



«Blindä Passagier» « Indoor-Routing für Alle (SBB)»

Technische Informationen für die Jury



1	Aktueller Stand des Sourcecodes	3
2	Ausgangslage	4
2.1	Problemstellung und Lösungsansatz	4
2.2	Grundsatzentscheidungen	4
3	Technischer Aufbau	5
3.1	Verwendeter Techstack	5
1.1.1	Angular + NodeJS	5
1.1.2	Python (Prototyp)	5
1.1.3	Figma (Frontend)	5
4	Implementation	6
4.1	Einsatz von Künstlicher Intelligenz	6
4.2	Erzeugen von Textanweisung	6
5	Abgrenzung / Offene Punkte	8
5.1	Geolokalisierung und Neuberechnung	8
5.2	Unterstützungsanfragen	8
5.3	Vereinfachung der Routenplanung	8
5.4	Opt-In für Helferfunktion	8
5.5	SBB Inclusive App	8
5.6	Kartenintegration mit Angular-API	8

1 Aktueller Stand des Sourcecodes

Unser Projekt "Blindä Passagier" ist auf Einladung zugänglich und kann in unserem GitHub-Repository eingesehen werden. Interessierte finden den aktuellen Sourcecode hier:

<https://github.com/OnlyOneCookie/Blinde-Passagier>

Bei fehlendem Zugriff kann eine Anfrage an antoine@martinspacheco.ch gestellt werden.

2 Ausgangslage

2.1 Problemstellung und Lösungsansatz

Unsere Aufgabe bestand darin, eine barrierefreie Lösung für das Indoor-Routing in Bahnhöfen zu entwickeln. Nach eingehenden Diskussionen mit dem Challengegeber und internen Überlegungen haben wir uns auf folgende Kernpunkte konzentriert:

- Fokus auf kleinere Bahnhöfe: Grosse Bahnhöfe verfügen bereits über Infrastruktur zur Unterstützung blinder Menschen. Unser Ziel ist es, eine Lösung für kleinere Bahnhöfe (3-5 Gleise) zu entwickeln, die ohne kostspielige Infrastrukturanpassungen auskommt.
- Berücksichtigung der Nutzereinschränkungen: Blinde Reisende sind oft mit Blindenstock und Föhrhund unterwegs, was die Bedienung eines Smartphones erschwert. Zudem wollen wir vermeiden, dass der essenzielle Hörsinn durch Kopfhörer beeinträchtigt wird.
- Flexibilität und Sicherheit: Unsere Lösung soll sowohl selbstständiges Reisen als auch die Möglichkeit zur Unterstützung durch Mitreisende bieten.

2.2 Grundsatzentscheidungen

Basierend auf diesen Erkenntnissen haben wir folgende Grundsatzentscheidungen getroffen:

- Integration in die SBB-App: Wir entwickeln eine Lösung, die in die bestehende SBB-App integriert wird und im Rahmen der Routenplanung abrufbar ist.
- Vorab-Wegweisung: Blinde Reisende erhalten bereits vor der Ankunft an einem fremden Bahnhof eine textbasierte Wegbeschreibung vom Ankunfts- zum Zielgleis, die per VoiceOver vorgelesen werden kann.
- Unterstützungsoption: Als Alternative bieten wir die Möglichkeit, um Hilfe durch Mitreisende zu bitten, die die blinde Person vom Ankunfts- zum Abfahrtsgleis begleiten.
- Flexibilität für Nutzer: Die Entscheidung zwischen selbstständiger Navigation und Begleitung liegt beim blinden Passagier. Bei Bedarf kann zwischen den Optionen gewechselt werden.
- Dynamische Neuberechnung: Falls sich eine blinde Person im Bahnhof verirrt, kann mittels Geolokalisierung eine neue Wegweisung generiert werden.
- Durch diesen Ansatz schaffen wir eine flexible, kostengünstige Lösung, die ohne zusätzliche Infrastruktur auskommt und sowohl in kleinen als auch grösseren Bahnhöfen einsetzbar ist. Unser Ziel ist es, blinden Reisenden mehr Selbstständigkeit und Sicherheit beim Navigieren in Bahnhöfen zu ermöglichen.

3 Technischer Aufbau

3.1 Verwendeter Techstack

Unser technischer Aufbau basiert auf einer Kombination von robusten Technologien und innovativen Lösungsansätzen, um eine effiziente und skalierbare Anwendung zu gewährleisten. Hier ein Überblick über die verwendeten Komponenten.

3.1.1 Angular + NodeJS

Das Herzstück unserer aktuellen Anwendung ist eine Angular-basierte Lösung, die sowohl Frontend- als auch Backend-Funktionalitäten umfasst. Die gesamte Logik zur Umwandlung der Objekte der Journey-Maps-API der SBB in verständliche Textanweisungen ist hier implementiert. Diese Komponente ist entscheidend für die Bereitstellung der barrierefreien Navigationsinformationen und bietet gleichzeitig eine interaktive Benutzeroberfläche.

