

Jasmine Latendresse (#40011419)

Airi Chow (#40003396)

### Weather System Specification

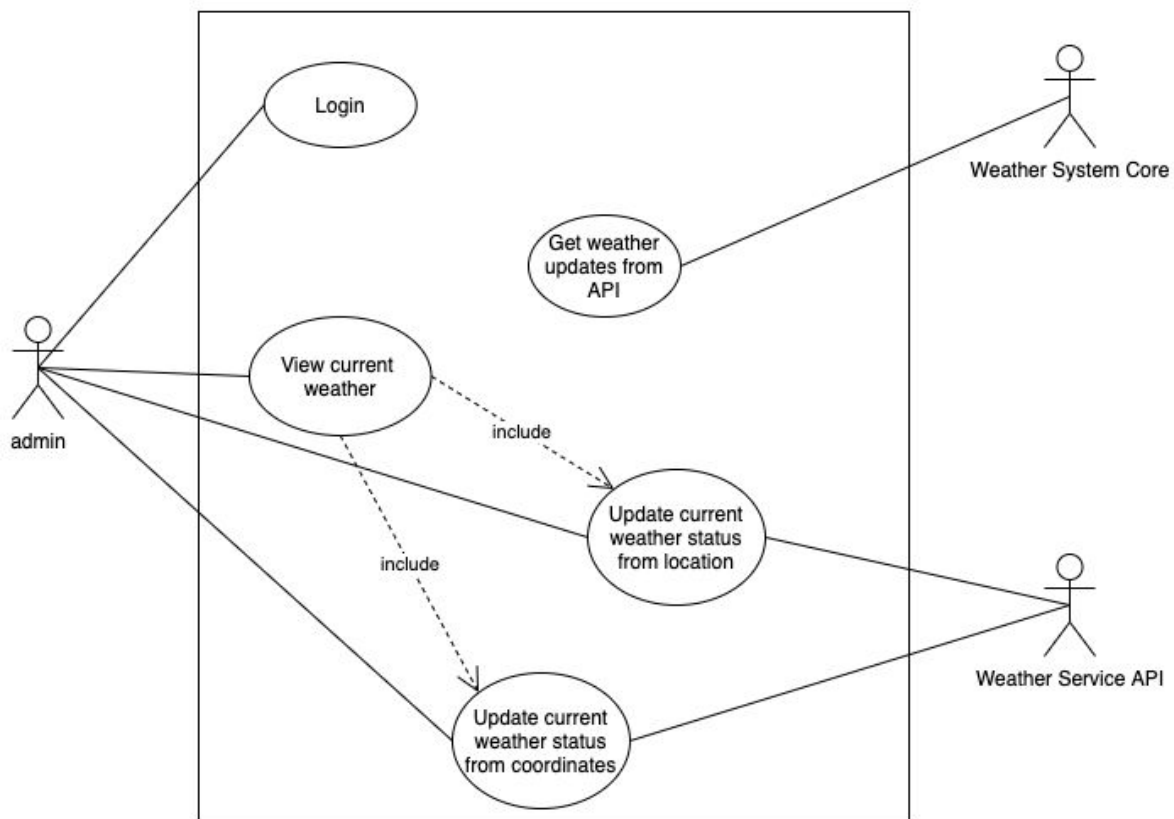
#### Description of System:

The Weather System displays the current weather of a location given the city's location or the geographical coordinates. In addition to the weather details, the time when this information was updated is displayed.

#### Functions of the System:

- Displaying as default Montreal's current weather in Celsius
- Displaying Last Updated time in Montreal's Time
- Searching location based on city
- Searching location based on longitude and latitude
- Toggling between Fahrenheit and Celsius

#### Use Case Diagram:



## Class / Package Diagrams:

<b>weatherSystemCore.py</b>
- longitude: float - latitude: float - weather_overview: String - weather_description: String - current_temperature: String - current_feels_like: String - current_min: String - current_max: String - current_time: int - current_city: String - units: String - formatted_url: String - response: int
+ get_current_weather_default(self, user): void

<b>weatherSystemAPI.py</b>
+ format_url_default(self, city, user): void + format_url_with_parameters(self, city, unit_format, user): void + format_url_with_coordinates(self, longitude, latitude, unit_format, user): void + format_temperature_object(self, temperature_json, status_code): void + load_config(self): String + location_error(self): String + user_error(self): String

<b>app.py</b>
+ index(): render(HtmlDocument) + search_location(): render(HtmlDocument) + login(): render(HtmlDocument) + logout(): redirect(HtmlDocument) + search_coordinates(): render(HtmlDocument)

## **API Specification:**

### *load\_config(self):*

Loads up the service api with config file consisting of the url with api token. User can modify the api token by modifying "weatherAPIConfig.json" file.

### *format\_url\_default(self, city, user):*

Takes city(string) and formats the url to set the requested url. The user parameter will be checked if it's None before calling the "Weather's System Core".

### *format\_url\_with\_parameters(self, city, unit\_format, user):*

Takes city(string) and unit\_format(string) to set the requested url. The user parameter will be checked if it's None before calling the "Weather's System Core".

### *format\_url\_with\_coordinates(self, longitude, latitude, unit\_format, user):*

Takes longitude(float), latitude(float) and unit\_format(string) to set the requested url. The user parameter will be checked if it's None before calling the "Weather's System Core".

### *format\_temperature\_object(self, temperature\_json, status\_code):*

Takes a response object in JSON format and parses the selected temperature attributes. This data is then written as a temperature object in "temperature.json". In the case that the status\_code is not 200, "temperatureError.json" is created.

## **Potential Status Code:**

**200** - Request was successfully processed.

**400** - "An error occurred, the location could not be found!":  
Coordinates entered do not match any city location.

**403** - "User is None! An error occurred, this is an invalid user. ": the user parameter is a None object.

**403** - "An error occurred, this is an invalid user.": the user parameter does not match the authorised users in "config.json" file

**404** - "An error occurred, the location could not be found!":  
Invalid location. Either misspell location name or city is not supported.