

Dokumentation

Accessible Maps – Visuelle Kartenkonzepte in Progressive Web Apps

von

Till Große, Tomasz Ludyga und Adrian Köhler

Datum

31. März 2022

Inhaltsverzeichnis

1 Einleitung	1
1.1 Motivation	1
2 Kontextanalyse	3
2.1 Verwandte Arbeiten	3
2.2 Kontextbeschreibung	3
2.3 Zielgruppe	4
2.3.1 Nutzer mit Sehbehinderung	4
2.3.2 Rollstuhlfahrer	6
2.4 Vergleich beliebter Karten-Websites	7
3 Konzept	9
3.1 Funktionale Anforderungen	9
3.1.1 Anforderungen aus PWA	9
3.1.2 Anforderungen aus Web-App	10
3.1.3 Zusätzliche Anforderungen	10
3.2 Qualitative Anforderungen	11
3.3 GUI-Entwurf	12
3.3.1 Kartenansicht und Suche	12
3.3.2 Einstellungen und Nutzerprofile	14
4 Umsetzung	17
4.1 Technische Umsetzung	17
4.1.1 Allgemeiner Aufbau der Anwendung	17
4.1.2 Struktur des Quellverzeichnisses	17
4.1.3 Konventionen im Aufbau der Komponenten und Services	18
4.1.4 Übersicht aller Skripte	19
4.1.5 Vorgegebene Bibliotheken und Plugins	21
4.1.6 Anpassungen für die Screen-Reader-Zugänglichkeit	23
4.1.7 Verwendete Screen Reader Attribute in HTML	23
4.2 Projektorganisation	23
5 Evaluation	25
5.0.1 Allgemeine Anforderungen	25
5.0.2 Zwei-Wege-Sprachkommunikation	25

5.0.3	Allgemeine Anforderungen	25
5.0.4	Videofähigkeiten	25
5.0.5	Textalternativen	25
5.0.6	Zeitbasierte Medien	25
5.0.7	Anpassbar	26
5.0.8	Unterscheidbar	26
5.0.9	Per Tastatur zugänglich	27
5.0.10	Ausreichend Zeit	27
5.0.11	Anfälle	27
5.0.12	Navigierbar	27
5.0.13	Eingabemodalitäten	27
5.0.14	Lesbar	27
5.0.15	Vorhersehbar	28
5.0.16	Hilfestellung bei der Eingabe	28
5.0.17	Kompatibel	28
5.0.18	Benutzerdefinierte Einstellungen	28
5.0.19	Autorenwerkzeuge	28
5.0.20	Dokumentation und Support	28
6	Diskussion und Ausblick	29
Anhang		31
A	Literaturverzeichnis	33
B	Abbildungsverzeichnis	34

1 Einleitung

Im Rahmen der Lehrveranstaltung *KP Mensch-Computer-Interaktion* an der TU Dresden wurde, aufbauend auf zwei bestehenden Projekten, eine barrierefreie Kartenapplikation entwickelt. Hierbei stand das Hervorheben von barrierefreien Zugängen für die Bedienung durch Menschen mit Einschränkungen im Vordergrund.

1.1 Motivation

Die Motivation für die Erarbeitung dieser Lösung bestand in erster Linie darin, Menschen mit Einschränkungen bei der Wahrnehmung einer Karte zu unterstützen. Dazu kann eine Anwendung für die an Barrierefreiheit orientierte Darstellung der Innenräumen in Gebäuden dienen. Die Unterstützung sollte einerseits durch die schnelle Bereitstellung von Informationen über verfügbare Barrierefreiheitsmerkmale innerhalb von Gebäuden geleistet werden. Ein Anwendungsbeispiel ist hier die Suche nach einer Toilette für Rollstuhlfahrer. Zum anderen soll die Nutzung für Menschen mit verschiedenen Arten von Einschränkungen wie Seh- oder Mobilitätseinschränkungen optimiert werden.

Komplexität der möglichen Einschränkungen der Benutzer und die Bedürfnisse dieser Zielgruppe bildet zusammen eine spannende technische Herausforderung. Unsere Motivation ist eine barrierefreie Nutzung sowohl durch visuell eingeschränkte Personen als auch durch Menschen mit anderen Einschränkungen sowie seine Pflegepersonen zu erstellen. Eine Benutzeroberfläche, die von der Mehrheit der Benutzer als gut und brauchbar beurteilt wird, berücksichtigt in den meisten Fällen nicht die Bedürfnisse und Anforderungen von sehbehinderten Menschen. Mit sehbehindert meinen wir im Rahmen dieser Arbeit die Menschen, die grundsätzlich sehen können, aber kleinere oder größere Defizite in Bezug auf Farbwahrnehmung, Sehbereich, Fokussierung oder andere Einschränkungen. Diese Zielgruppe ist relativ breit und benötigt individuell angepasste Möglichkeiten der Interaktion bezüglich verschiedenen Spezifikationen der Behinderung. Die Abweichung zwischen den allgemeinen graphische Designempfehlungen und den Bedürfnissen von Behinderten hat den Grund, dass die übliche Empfehlungen wie passende Farben, geringe Kontraste oder kleine und nicht ablenkende Textfelder, bei Menschen mit Schwierigkeiten in der visuellen Wahrnehmung das Gegenteil bewirkt. Sie können ähnliche kontrastarme Farben oft nicht unterscheiden oder

kleine und unscheinbare Zeichen nicht lesen. Deswegen, um seine Aufgaben realisieren zu können, wollen und müssen sehbehinderte Menschen in den meisten Fällen besondere Funktionalitäten wie z. B. Screen-Reader nutzen.

2 Kontextanalyse

In diesem Abschnitt der Arbeit wollen wir das Hintergrund der Anwendung beschreiben. Zu diesem Zweck werden die Zielgruppe, das Kontext der Kartenanwendung und unsere Beobachtungen analysiert.

2.1 Verwandte Arbeiten

Es wurden mehrere Arbeiten zur Thema der Klassifikation, Definition, Beschreibung und Analyse von Seh- und Bewegungsbehinderungen bereits publiziert. Eine breite Informationsquelle ist die Dokumentation "Forschungspraktikum Accessible Maps: Indoor Maps for Visually Impaired Users" von Qiqi Hui und Karl Kegel aus 2021. Die Arbeit behandelt das Thema der Sehbehinderung, kommt aus der Lehrveranstaltung *KP Mensch-Computer-Interaktion* an der TU Dresden und beschreibt die Klassifikation der Sehbehinderung im Kontext einer Kartenanwendung. Das Kapitel 2.3.1. hier beinhaltet die Informationen hauptsächlich aus dieser Quelle. Diese Informationen basieren hauptsächlich auf dem Forschung (KV Vaishali 2020). Die Softwarequalitäten für unsere Anwendung wurden an ISO 25010 orientiert.

2.2 Kontextbeschreibung

Unsere Kartenanwendung dient zur barrierefreien Darstellung von Innenräumen von Gebäuden. Barrierefreiheit bezeichnet eine Gestaltung der Umwelt, sodass sie auch von Menschen mit Beeinträchtigungen ohne zusätzliche Hilfen genutzt und wahrgenommen werden kann (Wikipedia). Diese Kartendarstellung soll also die verfügbare Barrierefreiheitsmerkmale innerhalb von Gebäuden adaptiv anzeigen. Die Adaptivität soll in diesem Kontext eine Anpassung der Darstellung zu den Präferenzen der Nutzer bedeuten. Die Zielgruppe beinhaltet Menschen mit verschiedenen Einschränkungsarten und Bedürfnissen, sowohl visuellen als auch anderen z. B. Rollstuhlfahrer.

2.3 Zielgruppe

Zu der Zielgruppe gehören Menschen mit Einschränkung und Assistenten von Menschen mit Einschränkungen.

2.3.1 Nutzer mit Sehbehinderung

Diese Gruppe von Benutzer besitzt unterschiedliche Menschen und Bedürfnisse. Es wird also eine Toleranz einer Breite von individuellen Präferenzen bezüglich graphischen Design benötigt.

