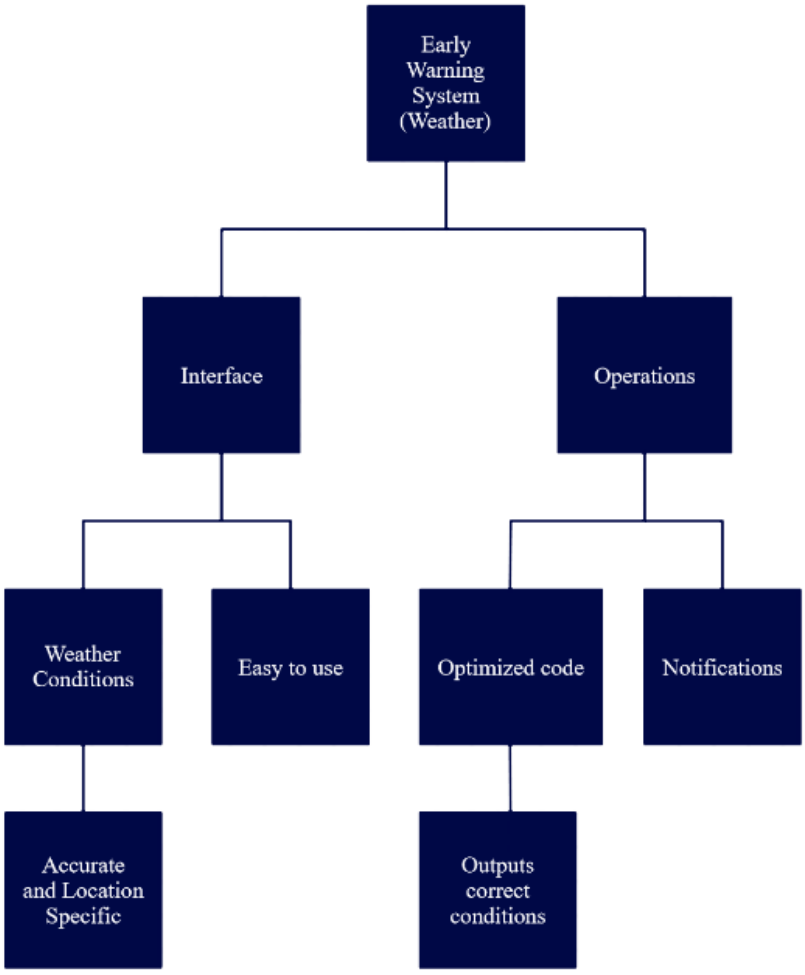

Figure 1. Function Tree

2.2.2 Objectives

Our objective is to make a project code that serves as an early warning system for the weather of a given region. For our objectives, we broke major ideas into individual sections: safety, cost, efficiency, and cool factor. For safety, there are two sub-sections: alerts users of weather conditions and non-breachable code. For the cost, our objective for our app is to be cheap or free. For efficiency, our objective for our app is to have fast code and be user-friendly, resulting in our app having many languages and be easy to use. Lastly, for the cool factor, our objective for our app is to have custom notifications and a custom interface that will suit the user's needs. Below is a visual representation of our objective tree:

