



Multimodal Parliament Explorer - Projektdokumentation

Kai Wöllstein Philipp Schneider Philipp Hein Philipp Landmann

21. März 2025

Inhaltsverzeichnis

1	Kurze Vorstellung des Projektes	2
1.1	Screenshots	2
1.2	Techstack	3
2	Organisation des Projektes	3
2.1	Aufteilung in Module	3
2.2	Nutzung von Tickets / Issues	4
2.3	Kommunikation im Team	4
3	Projektverlauf und Dokumentation der Eigenleistungen	4

1 Kurze Vorstellung des Projektes

Unser Abschlussprojekt - Multimodaler Parlament Explorer - ist eine Applikation, mit der man die vielen verschiedenen Reden im deutschen Bundestag visualisieren und analysieren kann.

Die Informationen dafür sind aus verschiedenen Quellen zusammengetragen und wir bieten den Mehrwert, diese Informationen möglichst übersichtlich und nützlich darzustellen.

1.1 Screenshots

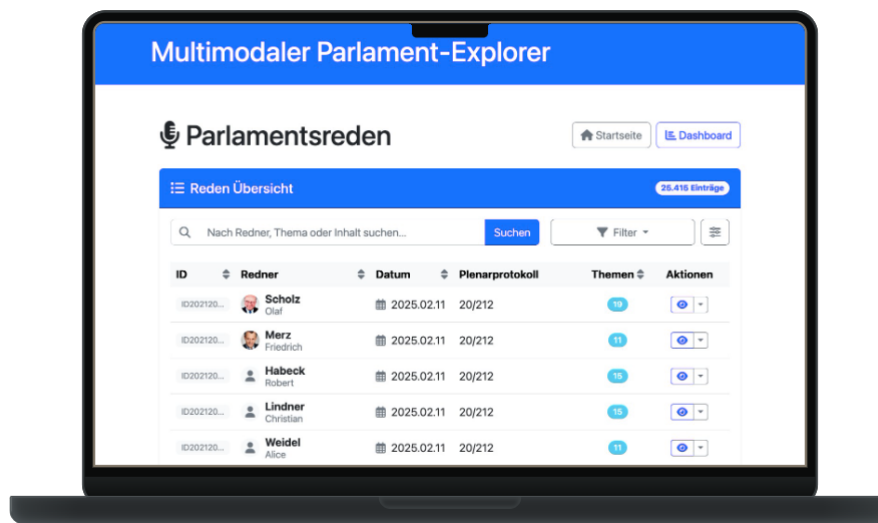


Abbildung 1: Screenshot 1

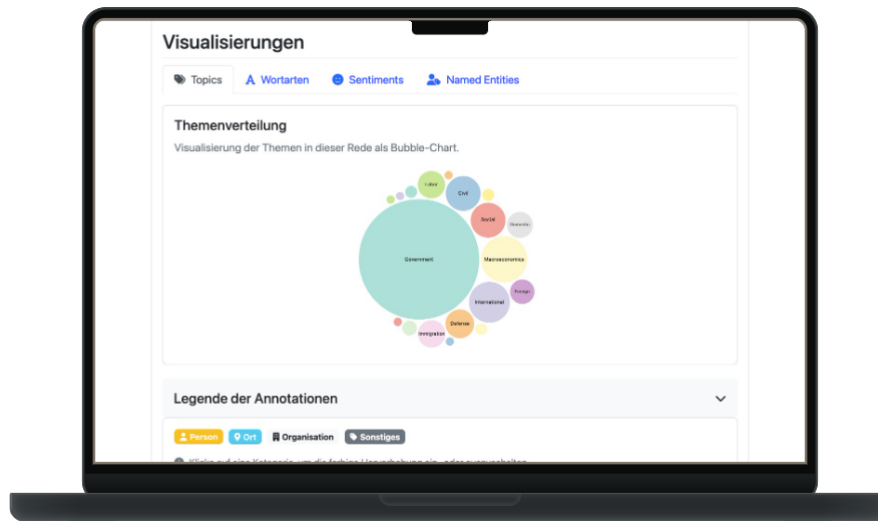


Abbildung 2: Screenshot 2

1.2 Techstack

- **Java 17** im Backend
- **Javalin** als Webserver
- **Freemarker** als Templating-Engine
- **MongoDB** als Datenbank
- **D3.js / TikZ** zum Generieren von Diagrammen

