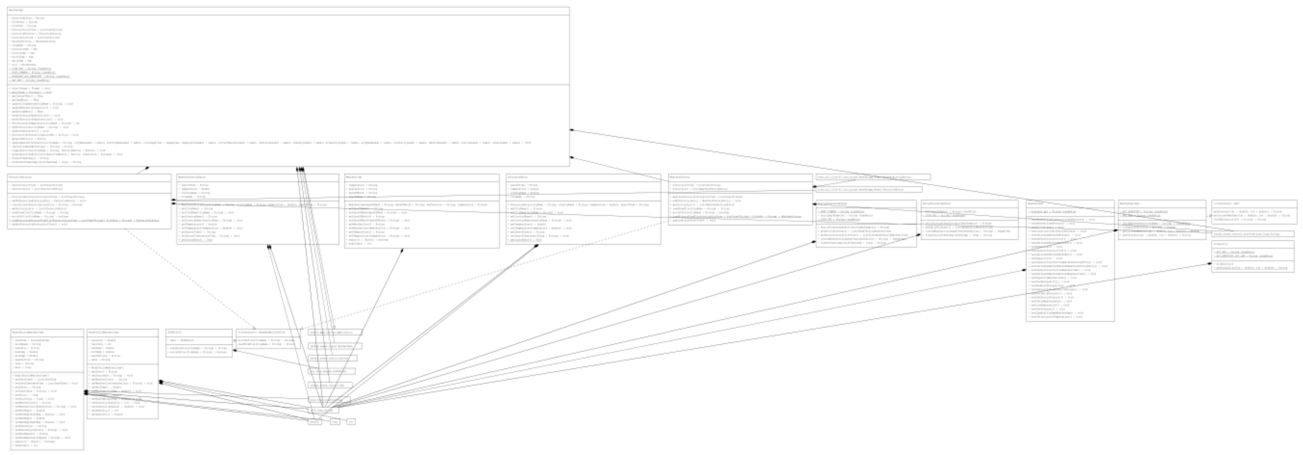
WeatherApp

1. **Introduction:** WeatherApp is an easy-to-use application that provides users with real-time weather information for their desired locations. The app seamlessly interfaces with a dedicated weather API to deliver a comprehensive and visually appealing weather experience, leveraging a robust architecture and featuring a sleek graphical user interface. Furthermore, some additional key features are available to enhance their app-using experiences.
2. **Software Structure**

a. **Data Source:** WeatherApp's core functionality is based on a carefully selected weather API, which ensures accuracy and reliability in delivering up-to-date weather data. Users can gain access to a wealth of information, including current conditions, forecasts, and detailed meteorological insights, by leveraging the power of this API. We use the OpenWeatherMap API for this app.

b. **UML Class Diagram:** Basically, I tried to keep the full structure in MVC form, So there are several files for model, then there are several files for controller and then there is a one particular file which is responsible for View. Moreover UML generator in IntelliJ IDEA has been used for the UML.

1. **Model:** There is a total of 5 total files in model. The class diagram for them is given below. (in the documentation folder the Model class diagram is also visible)
2. **Controller:** There are 7 total files in the controller. TheUML for them is given below. (in the documentation folder the controller class diagram is also visible)
3. **View:** Lastly there is a one file for this, which is named WeatherApp. The whole UML is given below (in the documentation folder the class diagram is also visible)



These things are responsible for the GUI, weather information for a fixed and location, daily, hourly forecast information, weather search history, favorite place’s history.

c. **Responsibilities of key Classes:**

1. **WatherApp:** Creating an interesting user interface and acting as an entry point, WeatherApp expands Application. By utilizing JavaFX features, it guarantees a smooth and visually appealing user interface. The relationships between these classes are shown in the UML class diagram, which provides the framework for WeatherApp's functionality. The sections that follow will cover important class duties, application features, and cooperative teamwork.

