

**«Tübinger Attacke»**

**«Indoor Routing»**

**Technische Informationen für die Jury**

Technische Informationen

Aktueller Stand des Sourcecodes

Link zu GitHub Repository: <https://github.com/talionderhobbit/indoor_routing.git>

Der Code beinhaltet ein Flutter Prototyp Frontend das der CI der SBB-Apps nachempfunden ist.

Ausgangslage

## Worauf habt ihr euch fokussiert?

Unsere Idee ist, dass Sehbehinderte von anderen mitfahrenden/aussteigenden PassagierInnen Unterstützung anfordern können.

Hilfsbereite PassagierInnen können Sehbehinderten beim Gang durch den Bahnhof zur Seite stehen.

Dabei integrieren wir diese Funktionalität in die bereits etablierten Apps der SBB.

Die SBB gab uns den Hinweis, dass sie den konkreten Standort von Reisenden über Beacons in den Zügen und Bahnhöfen bereits feststellen kann. Wir nutzen diese Annahme.

Zudem ist die bestehende SBB inclusive App bereits barrierefrei von Sehbehinderten nutzbar.

## Welche technischen Grundsatzentscheide habt ihr gefällt?

* Wir konzentrieren uns auf ein userfreundliches Frontend und vernachlässigen das Backend im Prototypen.
* Wir bauen eine native App, die möglichst nah an die original SBB App kommt.

Technischer Aufbau

## Welche Komponenten und Frameworks habt ihr verwendet?

* Wir nutzen die Flutter-Komponenten Library der SBB, um nah an der CI zu sein.
* Wir nutzen Dart in Kombination mit Flutter als Frontend-Framework.
* Wir nutzen das Bahnhofkarten-Framework der SBB um Karten und Wegbeschreibungen von Bahnhöfen zu rendern.

## Wozu und wie werden diese eingesetzt?

* Flutter dient als Framework, um die App einfach und effizient für iOS und Android bereitzustellen.

Implementation

## Was ist aus technischer Sicht besonders cool an eurer Lösung?

* Cross Plattform Verfügbarkeit (iOS und Android)
* Einfache Integration des Frontends in die SBB-App, da diese auch in Flutter geschrieben wurde und wir uns an das CI gehalten haben.
* Sehbehinderte können selbst entscheiden, wie sie die App nutzen (Kopfhörer mit Transparenz Modus, Screen-Reader, etc)
* Durch die native App können wir Funktionalitäten wie Bluetooth und WLAN des Handys verwenden, um den Zug und den Wagon des Sehbehinderten über Beacons festzustellen. Der Sehbehinderte muss nur Buttons drücken, keinen Text eingeben.

Ausblick

* Konnektivität der Helfenden und Sehbehinderten via direkter Bluetooth Verbindung, bei schlechter Netzabdeckung.
* Entwicklung von Backend Funktionalität, um echte Daten zu rendern.

Abgrenzung / Offene Punkte

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

## Aus Zeitgründen kein Fokus auf das Backend