Einschränkung der Fernsicht	leicht - Sehschärfe schlechter als 6/12 bis 6/18
	mäßig - Sehschärfe schlechter als 6/18 bis 6/60
	schwer - Sehschärfe schlechter als 6/60 bis 3/60
	Erblindung - Sehschärfe schlechter als 3/60
Einschränkung der Nahsicht	Sehschärfe in der Nähe schlechter als N6 oder M.08 bei 40 cm

Abbildung 2.1 Definition der Sehbehinderungen

2.3.1.1 Aktueller Stand und Trends

Nach Angaben der World Health Organization gibt es weltweit mindestens 2,2 Milliarden Menschen mit Sehbehinderungen (WHO). Die Sehbehinderung ist weltweit nicht gleichmäßig verteilt, wobei die Belastung Sehbehinderung in den am wenigsten entwickelten Regionen am größten ist. Außerdem ist die Sehbehinderung in den verschiedenen Altersgruppen ungleichmäßig verteilt, wobei der Schwerpunkt bei den über 50-Jährigen liegt, die über 50 Jahre alt sind. Auch die geschlechtsspezifische Verteilung der Sehbehinderung ist ungleichmäßig weltweit ungleich verteilt, wobei Frauen insgesamt einem höheren Risiko ausgesetzt sind als Männer. Laut einer globalen Statistiken über Sehbehinderungen sind 55 Prozent Frauen und 89 Prozent leben in Ländern mit niedrigem und Ländern mit mittlerem Einkommen. Laut der Statistik 2019 über Menschen mit schweren Behinderungen gibt es in Deutschland 349.036 sehbehinderte Menschen, davon sind 277.492 zusätzlich zu den Blinden sehbehindert. Der Anteil der sehbehinderten Patienten über 55 Jahren liegt bei über 85 Prozent, der Anteil der Patienten über 75 Jahren liegt bei über 56 Prozent. In Zukunft wird die Zahl der Patienten mit altersbedingten Augenkrankheiten altersbedingten Augenerkrankungen und Sehbehinderungen weiter zunehmen, da die Zahl der älteren Menschen aufgrund der alternden Bevölkerung und der steigenden Lebenserwartung weiter zunehmen.

2.3.1.2 Analyse der Ursachen

Es gibt viele Ursachen für Sehbehinderungen. Laut der Studie 2020 Visual Impairment & Blindness Global Data & results sind die vier wichtigsten Ursachen für Sehbehin-

derungen weltweit Refraktionsfehler, Katarakt, AMD und Glaukom. Die nachstehende Abbildung zeigt eine Simulation verschiedener häufiger Arten von Sehbehinderungen.

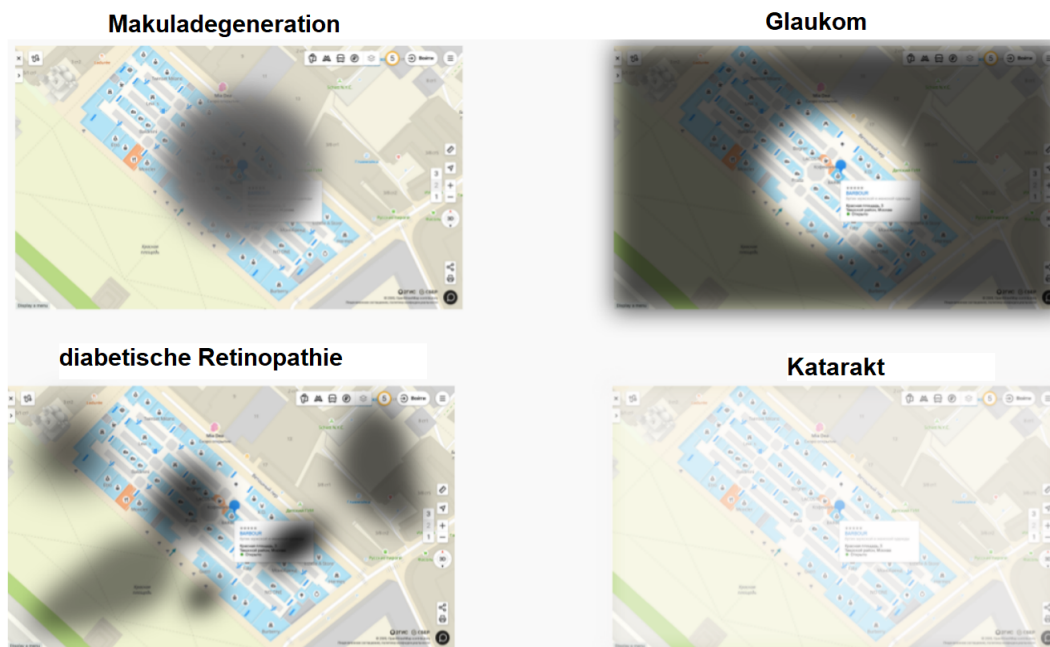


Abbildung 2.2 Simulation gängiger Arten von Sehbehinderungen

2.3.1.3 Häufige Arten von Sehbehinderungen

Albinismus ist ein allgemeiner Begriff für eine Gruppe genetischer Störungen, die durch Mutationen in Genen verursacht werden, die mit der Melaninsynthese zusammenhängen und zu einer verminderten oder fehlenden Melaninablagerung führen. Augenalbinismus kann zu Lichtscheu, Hypopigmentierung der Netzhaut und der Iris des Auges führen und Netzhautdysplasie, Nystagmus, Schielen und eine abnorme Entwicklung des Sehnervs verursachen, was zu verminderter Sehkraft oder sogar zum Verlust des Sehvermögens auf beiden Augen führt.

Die altersbedingte Makuladegeneration (AMD) ist eine altersbedingte Veränderung in der Struktur der Makula. Zu den möglichen klinischen Symptomen gehören leicht verschwommenes Sehen, ein verschwommener Bereich in der Nähe des Zentrum des Sehens, Farben, die nicht mehr so lebendig aussehen wie früher, und Schwierigkeiten beim Sehen bei schwachem Licht.

Ein Katarakt (auch bekannt als Graue Star) ist eine Trübung der Augenlinse, die das Sehvermögen beeinträchtigen kann. Die meisten Katarakte sind altersbedingt. Die Häufigkeit des Grauen Stars hängt mit dem Alter, der Region und der Ernährung zusammen. Die häufigsten Symptome des Katarakts sind trübes oder verschwommenes Sehen, Blendung oder schlechtes Nachtsehen.

Die diabetische Retinopathie ist eine der wichtigsten Komplikationen bei Diabetes. Sie wird häufig durch erhöhte Blutzuckerwerte verursacht. Mögliche klinische Anzeichen sind Flecken oder schwarze Linien, die in der Sicht schweben, verschwommenes Sehen, dunkle oder leere Bereiche in der Sicht.

Das Glaukom ist eine Gruppe von Krankheiten, die durch Atrophie und Depression der Sehnervenpapille Papille, Gesichtsfelddefekte und verminderte Sehschärfe. Pathologisch erhöhter Augeninnendruck Augeninnendruck und eine unzureichende Blutversorgung des Sehnervs sind die Hauptrisikofaktoren für die Entwicklung eines Glaukoms.

2.3.1.4 Farbenblindheit und Farbenschwäche

Farbenblindheit ist eine Störung, bei der einige oder alle Farben nicht richtig erkannt werden können. In Bezug auf das Verhältnis zwischen Männern und Frauen war die Inzidenzrate bei Männern höher als bei Frauen. Die Prävalenz der Farbenblindheit war bei Männern höher als bei Frauen, mit 8 % bei Männern und etwa 0,5 % bei Frauen. Bei der Mehrzahl der Farbenblindheit handelt es sich um partielle Farbenblindheit, und nur eine sehr geringe Anzahl von Farbenblindheit ist eine vollständige Farbenblindheit.

In unserer Netzhaut gibt es zwei Arten von Zellen, die Licht wahrnehmen. Das sind die Stäbchenzellen und die Zapfenzellen. Etwa 120 Millionen Stäbchenzellen sind für die Wahrnehmung von Licht und Dunkelheit zuständig, während etwa 6 bis 7 Millionen Zapfenzellen für die Wahrnehmung von Farbe verantwortlich sind. Es gibt drei Arten von Zapfen, den S-Zapfen, den M-Zapfen und den L-Wirbel. Sie sind empfindlich für kurze, mittlere und lange Wellen. Unser Gehirn verwendet den Input von diesen Zapfenzellen, um unsere Farbwahrnehmung zu bestimmen. Verschiedene Arten von Farbschwäche und Farbenblindheit können durch unterschiedlich starke Schädigung verschiedener Arten von Sehzapfenzellen entstehen.