2 Organisation des Projektes

2.1 Aufteilung in Module

Damit wir möglichst effektiv an dem Projekt arbeiten und möglichst schnell vorankommen, haben wir direkt am Anfang die Aufgabenstellung in mehrere Teile aufgeteilt und diese folgendermaßen untereinander im Team aufgeteilt:

- **Diagramme** Philipp Hein
- **Backend** Philipp Hein, Philipp Schneider
- **NLP** Kai Wöllstein
- **Frontend** Philipp Schneider
- **Dokumentation** Philipp Landmann

Natürlich ist die Zuteilung einigermaßen flexibel gehalten; einige Bereiche können sich überlappen, bzw. man kann auch in anderen Bereichen als dem eigenen Hauptbereich tätig werden.

21.03.2025	PS = Philipp Schneider	PL = Philipp Landmann	PH = Philipp Hein	KW = Kai Wollstein		
Meilensteine/Aufgaben	Wer?	Woche 1	Woche 2	Woche 3	Woche 4	Woche 5
1 - Projektplanung und Design						
a) Gantt-Diagramm erstellen	PH					
b) Package- Klassen und Use-Case-Diagramm	PH					
c) Mock-Up für Visualisierungen sowie User-Interface	KW					
2 - Backend						
a) Datenstruktur basierend auf Aufgabe 1	PH					
b) Implementation des MongoDB-Handler	PH					
c) Zugang/Konfig mit Klassenrepräsentation	PH					
d) Import der Parlamentstexte/Protokolle	PS					
e) Abbilden der Abgeordneten der 20. Wahlperiode + Foto	PH/PS					
f) automatische Erfassung/Download von Videobeiträgen pro Rede	PS					
g) RESTful Webservice - schnittstellen + swagger	PH					
2.1 Natural Language Processing						
1) Klassen und Methoden mit Docker UIMA Interface	KW					
2) alle eingelesenen Debatten mit NLP/UIMA-Pipeline vorverarbeiten	KW					
3) Speichern der Ergebnisse	KW					
4) UIMA TypeSystem	-					
3/3.1 Frontend/ Visualisierung und Abfrage von Parlamentsdebatten						
a) Visualisierungen implementieren	PS					
b) Visualisierungen einschränken/Suchfilter	PS					
c) generische Suchfunktion implementieren	PS					
d) Volltext Visualisierung implementieren	PS					
e) Video einbinden	PS					
3.2 - Export und Präsentation der Parlamentsdebatten						
a) Reden exporter als PDF mit Latex	PL					
b) Export mit toTex() methode Objektorientiert	PL					
c) Bausteine als Ressource abbilden	PL					
d) XML-Exporter Analog realisieren	PL					
4 - Dokumentation						
a) JavaDoc	Alle					
b) Benutzerhandbuch	PL					
c) Projektdokumentation	PL					
d) GitLabPage	PL					

Abbildung 3: Projektverlauf - als Gantt-Diagramm dargestellt

2.2 Nutzung von Tickets / Issues

Um die Übersicht über die vielen Teilaufgaben zu behalten, haben wir intensiv GitLab-Issues genutzt. Damit konnten wir einfach sehen, wie weit wir schon sind und was noch zu tun ist, bzw. entstandene Deadlocks erkennen.

Besonders bei den Issue-Boards kann man sehen, wie viele Issues pro Person noch gelöst werden müssen.

2.3 Kommunikation im Team

Die Kommunikation im Team wurde mit einer ca. wöchentlichen Videokonferenz gelöst, in der wir besprechen, was wir als nächstes machen und was wir bisher schon erreicht haben.

Zusätzlich hatten wir eine WhatsApp-Gruppe, sowie die GitLab Issues als Kommunikationskanäle.

3 Projektverlauf und Dokumentation der Eigenleistungen

In Abbildung 3 sieht man ein Gantt-Diagramm des Projektverlaufs. Man kann erkennen - Die Aufgaben sind ziemlich gleichmäßig auf die Gruppe aufgeteilt, und die Zuteilung, die wir am Anfang getroffen haben, hat sich bestätigt.