**Traffico**

**System Description**

**Document Version: V 1.2**

**Date: [31.10.2022]**

Mafruha Nuhary

Maruf Rayhan

Fahid Ahmed Khan

Contents

|  |  |  |
| --- | --- | --- |
| 1 | Introduction .................................................................................. | 3 |
| 2 | Overview..................................................................................... | 5 |
| 3 | Technology .................................................................................. | 6 |
| 4 | Prototype Description.................................................................................. | 7 |
| 5 | Interfaces and services..................................................................................... | 19 |
| 6 | Code and environment setup.......................................................................................... | 25 |

7 Conclusion................................................................................ 30

# Introduction

Software documentation is the information that explains the product to the individuals who create, deploy, and utilize it during the software development process. Software documentation is a technique for programmers and engineers to formally describe their product and the steps they took to develop it.

In this software project, an application is built which will monitor how the weather affects road maintenance and condition. This application will be a combination of a weather app and a road condition monitor. This application will not only provide information about the road status affected by the weather as well as the weather conditions.

This document will record the design process of the software. A detailed view of the software prototype and future decisions about building the software will be stated in this design document. Firstly, the whole prototype will be shown and then the different components, modules, and classes will be described. The document will serve to show the big picture.

## 1.2 Version Control

Please document all changes made to this document since the initial distribution.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date** | **Version** | **Author** | **Section** | **Amendment** |
| 31.10.2022 | V1.2 | Mafruha Nuhary |  | New sections were added based on the implementation |
| 08.12.2022 | V1.3 | Mafruha Nuhary |  | New feature description |

# Overview

Traffico allows users to receive weather predictions, hourly forecasts, and road conditions. The application fetches its data from open-source API, digitraffic. Through this users will easily obtain,

* Road maintenance data
* Road condition forecast
* Traffic messages

## Service description

The software designed and implemented will be used to track how the weather impacts road maintenance and condition. Particularly during the winter, the weather has a direct impact on the amount of maintenance needed and the state of the roads. The program will permit road weather predictions and condition forecasts separately. With this application, users will also be able to get whether announcements in different regions. Also, this application will enable users to make road reports so other users can benefit from it.

The user can select the type of traffic data they want to view from a list of alternatives, and their selection will be visualized. Users can choose from graphs and plots for visualizations.

# Technology

PyQt5 was used to create the GUI application Traffico, which displays information about traffic, road conditions, and maintenance.

PyQt is a GUI widget toolkit. Qt, one of the most effective and well-liked cross-platform GUI libraries, has a Python interface. The QtGui module provides all the graphical features, but the QtCore module only contains non-GUI functionality for working with files and directories, etc. There are also modules for working with XML (QtXml), SVG (QtSvg), SQL (QtSql), and other file types.

Dependencies used in this project,

* PyQt5
* matplotlib
* requests

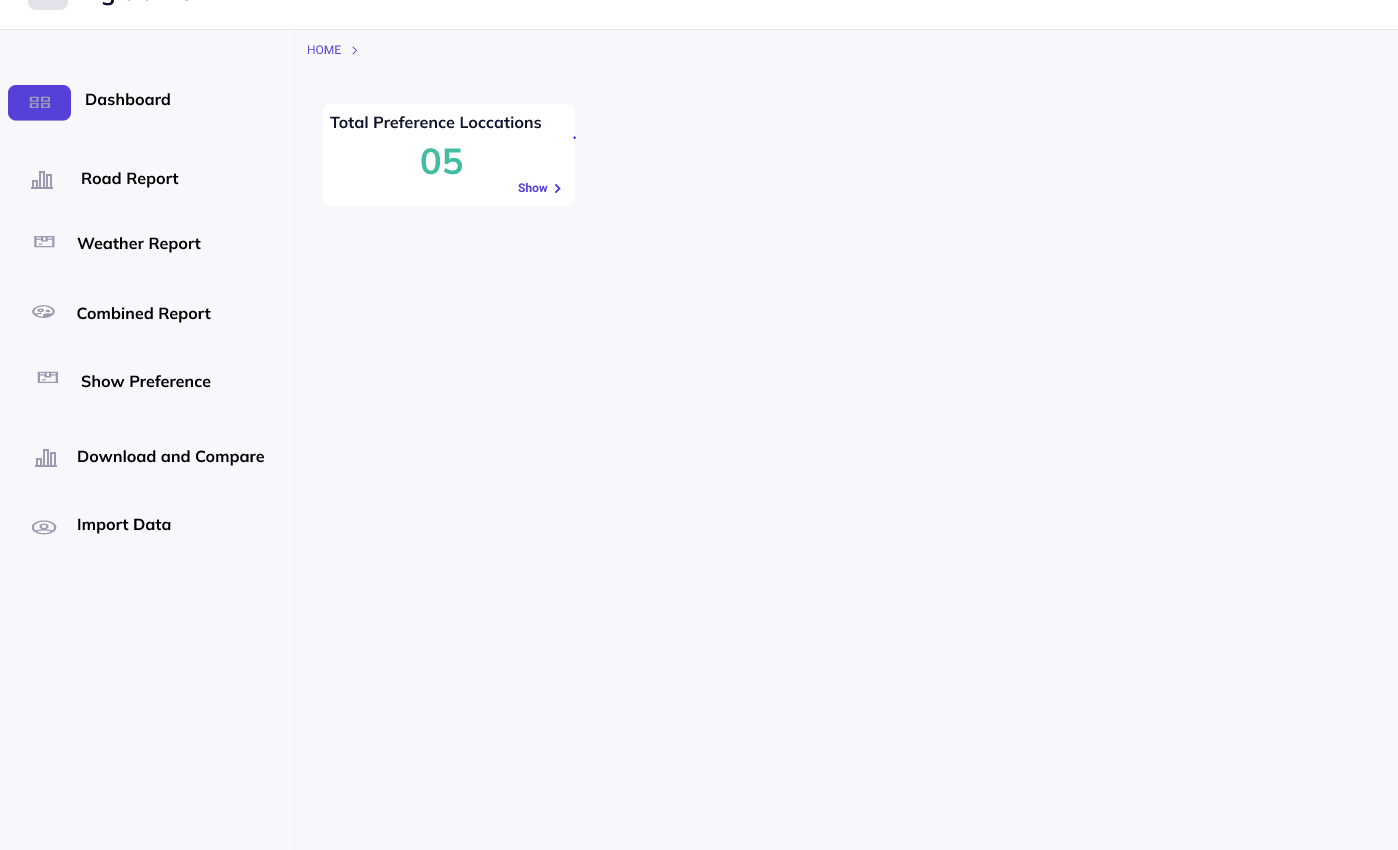
## Development tool

Developer tools, or simply "DevTools," are programs that let developers make, test, and debug software. The dev tools used in this project were Pycharm, and VS code. Other technologies used were, Figma for prototyping, creately for diagrams, UML, and class.

# Prototype Description

The main application in the later version differs a lot from the original prototype. But the initial prototype played an important role in deciding which of the layouts were important to implement. It also helped to eliminate unnecessary elements. In this section, the initial prototype will be discussed.

The main page of the application contains a menu bar. The dash bar gives the users various action options. Users can check road reports, weather reports, weather and road combined reports, the option to select their preferred location as default, and the option to download compare, and report.

