# Installing Requirements

The whole project is on github and should be downloaded („cloned“) as a such.

NOTE: Additional Informations inside the READ.ME file

Step-by-step guide:

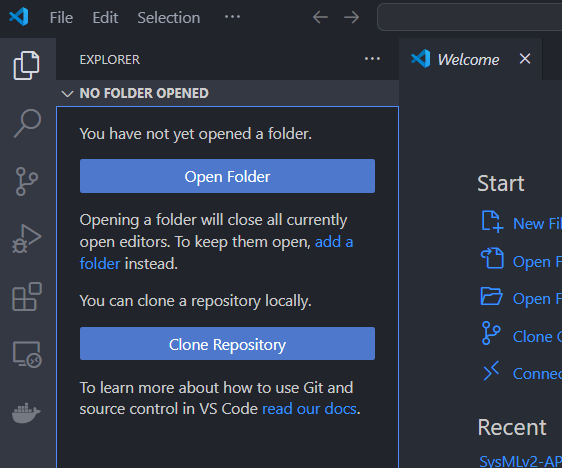
1. **Download and install Prerequisites:**

Prerequisits:

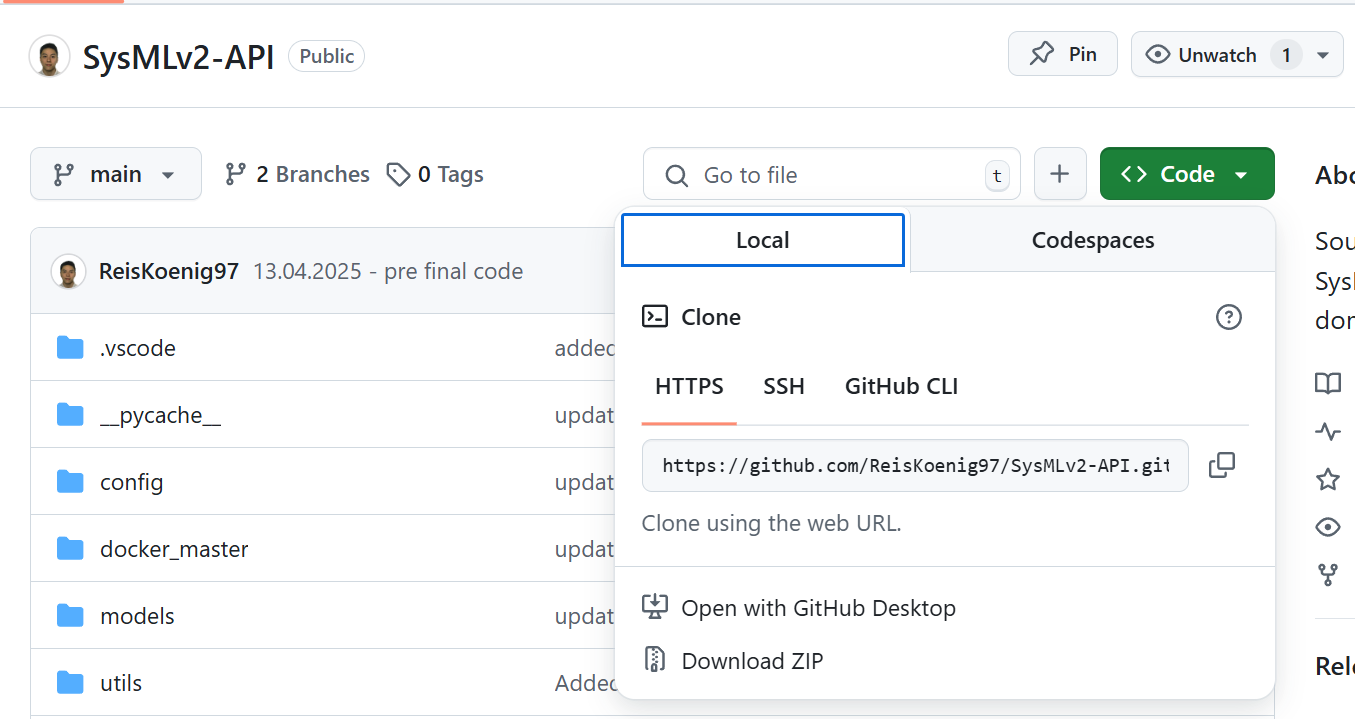
* **IDE** (Programmiertool) recommended: Visual Studio Code ([VS Code](https://code.visualstudio.com/download))
* [**Git**](https://git-scm.com/) for distributed version control system NOTE: To check if Git is installed: inside Terminal (e.g. PowerShell or cmd) „git --version“
* **Python** 3.12 NOTE: To check if python is installed: inside Terminal (e.g. PowerShell or cmd) „python --version“
* [**Docker**](https://www.docker.com/products/docker-desktop/) (Desktop version) (for SysON MBSE web-based modeling)
* Optionally **Conda** to install requirements (install via python: „pip install conda“ inside the terminal of the IDE)

