

**«Mechä Mode»**

**«SBB Online-Fahrplan Optimierung»**

**Technische Informationen für die Jury**

**Technische Informationen für die Jury**

Aktueller Stand des Sourcecodes

* <https://github.com/breadly7/verbindigs-mecher>

Ausgangslage

* Wir haben uns darauf fokussiert, die Online-Fahrplan-Optimierung der SBB zu verbessern, indem wir eine effiziente und benutzerfreundliche Lösung entwickelt haben.
* Wir haben uns entschieden, eine Microservices-Architektur zu verwenden, um die Skalierbarkeit und Wartbarkeit des Systems zu verbessern. Für das Backend haben wir Go verwendet, um von der hohen Performance und der einfachen Parallelisierung zu profitieren. Das Frontend wurde mit React entwickelt, um eine dynamische und reaktive Benutzeroberfläche zu gewährleisten.

Technischer Aufbau

* Backend:
  + Framework: Gin für die HTTP-Server-Implementierung.
  + Datenbank: SQLite für die Speicherung der Fahrplandaten.
  + Module: tripcomparator und triploader (beide selbst entwickelt) für die Verarbeitung und den Vergleich von Fahrplandaten.
* Frontend:
  + Framework: React für die Benutzeroberfläche.
  + Styling: Tailwind CSS für das Styling.
  + Build-Tool: Webpack für das Bundling der Ressourcen.
  + Weitere eingesetzte Libraries: react-select, react-router-dom

Implementation

* **Effiziente Datenverarbeitung**: Trotz der komplexen Datengrundlage konnten wir eine effiziente Methode entwickeln, um die Fahrplandaten zwischen verschiedenen Versionen zu vergleichen.
* **Hohe Performance**: Die Verwendung von Go im Backend ermöglicht eine hohe Performance und effiziente Parallelisierung, was besonders bei der Verarbeitung großer Datenmengen von Vorteil ist.
* **Robuste Fehlerbehandlung**: Wir haben eine robuste Fehlerbehandlung im Backend implementiert, um sicherzustellen, dass alle API-Endpunkte zuverlässig funktionieren.
* **Ansprechende Benutzeroberfläche**: Im Frontend haben wir mit React und Tailwind CSS eine dynamische und ansprechende Benutzeroberfläche geschaffen. Die grafische Darstellung der Unterschiede zwischen den Fahrplanversionen ist besonders gut gelungen und bietet den Nutzern eine klare und intuitive Übersicht.

Abgrenzung / Offene Punkte

* Wir haben uns entschieden, keine komplexen Benutzerverwaltungssysteme zu implementieren, da der Fokus auf der Optimierung des Fahrplans lag.
* Die Datenbank wurde bewusst als SQLite gewählt, um die Implementierung zu vereinfachen und die Entwicklung zu beschleunigen. Die Daten wurden uns auch bereits im SQLite-Format zur Verfügung gestellt. Eine Migration zu einer robusteren Datenbanklösung könnte in der Zukunft erfolgen (Performance-Verbesserung)