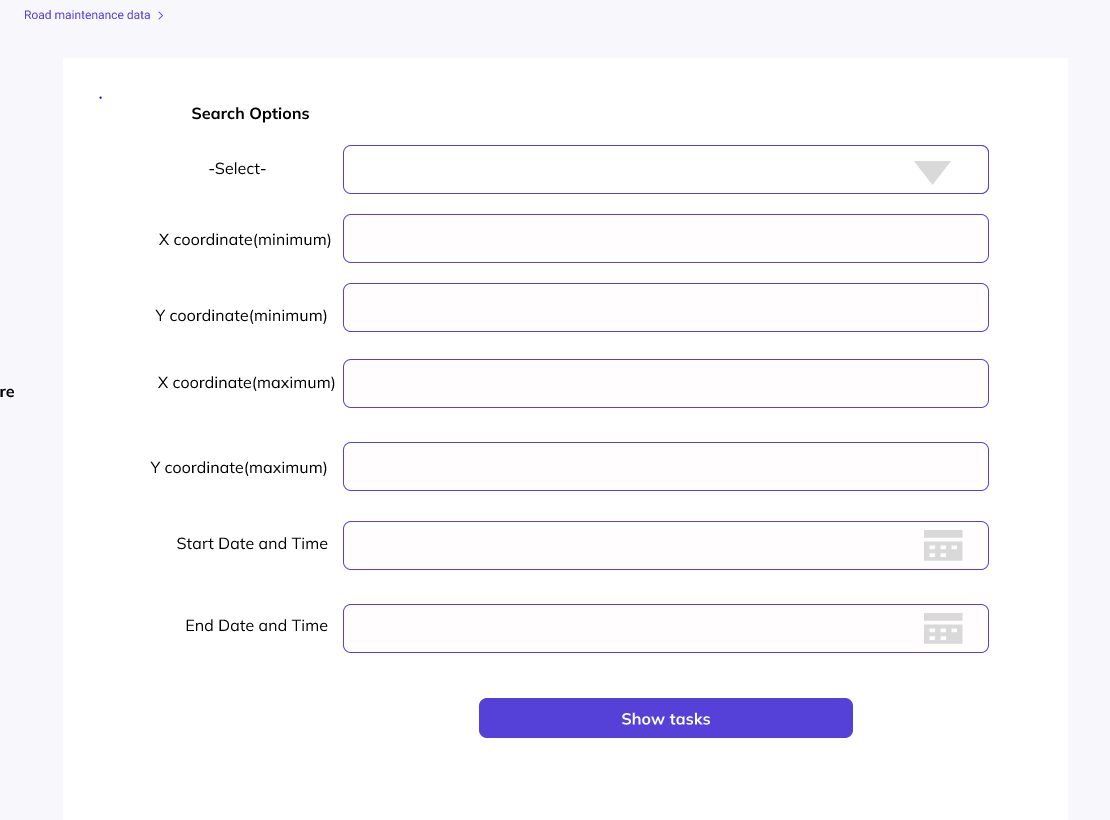


PICTURE 1: Main Page.

**Road Report**

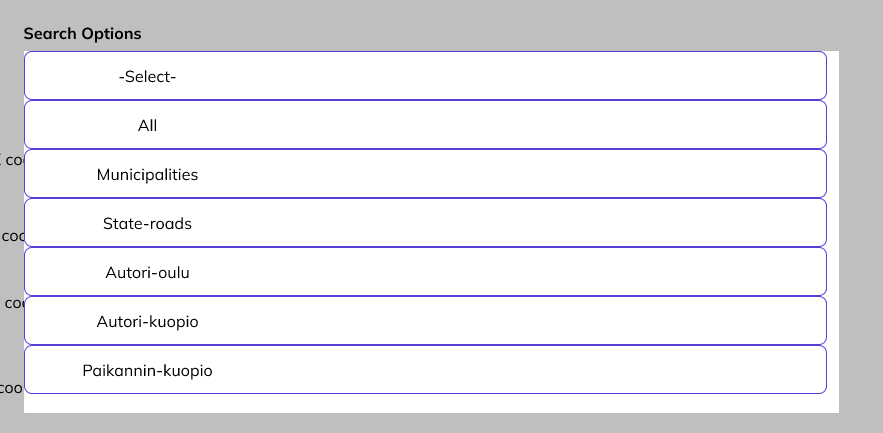
The road report option allows users to view different road conditions. Once the user clicks on this option, they will be given three options road maintenance, road condition, or recent traffic alerts.

1. Road maintenance: If the user selects the road maintenance option, it will first ask the user to fill out information regarding which road they want to view.

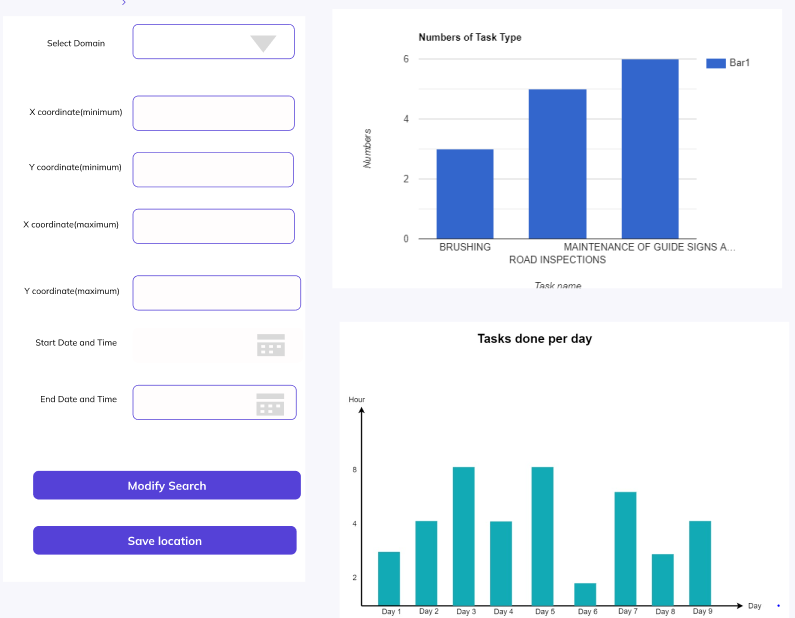


PICTURE 2: Road maintenance data.

Users have to fill in the place, maximum and minimum coordinates, and time limit. Users can select to view road maintenance information on a certain day and time through these options. The place selector has a scrollable list that looks like the picture below.

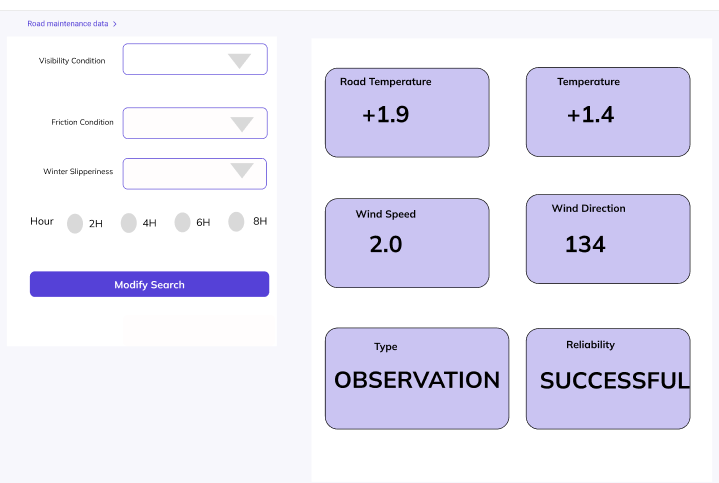
 PICTURE3: Selectors.

