Bedienungsanleitung

Zur Verwendung der Software muss eine passende conda-Umgebung erstellt werden. Hierfür liegt eine vorkonfigurierte `env.yml` Datei bei.

Führe folgende Schritte aus:

1. Erstellung einer passenden conda-Umgebung:

```
conda env create -f env.yml
```

2. Aktivierung dieser Umgebung, in der nun alles notwendige installiert ist:

```
conda activate UML
```

3. Aufruf des Webservers, welcher automatisch das passende Browserfenster öffnet:

```
python3 webserver.py
```

Alternativ kann eine bestehende Umgebung wie folgt aktualisiert werden:

1. Navigiere in deine conda-Umgebung:

```
conda activate <Umgebungsname>
```

2. Aktualisiere diese Umgebung:

```
conda env update -f env.yml
```

Info

Falls das Browserfenster sich nicht automatisch öffnen sollte, rufe im Browser (vorzugsweise Chrome oder Safari) folgende URL auf:

```
http://127.0.0.1:5001/
```

FLASK

Der Webserver wird mithilfe von Flask betrieben. Flask ist ein leichtgewichtiges Web-Framework für die Entwicklung von Webanwendungen in Python. Flask benötigt eine bestimmte Ordnerstruktur.

Die Ordnerstruktur und dessen Namen müssen zwingend wie gegeben bleiben!

```
root
|
|----static
|       |----node_modules
|
|----templates
|       |----index.html
|
|----webserver.py
```

static

Der static-Ordner in Flask wird verwendet, um statische Dateien wie CSS-Dateien, JavaScript-Dateien, Bilder und andere Ressourcen zu speichern, die von den Webseiten einer Flask-Anwendung verwendet werden. Innerhalb des static Ordners kann eine beliebige Ordnerstruktur vorliegen. Der in static liegende Ordner node_modules beinhaltet JavaScript-basierte Entwicklungstools und -bibliotheken, wie in unserem Fall z.B. chart.js und bootstrap.

templates

Der templates-Ordner beinhaltet lediglich die .html Dateien. Im Normalfall also nur eine index.html. Ich habe in dem Code jedoch Unterteilungen in verschiedene "Codeschnipsel" gemacht und diese in verschiedene Dateien ausgelagert, um eine bessere Übersichtlichkeit zu gewährleisten.

webserver.py

Diese Datei beinhaltet alle Funktionen, des Webservers. Von hier aus startet auch der Webserver.

JavaScript und deren Erweiterungen

Um die Erweiterungen zu benutzen werden diese in der directory `static/node_modules` aktuell gehalten.