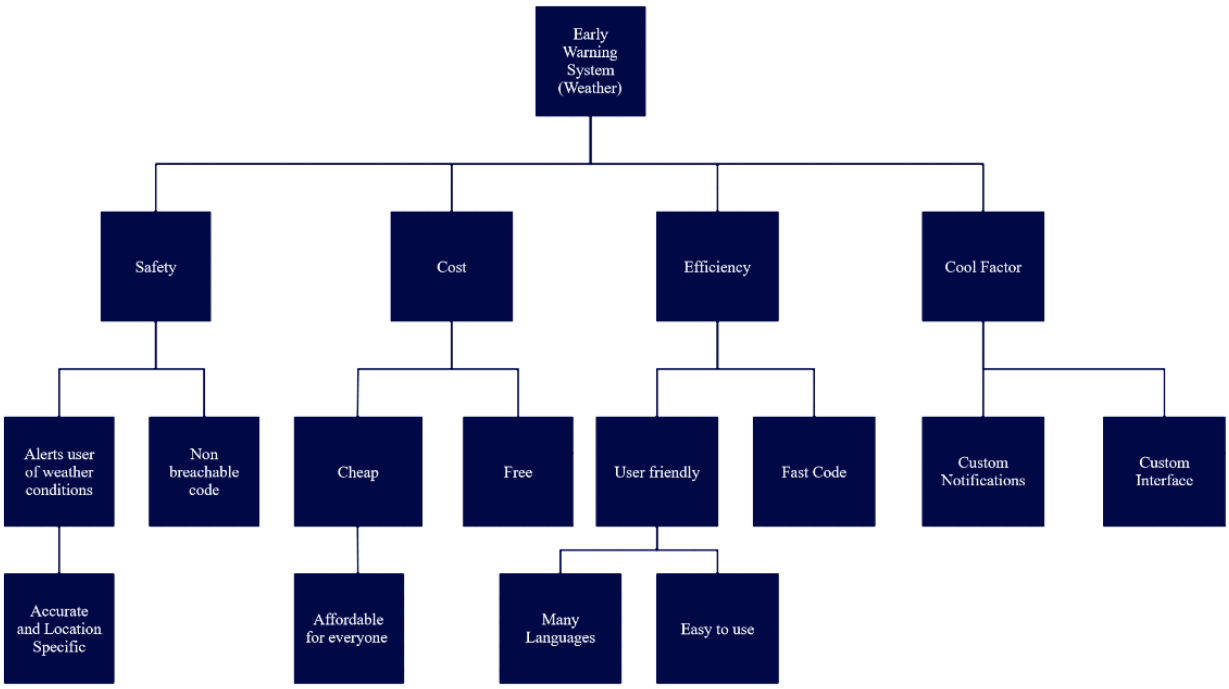

Figure 2. Objective Tree

2.2.3 Constraints

For our project's constraints, our professor provided us with several constraints. Our application should consider if not all, but at least four from the following factors of overarching characteristics of appropriate design: Economic Factors, Health and Safety, Regulatory Compliance, Reliability, Aesthetics, Sustainability, Environmental factors, Ethics, and Societal Impacts.

A. For economic factors, we consider how many people download the app and how much funding is required for mass upload of the application. This means a low-cost app is appropriate.
B. For health and safety, we want to keep computers safe; therefore we want a protected app that is not at risk of non-breachable hacks. For the app interface itself, the app should give a warning for severe weather conditions or any natural disasters.
C. For regulatory compliance, the app should follow the law of app making, and not disregard any rules against it for it to make public.

D. For reliability, our app should have code that is consistent and accurate displays each time the app is opened.
E. For aesthetics, our app should keep professionalism of the code and decent app interface.
F. For sustainability, the app should maintain its use over a long period of time.
G. For ethics, the app should be user friendly, meaning the app should not be hard to operate
H. For societal impacts, the app should help detect high-risk weather events.

3 Solution

3.1 Solution 1

Initially, we focused on crime reports for the early warning system but after research, we found out that it would be really hard to code on that topic because of its randomness so we decided to switch to a topic that can easily be estimated which was the weather. Although we had meetings and came up with an initial idea we had to drop it because of the complexity.

3.2 Solution 2

For Solution 2, we moved on to using an Early Warning System to predict natural disasters and has no weather-related functions such as days that are too hot or too cold. We were not satisfied with this solution because it did not contain all the elements or desired functions we wanted to accomplish; therefore we decided to better the solution and came up with our final solution.

3.3 Final Solution

For our final solution, we based it off on our second solution, but with improved features. The structure of our Early Warning System for our Final solution has a more complex and more featured design than our Solution 2; which includes weather warnings, possible earthquake warnings, and some natural disaster warnings.

Design Criteria	Accuracy	Reliability	Cost	Performance	User-friendly	Availability	Sum
Weighing Factor	0.19	0.17	0.13	0.17	0.15	0.19	1.0
Solution 1	6 / 1.14	6 / 1.02	9 / 1.17	6 / 1.02	5 / 0.75	9 / 1.71	6.81
Solution 2	8 / 1.52	6 / 1.02	9 / 1.17	7 / 1.19	6 / 0.90	9 / 1.71	7.51
Final Solution	9 / 1.71	9 / 1.53	10 / 1.30	8 / 1.36	7 / 1.05	10 / 1.90	8.85

Table 1. Decision Matrix

3.3.1 Components

No physical components were used due to the code-heavy nature of this project. Although in a real-world situation where we envision this to be a common phone application that would collect data for the web to make more accurate decisions. The data types used in this application were integers and floats with 4 bytes in integers and 4 bytes for float data types.

3.3.2 Features

Will take temperatures based on location and time of the user currently only using Kamloops temperatures due to the limitation of time and skills we have.

Will give a warning based on the current temperature when it gets too hot or too cold as well as warnings for major natural disasters.