1. **Cloning the project / download the project as a whole folder**

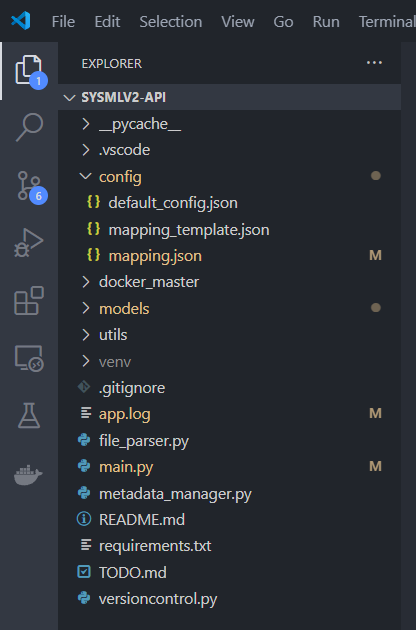
*Variant 1*: Cloning via GitHub and VS Code Terminal

1. Create a destination folder on your computer (that will be the github repository. This folder is connected to github. All project data e.g. files and folders will be uploaded.)
2. Open VS Code (or your IDE) and open the folder
3. Inside the terminal of the IDE: „git clone <https://github.com/ReisKoenig97/SysMLv2-API.git>“
4. OR: use „Clone Repository“ Button and provide the github link directly

*Variant 2:* Download project (folder) via GitHub manually

1. Go to the github link: <https://github.com/ReisKoenig97/SysMLv2-API> and download the git project as a ZIP folder 
2. Install the folder on your computer by unzipping the folder
3. Additionally you can set up that folder with you own GitHub Repository (recommended for version control)

Now you should end up with this folder structure:



1. **Installing requirements: python libraries**

Normally the program should install the library automatically by using the „requirements.txt“ file (which can be generated by python command:

* Python: "pip freeze > requirements.txt"

OR

* CONDA: "conda list --export > requirements.txt"

NOTE: If you change the program by adding/deleting libraries etc. you should generate a new requirements.txt and test it via virtual environment

NOTE: To run the code you have to run Docker Desktop as well (to check if the program installs the requirements automatically)

*OTHERWISE*

Install the libraries manually by using the commands inside VS Code

* „pip install …“ 🡪 replace … with the library that is often highlighted by the IDE
* OR
* „conda install …“ 🡪 replace … with the library that is often highlighted by the IDE (if pip does not install the library correctly)

