

MedPlanner

Web-Anwendungsentwicklung Sommersemester 2021

Egidia Cenko
Medieninformatik
e.cenko@oth-aw.de

Madina Kamalova
Medieninformatik
m.kamalova@oth-aw.de

Matthias Schön
Medieninformatik
m.schoen@oth-aw.de

Christoph Schuster
Medieninformatik
c.schuster1@oth-aw.de

Andrei Trukhin
Medieninformatik
a.trukhin@oth-aw.de

Zusammenfassung—Beschreibung der Software-Architektur für das Projekt MedPlanner.
Index Terms—TODO

I. ÜBERBLICK

A. Mission Statement

MedPlanner bietet die Möglichkeit, ärztliche Termine übersichtlich zu verwalten. Es handelt sich hierbei um eine Web-Anwendung, die gezielt auf das Selbstmanagement von Arztterminen abgestimmt ist. MedPlanner ist auf verschiedenen Geräten, einschließlich Computern, Smartphones und Tablets verfügbar. Mithilfe von MedPlanner können zukünftige Arzttermine eingetragen und geplant werden. Es ist vor allem für Privatpersonen gedacht, welche somit einen Überblick über die Vielzahl ärztlicher Untersuchungen behalten können. Zu den wesentlichen Features gehört, eigene Termine in den Kalender einzutragen und diesen mit Tags und Notizen zu versehen. Weiterhin ist es möglich, Kontaktinformationen für die eigenen Ärzte abzuspeichern. So kann man zum Beispiel immer die Telefonnummer, Adresse und, falls vorhanden, die Webseite der Arztpraxis einsehen, ohne extra vor einer Terminvereinbarung immer wieder nach den nötigen Informationen zu suchen. MedPlanner bietet außerdem die Funktion Erinnerungs-Mails für das Vereinbaren von Terminen zu erhalten, sog. *Reminder*. So bekommt der Patient 24 Stunden vor dem Arzttermin eine E-Mail zugesandt, indem nochmal die wichtigsten Informationen enthalten sind.

B. Architekturziele

Tabelle I
QUALITÄTSZIELE

Ziel	Erklärung
Benutzbarkeit	Intuitive Bedienbarkeit und schnelle Erlernbarkeit
Sicherheit	Inhalte sind vor unberechtigt Zugriff geschützt
Wartbarkeit	Leichte Erweiterbarkeit und Änderung
Leicht zu betreiben	Die Anwendung kann ohne größere Anpassungen genutzt werden

C. Kontextabgrenzung

TODO: Überblick über System als Blackbox?

D. Herausforderungen, Schmerzpunkte und zentrale Randbedingungen

TODO: Schwierigkeiten bei der Umsetzung? Bzgl Backend z.B. die Customized User Services mit Token-Authentifizierung; Allgemein gesagt die grundlegenden Elemente, damit ein Patient die Funktionalitäten der Web-Anwendung nutzen kann.

II. LÖSUNGSSTRATEGIE

A. Lösungsansätze

Folgende Auflistung zeigt die Lösungsansätze zu den Qualitätszielen zu MedPlanner.

- **Benutzbarkeit:**
 - Intuitives, modernes User Interface
 - Filterung für eine erhöhte Übersichtlichkeit der Informationen
- **Sicherheit:**
 - zustandslose Authentifizierung mittels Tokens
 - Passwortspeicherung in Form eines Hashwertes
- **Wartbarkeit:**
 - Modulare Implementierung in Python
 - #TODO: inwiefern ist das Frontend wartbar?
- **Leicht zu betreiben:**
 - üblicher Web-Browser als Client genügt
 - lokale Speicherung der RDBMS SQLite

B. Technologie-Stack

Im Frontend nutzt Medplanner Angular, im Backend das Framework Django in Kombination mit Django REST genutzt, um WEB-APIs für das Frontend bereitzustellen.

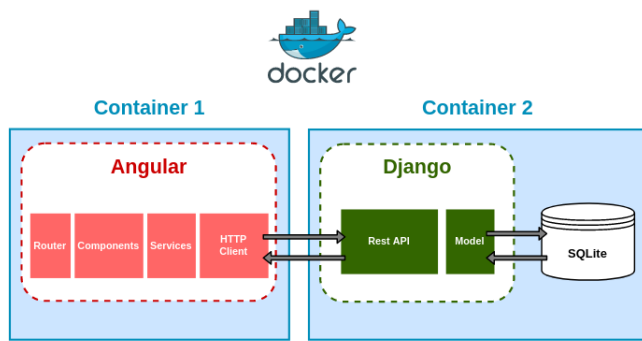


Abbildung 1. Überblick über die Funktionalität des Technologie-Stacks

Architekturentscheidungen hier. Angular als client-side MVC (Architekturprinzip), Architekturstile (micro-service), bestimmtes Vorgehen (user-centered-design)

III. FAZIT UND AUSBLICK

LITERATUR

- [1] <https://www.docker.com/>
- [2] <https://angular.io/>
- [3] <https://www.django-rest-framework.org/>
- [4] <https://www.mongodb.com/de>
- [5] <https://www.doctorbox.de/patienten.jsp>
- [6] <https://www.notion.so/>