2.3.2 Rollstuhlfahrer

Jeder Rollstuhltyp und jeder einzelne Benutzer hat möglicherweise unterschiedliche Bedürfnisse in Bezug auf den Zustand oder andere Kriterien. Rollstuhlfahrer erleben viele Situationen, die die Stabilität und die damit verbundene Leistung ihres Rollstuhls beeinflussen. Die Stabilität wird durch die Eigenschaften und Fähigkeiten des Benutzers, die Umgebungsmerkmale und -bedingungen sowie durch Rollstuhlmodifikationen und Zubehör beeinflusst. Rollstuhlfahrer benötigen wirksame Instrumente und Methoden zur quantitativen Bewertung und Vorhersage des Verhaltens des Systems Benutzer-Rollstuhl in einer Vielzahl von statischen und dynamischen Situationen. Solche Informationen sind sehr wichtig, um ein effizientes Management der damit verbundenen Risiken zu ermöglichen und die Stühle entsprechend anzupassen. (Moody 2012), (Neis 2014)

2.4 Vergleich beliebter Karten-Websites

Laut "Forschungspraktikum Accessible Maps: Indoor Maps for Visually Impaired Users" von Qiqi Hui und Karl Kegel aus 2021 der aktuelle Trend zeigt, dass Karten für Innenräume immer mehr mit Karten für Außenbereiche integriert werden. Auf der Abbildung unten sehen wir drei Websites, die relativ gut im Bereich Indoor-Mapping abschneiden, nämlich Google Maps, Baidu Maps, 2GIS und MazeMap.

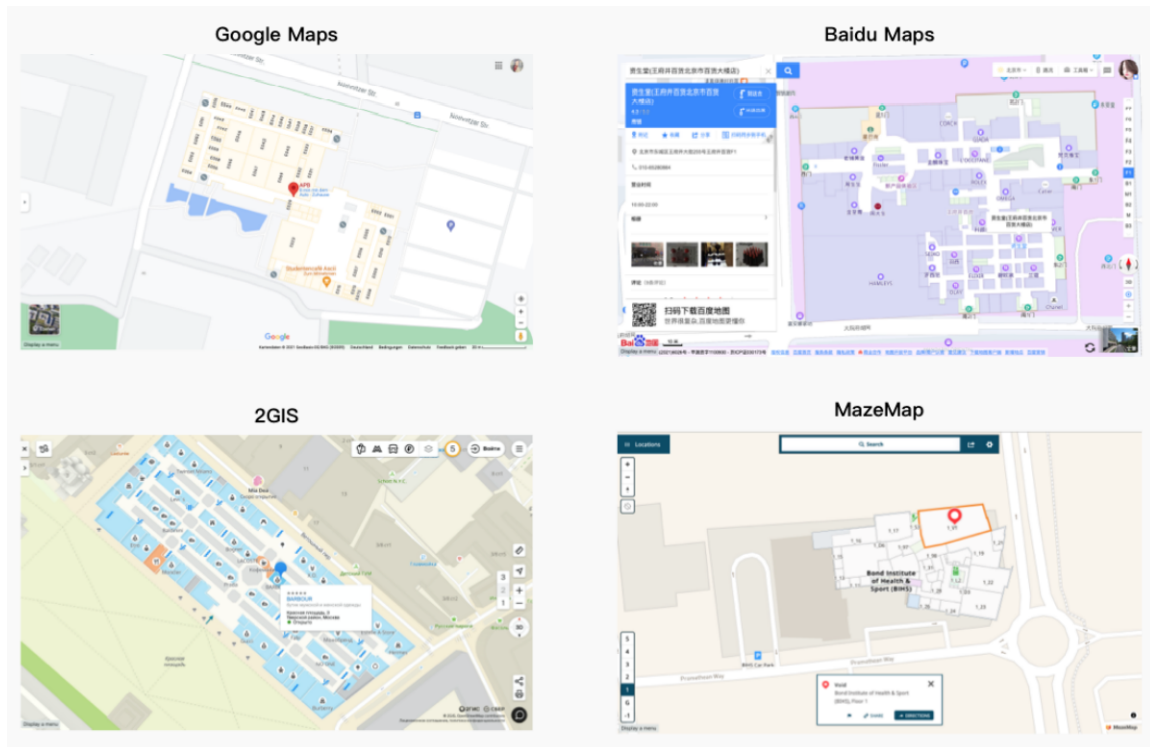


Abbildung 2.3 Populäre Karten-Websites

Im Vergleich zu Google Maps bietet Baidu Maps Baidu Maps bietet einen stärkeren Farbkontrast und die Möglichkeit, auf einen beliebigen Ort auf der Karte zu klicken, um Details anzuzeigen. Im Vergleich zu Baidu Maps hat 2GIS blaue Pfeile in die Kartenoberfläche eingefügt um den Benutzern den Eingang und die Richtung jedes Gebäudes anzuzeigen. Es gibt reichhaltigere Symbole in Innenraumkarte von 2GIS gibt es reichhaltigere Symbole sowie interaktives Feedback durch Farbkontrast Änderungen. Der letzte Anbieter ist MazeMap, der sich auf Innenraumkarten spezialisiert hat. MazeMap legt seinen ganzen Schwerpunkt auf die Optimierung von Innenraumkarten. Im Vergleich zu den ersten beiden Beispielen sind das UI-Design von MazeMap und Interaktionsdesign von MazeMap benutzerfreundlicher.

3 Konzept

Nachfolgend wird das zugrundeliegende Konzept der Arbeit vorgestellt. Dabei wird zuerst auf die Anforderungen bezüglich der benötigten Funktionen als auch auf die qualitativen Anforderungen der Kartenanwendung eingegangen. Anschließend soll der auf den vorher definierten Anforderungen basierende UI-Entwurf im Detail vorgestellt werden.

3.1 Funktionale Anforderungen

Bei den funktionalen Anforderungen wurden zuerst die Anforderungen aus den beiden bestehenden Projekten (Die PWA *'Accessible Maps'* und die Web-App *'Tactile Renderer'*) analysiert und erfasst. Ziel war es in erster Linie, die beiden bestehenden Projekte zusammenzufassen und bestehende Features in einer Anwendung zu vereinen. Anschließend wurden jedoch auch einige weitere, zusätzliche Anforderungen definiert, welche die Benutzerfreundlichkeit und Bedienbarkeit weiter verbessern sollen.

3.1.1 Anforderungen aus PWA

Der PWA *'Accessible Maps'* wurden neben einigen Grundfunktionalitäten einer Kartenanwendung auch die Möglichkeit der Auswahl von verschiedenen Nutzerprofilen entnommen. Die wichtigsten Anforderungen sind nachfolgend in Listenform zusammengefasst:

- Anzeige und Manipulation der Karte (Zoom, Verschieben)
- Suche von Gebäuden
- Indoor Ansicht von Gebäuden
- Markierung von Objekten in der Karte (Toiletten, Treppen, Aufzüge)
- Etagenwechsel
- Raum- und Objektauswahl
- Beschreibung Etagen, Räumen und Objekten

- Auswahl von verschiedenen Nutzerprofilen (Rollstuhl-Nutzer, Sehbehinderten- und 'Default'-Profil)
- Hervorheben verschiedener Features und Nutzung verschiedener Icons für Kartenobjekte je nach Nutzerprofil

3.1.2 Anforderungen aus Web-App

Zusätzlich zu den Funktionalitäten aus der PWA, umfasste die Web-App *'Tactile Renderer'* noch einige weitere Features, welche ebenfalls übernommen werden sollten und nachfolgend zusammengefasst werden:

- Detaillierte Feature- und Symbolauswahl
- Legende
- Indoorsuche
- Anpassung des Aussehens der Anwendung
 - Farbprofile
 - Farbstärke (Indoor)
 - Umgebungstransparenz
 - Kartenhintergrund (Hell-Dunkel)
 - Linienstärke von Umrissen

3.1.3 Zusätzliche Anforderungen

Zusätzlich zu den Funktionen aus den beiden bestehenden Repos wurden einige weitere Funktionsvorschläge vorgebracht, welche die Benutzerfreundlichkeit und Bedienbarkeit der finalen Anwendung weiter verbessern sollen.