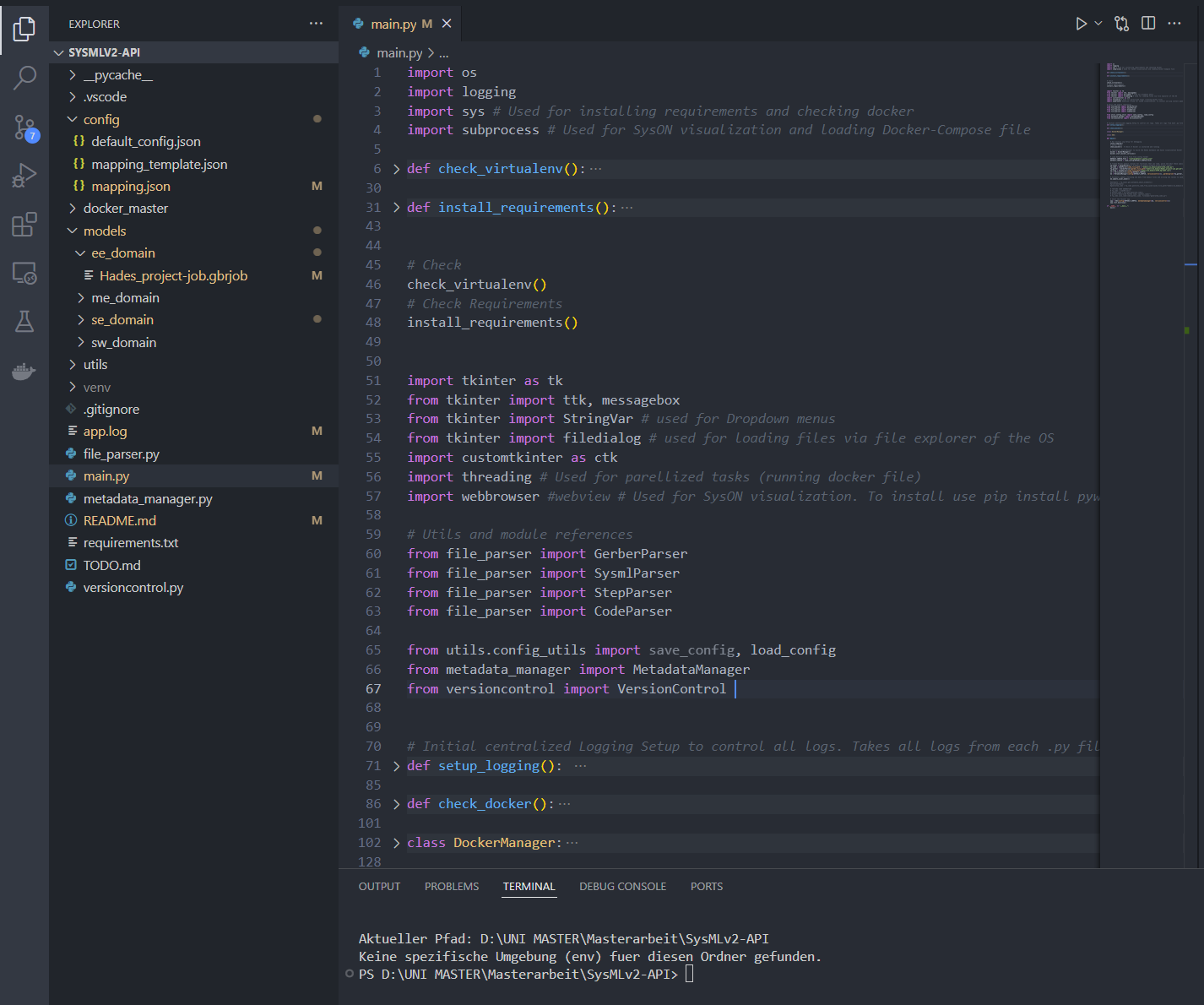
Currently the libraries that need to be installed:

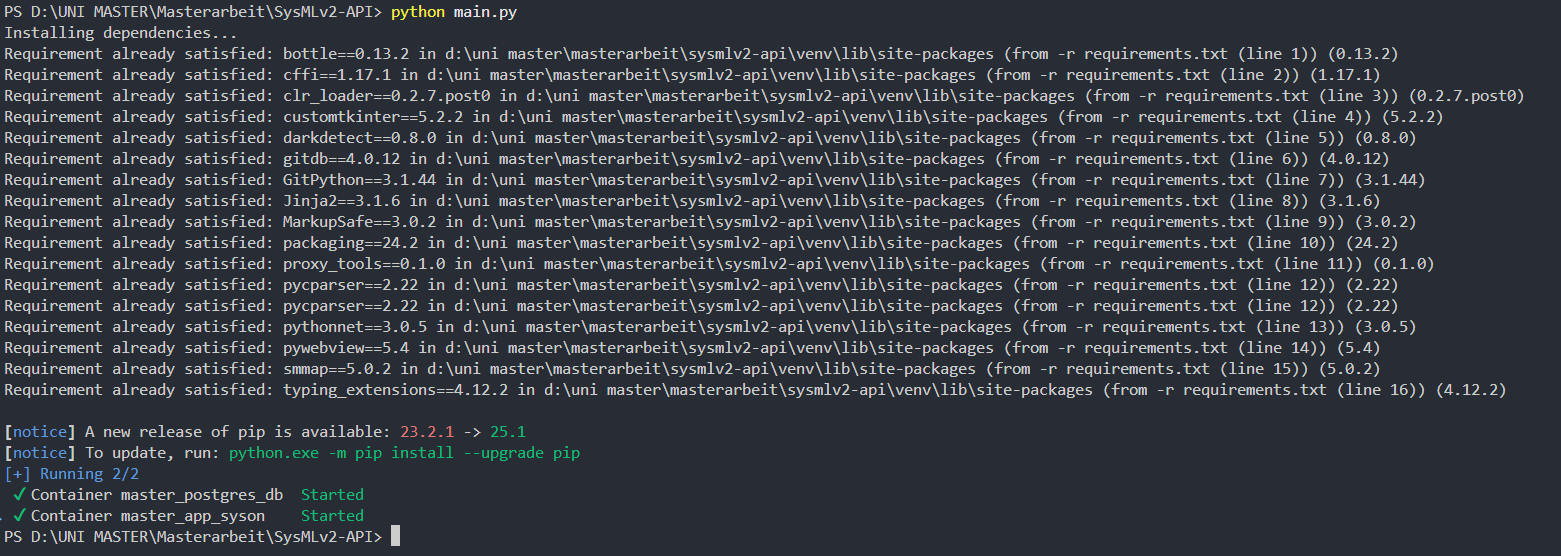
* Tkinter
* Customtkinter
* Threading
* Webbrowser
* Jinja2
* Git
* Re

If you feel unsure: search the library and check if that library is inside the in-built python libraries

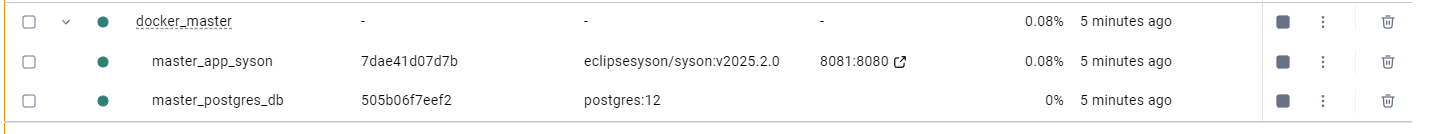
# Running the program

To run the program there are two main ways:

1. Using the VS Code „Run“ Button. Select the „main.py“ and click on the „Run“ Button 
2. Using the Command inside the terminal: „python main.py“

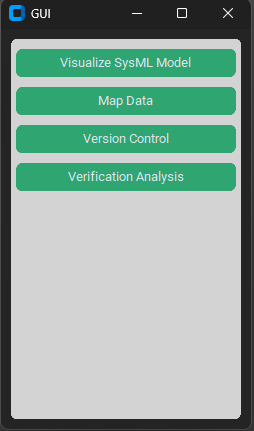


After running the program inside Docker Desktop it should look like this:



