

# Entwicklung einer Webanwendung unter Verwendung des Angular-Frameworks, welche die Verwaltung von Schichtwechseln für Mitarbeiter unterstützt

Studienarbeit T3\_3101

des Studienganges Angewandte Informatik an der  
Dualen Hochschule Baden-Württemberg Mosbach



von

Nora Leuchner

Matrikelnummer, Kurs 4753795, INF21B

Gutachter der Dualen Hochschule Philipp Abele

23. Juni 2024

### Eigenständigkeitserklärung

*Ich versichere hiermit, dass ich meine Studienarbeit mit dem Thema:  
„Entwicklung einer Webanwendung unter Verwendung des Angular-Frameworks, welche  
die Verwaltung von Schichtwechseln für Mitarbeiter unterstützt“  
selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel be-  
nutzt habe.  
Ich versichere zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fas-  
sung übereinstimmt.*

*Pittsiedt, 23.06.2024*

Ort, Datum

*Nora Lechner*

Unterschrift

## Kurzfassung

Diese Studienarbeit behandelt die Entwicklung einer Anwendung zum Verwalten von Schichtwechseln unter der Verwendung des Google Design Sprints. Zunächst wurde die UX entwickelt und anschließend darauf basierend ein Benutzerkonzept, um die Nutzerfreundlichkeit und Effektivität der Anwendung zu gewährleisten. Im Anschluss daran wurde eine Nutzerumfrage durchgeführt, um Feedback von den Anwendern zu sammeln. Diese Umfrage zielte darauf ab, herauszufinden, wie das Design bei den Nutzern ankommt, ob die Anwendung intuitiv ist und welche Verbesserungsvorschläge oder zusätzlichen Informationen von den Nutzern gewünscht werden. Die Ergebnisse der Umfrage wurden sorgfältig ausgewertet. Es stellte sich heraus, dass die Nutzer das Design im Allgemeinen gut fanden, jedoch einzeln Verbesserungsvorschläge und der Wunsch nach zusätzlichen Informationen geäußert wurde. Basierend auf diesen Erkenntnissen wurde das Benutzerkonzept überarbeitet, um besser auf die Bedürfnisse und Wünsche der Nutzer einzugehen. In der letzten Phase der Entwicklung wurde anhand des überarbeiteten Benutzerkonzepts ein Prototyp mit Angular erstellt. Dieser dient als Grundlage für die weitere Entwicklung und Verfeinerung der Anwendung, um eine optimale Nutzererfahrung zu gewährleisten.

## Abstract

This student research project deals with the development of an application for managing shift changes using the Google Design Sprint. First, the UX was developed and then, based on this, a user concept was developed to ensure the user-friendliness and effectiveness of the application. Following this, a user survey was conducted to gather feedback from users. This survey aimed to find out how the design was received by users, whether the application was intuitive and what suggestions for improvement or additional information users would like to see. The results of the survey were carefully analyzed. It turned out that the users generally liked the design, but that there were individual suggestions for improvement and a request for additional information. Based on these results, the user concept was modified to better meet the needs and wishes of the users. In the final phase of development, a prototype was created with Angular based on the updated user concept. This serves as the basis for further development and refinement of the application to ensure an optimal user experience.

# Inhaltsverzeichnis

<b>Abbildungsverzeichnis</b>	<b>I</b>
<b>Abkürzungsverzeichnis</b>	<b>II</b>
<b>1 Einleitung</b>	<b>1</b>
<b>2 Grundlagen</b>	<b>3</b>
2.1 Google Design Sprint . . . . .	3
2.2 User Experience . . . . .	4
2.3 Technologien . . . . .	7
2.3.1 Das Angular-Framework . . . . .	7
2.3.2 MongoDB . . . . .	8
2.3.3 Node.js . . . . .	9
<b>3 Entwicklung der User Experience mit Verwendung des Google Design Sprints</b>	<b>10</b>
3.1 Phase 1: Verstehen und Definieren . . . . .	10
3.2 Phase 2 und 3: Skizzieren und Auswahl der Idee . . . . .	11
3.3 Phase 4: Entwicklung des Bedienungskonzepts . . . . .	14
3.4 Phase 5: Nutzerumfrage . . . . .	15
3.4.1 Methodik und Durchführung der Nutzerumfrage . . . . .	16
3.4.2 Auswertung und Analyse der Nutzerumfrage . . . . .	16
<b>4 Weiterentwicklung des Benutzerkonzepts auf Basis der Ergebnisse der Nutzerumfrage</b>	<b>19</b>

