



«Tübinger Attacke» «Indoor Routing»

Technische Informationen für die Jury



Technische Informationen

Aktueller Stand des Sourcecodes

Der aktuelle Stand des Sourcecodes ist in einem GitHub-Repository verfügbar: [indoor_routing Repository](#). Der Code enthält ein Flutter-basiertes Prototyp-Frontend, das an die Corporate Identity (CI) der SBB-Apps angelehnt ist.

Ausgangslage

Worauf habt ihr euch fokussiert?

Unsere Idee zielt darauf ab, sehbehinderten Reisenden die Möglichkeit zu bieten, Unterstützung von mitfahrenden oder mit-aussteigenden PassagierInnen anzufordern. Diese Unterstützung erfolgt durch PassagierInnen, die in ihrem Profil hinterlegt haben, dass sie bereit sind, zu helfen. Dadurch können hilfsbereite PassagierInnen sehbehinderten Menschen bei der Orientierung und beim Weg durch den Bahnhof zur Seite stehen.

Diese Funktionalität soll nahtlos in die bereits etablierten Apps der SBB integriert werden. Die SBB hat uns darüber informiert, dass sie den Standort von Reisenden bereits über Beacons in Zügen und Bahnhöfen ermitteln kann. Diese Technologie bildet die Grundlage für unsere Lösung. Darüber hinaus ist die bestehende SBB Inclusive App bereits barrierefrei und für sehbehinderte NutzerInnen optimiert.

Welche technischen Grundsatzentscheide habt ihr gefällt?

Wir haben uns bewusst entschieden, den Fokus auf ein benutzerfreundliches und barrierefreies Frontend zu legen. Unser Ziel ist es, eine App zu entwickeln, die der originalen SBB-App möglichst nahekommt und unsere Funktionalität so integriert, als wäre sie bereits vollständig in die bestehende App eingebettet.

Technischer Aufbau

Welche Komponenten und Frameworks habt ihr verwendet?

Wir setzen die Flutter-Komponenten-Bibliothek der SBB ein, um die CI optimal umzusetzen. Als Frontend-Framework nutzen wir Dart in Kombination mit Flutter. Zusätzlich greifen wir auf das Bahnhofskarten-Framework der SBB zurück, um Karten und Wegbeschreibungen innerhalb von Bahnhöfen zu rendern.

Wozu und wie werden diese eingesetzt?

Flutter dient als Framework, um die App effizient sowohl für iOS als auch Android bereitzustellen. Die Flutter-Komponenten-Bibliothek ermöglicht uns eine realistische Nachbildung der Funktionalitäten innerhalb der SBB-Apps. Das SBB-Bahnhofskarten-Framework ermöglicht es, detaillierte Karten und Routen innerhalb der Bahnhöfe anzuzeigen, was insbesondere den hilfsbereiten PassagierInnen bei der Navigation mit sehbehinderten NutzerInnen unterstützt.

Implementation

Was ist aus technischer Sicht besonders cool an eurer Lösung?

Unsere Lösung besticht durch ihre Cross-Plattform-Verfügbarkeit, da sie sowohl auf iOS als auch auf Android funktioniert. Durch die Nutzung der

Flutter-Komponenten der SBB ist eine einfache Integration des Frontends in die bestehenden SBB-Apps möglich, wodurch konkrete Use Cases besonders realistisch innerhalb der aktuellen SBB-Apps nachgebildet werden können.

Durch die native Implementierung der App können wir zudem auf hardwareseitige Funktionalitäten wie Bluetooth und WLAN zugreifen. Dies ermöglicht eine präzise Ermittlung des Zuges und des Waggons des sehbehinderten Nutzers über Beacons, ohne dass dieser manuelle Eingaben vornehmen muss. Darüber hinaus wird in Zukunft eine direkte Verbindung zwischen Helfenden und sehbehinderten NutzerInnen bei schlechter Internetverbindung ermöglicht, beispielsweise in Tunneln oder Tiefbahnhöfen.

Abgrenzung / Offene Punkte

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

Aufgrund der begrenzten Zeit während des Hackathons haben wir uns entschieden, kein vollständiges Backend zu implementieren und stattdessen den Fokus auf die Entwicklung eines benutzerfreundlichen Frontends zu legen.