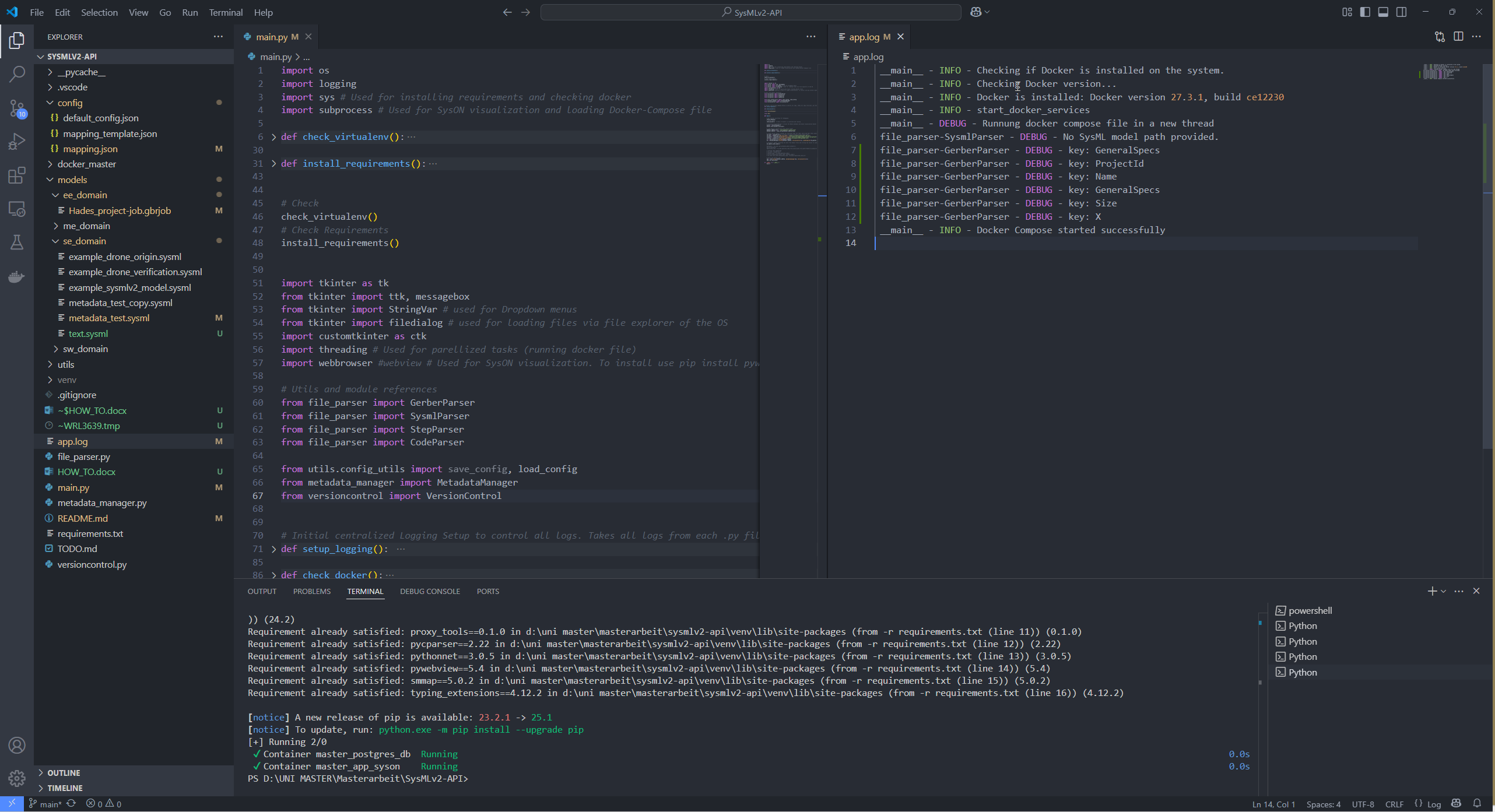
NOTE: If you stop the program, be aware that the docker container are still running! You have to turn them off manually.