2. **Handling Daily Forecasts:** The DailyForecastHandler coordinates the collection, processing, and display of daily weather forecast data. It works closely with DailyWeatherItem, a data container that uses well-defined set and get methods to ensure data integrity.
3. **Hourly Forecast Handling:** In order to provide a precise user experience, HourlyForecastHandler processes and presents granular weather information on an hourly basis. This involves managing detailed hourly weather forecasts.
4. **History Handling:** In order provide user to a good experience a WeatherHistory class has been introduced.
5. **Favorite Location Handling:** In order to enhance the user connectivity with the app favorite locations feature have been introduced and FavoritesHisotry class has been working for this.

3. Key Features and Fulfilled work:

- **Graphical User Interface (GUI):** WeatherApp has a user-friendly interface that has been meticulously designed. Improve user interaction. Our team has ensured that the interface is distinctly unique, providing a seamless and engaging experience if built on JavaFX.
- **Weather information:** User can see in total more than 7 weather information data types and date with time and country for a fixed location (Tampere) and also there preferable one while using the search operation.
- **Custom Icons:** WeatherApp uses a custom collection of icons to add a touch of individuality to the visual depiction of weather conditions, deviating from standard weather icon sets.
- **Class design and implementation:** Proper object-oriented MVC structure has been maintained with well-documented interfaces and classes and also provided comment and generated Javadoc.
- **State Persistence:** WeatherApp remembers the user's preferences, which goes beyond the call of duty. Upon shutting down, the application stores its preferred locations and current location on disk, which it then automatically restores when it restarts.
- **Interface using:** Both iAPI and iReadAndWriteToFile interfaces have been used for gathering data from api and handling JSON files.
- **Version history:** Version history has been well handled through GitLab.
- **Error Handling:** The program is equipped with strong error-handling features that guarantee uninterrupted use even in the event of unforeseen file processing problems.
- **Unit Tests:** To ensure the accuracy and dependability of the program's functionality, we've put in place several unit tests.

Extra Features and work:

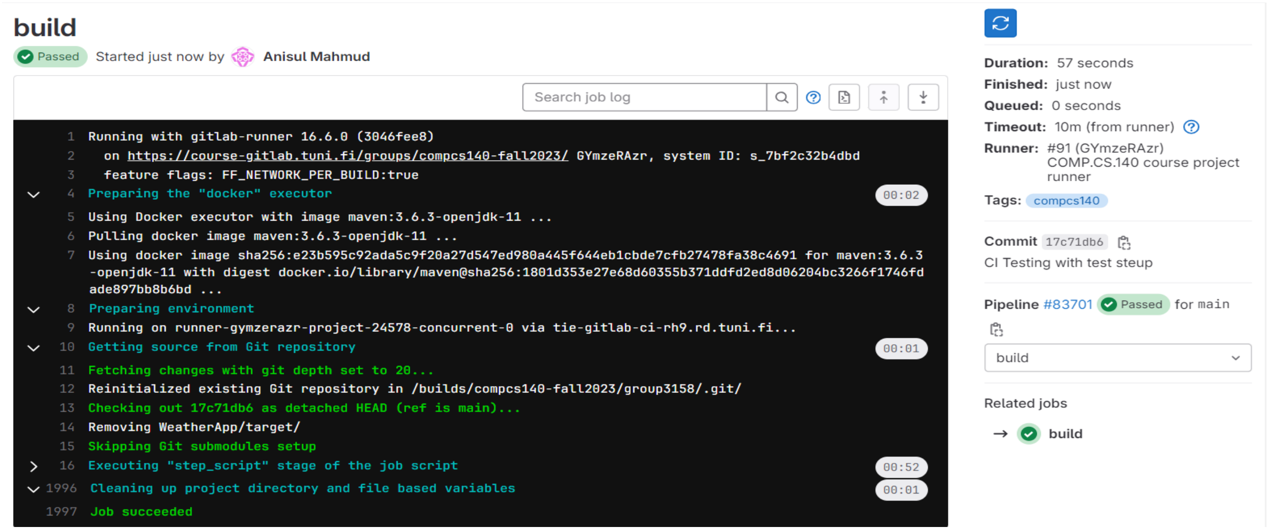
a. **Daily Forecast:** User can see the total 5-day forecast for their searched location moreover for the fixed location (Tampere). With the basic information with more than 7 data ypes with country, date, and time, the user cann see the searched location's ed locations's 5-day forecast and the user do not need to do anything after searching new location the daily forecast will be updated with search operation.