Once the user is satisfied they can proceed by pressing the “show tasks” button. The following page will show the road maintenance results based on the user-provided values. Users will be provided with two different types of graphs to show the maintenance of certain roads. Users will also be able to edit or modify their results by providing new information to the form again which will also be visible on this page. The application also keeps records of the places searched by the users.



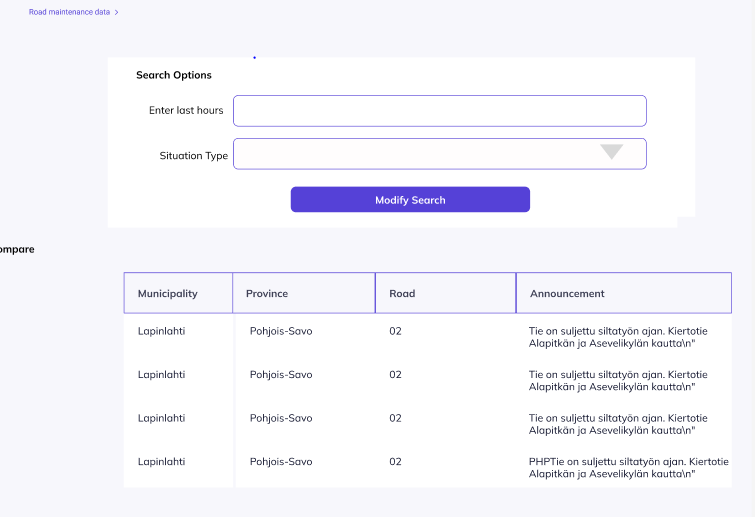
PICTURE 4: Road maintenance result.

1. Road condition: Once the user selects the road condition option from the road report they will be guided to a new page where there are fields for selecting visibility, friction conditions, winter slipperiness, and duration as the last 2hr to 8hr. Once the users have selected the required fields, they can search for results with the “search” button. The results will be the following, road temperature, actual temperature, wind speed, the direction of the wind, type, and reliability. Users can modify their search with the new information in the required fields on this page.



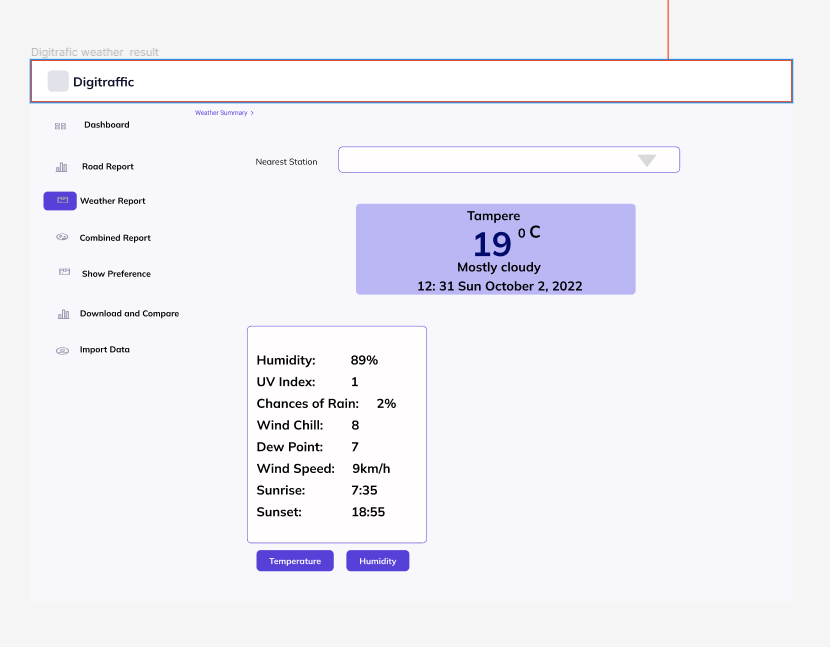
PICTUR5: Road condition result.

1. Traffic message: The traffic message option only requires two pieces of information from the users, time, and situation type.

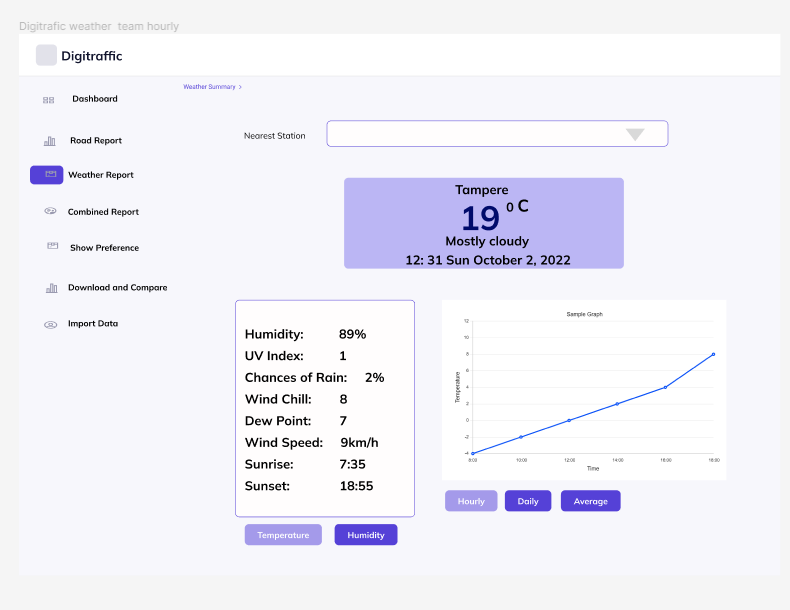


PICTURE 6: Traffice message.

**Weather report**

Users can select the nearest station for a weather forecast through this option. Once a location is selected the application will show the latest weather conditions. It includes information about humidity and UV index at sunset and sunrise. On this page, users will also find options, temperature, and humidity. Clicking one of these buttons will open a graph showing temperature or humidity throughout the day. Users can modify this graph by selecting hourly, daily, or average information.