## Inhaltsverzeichnis

<b>5</b>	<b>Entwicklung eines Prototypen des Benutzerkonzepts mit dem Angular-Framework</b>	<b>21</b>
5.1	Backend . . . . .	21
5.2	Frontend . . . . .	22
<b>6</b>	<b>Fazit und Ausblick</b>	<b>24</b>
	<b>Literaturverzeichnis</b>	<b>II</b>
	<b>Anhang</b>	<b>III</b>

# Abbildungsverzeichnis

1.1	Beispiel der Schichtwechsel-Liste aus WhatsApp für Januar . . . . .	1
2.1	Ablauf eines GDS . . . . .	3
2.2	Beispielhafte Visualisierung der Eye Tracking Methode anhand einer Heat Map . . . . .	6
2.3	Komponenten in Angular . . . . .	7
2.4	@Component-Dekorator in TypeScript . . . . .	7
2.5	Beispiel eines Dokuments in einer MongoDB . . . . .	9
3.1	Erste Version der Übersichtsseite für den Desktop . . . . .	11
3.2	Erste und zweite Version der Übersichtsseite für das Handy . . . . .	12
3.3	Dritte Version der Übersichtsseite für das Handy, im Light Mode und Dark Mode . . . . .	13
3.4	Farbpalette für den Light Mode und Dark Mode . . . . .	14
3.5	Bedienungskonzept der Anwendung . . . . .	15
3.6	Bewertung des visuellen Designs der Anwendung . . . . .	17
4.1	Weiterentwickeltes Benutzerkonzept . . . . .	20
5.1	Login- und Übersichtsseite im Dark Mode auf einem 400x600px großen Bildschirm . . . . .	22

# Abkürzungsverzeichnis

**GDS** Google Design Sprint

**UX** User Experience

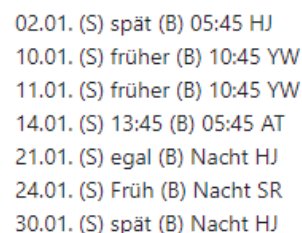
**TS** TypeScript

**JSON** JavaScript Object Notation



# 1 Einleitung

In der modernen Arbeitswelt spielen flexible Arbeitszeitmodelle eine zunehmend wichtiger werdende Rolle, um den vielfältigen Anforderungen sowohl der Arbeitgeber als auch der Arbeitnehmer gerecht zu werden. Laut einer Studie der International Labour Organization wird Flexibilität in der Arbeitszeitgestaltung als entscheidender Faktor für die Vereinbarkeit von Beruf und Privatleben angesehen und kann die Zufriedenheit und Produktivität der Mitarbeiter signifikant erhöhen [1, S.12 ff.]. Ein zentrales Element dieser Flexibilität ist die Möglichkeit für Mitarbeiter, Schichten untereinander zu tauschen. Oft werden Schichttauschvereinbarungen privat organisiert, beispielsweise über Messenger-Dienste wie WhatsApp. In vielen Unternehmen organisieren Mitarbeiter ihre Schichtwechsel eigenständig und informell, oft über Messaging-Dienste wie WhatsApp. Dies gilt auch für die Mitarbeiter dieser Firma, die ihre Schichtwechsel in einer WhatsApp-Gruppe privat koordinieren. Ein typisches Beispiel einer solchen Liste für den Monat Januar sieht wie folgt aus:



02.01. (S) spät (B) 05:45 HJ  
10.01. (S) früher (B) 10:45 YW  
11.01. (S) früher (B) 10:45 YW  
14.01. (S) 13:45 (B) 05:45 AT  
21.01. (S) egal (B) Nacht HJ  
24.01. (S) Früh (B) Nacht SR  
30.01. (S) spät (B) Nacht HJ

Abbildung 1.1: Beispiel der Schichtwechsel-Liste aus WhatsApp für Januar

Hierbei steht (S) für „Suche“ und (B) für „Biete“, gefolgt von den jeweiligen Schichtzeiten und den Initialen der Personen.

Diese manuelle Methode bringt mehrere Herausforderungen und Ineffizienzen mit sich. Die fortlaufende Aktualisierung der Liste in WhatsApp führt oft zu einer unübersichtlichen Flut an Informationen, bei der Änderungen oder neue Einträge leicht übersehen

werden können. Außerdem erfordert die aktuelle Methode von den Mitarbeitern, die gesamte Kommunikation und Liste in WhatsApp ständig zu überwachen, was zeitaufwendig und unpraktisch ist. Diese Herausforderungen beeinträchtigen die Effizienz der Schichtorganisation und führen zu mehr Unzufriedenheit bei den Mitarbeitern.

