

# Project X

- Requirements
- Deliverables
  - Use cases
  - Architectural diagrams
  - UML diagrams
  - Sequence diagrams
  - Python code
- Demo
  - `pip install geopy`
  - `pip install geopy/opencage`

# Requirements

- Create an GUI to accept a city name then get the Ultraviolet Index of the city at noon time of the current day. If the location is away from city, use coordinates instead of city.
- Get pollution data which includes Ozone thickness, Carbon Monoxide, Nitrogen Dioxide and Sulfur Dioxide Data

# Requirements

- Create an GUI to accept a city name then get the Ultraviolet Index of the city at noon time of the current day. If the location is away from city, use coordinates instead of city.
- Get pollution data which includes Ozone thickness, Carbon Monoxide, Nitrogen Dioxide and Sulfur Dioxide Data

- Get weather information which includes observation time, weather description, temperature, humidity, wind, visibility, pressure, sunrise, sunset time and a weather icon of the city
- Displays all the data/information on the GUI.
- Relevant data can be announced by computer as an option.

# Use cases

## 1.) Gets Ultraviolet Index:

User inputs city name, clicks 'get city UVI' button, system converts city to coordinates, sends request to website, website returns UVI data, system reads, decodes and then displays on GUI and announces if the option is selected.

User inputs city name, clicks 'get city UVI' button, system converts city to coordinates but failed, system displays message to ask user to retry or uses coordinates to get UVI.

User inputs coordinates, clicks 'get coordinate UVI' button, system sends request to website, website returns UVI data, system reads, decodes and then displays on GUI and announces if selected.

## 2.) Gets Pollution Data:

User inputs coordinates then clicks 'Get pollution by Coord' button to get Ozone thickness, CO, SO2 and NO2 data from Website and displays.

## 3.) Gets Weather Data:

User inputs city name, clicks 'get city weather' button, system sends request to website, website returns weather data, system reads, decodes and then displays on GUI and announces relevant info. if selected.

## 4.) Gets Coordinates

User inputs city name, clicks 'get Coordinates of City' button, system converts city to coordinates and displays on GUI and announces the latitude and longitude if selected.

## 5.) Gets information about this App:

User clicks 'Help' followed by 'About' in pull down menu, system displays ProjectX.pdf's contents.

## 6.) Gets Help of this App:

User clicks 'Help' followed by 'Help' in pull down menu, system displays Help.txt.

## 7.) Turn speaker on and off

User click 'File' followed by 'Speaker On'/'Speaker Off' in pull down menu, system turns speaker on or off.

# Fully dressed use case

## 1.) Gets Ultraviolet Index:

**Goal:** User wants to get UVI of a location using ProjectX App.

**Primary actor:** User

**Scope:** ProjectX App on Windows

**Level:** User

**Precondition:** ProjectX App is running on Windows, the "Ultraviolet Index" tab is posted on screen.

**Success end condition** The Ultraviolet Index of a specific location is displayed on ProjectX App's GUI.

**Failure end condition:** The Ultraviolet Index of the specific location is not displayed and error message is displayed.

**Trigger:** User clicks one of the two buttons on "Ultraviolet Index" tab of ProjectX GUI.



## **Main success scenario:**

- 1. User selects city in combo box of "Latest Ultraviolet Index for" Frame**
- 2. User clicks "Get UVI by City" button**
- 3. UviTab passes the city name to Coordinates**
- 4. Coordinates returns Latitude and Longitude back to UviTab**
- 5. UviTab sends coordinates returned from Coordinates to OpenWeatherMapUvi**
- 6. OpenWeatherMapUvi returns UVI data back to UviTab**
- 7. UviTab displays UVI Data on "Coordinates & UVI" Frame**
- 8. If Speaker is "ON", UVI Data will be announced**

## **Extensions (error scenarios)**

- 4a. Coordinates returns Exception back to UviTab**
  - 4a.1 System displays invalid city message**
    - 4a.1.a User re-enter city name**
    - 4a.1.b User uses Coordinates method in "Coordinate & UVI" Frame**

## **Variations (alternative scenarios)**

- 1. User keys-in a city name in combo box of "Latest Ultraviolet Index for" Frame**
- 9. User enters Latitude and Longitude in "Latest Ultraviolet Index for" Frame**
- 10. User clicks "Get UVI by Coordinates" button**
- 11. UviTab sends Latitude and Longitude from "Latest Ultraviolet Index for" Frame to OpenWeatherMapUvi**
- 12. OpenWeatherMapUvi returns UVI data back to UviTab**
- 13. UviTab displays UVI Data on "Coordinates & UVI" Frame**
- 14. If Speaker is "ON", UVI Data will be announced**

## **Extensions (error scenarios)**

- 12a. OpenWeatherMapUvi returns Exception back to UviTab**
  - 12a.1 System displays invalid Coordinates message**
  - 12a.2 user re-enters Latitude and Longitude**