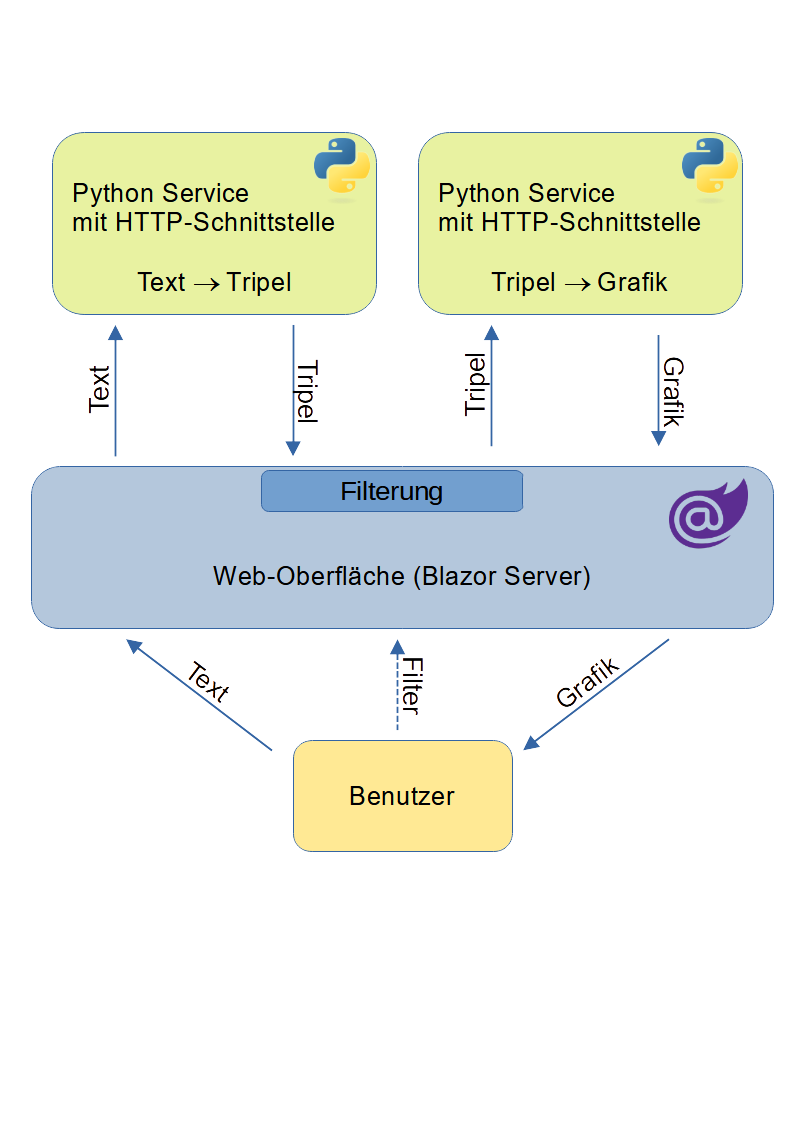
1. **Grobe Lösungsskizze**

Es soll eine Anwendung zur automatischen Generierung von *Concept Maps* aus einem vom Benutzer eingegebenen Text entwickelt werden.

Die Benutzeroberfläche bildet eine Blazor (Server) Webanwendung, welche in C# geschrieben wird. Dies erleichtert die Entwicklung enorm, weil kein JavaScript-Code und keine separaten APIs für die Schnittstelle *Browser → Server* implementiert werden müssen.

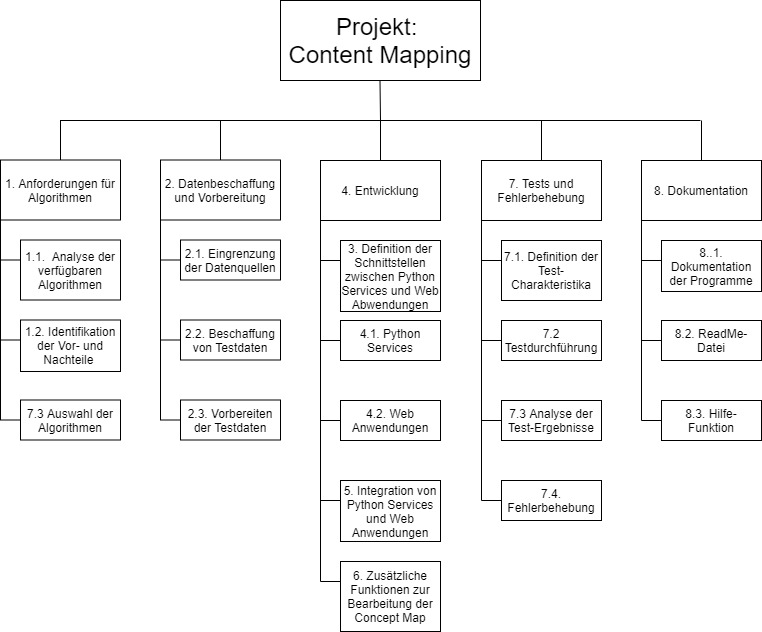
Um die *Concept Map* zu generieren werden die Python-Services im Hintergrund aufgerufen:

* Zunächst werden aus dem Text die Tripel extrahiert. Jedes Tripel stellt die Beziehung von Wörtern untereinander dar. Dazu wird der Service auf der linken Seite aufgerufen.
* Anhand diesen Tripel kann dann die *Concept Map* erstellt werden. Der Benutzer soll jedoch die Möglichkeit haben, die Erstellung mittels Filtern zu beeinflussen. Die vorher ermittelten Tripel könnten beispielsweise mittels Blocklist, Allowlist o.ä. gefiltert werden, bevor sie an den zweiten Service an der rechten Seite übergeben werden.