Das Ziel dieser wissenschaftlichen Arbeit ist es, ein benutzerfreundliches und effizientes UX Design Konzept zur Organisation von Schichtwechseln zu entwickeln. Dieses Konzept soll die aktuellen Probleme der ineffizienten WhatsApp-basierten Methode adressieren und eine strukturierte, leicht zugängliche und übersichtliche Alternative bieten.

## 2 Grundlagen

### 2.1 Google Design Sprint

Der Google Design Sprint (GDS) ist eine von Jake Knapp bei Google Ventures entwickelte Methodik zur schnellen und effizienten Problemlösung sowie Produktentwicklung, insbesondere für Herausforderungen, die sich aus der dynamischen Natur des Marktes und den sich verändernden Produktanforderungen ergeben. Ziel dieser Methode ist es, innerhalb eines Zeitraums von fünf Tagen einen Prototyp zu entwickeln und zu evaluieren. Diese Methode bietet den Vorteil, dass nicht auf die Markteinführung gewartet werden muss, um Feedback zu erhalten. Stattdessen können dringende Fragen sofort beantwortet werden [2, S.98 f.].

Die fünftägige Methode, wie in Abbildung 2.1 dargestellt, verläuft wie folgt:

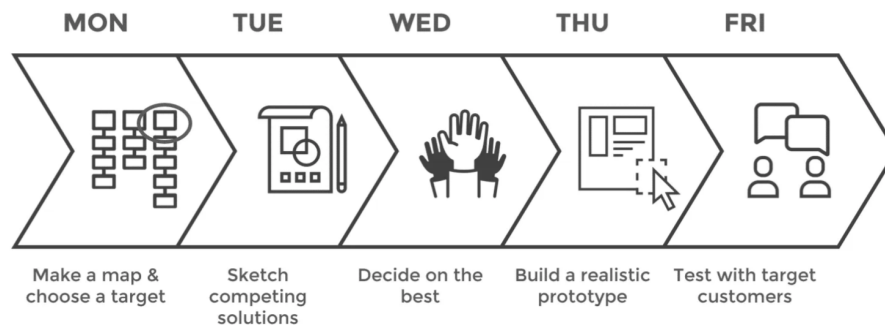


Abbildung 2.1: Ablauf eines GDS [3]

Am ersten Tag geht es darum, das Problem zu verstehen und den Fokus für die Woche festzulegen. Das Team definiert das langfristige Ziel und identifiziert die Herausforderung.

Am zweiten Tag konzentriert sich das Team auf die Bewältigung bereits bekannter Herausforderungen. Anders als bei herkömmlichen Brainstorming-Sitzungen arbeiten die Team-

## 2.2. USER EXPERIENCE

mitglieder einzeln an Lösungsansätzen und folgen einem strukturierten vierstufigen Prozess, um das kritische Denken zu fördern.

Am dritten Tag trifft das Team Entscheidungen darüber, welche Idee als Prototyp entwickelt und getestet werden sollen. Dabei kommt die fünfstufige „Sticky Decision“-Methode zum Einsatz, um die besten Lösungen zu identifizieren. Anschließend wird ein detaillierter Prozessplan für den Prototypen erstellt.

Am vierten Tag wird ein realitätsnaher Prototyp entwickelt. Das Ziel ist es, eine testbare Version der Lösung zu erstellen, die am nächsten Tag mit echten Nutzern evaluiert werden kann.

Der letzte Tag ist für das Testen des Prototyps reserviert. Das Team sammelt Feedback von echten Nutzern und erhält Einsicht, ob die Lösung in der Praxis funktioniert und welche Anpassungen nötig sind. Dieses Feedback ist entscheidend, um die Stärken und Schwächen der entwickelten Lösung zu identifizieren [2, S.22 ff.].

## 2.2 User Experience

Das Design der User Experience (UX) konzentriert sich auf jedes Element und jede Funktion, die der Benutzer bei einer Anwendung sieht, mit dem Ziel, eine möglichst angenehme und effiziente Erfahrung zu ermöglichen [4, S.12]. Laut der International Organization for Standardization (ISO 9241-210) wird UX definiert als „Wahrnehmungen und Reaktionen einer Person, die sich aus der Verwendung und/oder der erwarteten Verwendung eines Produkts, Systems oder einer Dienstleistung ergeben“ [5]. Während sich das User Interface auf das Erscheinungsbild der Anwendung konzentriert, beispielsweise auf Schriftarten und Farben, geht die UX tiefer und umfasst das gesamte Erlebnis des Nutzers [6, S.8].