- Zentrieren der Karte über einen Button (wenn der gesuchte Bereich verlassen wird und nicht mehr im Anzeigebereich liegt)
- Anzeige von Raumnummern direkt auf der Karte
- Zweisprachigkeit (Umschalten zwischen Deutsch und Englisch ermöglichen)
- Intelligente Suche (Vorschläge, Tippfehler ignorieren)
- Anzeige von Gehwegen in Gebäuden
- Profilabhängige Indoor Routen

3.2 Qualitative Anforderungen

Zusätzlich zu den funktionalen Anforderungen wurden noch einige qualitative Anforderungen erstellt, welche eine effektive, einfache und uneingeschränkte Bedienung der Anwendung garantieren sollen beziehungsweise eine spätere Weiterentwicklung des Projektes vereinfachen sollen.

- Adaptivität der Anwendung anhand von
 - Endgerät und Eingabemöglichkeiten (PC, Smartphone, Tablet)
 - Displaygröße (Bspw. Größe des Browserfensters)
- Zugänglichkeit der Anwendung auch mit Screenreadern
- Anpassungsmöglichkeiten durch Benutzerprofile und Feineinstellungen
- Eindeutigkeit und Lesbarkeit von Symbol-Markierungen
 - Kontrastierende Umrisse
 - Verwendung eindeutige Symbole
 - Vereinigung von mehreren nah gelegenen Symbolen zu einem (je nach Zoomstufe)
- Flüssige Bedienung und Minimierung von Verzögerungen
- Fehlerfreiheit und Zuverlässigkeit
- Einfache Weiterentwicklung und Wartbarkeit ermöglichen
 - Auslagerung *hardgecodeter* Texte
 - Refaktorisierung von bestehenden Code

3.3 GUI-Entwurf

Im Folgenden soll der auf den Anforderungen basierende GUI-Entwurf vorgestellt werden. Dabei wurde sich vom grundlegenden Aufbau an der PWA 'Accessible Maps' orientiert. Diese wurde jedoch um einige Funktionalitäten erweitert beziehungsweise wurden bestehende Funktionen verbessert.

3.3.1 Kartenansicht und Suche

In der Standard-Kartenansicht (ohne ausgewähltes Gebäude) kann der Nutzer durch zoomen oder verschieben des Kartenausschnitts frei in der Karte navigieren. Am oberen Rand findet er sowohl die Suche als auch die Einstellungen.

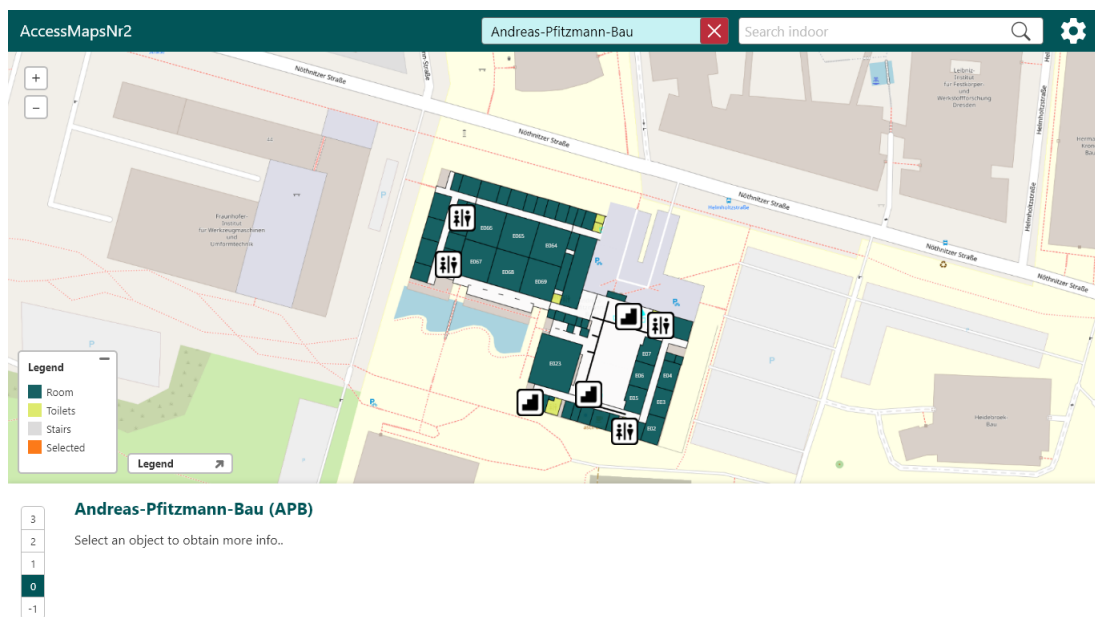
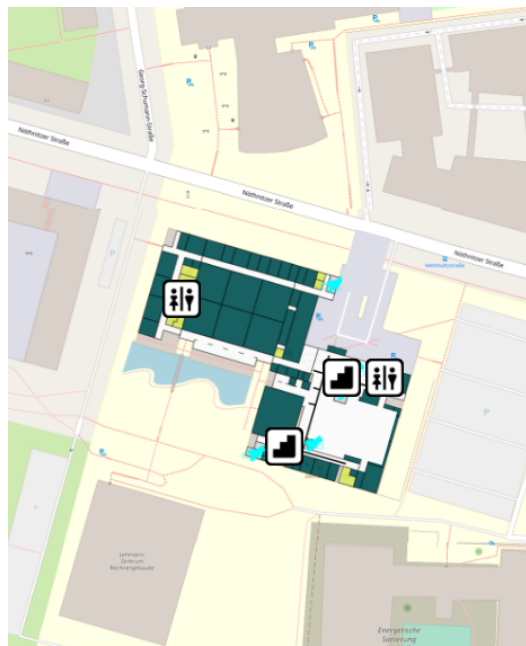


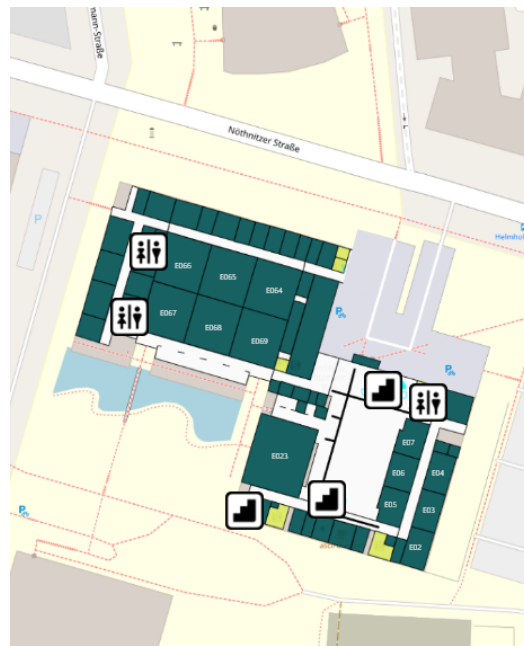
Abbildung 3.1 Ansicht eines Gebäudes

Sobald er auf ein Gebäude klickt oder er über die Suche zu einem gelangt wird ihm die Innenansicht des Gebäudes eingeblendet. Auf dieser sind Markierungen für ausgewählte Objekte (z.B.: Toiletten, Aufzüge oder Treppen) zu sehen. Ihm werden ebenfalls die einzelnen Raumnummern angezeigt. Um eine Überladung mit Informationen zu vermeiden und die Karte übersichtlich zu gestalten, wurde sich dazu entschieden, die einzelnen Markierungen je nach Zoomstufe zusammenzufassen und die Raumnummern erst ab einer angemessenen Zoomdistanz anzuzeigen. Die Raumnummern wurden im Laufe der Entwicklung des Weiteren mit einem weißen Hintergrund dargestellt, um einen ausreichenden Kontrast zwischen Schrift und Farbe des Raumes sicherzustellen (was insbesondere bei wechselnden Farbprofilen ein Problem darstellte).