Die Bereitstellung (Deployment) soll mittels Docker-Containern ermöglicht werden. Mittels Docker Compose wäre das Deployment aller Container gemeinsam mit einem Konsolenbefehl machbar.

Ggf. könnten die Python-Services auch in einen gemeinsamen Service bzw. Container zusammengeführt werden – dies wird noch während der Umsetzung evaluiert.

1. **Definition Work Breakdown Structure**



1. **Team (Adriana)**

**3.1 Wer kümmert sich um was?**

Sammeln von relevanten Texten/Informationen für Testcase

* welche Algorithmen gibt es (Vor/Nachteile etc.)

**NLTK (Hauptverantwortlich: Adriana Mikuteit)**

Paket 1 – Erstellung von Triplets

Paket 2 – Generierung der Grafik (**save fig as svg)**

**PyTorch (Hauptverantwortlich: Korvin Walter)**

Paket 1 – Erstellung von Triplets

Paket 2 – Generierung der Grafik (**save fig as svg)**

**Webanwendung (Hauptverantwortlich: Sven Nicolai)**

Paket 1 – Entwicklung der Webanwendung

Paket 2 - Integration von Concept Maps in die Webanwendung

Nachträgliche Anpassung der Concept Map (Nice-to-Have)

**Evaluierung und Test**  **(Hauptverantwortlich: Korvin Walter)**

* Auswertung (Quantitativ/Qualitativ)
* Katalog zur Bewertung der erstellten Concept Map
* Testcases

**3.2 Way-of-Working (WAS MUSS DA ALLES REIN? WIE WOLLEN WIR DIE AUFGABEN TRACKEN?)**

* Agile Entwicklungsmethode mit zweiwöchigen Statusreports (1 Seite)
* verwendet Tools wie GitHub (DISCORD/ARBEITSTEILUNG?)
* Die Webanwendungs-Entwicklung erfolgt unter Verwendung von Blazor.

1. **Grobe Zeitplanung (auf Milestones achten) -> alle**

WO/WAS WÄREN Milestones?

Datenextraktion - X Wochen Datum:

Datenvorverarbeitung - X Wochen Datum:

Modellierung - X Wochen Datum:

Visualisierung - X Wochen Datum:

Evaluierung und Test - X Woche Datum:

Webanwendungs-Entwicklung - Wochen Datum:

Integration von Concept Maps in die Webanwendung - Datum:

Benutzerinteraktion mit Concept Maps - Datum:

Milestone 1 – DemoVersion

1. **Beschreibung Datensammlung** (Was genau versteht er hier runter?) -> alle, Frage per Mail stellen

* Datensammlung aus einer bestimmten Domäne (z.B. Medizin)
* ~~Verwendung von RDF-Daten zur Generierung der Concept Maps (WIE?, ich kenne nur die Art über Texte und dann Extraktion etc.)~~
  1. Umfang (nur Datensammlung oder das Projekt?)
* Generierung von Concept Maps aus Text-Datei (bzw. RDF)
* User kann Concept Maps in der Webanwendung anzeigen und bearbeiten

5.2 Charakteristik (nur Datensammlung oder das Projekt?)

- englisch

5.3. Methode: (nur Datensammlung oder das Projekt?)

- Algorithmen

1. Evaluation & Test -> alle

Für den Test und die Evaluation der Concept Maps des fertigen Programms sollen quantitative und qualitative Kriterien verwendet werden, um den Erfolg des Projekts zu verifizieren. Hierdurch sollen auch die zwei Resultate der beiden Python Services (NLTK und PyTorch) mit einander verglichen werden, um mögliche Vor- und Nachteile der beiden Implementierungen identifizieren. Hierzu wird parallel zum laufenden Projekt ein Kriterien-Katalog entwickelt und ein Set von Testcases erstellt. Quantitative Kriterien könnten z.B. die korrekte Anzahl von identifizierten Objekten und Relationen zu einem Test-Text oder eine Verteilungsanalyse der Relationen sein. Qualitative Bewertungen können durch die subjektive Bewertungen der Maps z.B. in Hinsicht auf Übersichtlichkeit erfolgen.

Des Weiteren wird auch die Web-Anwendung mittels qualitativer Kriterien auf die Benutzerfreundlichkeit und mittels der Verwendung von Testcases auf ihre Zuverlässigkeit getestet.

Auswertung der Concept Maps (Quantitativ/Qualitativ)

Katalog zur Bewertung der erstellten Concept Map

Testcases werden definiert und auf Basis der Bewertungsmatrix als Erfolgreich/nicht erfolgreich klassifiziert.

Die Concept Maps werden auf ihre Genauigkeit und Relevanz in Bezug auf die Texte, aus denen sie generiert wurden, überprüft.

Die Webanwendung wird auf ihre Benutzerfreundlichkeit und Zuverlässigkeit getestet.

1. Aufbereitung & Visualisierung →alle

Die Dokumentation inklusive Readme-Datei wird bereits parallel zur Entwicklung der Programmteile geschrieben. Sowohl für die Zwischenpräsentation als auch der Abschlusspräsentation werden Demo-Cases vorbereitet, anhand derer die Funktionalitäten des Programms gezeigt werden können. Die quantitativen und qualitativen Ergebnisse der abschließenden Programm-Evaluation werden für die Abschlusspräsentation übersichtlich dargestellt.

Showcase vorbereiten (Democase)

Dokumentation wird parallel gepflegt (Readme)

Präsentation der Ergebnisse

Aufbereitung der Testergebnisse