b. **Hourly Forecast:** The user can view hourly forecast details in 3-hour steps for five days, along with lost weather data and a customized icon. Just like the daily forecast, this information is automatically updated when a new location is searched.

c. **Location History:** User can also see their searched history, and the time and the temperature when they searched the location. The search history will be saved in a json file and user can see the searched history and this is also will be auto updated.

d. **Favorite Locations:** User can save their favorite location with using (*). User favorite location will be saved in a JSON file and whenever the user s them in the favorite location's latest time and temperature, not the one when they add the location in favorite. And this will also be auto updated, nothing needs to be done.

e. **Continuous Integration (CI):** Through the use of a pre-defined build task for Maven compilation in its gitlab-ci.yml file, this WeatherApp project leverages Continuous Integration (CI). Also, a test task integrated into the CI/CD pipeline marked with compcs140 is functioning properly. Below is the last successful CI build with test task.



build

Passed Started just now by Anisul Mahmud

```
Running with gitlab-runner 16.6.0 (3046fee8)
  on https://course-gitlab.tuni.fi/groups/compcs140-fall2023/ GYmzeRAzr, system ID: s_7bf2c32b4dbd
  feature flags: FF_NETWORK_PER_BUILD:true
Preparing the "docker" executor                                                    00:02
Using Docker executor with image maven:3.6.3-openjdk-11 ...
Pulling docker image maven:3.6.3-openjdk-11 ...
Using docker image sha256:e23b595c92ada5c9f20a27d547ed980a445f644eb1cbde7cfb27478fa38c4691 for maven:3.6.3
-openjdk-11 with digest docker.io/library/maven@sha256:1801d353e27e68d60355b371ddfd2ed8d06204bc3266f1746fd
ade897bb8b6bd ...
Preparing environment
Running on runner-gymzerazr-project-24578-concurrent-0 via tie-gitlab-ci-rh9.rd.tuni.fi...
Getting source from Git repository                                                 00:01
Fetching changes with git depth set to 20...
Reinitialized existing Git repository in /builds/compcs140-fall2023/group3158/.git/
Checking out 17c71db6 as detached HEAD (ref is main)...
Removing WeatherApp/target/
Skipping Git submodules setup
Executing "step_script" stage of the job script                                    00:52
Cleaning up project directory and file based variables                             00:01
Job succeeded
```

Duration: 57 seconds
Finished: just now
Queued: 0 seconds
Timeout: 10m (from runner)
Runner: #91 (GYmzeRAzr) COMP.CS.140 course project runner
Tags: compcs140

Commit 17c71db6
CI Testing with test steup

Pipeline #83701 Passed for main

4. **Project Functionality:** WeatherApp's functionality extends beyond that of conventional weather applications, providing a comprehensive and user-focused experience. The application seamlessly manages current weather information, daily and hourly weather forecasts, and the current forecast for a favorite location by utilizing the robust roles of key classes like **'DailyForecastHandler'**, **`HourlyForecastHandler'**, **'WeatherHistory'**, **'FavouritesHistory'**, and **'WeatherApp'** with losses of model class. Users can expect a comprehensive summary that includes all the fine details needed for accurate planning. With the help of JavaFX capabilities, the visually appealing WeatherApp class-led graphical user interface creates a realistic experience. With extra features like state persistence, the ability to save favorite locations, and a customizable icon set, WeatherApp goes above and beyond the standard.
   The user experience is guaranteed by the program's ability to handle errors that arise during file processing, and reliability is ensured by the inclusion of unit tests. With its extensive functionality and user-friendly interface, WeatherApp stands out for providing accurate and comprehensive weather insights.

