

**«Hert Am Struggle»**

**«Digitale Arztrezepte»**

**Technische Informationen für die Jury**

Aktueller Stand des Sourcecodes

* Link zu Github Repository

Hier ist der aktuelle Link zu unseren 2 Github-Repos. Front -und Backend sind aufgeteilt.

<https://github.com/orgs/HertAmStruggle/repositories>

Ausgangslage

* Worauf habt ihr euch fokussiert?
* Welche technischen Grundsatzentscheide habt ihr gefällt?

Wir wussten schnell, dass diese Challenge die interessanteste für uns ist. Unser Team besteht aus 4 Medizininformatikern und einem regulären Informatiker. Wir haben uns auf zwei Grundprobleme fokussiert, mit denen man heutzutage zu kämpfen hat. Einerseits das Problem mit dem schwierigen Handling von Arztrezepten. Oft sind sie unleserlich was sowohl für den Patienten aber auch für das Apothekenpersonal mühsam ist. Auch kann es zu Verwechslungen führen bei normalen Rezepten, dies ist potenziell gefährlich für den Patienten. Bei unserer Lösung haben wir auf eine eindeutige Identifikation des Patienten (AHV), Arztes (HIN-Email) und der Medikamente (ATC-Codes) gesetzt um solche Verwechslungen zu minimieren.

Auch wollten wir das Problem angehen, dass man Arztrezepte nicht mehr bei einer Apotheke abgeben muss, sondern digital dabeihaben kann. Somit ist man nicht mehr an eine Apotheke gebunden.

Auch war uns klar, dass wir eine Web-Applikation machen wollen. Ärzte sind bekanntlich schwierig zu überzeugen neue Digitale Lösungen zu verwenden. Eine Website ist viel einfacher einzusetzen als eine Applikation oder sogar ein Informationssystem. Das Ziel war also eine Win-Win-Win Situation zu erreichen bei der alle 3 Parteien profitieren können.

Technischer Aufbau

* Welche Komponenten und Frameworks habt ihr verwendet?
* Wozu und wie werden diese eingesetzt?

Allgemein sind wir alle versiert in Java, für alle ist es die Sprache bei denen wir die meiste Erfahrung haben.

**Backend:** Hier haben wir auf das Java Spring Boot API-Framework gesetzt. Das Framework wurde genutzt, um dem Frontend Daten und Funktionen zur Verfügung zu stellen und zu managen. Die Daten werden in einer Java-Klasse organisiert und gespeichert. Wir haben keine Persistierung der Daten vorgenommen. Dazu haben wir «Dummy-Daten» erstellt um den Prototyp funktional zu machen.

**Frontend:** Noch machen

Implementation

* Gibt es etwas Spezielles, was ihr zur Implementation erwähnen wollt?
* Was ist aus technischer Sicht besonders cool an eurer Lösung?

Wir sind als Team sehr zufrieden mit dem Grundsatz, dass ein Rezept nicht nur an einer Apotheke verwendet werden kann, sondern überall wo unser System unterstützt wird. Dies bringt einen extremen Mehrwert (sowohl Sicherheit wie auch Komfort) für den Patienten. Diesen Grundsatz konnten wir so programmieren, dass er funktional ist. Aber bis zu einem System, dass im Gesundheitswesen einsetzbar ist, bleibt noch ein sehr langer weg.

Abgrenzung / Offene Punkte

* Welche Abgrenzungen habt ihr bewusst vorgenommen und damit nicht implementiert? Weshalb?

Wir haben uns früh bewusst gegen eine Datenbank entschieden. Uns reichte die Zeit nicht einen funktionierenden Prototyp zu coden und die Daten zu persistieren. Hier mussten wir klar eine Priorität setzen und abgrenzen. Damit sind aber auch viele initiale Features weggefallen, die man noch zu einem späteren Zeitpunkt implementieren könnte. Mit einer vollwertigen Medizin-Datenbank kann der Arzt entlastet werden mit einfachen Medikamenten suche. Auch kann er bei der Entscheidung unterstützt und auf allfällige Fehler hingewiesen werden (zum Beispiel eine Indikation zweier Medikamente).