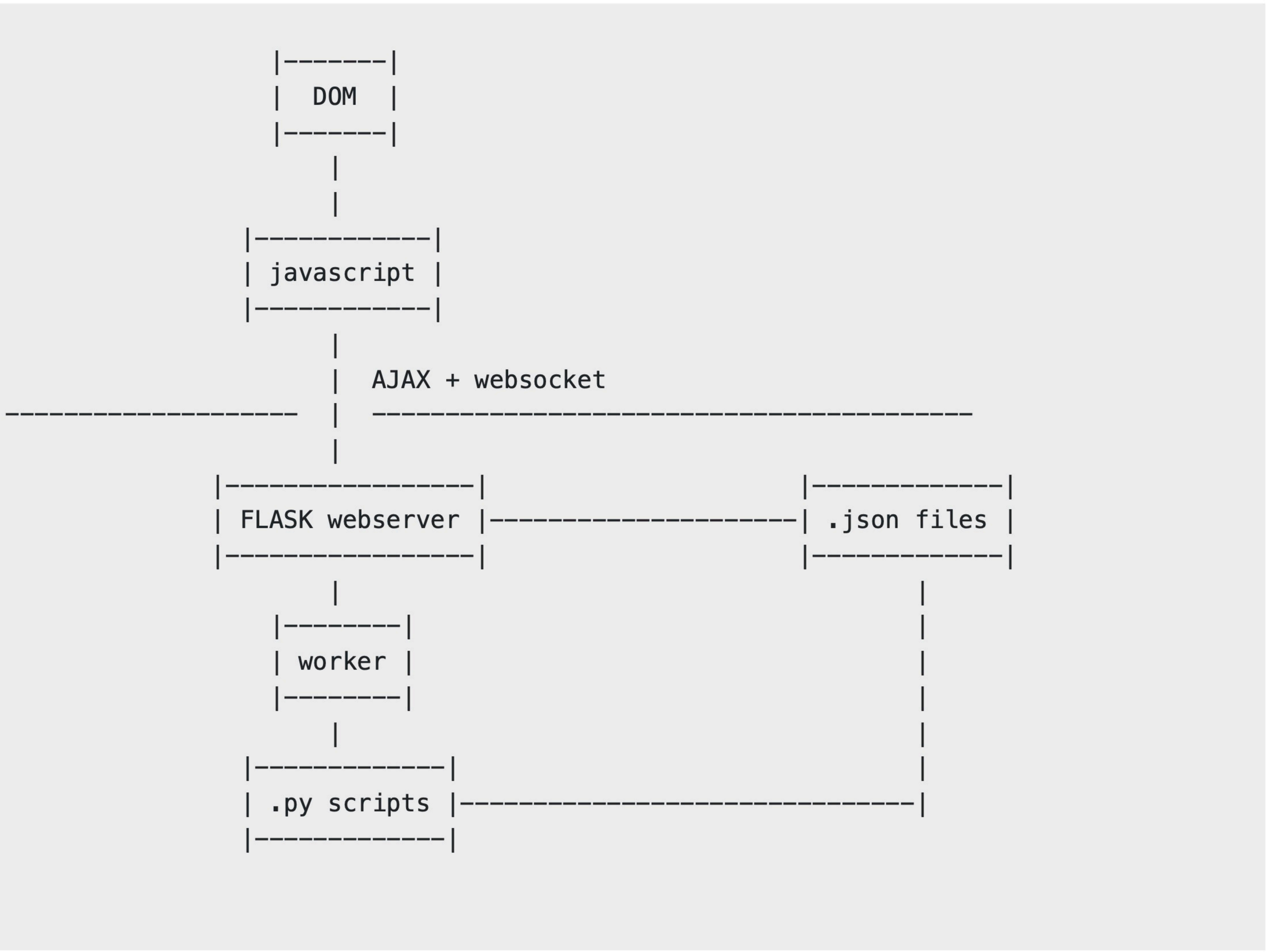
chart.js

Als Darstellung der Daten kann chart.js verwendet werden. Es ist eine gut zu bedienendes und sehr anpassbare Bibliothek zur Erstellung von Diagrammen jeglicher Art. Graphen von Chart.js sind im laufenden Prozess aktualisierbar. Link: https://www.chartjs.org/

bootstrap

Zur Erleichterung können von bootstrap, ähnlich wie bei Streamlit, fertige Elemente eingebunden werden. Also zum Beispiel slider oder buttons etc. Link: https://getbootstrap.com/docs/5.3/forms/range/ (Beispiel des Sliders)

grundlegende Softwarearchitektur



Links

Tabelle für: ToDo's & SUS-Score:
https://docs.google.com/spreadsheets/d/128ToeKJUO-rZcMXk5bAOppTXJHs30mdFSz-yu5_1URE/edit#gid=0

Milestone-Presentation:

https://docs.google.com/presentation/d/1SL-duqA5HIOCD0AuGBo_jJ9eiIqQ4VWo91K07jfmLDw/edit#slide=id.g25316d8211d_0_0

Final-Presentation:

https://docs.google.com/presentation/d/18Zy3pfBa0rSG0zK5OTw2aYTbENm83aqC2pLiTxilP2s/edit?usp=sharing