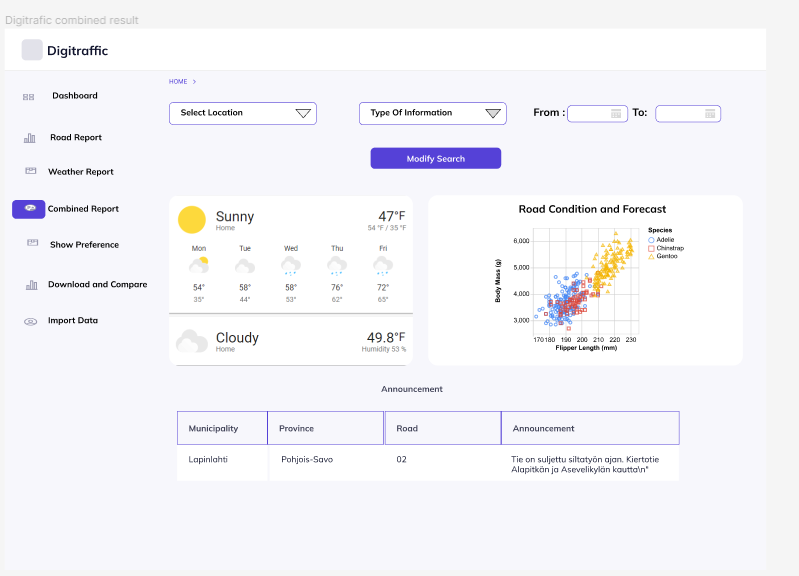
PICTURE 7: Weather result.



PICTURE 8: Weather report with graph.

**Combine report**

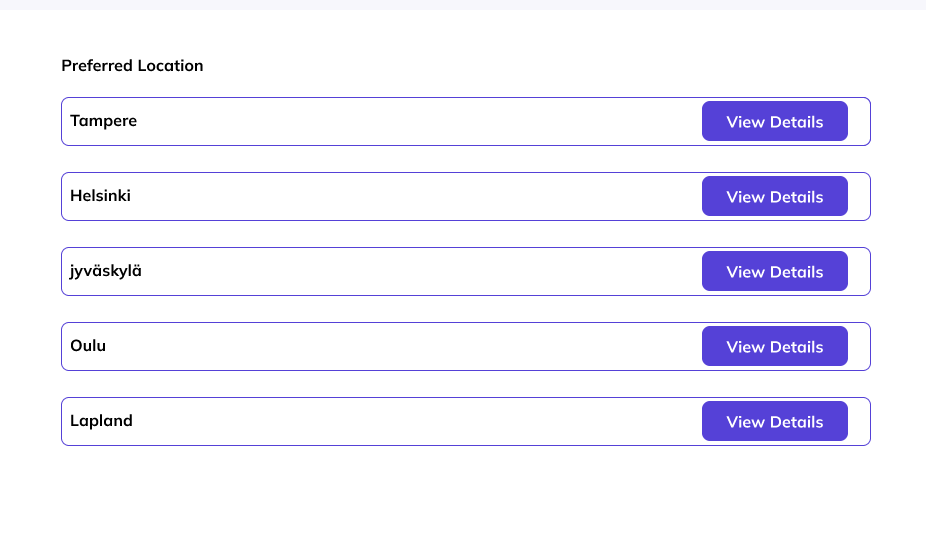
In the combined report the users can get their desired result by selecting the location, type of information, and duration. This gives the users a weather overview, road conditions based on the weather, and the latest announcement.



PICTURE 9: Combine report results.

**Show Preference**

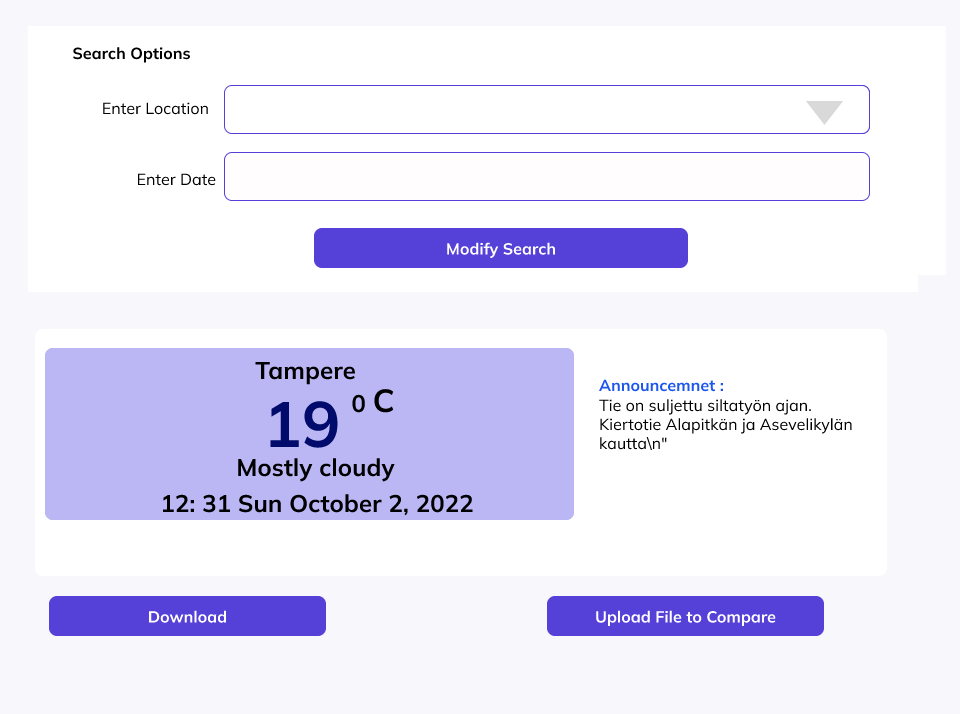
This option allows users to select a default option.



PICTURE 10: Sow preference page.

**Download and Compare**

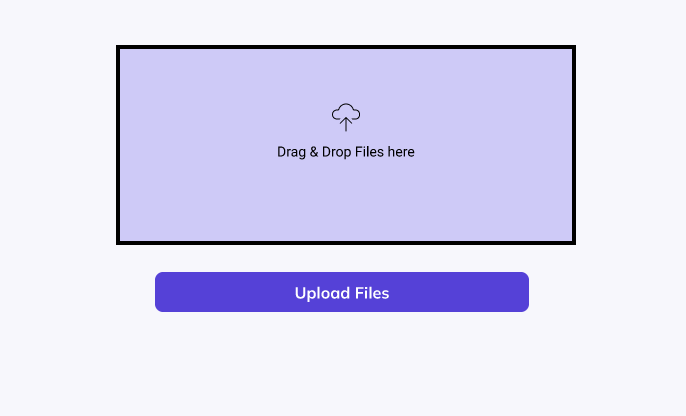
Users also have the option to download weather data. The application can also compare the date if a user uploads a previously downloaded date.



PICTURE 11: Download and compare the page.

**Import data**

With the help of import data, users can upload files. They can make a report, and send in a picture with this option.