Die verschiedenen Raumarten (Raum, Treppengang, Toiletten) werden mithilfe von Farben kenntlich gemacht, welche über eine Legende am linken, unteren Rand erklärt werden. Die Legende kann zudem eingeklappt werden um Nutzern, welche bereits mit der Farbgebung vertraut sind, eine übersichtlichere Karte zu bieten.



(a) Markierungen zusammengefasst



(b) Hereingezoomte Ansicht

Abbildung 3.2 Anzeige von Markierungen und Raumnummern bei verschiedener Zoomdistanz

Im unteren Bereich findet der Nutzer zudem das aktuell fokussierte Gebäude, die Möglichkeit zum Wechseln der angezeigten Etage und weitere Informationen zum angeklickten Objekt. Während dieser Bereich bei der PWA 'Accessible Maps' noch in 'angeklicktes Objekt' und 'Etageauswahl' unterteilt war, wurde er von uns zu einem Bereich zusammengefasst um die Anwendung übersichtlicher zu gestalten und deutlich zu machen, dass sowohl das angewählte Objekt als auch die Etageauswahl zum aktuell ausgewählten Gebäude gehören.

Im oberen Bereich ist die Suche zu finden, welche um die Funktionalität der Indoor-Suche von Räumen erweitert wurde. Der linke Suchbalken (*Ausgewähltes Gebäude*) wird dabei zuerst für das Suchen von Gebäuden genutzt und sobald ein Gebäude ausgewählt oder gefunden wurde wird der Name des Gebäudes dort eingetragen. In Abbildung 3.3 wurde das Gebäude *APB* bereits gefunden und es wird anschließend der zweite Suchbalken *Ausgewählter Raum* freigegeben, welcher eine Suche innerhalb des Gebäudes ermöglicht.



Abbildung 3.3 Suche mit bereits ausgewählten Gebäude

Eine weitere diskutierte Variante war es, die beiden Suchbalken zu einem zu verbinden (wie in Abbildung 3.4 dargestellt). Dies wurde jedoch aufgrund einer erschwerten barrierefreien Umsetzung verworfen. Ebenso wurde sich gegen eine Autovervollständigung der Suche entschieden, da die technische Umsetzung dieser den Umfang des Komplexpraktikums überschritten hätte.

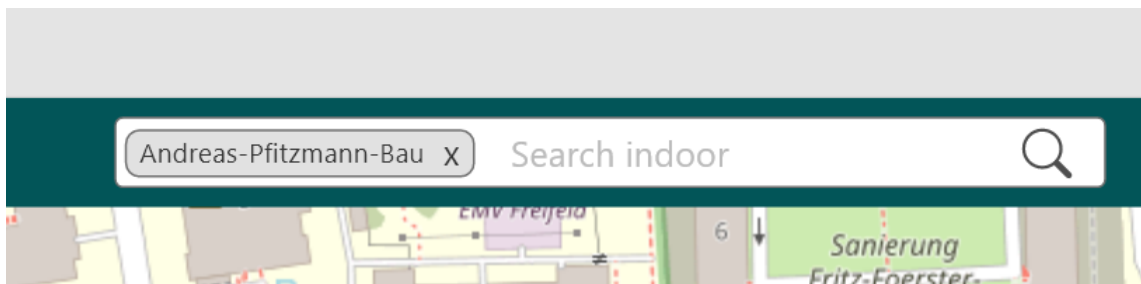


Abbildung 3.4 Kombiniertes Suchbalken

3.3.2 Einstellungen und Nutzerprofile

Bei den Einstellungen der Anwendung galt es die umfangreichen Detaileinstellungen der Web-App in die bestehende Profilauswahl der PWA zu integrieren. Um dies zu erreichen wurde sich dafür entschieden die Profilauswahl als eine Art schnelle Vor-konfiguration ("*profil quickswitch*") zu implementieren (Abbildung 3.5), welche, je nach ausgewählten Profil, passende Voreinstellungen trifft. So bekommen beispielsweise Rollstuhlfahrer bei Auswahl des entsprechenden Profils Rampen angezeigt, während Nutzer mit visuellen Einschränkungen taktile Linien eingeblendet bekommen (Abbildung 3.6a). Nach Auswahl eines Profils kann der Nutzer die Darstellung der Karte noch weiter für sich personalisieren (Abbildung 3.6b). So können etwa verschiedene (nicht benötigte) Features ausgeblendet werden oder die Kartendarstellung (Farben, Linienstärke, usw.) weiter an die individuellen Bedürfnisse angepasst werden.

Der letzte Punkt *Sprache* ermöglicht es, die Sprache der Anwendung auf Deutsch oder Englisch umzustellen.

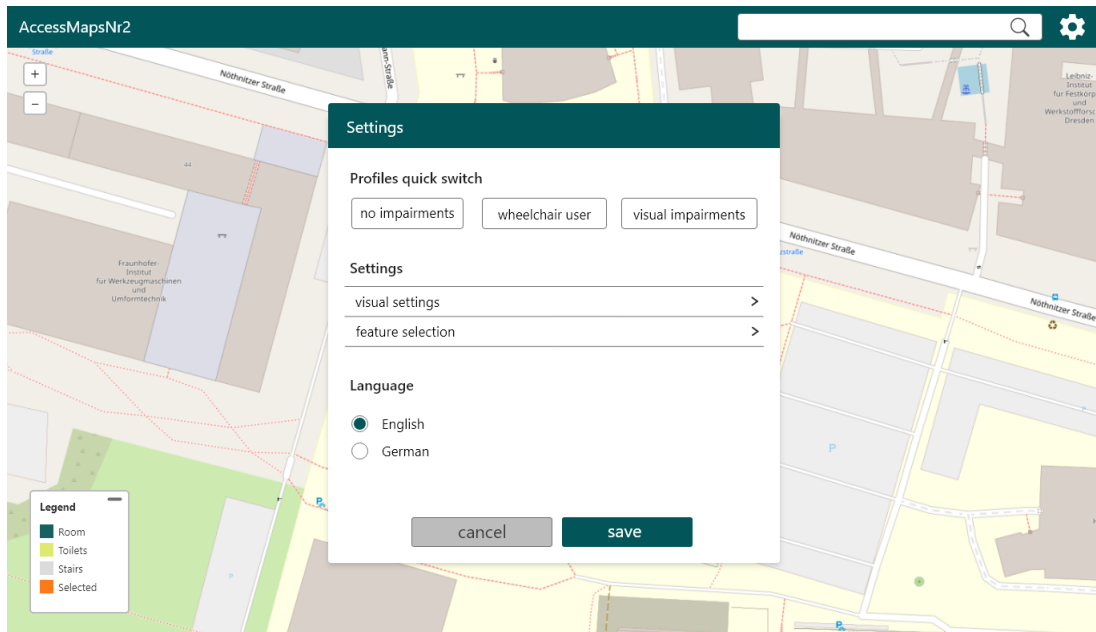
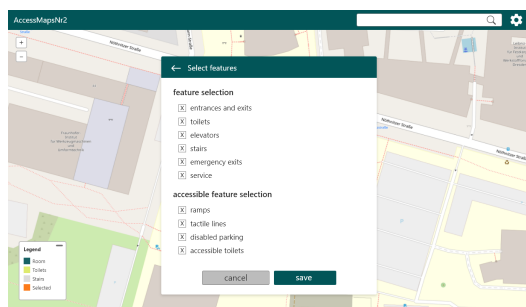
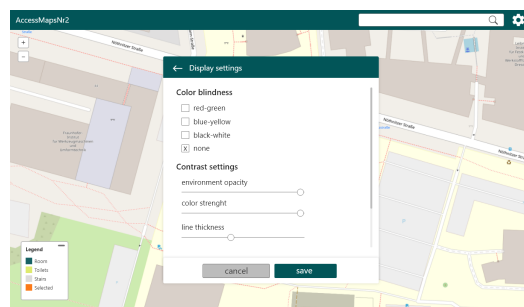


Abbildung 3.5 Kombiniertes Suchbalken



(a) Markierungen zusammengefasst



(b) Hereingezoomte Ansicht

Abbildung 3.6 Anzeige von Markierungen und Raumnummern bei verschiedener Zoomdistanz

4 Umsetzung

4.1 Technische Umsetzung

Im Folgenden wird die technische Umsetzung vom Allgemeinen bis hin zum Detaillierten vorgestellt.