Sie dient als umfassendes Werkzeug für Entwickler und Stakeholder, um die Leistungsfähigkeit unseres Backends zu demonstrieren, zu testen und zu optimieren.

3.1.2 Python (Prototyp)

In der initialen Entwicklungsphase wurde ein Python-Backend für API-Tests verwendet. Obwohl die Anwendung mittlerweile vollständig auf Angular migriert wurde, enthält der Python-Code noch die ursprüngliche Logik. Es ist wichtig zu beachten, dass dieser Teil nicht die aktuellste Frontend-Version nach CI/CD-Richtlinien der SBB beinhaltet und primär als Referenz und für Vergleichszwecke dient.

Diese Struktur ermöglicht es uns, die Stärken von Angular für eine moderne, responsive Webanwendung zu nutzen, während wir gleichzeitig von den frühen Erkenntnissen aus unserer Python-Entwicklung profitieren. Die Entscheidung, die gesamte Logik in Angular zu implementieren, verbessert die Konsistenz und Wartbarkeit unserer Anwendung erheblich.

3.1.3 Figma (Frontend)

Um eine realistische Integration in die bestehende SBB-App zu visualisieren, haben wir uns für Figma als Design-Tool entschieden. Diese Entscheidung basiert auf der Erkenntnis, dass eine vollständige Nachbildung der SBB-App im Rahmen des Hackathons nicht realisierbar wäre. Unser Figma-Modell zeigt, wie unsere Lösung nahtlos in die produktive App integriert werden könnte. Die App repräsentiert dieses Modell, die kundenseitige Ansicht unserer Anwendung und ermöglicht eine anschauliche Präsentation des Konzepts.

4 Implementation

Bei der Implementierung unserer Lösung haben wir uns auf innovative Ansätze und effiziente Technologien konzentriert, um eine zuverlässige und zukunftsichere Anwendung zu entwickeln. Zwei Hauptaspekte stechen dabei besonders hervor.

4.1 Einsatz von Künstlicher Intelligenz

Nach sorgfältiger Abwägung haben wir uns gegen den Einsatz von Künstlicher Intelligenz zur Generierung von Textanweisungen entschieden. Diese Entscheidung basiert auf mehreren wichtigen Überlegungen:

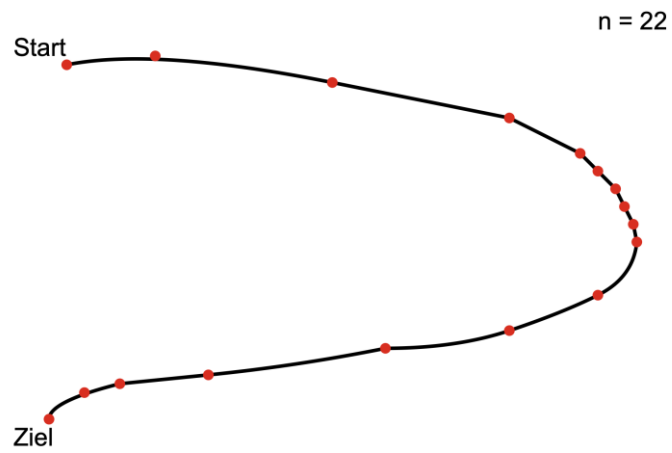
- **Nachvollziehbarkeit:** Ohne ein klares Verständnis des KI-Generierungsprozesses wäre es schwierig, die Korrektheit der Anweisungen zu überprüfen und zu garantieren.
- **Zuverlässigkeit:** In unserem spezifischen Anwendungsfall ist eine konsistente, wenn auch möglicherweise weniger elegante Lösung, einer potenziell unberechenbaren KI-Lösung vorzuziehen.
- **Langfristige Funktionalität:** Unser algorithmischer Ansatz kann ohne ständige Wartung deutlich länger funktionieren als eine KI-basierte Lösung, was besonders vorteilhaft für eine potenzielle spätere Integration in die SBB-App ist.
- **Ressourcenschonung:** Dieser Ansatz deckt sich sehr gut mit der strategischen Ausrichtung der SBB mit ihrem Motto: **“Ein Tick grüner”**, da AI deutlich mehr Strom konsumiert, als eine einfache Funktion.