5. **Conditions for classes:** All key classes strictly follow predefined pre and post conditions. All key classes ensure accurate data processing for a structured output. With all other classes, The WeatherApp class, responsible for the graphical interface, establishes preconditions and post conditions, ensuring smooth user interactions and confirming the application's reliability.

Upcoming sections will elaborate on the class's specific functionalities, highlighting meticulous design considerations in WeatherApp.

6. Division of work:

1. **Md Anisul Islam Mahmud, anisul.mahmud@tuni.fi, 152152220,**

- Mentor in charge of both the front end, back end, and the whole project.
- Guided the methodical development of products and order modules, playing a key role in their notable progress and enhancement.
- Working with all the features, testing, gui,CI, api handling and GitLab etc.

2. **Md Nasir Uddin Shuvo, nasir.shuvo@tuni.fi, 152134471**

- Working in frontend, backend.
- Worked in module, GitLab issues, testing, error handling.
- Contributed to a systematic project breakdown, emphasizing an organized project management approach.

3. **Israt Jahan Runa, israt.runa@tuni.fi, 152134329**

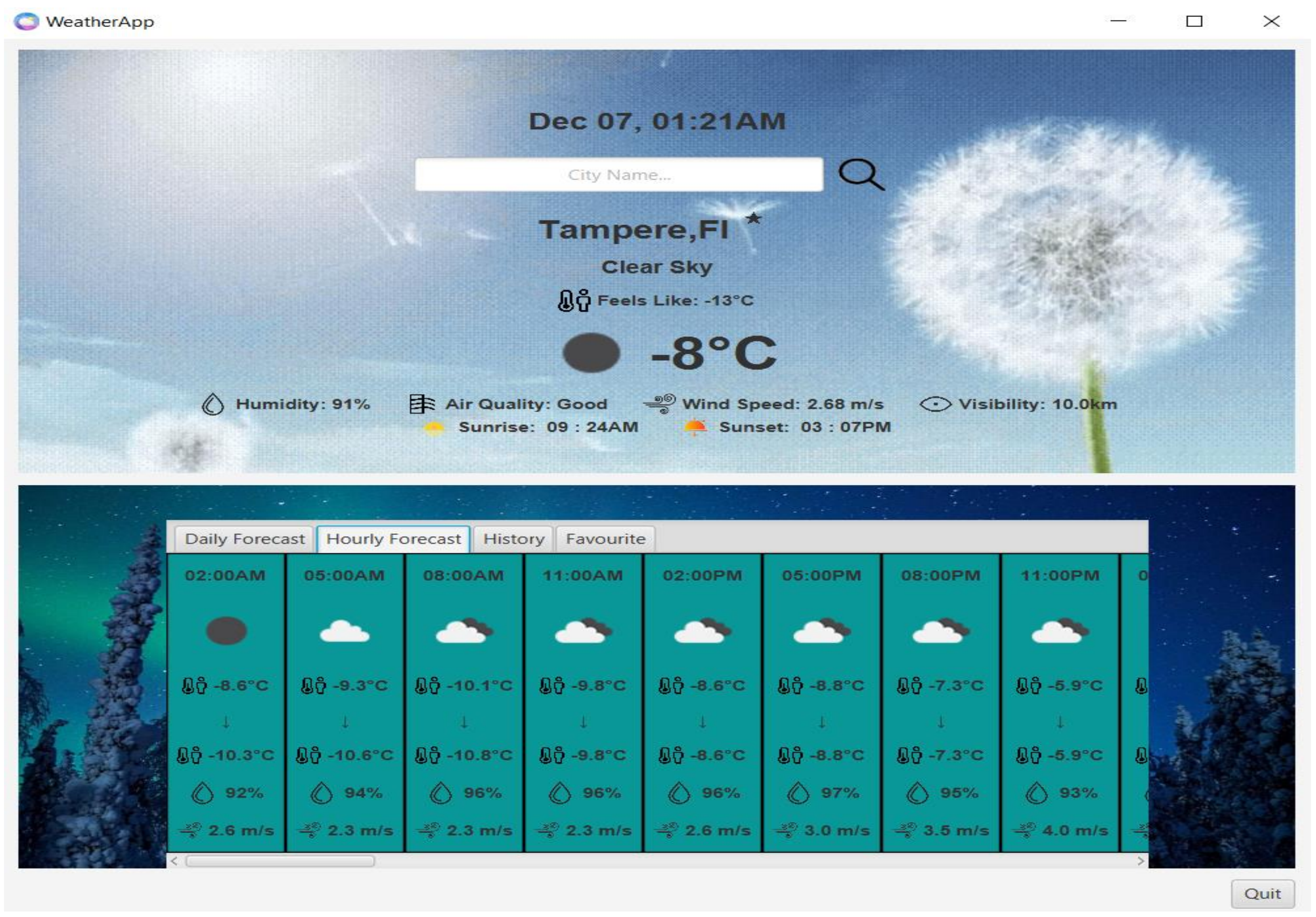
- Predominant role in meticulous test case development and execution.
- Actively engaged in backend tasks, including comprehensive API testing with Postman.