4.1.1 Allgemeiner Aufbau der Anwendung

Der Code der Anwendung lässt sich in *Frontend* und *Backend* unterteilen. Zum Backend zählen die JavaScript-Dateien im Verzeichnis *Server* und die *Index.js*-Datei im Root-Verzeichnis. Das Backend basiert auf Node.js mit dem serverseitigen Anwendungsframework *express*.

Das Frontend ist in den Verzeichnissen *src* (*Source*) und *scss* (*Sass*) zusammengefasst. *scss* enthält Style-Dateien in der Sprache SASS mit SCSS-Syntax. Das Verzeichnis *src* enthält TypeScript-Dateien für die Komponenten des Frontends und deren Anwendungslogik.

In der Datei *package.json* sind die Module der Anwendung registriert, sowie Skripte zum starten des lokalen Servers. Von Bedeutung ist hierbei das Skript *build-start*, welches zwei Skripte zusammenfasst. Das ausführen dieses Skripts sorgt dafür, dass im ersten Teil sowohl die SASS-, als auch die TypeScript-Dateien in CSS bzw. JavaScript übersetzt (transpiliert) werden. Die Ausgabe der Übersetzung befindet sich anschließend im Verzeichnis *Public*.

Im zweiten Teil des Skripts wird der Start-Befehl des Node.js-Servers ausgeführt und eine *express*-Anwendung mit den Dateien im Verzeichnis *public* gestartet. Hierbei ist zu beachten, dass sich statische HTML-Elemente in der Datei *index.html* befinden.

4.1.2 Struktur des Quellverzeichnisses

Im Folgenden wird ein kurzer Überblick über das *src*-Verzeichnis gegeben.

Components	Die Elternkomponente der Anwendung bildet <i>geoMap</i> . Für die Verwaltung der Logik für die verschiedenen Ebenen innerhalb von Gebäuden wird <i>indoorLayer</i> verwendet. Diese beiden Komponenten sind allerdings nur teilweise bis nicht in der Nutzeroberfläche sichtbar. Alle weiteren sichtbaren Komponenten befinden sich im Unterverzeichnis <i>ui</i> .
Data	In diesem Verzeichnis befinden Dateien, die textliche Inhalte der Komponenten in Form von z.B. Strings speichern.
Models	Modelle, wie z.B. Schnittstellen oder Datenobjekte der Anwendungen werden hier gesammelt.
Services	Skripte, die sich mit dem Austausch oder der Verarbeitung von Daten beschäftigen liegen in diesem Verzeichnis. Diese sog. <i>Services</i> liefern Information an die Komponenten der Nutzeroberfläche im Verzeichnis <i>ui</i> .
Utils	In diesem Verzeichnis werden Skripte gespeichert, die als Werkzeuge von Services oder Komponenten dienen.
Startdateien	Die Dateien <i>main</i> , <i>style</i> und <i>ui</i> dienen dem Start der Elternkomponente, aller Kinder und den Services.

4.1.3 Konventionen im Aufbau der Komponenten und Services

Wrapper-Komponenten: Die Elternkomponente *geoMap* wird selbst als *Wrapper* instanziiert und enthält ihrerseits das eigentliche Leaflet-Object und den *indoorLayer* des Gebäudes. *geoMap* verwaltet alle Informationen rund um den Zustand der Karte, deren Verhalten und alle inneren Objekte. Selbes gilt für die Komponente *indoorLayer*.

Rendering: Falls möglich werden die Komponenten dynamisch mit den vorgefertigten Strukturen in *index.html* im DOM kombiniert.

Datenverwaltung: Nach dem Vorbild von *React* sind die meisten Komponenten durch *geoMap* kontrolliert, es gibt aber auch einige Komponenten, die einen eigenen Zustand speichern. Die Verarbeitung der Daten wird von den Services übernommen und findet nicht in den Komponenten statt. Die Anwendung speichert wichtige Daten wie z.B. das aktuelle Profil, Gebäude oder die Etage im lokalen Speicher des Webbrowsers. Wenn Komponenten Zustände speichern dürfen nur sie selbst diese verändern. Wenn mehrere Komponenten auf den gleichen Zustandsobjekt zugreifen müssen, muss dieses von deren gemeinsamer Elternkomponente verwaltet werden. Die gleichen Zustände dürfen nicht doppelt gespeichert werden.

Exports/Imports: Alle Komponenten außer *geoMap* und *indoorLayer* sind als einfache Skripte (ohne Klassen) implementiert und verwenden einen Default-Export für Funktionen, die von außerhalb genutzt werden.

4.1.4 Übersicht aller Skripte

Im folgenden werden die Skript-Dateien der Unterordner in *src* kurz beschrieben:

Tabelle 4.1 /components

geoMap	Elternkomponente, verantwortlich für den globalen Zustand der Anwendung und fungiert als Wrapper für die Leaflet-Karten-Instanz und den Indoor Layer
indoorLayer	repräsentiert den selbstkreierten Layer der für Innenräume gerendert wird. Verwaltet werden Aktionen auf dem Layer, Zustände des Layers, Hinzufügen von Markern etc. Die Komponente selbst ist ein Wrapper für einen nativen Layer von Leaflet

Tabelle 4.2 /components/ui

centeringButton	rendert den Center-Button mit Event-Handler
descriptionArea	HTML der Komponente ist bereits in index.html enthalten, hier wird nur die Update-Funktion definiert
legend	rendert die Legende basierend auf Input-Daten für Farbe und Text
levelControl	rendert eine Liste an Buttons für den Wechsel der Etagen mit Event-Handlern
loadingIndicator	Ladekreis zum Anzeigen der Beschäftigung während des Ladens, mitverantwortlich für das Anzeigen von Fehlermeldungen
toast	rendert den Toast-Container für das Anzeigen von Hinweisen und Fehlermeldungen

Tabelle 4.3 /components/userProfileModal

userProfileModal	Hauptsript des Einstellungsmodals, das die beiden anderen Skripte verknüpft, rendert Hauptansicht des Modals basierend auf Daten-Input, Hauptmanipulation des Anwendungszustandes
userFeatureSelectionModal	Submodal, welches Auswahl an Features rendert, basieren auf dem ausgewählten Profil (via Daten-Input), Kommunikation mit Feature Service
userVisualSettingsmodal	rendert Farbprofile und Slider für visuelle Einstellungen (via Daten-Input), Kommunikation mit Color Service

Tabelle 4.4 /data

buildingAccessibilityProperties	Liste aller verfügbaren Eigenschaften, die ein Gebäude haben kann, notwendig für das Erkennen und Filtern der OSM-Daten
featureAccessibilityProperties	Liste aller verfügbaren Features, notwendig für das Erkennen und Filtern der OSM-Daten
levelAccessibilityProperties	Liste aller verfügbaren Eigenschaften einer Etage, notwendig für das Erkennen und Filtern der OSM-Daten
languageSettings	Liste aller verfügbaren Sprachoptionen
userFeatureSelection	Initiale Liste an Features, die der Nutzer auswählen kann, Features sind Profile zugeordnet in denen sie angezeigt werden
userGroups	Liste aller verfügbaren Nutzergruppen mit Icons
userSettings	Zuordnung der Einstellungsarten zum jeweiligen Modal mit Sprache und Icon

Tabelle 4.5 /models

accessibilityPropertiesInterface	Interface für Accessibility Properties
buildingInterface	Interface für ein Gebäude
languageSettingsEnum	Aufzählung der Sprachen
userFeatureEnum	Aufzählung der Features
userGroupEnum	Aufzählung der Nutzergruppen
userSettingsEnum	Aufzählung der Einstellungsgruppen, Entsprechen jeweils einem Modal

