

We have a class called WeatherApp which has and holds all the data we need for the UI. Then we have four controller classes which use WeatherApp's data and makes the actual UI. These four different controllers make up a tab each in our UI. The tabs are Weather, Road condition, Combined (weather+road condition) and settings. The preference class saves user's preferences by saving the tab the user is on and the buttons that are pressed at the moment on that tab.

We are using two classes to fetch the data from the APIs used. FMIManager gets and parses the data from the FMI API and DigiTrafficManager does the same to the DigiTraffic API. Then we have one class for each API to get and save the data we need in this project.

Classes and their dependencies are seen in this class diagram (Figure 1).

Class responsibilities:

WeatherController:

Controls the UI of the weather forecast tab

RoadController:

Controls the UI of the road maintenances and tasks tab

CombinedController:

Controls the UI of the combined tab.

SettingsController:

Controls the UI of the settings tab.

Every controller's functions are quite self-explanatory.

WeatherApp:

Root class of the application. Holds the data of the application.

FmiApi:

Gets the wanted data from FMI API.

FMIManager:

Gets and parses data from fmi api into wanted weatherData model.

DigiTrafficApi:

Gets and parses data from the digitraffi api into wanted roadData model.

DigiTrafficManager:

Reads the data from API and parses it.

Preference:

Saves the user's given options to a map by saving the current tab and selected options.

We decided to use a class WeatherApp to save and hold all the data. This way it's easy to use different UI controllers using the same data. The APIs have models for fetching the data from the DigiTrafficApi and FmiApi are managed through API managers.

For tracking project status and assigning tasks to developers we are using Trello and Kanban tables. We have decided to use JavaFx as an UI-library in the project and for common tools we have chosen SceneBuilder and NetBeans IDE.

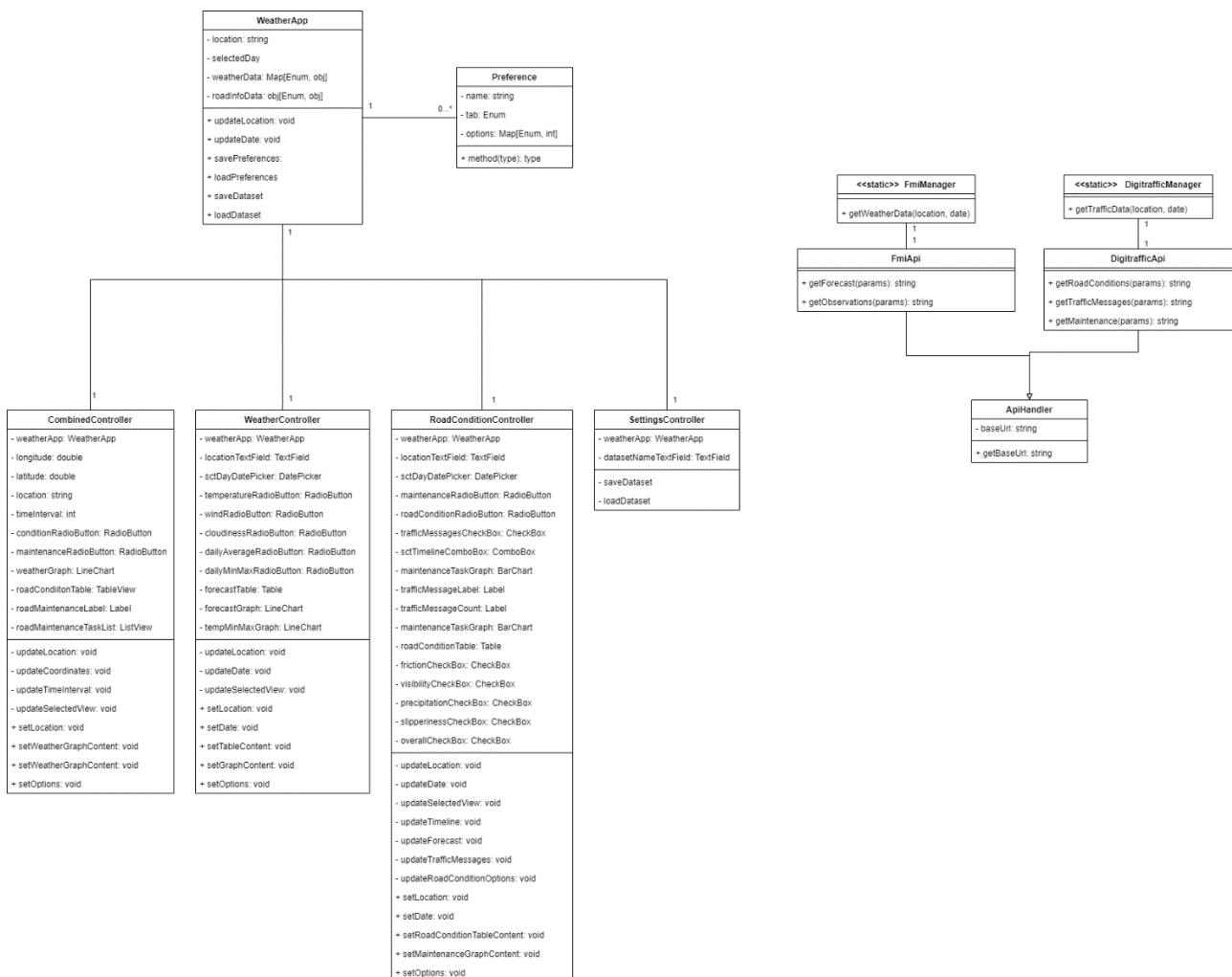


Figure 1 Application class diagram