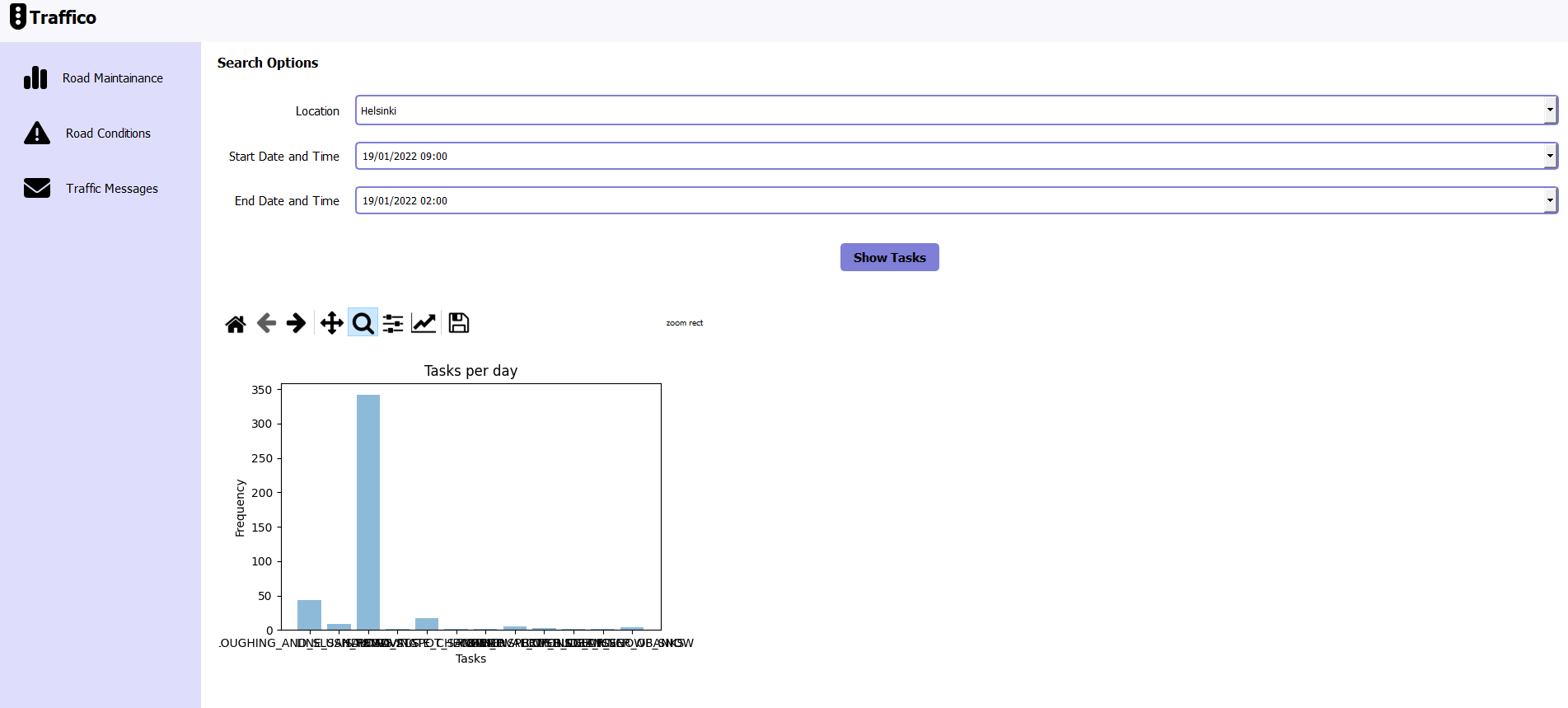
PICTURE 12: Import data page.

# Interfaces and services

In this section implemented project will be discussed, Through this, the project interfaces and services will be showcased. The application was kept clean with a minimalistic design. The main page of the application includes a menu bar through which the users will be able to navigate different services of the application.

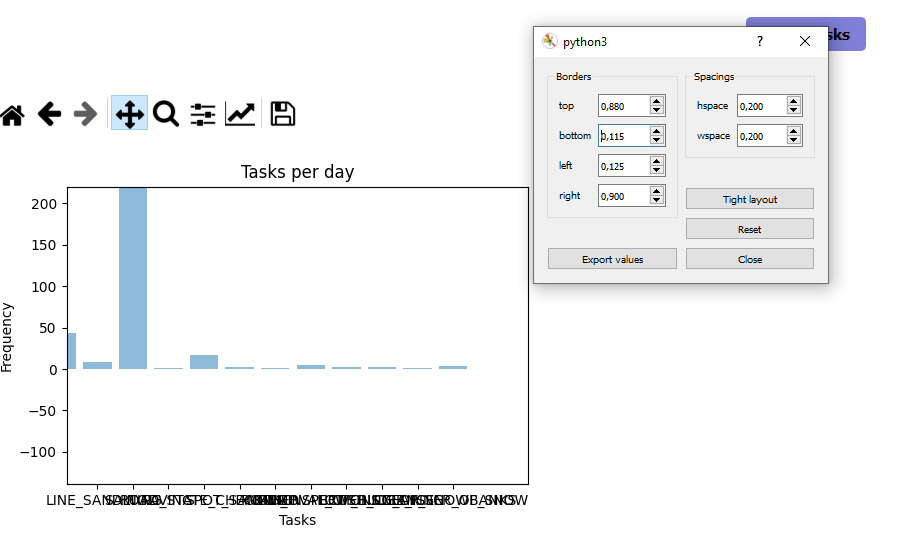
**Road Maintainance**

The first option for the users is road maintenance. As the name suggests through this users will be able to receive information about the road region-wise. To get the data users must input location and starting and ending date and time to the “search option”.



PICTURE 13: Road maintenance option.

Users will have various options for viewing the data. Users can zoom in to see any section of the data closely with the zooming icon. Users can drag and slide the graph to their liking for better observation. Users can also change the appearance of the layout.

PICTURE 14: Layout filter.

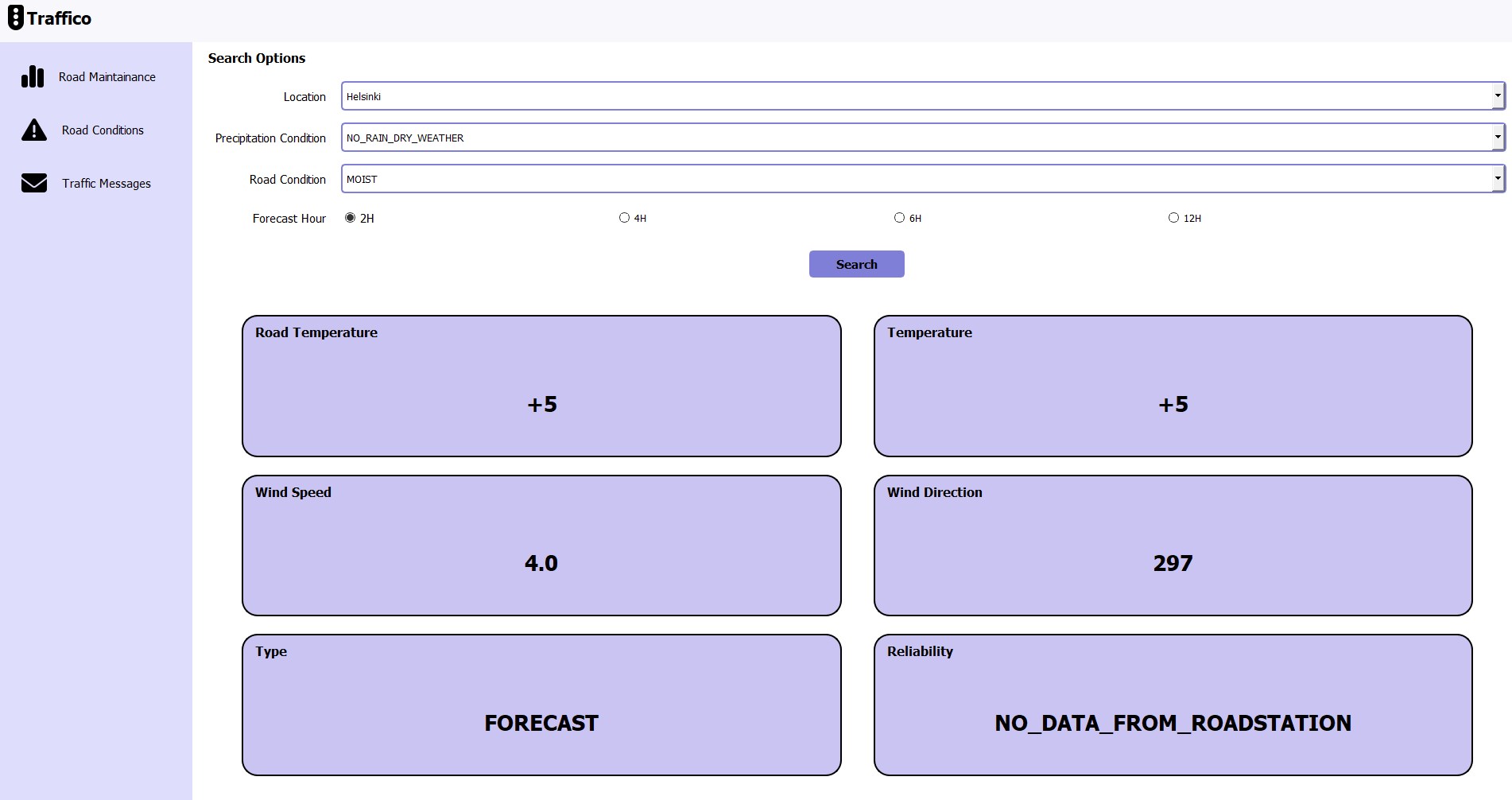
The figure icon allows the user to change the appearance of the figure. There is various option for changing how the graph will look. With this option, users can choose what type of graph they want to see. For example, linear, log, Syslog, or logit.