3.3.3 Environmental, Societal, Safety, and Economic Considerations

Our team took into consideration into societal and cost the most, as we want this to be a free product for everyone to use. Having a free product means spending as little as possible, but also making it available to the vast majority of the population which is one of our main goals. Societally, we wanted to consider everyone and their struggles when it comes to using technology so our interface would be simple and output and input as little as possible to ensure everyone has no issues. Environmental considerations were discussed but not deemed important enough as this product will be entirely digital having no harm to the environment. Safety was another point we wanted to consider as the safety of the user is the main function of our product. We would ensure no data from the user is taken other than the location to provide accurate weather conditions and have a non-hackable code with no possible breaches.

3.3.4 Limitations

Although our solution is based on a complex-structured Early Warning System, the application is only currently useful in making early warnings in a certain area, which is Kamloops in this case. Also, this is only a prototype application and will require further work to make it more user-friendly. In order for the application to work to its full capacity, it requires an update for the data stored for Kamloops' weather condition, seismic activity, and natural disaster warnings for a specific year. As well as only using random number generators for statistics for natural disasters as having specific disaster warnings are not currently possible.

4 Team Work

Since this is a group project, you must have a fair distribution of tasks among yourselves. To this end, you must hold meetings to discuss the distribution of tasks and to keep a track of the project progress.

4.1 Meeting 1

Time: February 18, 2021, 2:00 pm to 3:13 pm

Agenda: Brainstorming and picking an area for the EWS

Team Member	Previous Task	Completion State	Next Task
Andrei Vivar	N/A	N/A	Project research
Raiden Yamaoka	N/A	N/A	Code & Project research
Jack Ukitetu	N/A	N/A	Code research

Table 2. Meeting 1.

4.2 Meeting 2

Time: February 25, 2021, 2:23 pm to 4:00 pm

Agenda: Starting the reports and research (background information)

Team Member	Previous Task	Completion State	Next Task
Andrei Vivar	Project research	30%	Working on the code
Raiden Yamaoka	Code & Project research	40%	Working on the report
Jack Ukitetu	Code research	40%	Working on the code

Table 3. Meeting 2.

4.3 Meeting 3

Time: March 3, 2021, 6:00 pm to 8:15 pm

Agenda: Working on the reports and code

Team Member	Previous Task	Completion State	Next Task
Andrei Vivar	Task 1	70%	Working on the code
Raiden Yamaoka	Task 2	70%	working on the code
Jack Ukitetu	Task 3	60%	Working on the report

Table 4. Meeting 3.

4.4 Meeting 4

Time: March 25, 2021, 4:00 pm to 5:50 pm

Agenda: Working on the reports and code

Team Member	Previous Task	Completion State	Next Task
Andrei Vivar	Working on the code	80%	Finishing report
Raiden Yamaoka	Working on the code	80%	Reviewing project
Jack Ukitetu	Working on the report	70%	Finishing code

Table 5. Meeting 4.

4.5 Meeting 5

Time: April 9, 2021, 3:00 pm to 6:00 pm

Agenda: Finishing the reports and code

Team Member	Previous Task	Completion State	Next Task
Andrei Vivar	Finishing report	100%	N/A
Raiden Yamaoka	Reviewing project	100%	N/A
Jack Ukitetu	Finishing code	100%	N/A

Table 6. Meeting 5.