Tabelle 4.6 /services

accessibilityService	Service für das Bekommen von Accessibility-Informationen, Speichert Eigenschaften pro Etage
buildingService	Service für das Bekommen von Gebäudeinformationen, Implementiert Gebäudesuche, Indoor-Suche, Weitergabe von GeoJSON
colorService	Service für das Bekommen und Speichern von Farbinformationen
featureService	Service für das Bekommen von Feature-Informationen, Verwaltet Features und wandelt diese in z.B. Marker um
httpService	Service für das Fetchen der OSM-Daten und das Aufteilen in verschiedene Gebäude- und Indoor-Daten
languageService	Service für das Bekommen und Speichern von Sprachinformationen
levelService	Service für das Bekommen von Etageninformationen (Daten, Namen, Beschreibung)
userService	Service für das Bekommen und Speichern des Nutzerprofils

Tabelle 4.7 /utils

extractLevels	extrahiert Levels aus einem Input-String
featureDescriptionHelper	extrahiert Beschreibung aus einem GeoJSON-Informationen
getArrayDepth	errechnet Array-Tiefe
hasCurrentLevel	überprüft ob aktuelle Etage im Feature enthalten ist
highlightSelectedPath	ändert CSS Fill-Attribut eines Pfades
makeAccessible	Sammlung von Methoden, die das UI mit Aria-Attributen für den Screen-Reader vorbereitet
translate	Übersetzt statische Elemente des UI entsprechend der aktuellen Sprache

4.1.5 Vorgegebene Bibliotheken und Plugins

Folgende Erweiterungen wurden genutzt:

- Bootstrap
- Leaflet
- GeoJSON

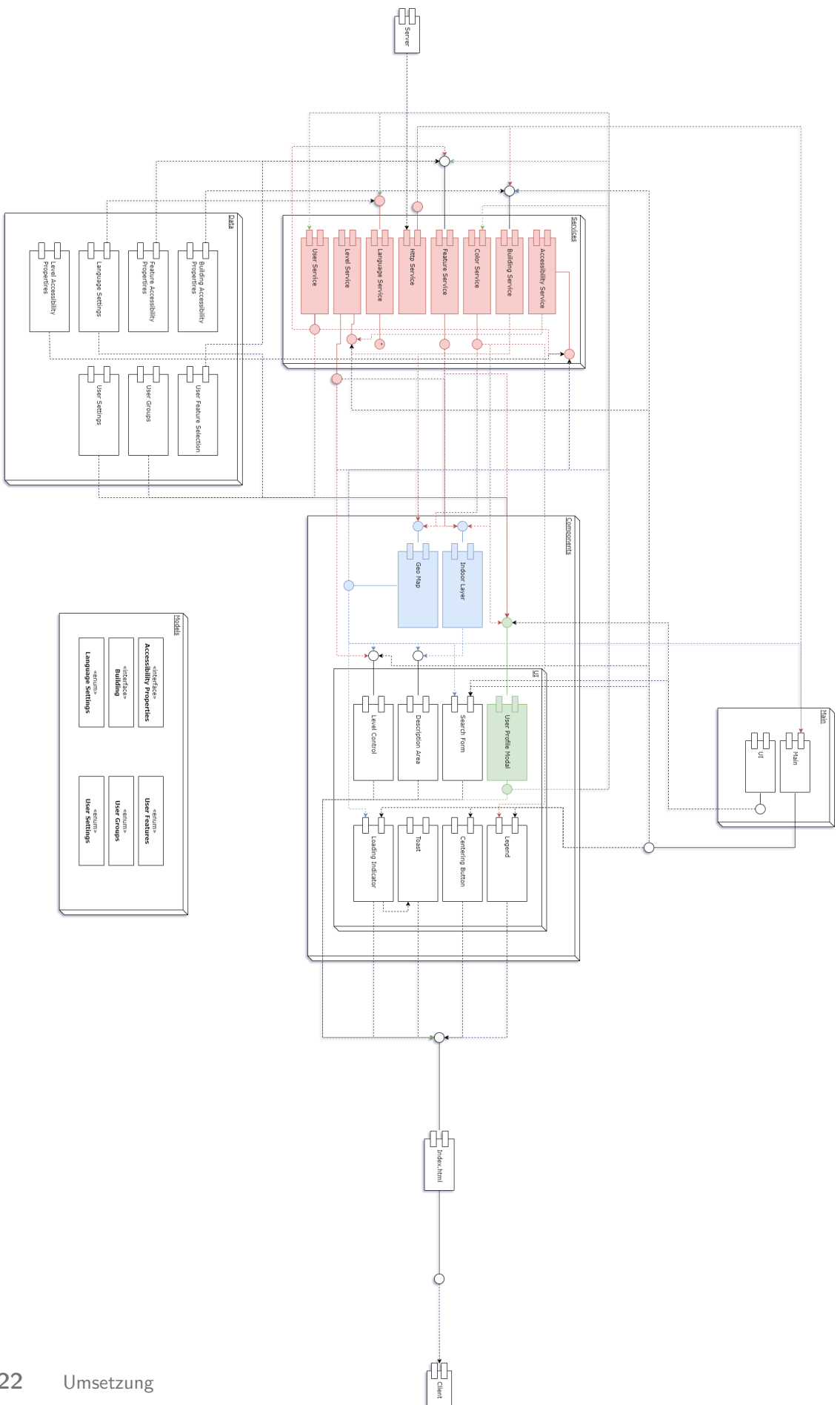


Abbildung 4.1 Komponentendiagramm der Anwendung

- Babel
- Node
- Leaflet.markercluster (Plugin)

4.1.6 Anpassungen für die Screen-Reader-Zugänglichkeit

Die Tabreihenfolge des Screen-Readers durchläuft zuerst die Navigationsleite, dann die Kontrollbuttons der Karte (Zoom in / Zoom out und Center) anschließend die Etagenkontrolle und das Beschreibungsfeld am unteren Bildschirmrand. Sobald die Anwendung aktualisiert wird (z.B. als Reaktion auf Nutzereingaben) springt der Fokus auf das Beschreibungsfeld, um Nutzern mit Screen-Reader neue Informationen zu präsentieren. Elemente, die sich in der Karte befinden (Räume, Marker, Gebäude etc.) sind dem Screen-Reader nicht zugänglich.

4.1.7 Verwendete Screen Reader Attribute in HTML

Im Folgenden werden die verwendeten HTML-Attribute aufgelistet und deren Funktion kurz beschrieben:

- Fokussierbarkeit mit *tabindex = 0* oder *href="#"*
- keine Fokussierbarkeit mit *tabindex = -1*
- Zusatzinformationen durch *aria-label*
- Zugriff entfernen mit *aria-hidden*
- Verweisen auf beschreibende Labels mit *aria-labelledby*
- Geklappten Zustand markieren mit *aria-expanded*
- Deaktivierten Zustand markieren mit *aria-disabled*
- Auf verwaltende Komponente verweisen mit *aria-controls*
- Funktionale Rollenverteilung mit: *role = status | complementary | search | button | presentation*

4.2 Projektorganisation

Folgende Standards für Softwareentwicklung in Teams wurden eingehalten:

- Gitlab
- Git
- Feature Branches
- Issues

- Milestones
- Wiki
- Kategorien für Issues
- Merge Requests

Nach dem Aufteilen der Aufgaben wurden die Features wie folgt von den Entwicklern übernommen:

- Anforderungserfassung (Tomasz)
- GUI-Entwurf (Adrian)
- Refaktorisierung des Codes (Till)
- Indoor Suche implementieren (Till)
- Legende für Farben (Tomasz)
- Modal für Feature-Auswahl (Adrian)
- Karte zentrieren (Tomasz, Till)
- Kartenmanipulation: Features, Farbprofile, etc. (Till)
- Zweisprachigkeit (Adrian)
- Zoomdistanz für Symbole (Adrian)
- Screen Reader Kompatibilität (Alle)
- Responsivität (Alle)

5 Evaluation

Gegen Ende der Umsetzung des Projektes wurde eine Evaluation auf Grundlage des BITV-Tests durchgeführt, um möglicherweise verbleibende Probleme zu ermitteln. Die Ergebnisse dieser Evaluation sind im Folgenden im Detail aufgeführt. Als Screen-Reader wurde NVDA für Windows benutzt, das Plugin *focusHighlight* hat sich als sehr hilfreich erwiesen um den Fokus visuell darzustellen.