PICTURE 15: Graph layout option.

**Road Condition**

The road condition option allows users to receive weather conditions. Users can search for weather updates using the parameters, location, Precipitation condition, road condition, and duration. Users have the option to choose from 4 different durations.

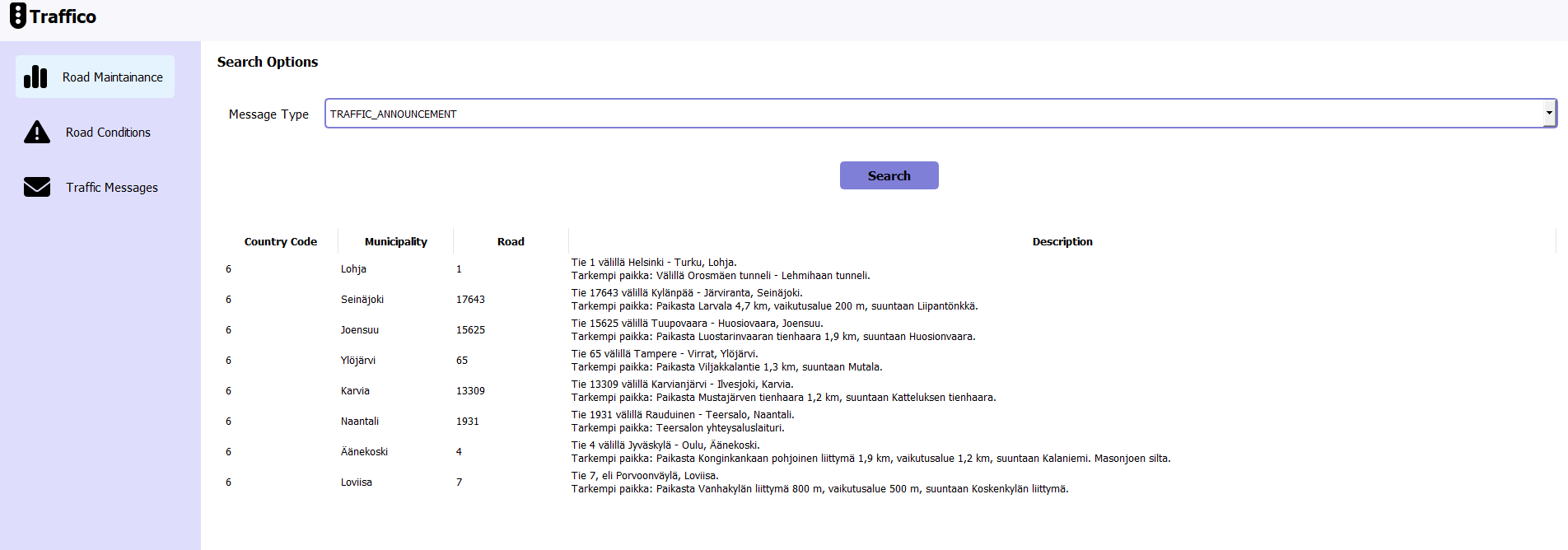


PICTURE 16: Road condition.

Through this users will get the road temperature, actual temperature, wind speed, wind direction, and reliability.