7. Short User Manual:

- **Launch and Engage:** Open the WeatherApp portal to explore a colorful, user-friendly graphical universe that has been painstakingly created by the creative minds behind the scenes, the `WeatherApp' class. The user interface is smooth and visually appealing right from the start of your weather exploration adventure. Follow this step to run this in a IDE: -> mvn clean install -> click the start or mvn clean javafx:run

Further A video will be provided for this.

- **Weather Discovery:** Use the magical wand with its many options to easily discover the beauty of daily forecasts.
- **Detailed Forecast Expedition:** With a detailed forecast that presents hourly, daily aggregates with favorite and historical options, set out on an amazing voyage of discovery. It's all about the adventure, not just the weather!
- **Glitch-Error free Experience:** Our skilled team weaves collaborative magic to fortify the WeatherApp ship, ensuring a smooth and flawless user journey.



Dec 07, 01:21AM

Tampere,FI

Clear Sky

Feels Like: -13°C

-8°C

Humidity: 91% Air Quality: Good Wind Speed: 2.68 m/s Visibility: 10.0km

Sunrise: 09 : 24AM Sunset: 03 : 07PM

02:00AM	05:00AM	08:00AM	11:00AM	02:00PM	05:00PM	08:00PM	11:00PM
-8.6°C	-9.3°C	-10.1°C	-9.8°C	-8.6°C	-8.8°C	-7.3°C	-5.9°C
↓	↓	↓	↓	↓	↓	↓	↓
-10.3°C	-10.6°C	-10.8°C	-9.8°C	-8.6°C	-8.8°C	-7.3°C	-5.9°C
92%	94%	96%	96%	96%	97%	95%	93%
2.6 m/s	2.3 m/s	2.3 m/s	2.3 m/s	2.6 m/s	3.0 m/s	3.5 m/s	4.0 m/s

8. Missing Features or Bugs

- **Absence of Features:** As we continue to improve WeatherApp, we have to acknowledge its lackings. The team specifically experimented with integrating a map, but we reserved this for future use. Though so many other features like login user, are in the pre-stage of implementation. Stay updated for new features.
- **Known Bugs:** There are no known bugs currently in this project.

9. Conclusion:

Overall, WeatherApp's development demonstrates a dedication to user satisfaction and meticulous design. The masterfully crafted classes with WeatherApp create a cohesive whole for an immersive weather forecasting experience. The graphical interface, driven by JavaFX, balances fluid interaction with visual appeal.

As an example of the unwavering pursuit of software excellence, WeatherApp prioritizes the user experience throughout the refinement process while acknowledging challenges.