**Herangehensweise**

* Warum haben wir uns für Angular entschieden?
  + Neben der Frage, ob wir das ganze klassisch mit HTML und CSS umsetzen, kam uns zusätzlich die Frage auf, ob wir stattdessen ein Framework verwenden wollen. Nach einer Diskussionsrunde haben wir uns für Angular entschieden. Hauptgrund war, dass alle Projektteilnehmer bereits erste Erfahrung mit diesem Framework hatte und das dieses Projekt auch im Fokus stand, weitere Erfahrungen in Angular sammeln zu können.
* Versionierung
  + Damit sich die Möglichkeit bietet, dass jeder unabhängig des anderen an dem Projekt arbeiten und Programmstände einfach zusammengeführt werden können, haben wir uns entschieden, das Projekt auf GitHub hochzuladen und somit Git als Versionierungstool zu benutzen.
* Deployment
  + Da wir eine einfache Installation und eine effektive Entwicklungsinfrastruktur gewährleisten wollten, haben wir uns entschieden, alle Dependencies in Docker laufen zu lassen. Dadurch ist es möglich, durch einen einzigen Command die Anwendung zum Laufen zu kriegen.

**Herausforderungen**

Nicht alle von uns gesteckten Ziele konnten aufgrund des Zeitmangels umgesetzt werden:

- Prüfen, ob eine Session abgelaufen ist

- Einen 100%igen Restful Service implementieren.

- Aneignen des gleichen Wissenstand in allen Gebieten durch alle Teilnehmer.

- Viel Zeit ging auch durch das Erlernen neuer Technologien, wie z. B. Docker, PHP verloren

**Zeitplan**

Ca. 30% des Projekts wurde in der Berufsschule erledigt, der Rest musste in der Freizeit fertiggestellt werden.

**Aufgabenverteilung**

Developer:

|  |  |
| --- | --- |
| Frontend | Backend |
| Samuel S. | Thomas S. |
| Marco H. | Michael W. |

**Samuel S. – Frontend Styling**

Implementierung des Designs von Login und Hauptseite, sowie Styling von Elementen wie Header und Footer

**Marco H. – Frontend Logik**

Authentifizierung, Kommunikation der Komponenten, Anlegen und Logikimplementierungen der Services, Login, Models und teilweise Komponentenbuilding, sowie auch deren Logikimplementierung

**Thomas S. – Backend**

Grundlegende Backend-Architektur nach MVC-Pattern, Ausarbeitung des Datenbankzugriffs mit PDO, Anlegen und Ausarbeiten der Repository Funktionen vorallem für den MemberController (+Mapping), JWT Generierung und Abgleichung, CORS Implementierung, Implementierung von Basisklassen und Methoden (bspw. für http-Errors), Änderung der Datenbank (ON CASCADE delete), Troubleshooting Docker

**Michael W. – Backend**

Speicherung und Validierung der Login Daten (inklusive Passwort salting, peppering und hashing), Anbindung der Login-Daten mit der Datenbank, Implementierung von Basisklassen und Methoden (bspw. für http-Errors), Anlegen des Github-Repos, Anlegen von Dokumentationsdateien (im .md Dateiformat), Einrichtung von Docker (docker-compose + dockerfiles)