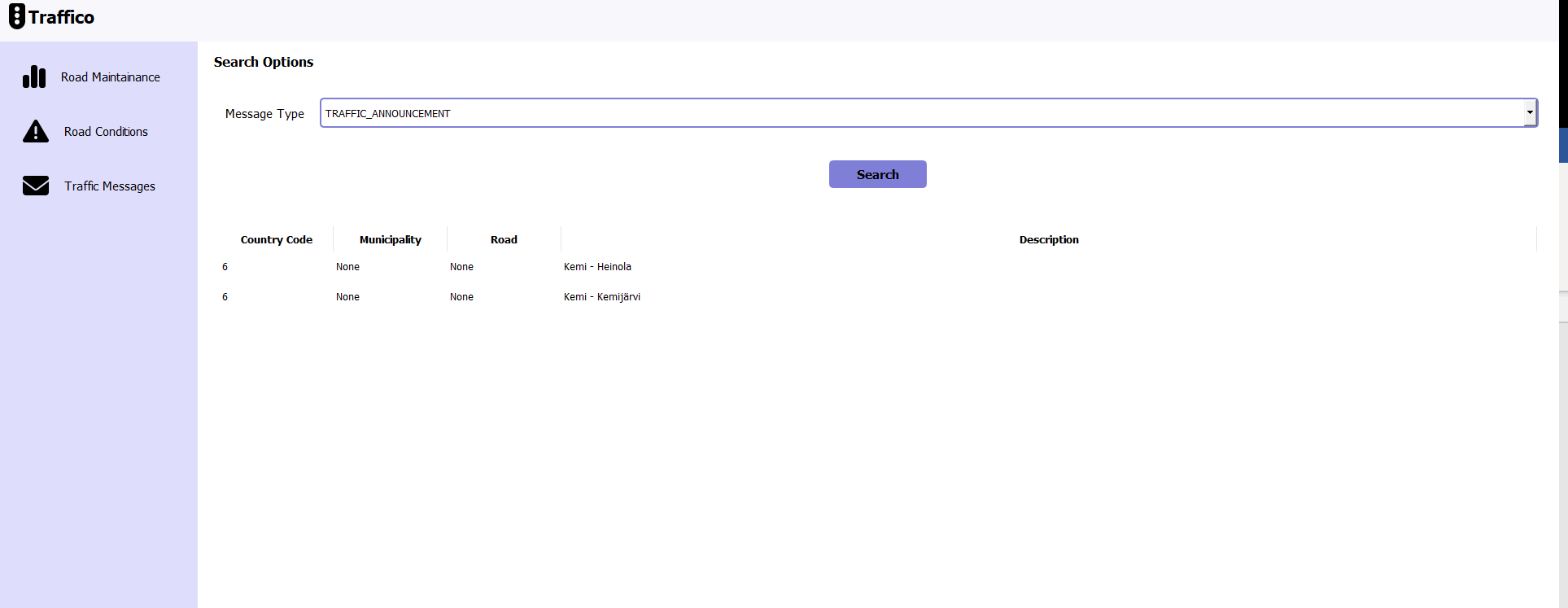
**Traffic Message**

From the traffic message page, if the user chooses the “traffic announcement” option, they will receive the latest weather information and announcement. Users will be able to see announcements for a different region on a single page.



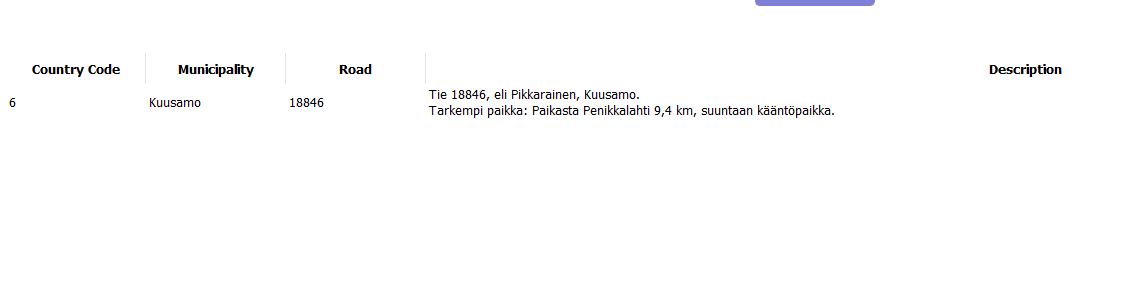
PICTURE 17: Traffic announcement.

If the user wants to see if there is any exempted transport they can select the “exempted\_transport” option from the message type dropdown.



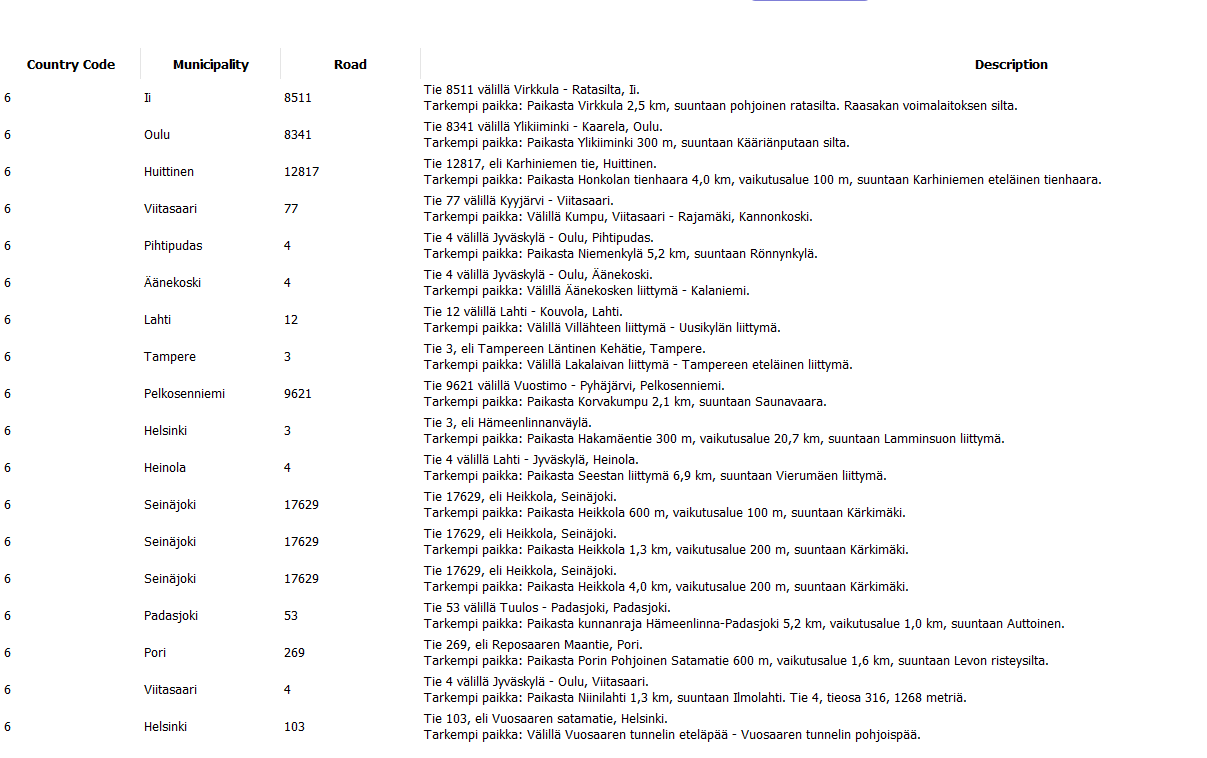
PICTURE 18:Exempted transport.

Similarly to view weight restriction users have to select the “weight\_restriction” option.



PICTURE 19: Weight restriction.