4.2 Erzeugen von Textanweisung

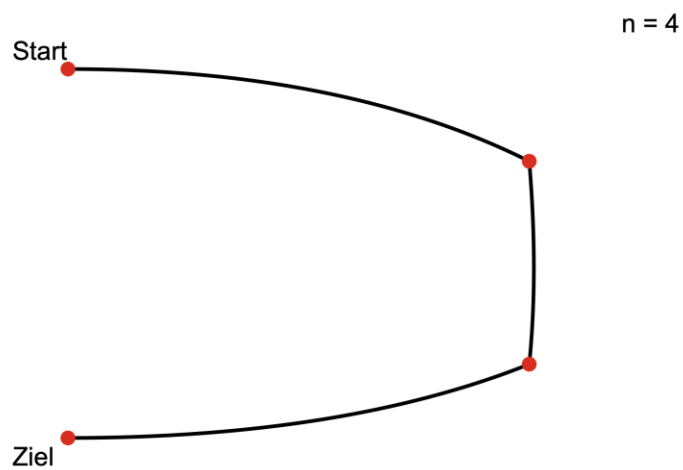
Die Entwicklung eines Algorithmus zur Generierung von Textanweisungen basierend auf einer Serie von geolokalisierter Punkten erwies sich als besonders lehrreich und faszinierend. Unser Prozess lässt sich wie folgt zusammenfassen:

- **Datengrundlage:** Wir erhalten von der SBB-API bei einer Routenabfrage eine Reihe von Koordinaten, dargestellt als rote Punkte auf einer Karte. Diese Punkte bilden die Grundlage für die Wegdarstellung in der SBB-App.
- **Datenreduktion:** Durch mathematische Berechnungen reduzieren wir diese Punktereihe auf die wesentlichen Informationen. Dieser Schritt ist entscheidend, um prägnante und verständliche Textanweisungen zu generieren.
- **Anweisungsgenerierung:** Basierend auf den reduzierten Daten erstellen wir klare und präzise Textanweisungen, die für blinde Nutzer leicht zu verstehen und zu befolgen sind.

Dieser Ansatz ermöglicht es uns, komplexe räumliche Informationen in einfache, aber effektive Navigationsanweisungen umzuwandeln. Die Kombination aus mathematischer Optimierung und algorithmischer Verarbeitung bildet das Fundament für eine robuste und benutzerfreundliche Lösung, die den speziellen Anforderungen blinder Reisender gerecht wird.



Durch mathematische Berechnungen reduzieren wir die Punktereihe auf das Wesentliche, sodass wir eine Textanweisung liefern können.



5 Abgrenzung / Offene Punkte

In der Entwicklung unserer Lösung haben wir einige bewusste Entscheidungen getroffen und bestimmte Aspekte ausgegrenzt, um den Fokus auf die Kernfunktionalität zu legen. Hier ein Überblick über diese Punkte und offene Themen.

5.1 Geolokalisierung und Neuberechnung

Wir setzen voraus, dass Geolokalisierung auf den Endgeräten der Nutzer verfügbar ist. Diese Funktion ist entscheidend, um Nutzern im Falle einer Verirrung im Bahnhof eine Neuberechnung der Route zu ermöglichen.

5.2 Unterstützungsanfragen

Die technische Umsetzung von Unterstützungsanfragen wurde nicht implementiert, da hierfür eine voll funktionsfähige Version der SBB-App erforderlich wäre. Dies bleibt ein offener Punkt für zukünftige Entwicklungen.

5.3 Vereinfachung der Routenplanung

Zur Reduzierung der Komplexität haben wir bewusst detaillierte Angaben zu Verbindungen, die normalerweise in einer Routenplanung verfügbar wären, ausgelassen. Stattdessen gehen wir von der vereinfachten Annahme aus, dass der Nutzer in der Gleismitte aus dem Zug austritt.

5.4 Opt-In für Helferkfunktion

Die Unterstützungsfunktion für Helfer würde in einer vollständigen Implementierung eine Aktivierung in den Einstellungen der SBB-App erfordern (Opt-In). Aus Präsentationsgründen haben wir diese Einstellungsansicht in unserem Prototyp nicht integriert.

5.5 SBB Inclusive App

Wir haben uns gegen die Verwendung der SBB Inclusive App entschieden, da sie keine Möglichkeit bietet, komplexe Reisen mit Umstiegen einzugeben. Unsere Lösung zielt darauf ab, auch solche Szenarien abzudecken.

5.6 Kartenintegration mit Angular-API

Ursprünglich war geplant, in unserer Angular-Anwendung neben der Wegbeschreibung auch eine Karte anzuzeigen. Diese Funktion wurde nicht umgesetzt, da wir den erforderlichen API-Schlüssel zu spät angefordert haben. Die Integration dieses Moduls hätte zur besseren Veranschaulichung und Überprüfung der Qualität unserer Textanweisungen beigetragen.

Diese Abgrenzungen und offenen Punkte bieten Potenzial für zukünftige Erweiterungen und Verbesserungen unserer Lösung. Sie zeigen auch die Komplexität der Aufgabe und die Notwendigkeit, in späteren Entwicklungsphasen eng mit der SBB und potenziellen Nutzern zusammenzuarbeiten, um eine vollständig integrierte und umfassende Lösung zu schaffen.