# Using the program / How the program works

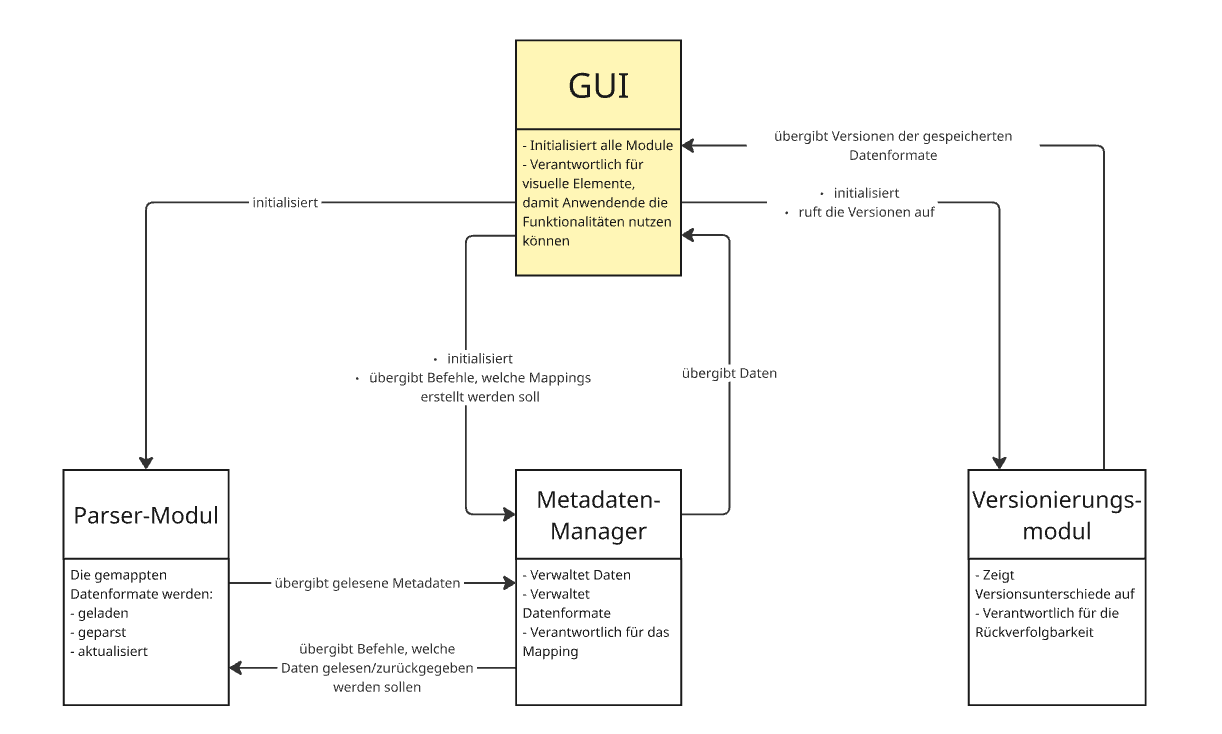
After the program starts it should provide a clean GUI

At **the current state there are 4 Buttons/Functions**

* **Visualize SysML Model**: Opens a web browser with the SYSON Addon (docker container syson container + syson data storage container). The user can visualize SysML models by uploading the sysml file inside the syson addon
* **Map Data**: Opens a popup where the user makes the main mapping elements by providing sysml file and domain file.
* **Version Control**: Opens a popup where the user can choose their sysml and domain files to check the versions that are tracked with git
* **Verification Analysis**: Opens a popup where the user selects a sysml file that MUST have a constraint definition and constraint usage. In the current state of the program the program can ONLY verify a specific sysml file that is used for demonstration (part-hardcoded)
* To Debug the code / see what inside the Code happens:
  + Use the file “app.log” 🡪 I recommend to split the window inside VS Code when running the program



The Code is divided into functional classes that are separated:

NOTE: The GUI element is the main.py file 

# Advancing the code

To advance the code use the comments inside each class and function to understand the code

* To add additional domain file parser use the file\_parser.py
  + Each class inside that python file contains functions to read and parse the specific standardized domain file
  + The main way to read/parse and update a domain file is set in that class and is mainly achieved by the python library “regex” aka “re” which simply searches specific keywords
  + NOTE: Some domain files like JSON or XML already have libraries that can be used to dissect information. I recommend to use that approach first!

# (My own opinion on) Possible Improvements

* Better SysML file parsing using different parsing methods/techniques that aren’t keyword related
* Transfering the custom GUI to webbrowser based and menu