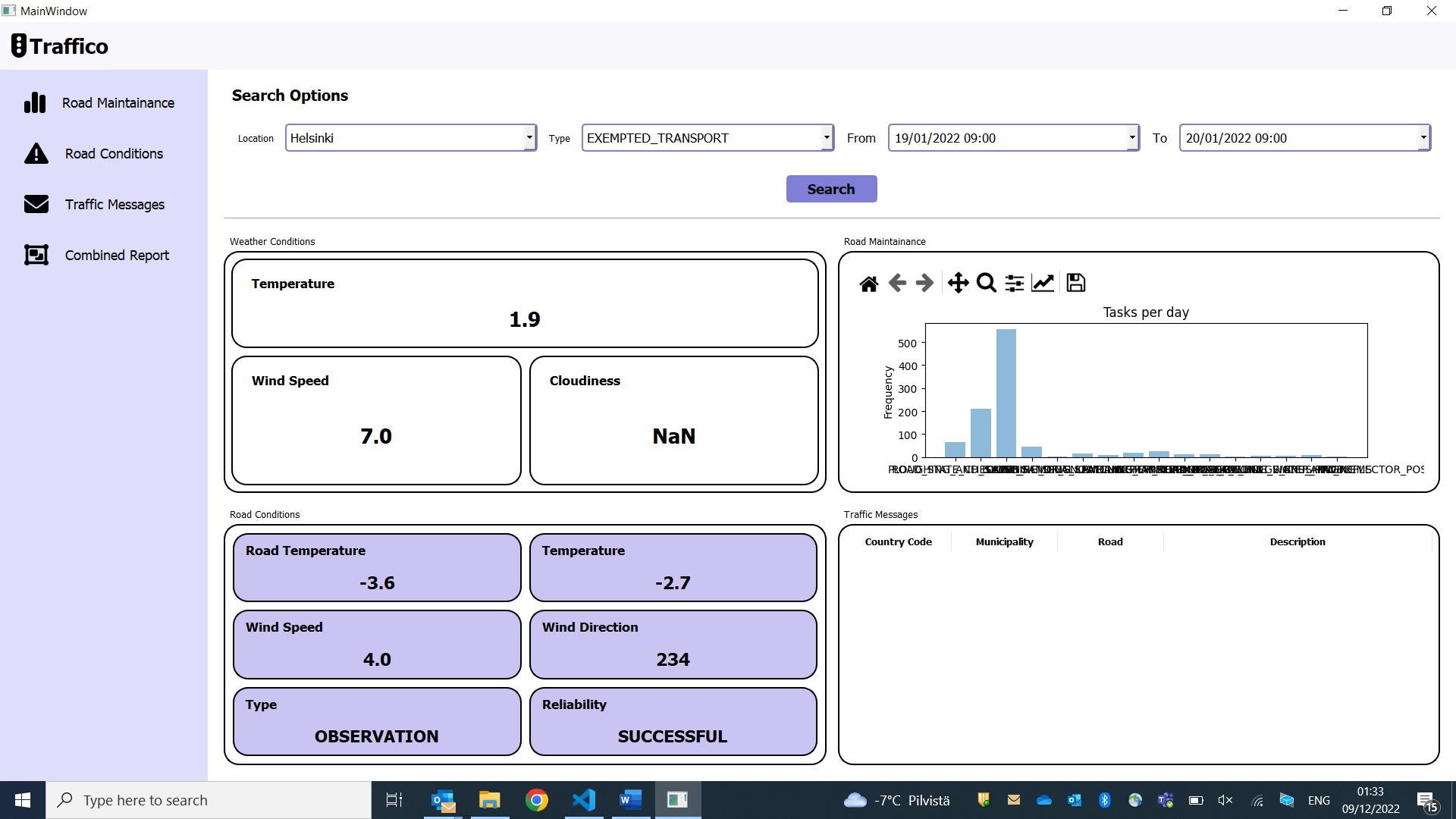
Users can also view any ongoing road maintenance work through the “road\_work” option from the option list.



PICTURE 20: Road work.

**Combine Report**

Combine FMI is shown in the same window. Combine road condition, road maintenance, and whether the message and condition are shown side by side Through the FMI endpoint. Here users can get the desired information by selecting the location, type, and duration.

****

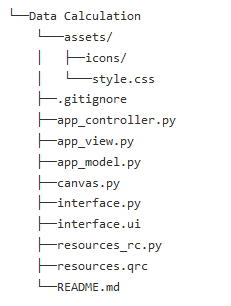
PICTURE 21. Combine report.

# Code And environment setup

In this section, different components of the code will be discussed. It will also be shown how to run the application.

### **File Structure**

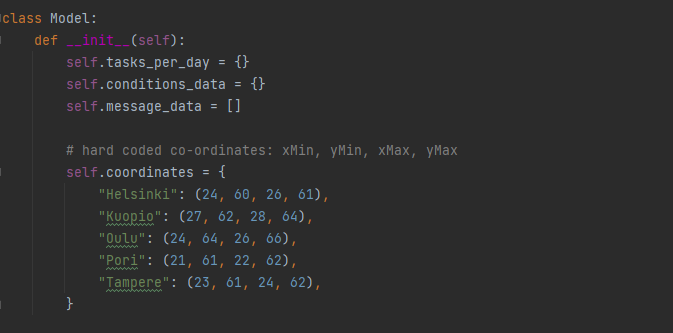
The file structure of the code base,



PICTURE 22: file structure.

App\_controller.py: In a virtual filesystem, controller classes serve as a template for directories. Callable attributes like methods are "files" in the "folder" that is "created" in the containing class when a class is assigned to an attribute. Because this child controller is statically mounted, the constructor will just receive the current RequestContext; dispatch will take care of instantiating any class children as requested during descent on each request.

App\_model.py: This file creates the model of the application. The content of this file looks like this,



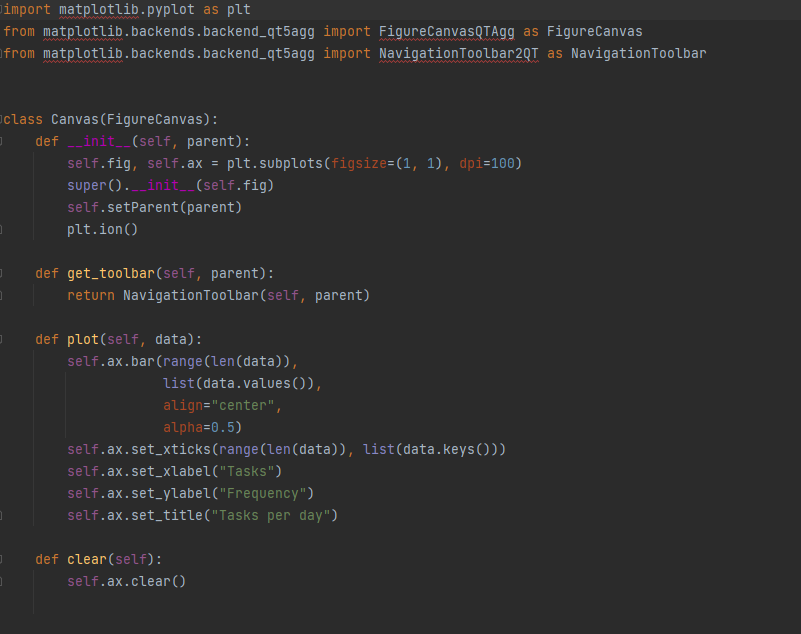
PICTURE 23: model.py file.

App\_view.py: This file creates the different windows of the application.



PICTURE24: Code snippet of App\_view.py file.

Canvas\_py: For showing the graph.



PICTURE 25: canvas.py file

Interface.UI: The GUI window is designed using the Qt Designer. The GUI is saved as an interface.UI file.

Interface.py: From the \*.ui file, the interface.py file is generated using the following command:

pyuic5 -x interface.ui -o interface.py

The interface.py file contains all the GUI elements of the app.

resources\_rc.py: The resources.qrc file, which contains assets such as icons, is also converted to resources\_rc.py file using:

pyrcc5 resources.qrc -o resources\_rc.py

**Setting the environment**

The first step to run the application is to install python. After successfully installing python the project dependencies needs to be installed. For installing the dependencies “pip” command can be used. Installing PyQt5,

pip3 install PyQt5

After installing all the necessary dependencies using the following command the application can be run,

python app\_controller.py

# 7 Conclusion

The main idea for this project was to build an application that will give weather information as well as road condition based on the weather. When creating the prototype of the application there were a lot of ideas were explored. But in the final implementation besides the main functionalities, a few of the UI were changed. The final UI looks simpler and user-friendly which was one of the targets of the project.