In den letzten Jahrzehnten hat die Bedeutung der User Experience stark zugenommen, da Unternehmen erkannt haben, dass ein positives Nutzungserlebnis entscheidend für den Erfolg eines Produkts ist [7]. Bei einem schlechten UX-Design fällt es den Benutzern schwer, die Anwendung zu nutzen, und sie wechseln, sobald sie eine ähnliche, bessere App gefunden haben, die die gleiche Aufgabe erfüllt. Deshalb ist es für ein gutes UX-Design wichtig, dass der Designer die Bedürfnisse der Benutzer vorhersieht und erfüllt, indem er ihre Sichtweise einnimmt. Ein gutes UX-Design kann die Produktivität steigern, die Zufriedenheit der Kunden erhöhen und die Verkaufszahlen verbessern. Zudem können

## 2.2. USER EXPERIENCE

durch eine optimierte Benutzererfahrung die Kosten für Support und Wartung gesenkt werden [6, S.8 ff.].

Die UX umfasst verschiedene Komponenten, die zusammenarbeiten, um ein positives Nutzungserlebnis zu schaffen. Dazu gehören Usability, Ästhetik, Interaktion und Emotionalität.

Die Usability bezieht sich auf die Benutzerfreundlichkeit und Effizienz eines Produkts. Ein System mit hoher Usability ermöglicht es den Nutzern, ihre Ziele schnell und ohne Frustration zu erreichen. Dies beinhaltet Aspekte wie Lernfähigkeit, Effizienz der Nutzung und Zufriedenheit der Nutzer [8, S.23 ff.].

Die Ästhetik bezieht sich auf das Erscheinungsbild des Produkts. Ein ansprechendes visuelles Design kann die Wahrnehmung der Qualität und den Gesamteindruck eines Produkts erheblich verbessern. Studien zeigen, dass ästhetisch ansprechende Designs oft als funktionaler und benutzerfreundlicher wahrgenommen werden [9].

Die Interaktion beschreibt, wie Nutzer mit einem Produkt oder einer Dienstleistung umgehen. Dies umfasst die Art und Weise, wie Eingaben getätigt werden, wie das System auf Nutzeraktionen reagiert und wie intuitiv diese Interaktionen gestaltet sind. Gute Interaktionsdesigns minimieren die kognitive Belastung und fördern eine reibungslose und intuitive Nutzung [10].

Emotionalität in der UX bezieht sich auf die emotionalen Reaktionen und das subjektive Erleben der Nutzer während der Interaktion mit einem Produkt. Positive emotionale Erfahrungen können die Benutzerbindung und Zufriedenheit erhöhen. Studien belegen, dass Produkte, die positive Emotionen hervorrufen, häufiger genutzt und weiterempfohlen werden [11].

Es gibt eine Vielzahl von Methoden zur Bewertung der UX, zu den gebräuchlichsten Methoden gehören:

Usability-Tests beinhalten die Beobachtung und Analyse der Nutzerinteraktionen mit einem System, um Nutzerprobleme zu identifizieren. Durch das Testen in realistischen Nutzungsszenarien können konkrete Probleme und deren Schweregrad ermittelt werden, was zur Verbesserung der Benutzerfreundlichkeit beiträgt. Diese Methode ist besonders effektiv, um zu verstehen, wie Nutzer tatsächlich mit dem System interagieren [8, S.165 ff.].

## 2.2. USER EXPERIENCE

Mit Nutzerbefragungen und Interviews können qualitative Daten direkt von den Nutzern gesammelt werden. Diese Methode ermöglicht es, tiefere Einblicke in die Meinungen, Erfahrungen und Verbesserungsvorschläge der Nutzer zu gewinnen, was zur Optimierung der Interaktionen und der allgemeinen Zufriedenheit beitragen kann. Durch gezielte Fragen können spezifische Aspekte der Nutzererfahrung identifiziert und adressiert werden [12].

Bei der Eye-Tracking-Technologie wird die Augenbewegungen von Nutzern verfolgt, wie in Abbildung 2.2 dargestellt.

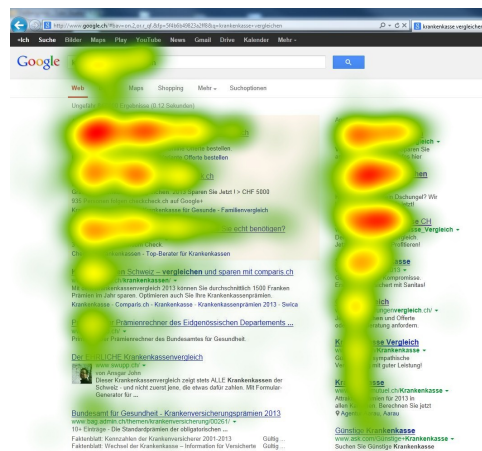


Abbildung 2.2: Beispielhafte Visualisierung der Eye Tracking Methode anhand einer Heat Map [13]