5 Project Management

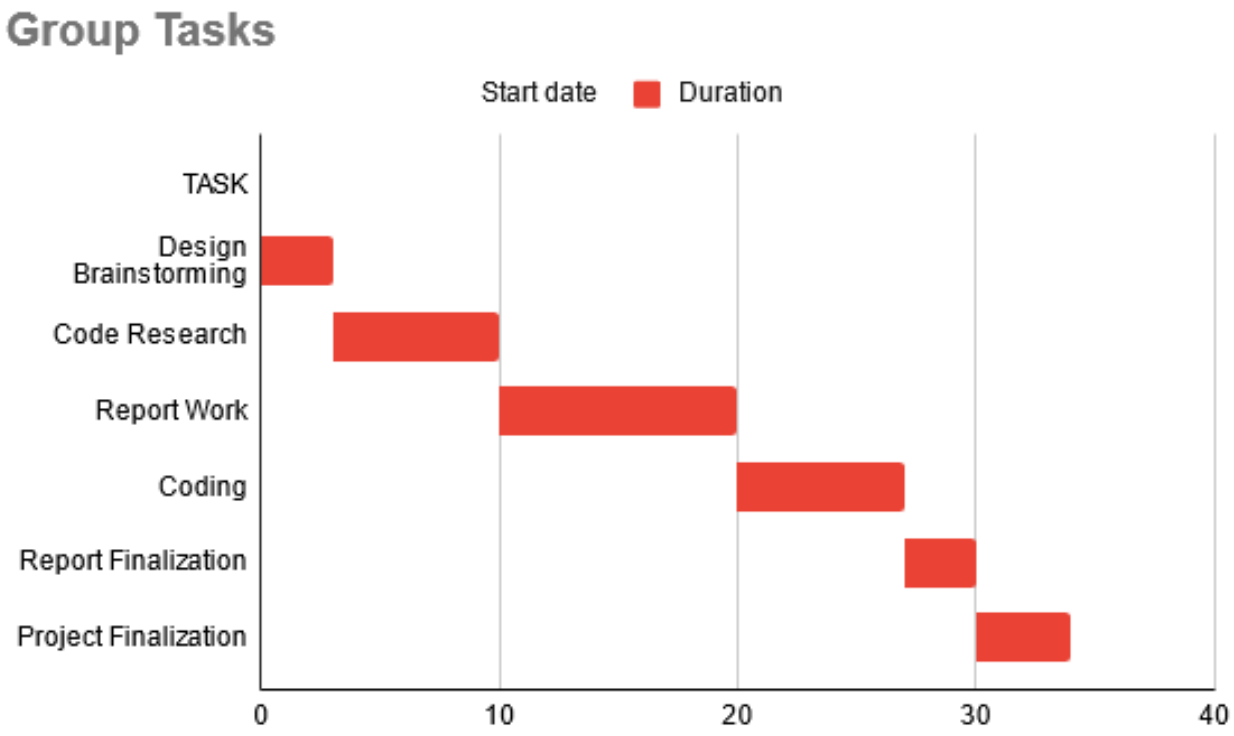

Figure 3. Gantt Chart

As a group our critical path of tasks was; Design Brainstorming, Code research, Report work, Coding, Report finalization and Project finalization. We gave ourselves two days slack for Design Brainstorming as it was not a lengthy task. Code research was given no additional slack days as all the code was found in the given week. Report work was given five slack days as the report is a key part of the completion of the project. The entire coding process was given three slack days to ensure that we had enough time to fix any bugs and change any code that was inefficient. Report and project finalization were both given an additional two days for review by all group members if needed,

6 Conclusion and Future Work

- As a group, we achieved multiple big milestones. We coded a function app that has many features including a Kamloops and monthly based weather checker as well as learnt many cooperative methods in how to work well as a group. Our final design has many functions that we wanted to include but is still missing some features that we wanted to include. The features that we have included are; an extreme weather condition checker that searches for things such as earthquakes, tornadoes and weather that is either too hot or too cold using the month and Kamloops-based data. The objectives we achieved are based on safety, cost, efficiency, and cool factor. For safety the main goals we accomplished were that our code is non-breachable and provides an accurate warning based on the weather conditions, however, we did not find a way to base it off locations outside of Kamloops. Secondly for cost, we achieved our only goal in regards to cost and made our product free for anyone to use all while doing it at no personal costs. Third, our code is only missing one of our efficiency goals, that being we only have it in one language making it not as available to everyone as we wanted. However, all of our other goals were accomplished as the code runs at amazing speeds with few halts and has a very basic interface so that anyone can use it without any issues. Finally cool factor, our code has our own personally designed custom interface looking as clean as possible using the given DEV C++ interfaces. Although we could not add custom notifications as we did not design them it was part of our plan to have them in the application.
- Our final design is intended to be used as a mobile phone application therefore more planning and research would have to be done in order for it to be used as a true application. Improvements to the code for next time would be adding multiple location-based temperatures monthly to keep accurate and safe measures for those using the app. Instead of using a random number generator for the earthquakes using seismic activity in that specific area to alert users to unusual activity. A similar system would be put in place for tornadoes. As well as coding notifications based on the weather conditions for the area that the user is in, increasing safety measures. In the application interface, a better-looking design would be used including possible graphics and images as well as adding colour and image changing options for the user.

7 References

T. Onur and M. Seemann, “Probabilities of Significant Earthquake ... - IIT Kanpur,” *PROBABILITIES OF SIGNIFICANT EARTHQUAKE SHAKING IN COMMUNITIES ACROSS BRITISH COLUMBIA: IMPLICATIONS FOR EMERGENCY MANAGEMENT*, 04-Aug-2004. [Online]. Available: https://www.iitk.ac.in/nicee/wcee/article/13_1065.pdf. [Accessed: 02-Apr-2021].

used in the random number generator for earthquake chances.

W. N. Inc., “Kamloops, BC Weather,” *The Weather Network*. [Online]. Available: https://www.theweathernetwork.com/ca/monthly/british-columbia/kamloops?year=2020&month=7&dispt=calendar-container-monthly. [Accessed: 13-Apr-2021].

Used in data collection for weather conditions daily.