5.0.1 Allgemeine Anforderungen

- Aktivierung von Barrierefreiheitsfunktionen: erfüllt
- Biometrie (entfällt)
- Erhaltung von Barrierefreiheitsinformationen bei Konvertierung (entfällt)

5.0.2 Zwei-Wege-Sprachkommunikation

5.0.3 Allgemeine Anforderungen

- entfällt

5.0.4 Videofähigkeiten

- entfällt

5.0.5 Textalternativen

- Alternativtexte für Bedienelemente: erfüllt
- Alternativtexte für Grafiken und Objekte: erfüllt
- Leere alt-Attribute für Layoutgrafiken: erfüllt
- Alternativen für CAPTCHAs (entfällt)

5.0.6 Zeitbasierte Medien

- entfällt

5.0.7 Anpassbar

- HTML-Strukturelemente für Überschriften: teilweise erfüllt (keine H1, in Einstellungen folgt dann gleich H5)
- HTML-Strukturelemente für Listen: teilweise erfüllt (Einstellungen (bspw. anzuzeigende Features) sind keine Listen)
- Inhalt gegliedert: erfüllt
- HTML-Strukturelemente für Zitate (entfällt)
- Datentabellen richtig aufgebaut: erfüllt
- Zuordnung von Tabellenzellen (entfällt, Legende nicht komplex genug)
- Kein Strukturmarkup für Layouttabellen: erfüllt
- Beschriftung von Formularelementen programmatisch ermittelbar: erfüllt
- Sinnvolle Reihenfolge: erfüllt
- Ohne Bezug auf sensorische Merkmale nutzbar: eher erfüllt (Legende bietet lediglich Farben zur Unterscheidung an, ist jedoch auch in Schwarz-Weiß bedienbar & Karte bietet zusätzlich Icons an)
- Keine Beschränkung der Bildschirmausrichtung: teilweise erfüllt: bei Smartphoneansicht (quer) ist Karte schlecht sichtbar
- Eingabefelder zu Nutzerdaten vermitteln den Zweck (entfällt)

5.0.8 Unterscheidbar

- Ohne Farben nutzbar: erfüllt
- Ton abschaltbar (entfällt)
- Kontraste von Texten ausreichend: erfüllt
- Text auf 200 % vergrößerbar: erfüllt
- Verzicht auf Schriftgrafiken: erfüllt
- Inhalte brechen um: erfüllt
- Kontraste von Grafiken und grafischen Bedienelementen ausreichend: teilweise erfüllt (Fokushervorhebung schlecht erkennbar)
- Textabstände anpassbar: erfüllt
- Eingblendete Inhalte bedienbar: erfüllt

5.0.9 Per Tastatur zugänglich

- Ohne Maus nutzbar: erfüllt
- Keine Tastaturfalle: erfüllt
- Tastatur-Kurzbefehle abschaltbar oder anpassbar (entfällt)

5.0.10 Ausreichend Zeit

- Zeitbegrenzungen anpassbar: nicht erfüllt (Error-Feld z.B. nach fehlgeschlagener Suche verschwindet nach einiger Zeit von selbst)
- Bewegte Inhalte abschaltbar: erfüllt (Fokus auf Gebäude <5s)

5.0.11 Anfälle

- Verzicht auf Flackern: erfüllt

5.0.12 Navigierbar

- Bereiche überspringbar: erfüllt
- Sinnvolle Dokumenttitel: erfüllt
- Schlüssige Reihenfolge bei der Tastaturbedienung: erfüllt
- Aussagekräftige Linktexte: erfüllt
- Alternative Zugangswege (entfällt)
- Aussagekräftige Überschriften und Beschriftungen: erfüllt
- Aktuelle Position des Fokus deutlich: nicht erfüllt (zu undeutlich)

5.0.13 Eingabemodalitäten

- Alternativen für komplexe Zeiger-Gesten: erfüllt
- Zeigergesten-Eingaben können abgebrochen oder widerrufen werden: erfüllt
- Sichtbare Beschriftung Teil des zugänglichen Namens: erfüllt
- Alternativen für Bewegungsaktivierung (entfällt)

5.0.14 Lesbar

- Hauptsprache angegeben: erfüllt
- Anderssprachige Wörter und Abschnitte ausgezeichnet: erfüllt

5.0.15 Vorhersehbar

- Keine unerwartete Kontextänderung bei Fokus: erfüllt
- Keine unerwartete Kontextänderung bei Eingabe: erfüllt
- Konsistente Navigation: erfüllt
- Konsistente Bezeichnung: erfüllt

5.0.16 Hilfestellung bei der Eingabe

- Fehlererkennung: eher nicht erfüllt (für ausgewähltes Gebäude gibt es keine; für ausgewählten Raum schon, aber weit weg)
- Beschriftungen von Formularelementen vorhanden: erfüllt
- Hilfe bei Fehlern: eher erfüllt ("Not STI conform" potenziell verwirrend)
- Fehlervermeidung wird unterstützt (entfällt)

5.0.17 Kompatibel

- Korrekte Syntax: erfüllt
- Name, Rolle, Wert verfügbar: erfüllt
- Statusmeldungen programmatisch verfügbar: erfüllt

5.0.18 Benutzerdefinierte Einstellungen

- Benutzerdefinierte Einstellungen: erfüllt

5.0.19 Autorenwerkzeuge

- entfällt

5.0.20 Dokumentation und Support

- entfällt, da Dokumentation ausschließlich außerhalb des Webangebots verfügbar

6 Diskussion und Ausblick

Für die Weiterentwicklung der Anwendung könnten noch weitere Informationen zur Barrierefreiheit in die Anwendung eingepflegt werden bzw. mit anderen Nutzern geteilt werden. Bezüglich der Barrierefreiheit ist es notwendig, dass bis zum finalen Release weitere Tests und Anpassungen durchgeführt werden. Zusätzliche Features könnten zum einen das Anzeigen der eigenen Position im Gebäude sein und auch die Navigation im Gebäude bzw. außerhalb. Diese muss natürlich in enger Abstimmung mit dem Nutzerprofil durchgeführt werden. Für die bessere Skalierbarkeit der Anwendung, sowie Performance empfiehlt sich ein Umstieg auf die React-Bibliothek. An vielen Stellen ist die Anwendung darauf bereits vorbereitet und verwendet ähnliche Designansätze. Gerade das Aktualisieren der gesamten Anwendung ist mit React leichter zu bewerkstelligen, als mit Neuladen der Seite oder gezielten UI-Updates mit Vanilla JavaScript.

Anhang

A Literaturverzeichnis

- [KV Vaishali 2020] KV Vaishali, P V.: Understanding definitions of visual impairment and functional vision. (2020)
- [Moody 2012] Moody, Heelis Chichi F.: Improving wheelchair prescription: an analysis of user needs and existing tools. (2012)
- [Neis 2014] Neis, Pascal: Measuring the Reliability of Wheelchair User Route Planning based on Volunteered Geographic Information. (2014)
- [WHO] WHO: *Blindness and vision impairment*. – URL: <https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment>. – Abgerufen 29 März 2022
- [Wikipedia] Wikipedia: *Barrierefreiheit*. – URL: <https://de.wikipedia.org/wiki/Barrierefreiheit>. – Abgerufen 29 März 2022

B Abbildungsverzeichnis

2.1	Definition der Sehbehinderungen	4
2.2	Simulation gängiger Arten von Sehbehinderungen	5
2.3	Populäre Karten-Websites	7
3.1	Ansicht eines Gebäudes	12
3.2	Anzeige von Markierungen und Raumnummern bei verschiedener Zoom- distanz	13
3.3	Suche mit bereits ausgewählten Gebäude	14
3.4	Kombinierter Suchbalken	14
3.5	Kombinierter Suchbalken	15
3.6	Anzeige von Markierungen und Raumnummern bei verschiedener Zoom- distanz	15
4.1	Komponentendiagramm der Anwendung	22