Durch die Messung von Fixationen, also den Momenten, in denen das Auge auf einem Punkt verharret, kann festgestellt werden, welche Elemente einer Benutzeroberfläche besondere Aufmerksamkeit erregen. Sakkaden, die schnellen Augenbewegungen zwischen Fixationen, helfen dabei zu verstehen, wie Nutzer Informationen auf einer Seite scannen. Diese Daten liefern wertvolle Einblicke in die Nutzerinteraktion und ermöglichen die Optimierung von Design und Layout basierend auf tatsächlichen Nutzerverhalten [14, S.3 ff.].

Durch die Kombination dieser Methoden können umfassende Einblicke in die Stärken und Schwächen der User Experience gewonnen werden, die zur kontinuierlichen Verbesserung des Produkts beitragen.

## 2.3 Technologien

### 2.3.1 Das Angular-Framework

Angular ist eine Plattform zur Entwicklung von Webanwendungen, die auf TypeScript (TS) basiert. Mit Angular können Single Page Applications für Web-, Mobil- und Desktop-Anwendungen entwickelt werden. Es ist Open Source und wird von Google unterstützt, was bedeutet, dass es kostenlos verwendet werden kann und von einer großen Entwickler-Community gepflegt wird. Ursprünglich war es als AngularJS bekannt und nutzte JavaScript als Hauptprogrammiersprache. Die Codebasis wurde jedoch komplett neu geschrieben, wobei JavaScript durch TS ersetzt wurde [15].

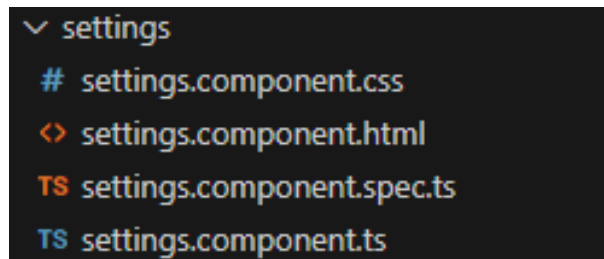


Abbildung 2.3: Komponenten in Angular

Die grundlegenden Bausteine des Angular-Frameworks sind die Komponenten, welche eine isolierte und somit wieder verwendbare Einheit sind. Eine Komponente besteht aus einer HTML- und CSS-Datei (siehe Abbildung 2.3), die das Aussehen der Benutzeroberfläche definieren. Die zugehörige Klasse enthält den TS Code, der das Verhalten der Komponente steuert [15].

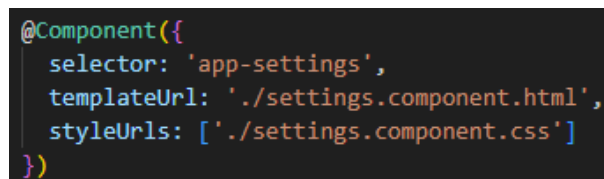


Abbildung 2.4: @Component-Dekorator in TS

Der @Component-Dekorator in TS fügt der Klasse zusätzliche Metadaten hinzu, wie den Pfad zum Template und zu den Styles (siehe templateUrl und styleUrls in der Abbildung 2.4) [15].

## 2.3. TECHNOLOGIEN

Durch Unit-Tests in Angular ist es möglich, isoliert einzelne Teile der Anwendung zu testen, um sicherzustellen, dass diese gemäß den Erwartungen funktionieren. Diese Tests sind in `.spec.ts` Dateien organisiert und umfassen Testfälle für Komponenten, Dienste und Module. Sie werden üblicherweise mit Testing-Frameworks wie Jasmine und Karma ausgeführt, die in die Angular-Entwicklungsumgebung integriert sind. Neben Unit-Tests gibt es auch End-to-End-Tests, die das Verhalten der Anwendung in einer realen Umgebung simulieren [16, S.340 ff.].

Der Routing-Service ermöglicht es verschiedene Ansichten basierend auf der URL der Anwendung zu laden. Durch die Definition von Routen und deren Zuordnung zu spezifischen Komponenten ermöglicht Angular eine klare Strukturierung und Navigation innerhalb der Anwendung, was insbesondere für komplexe Single Page Applications entscheidend ist. Routen können parameterisiert werden oder der Zugriff auf bestimmte Routen kann durch Guards gesteuert werden [17].

### 2.3.2 MongoDB

MongoDB ist eine dokumentenorientierte NoSQL-Datenbank und entstand 2007 aus einem Platform-as-a-Service-Projekt von dem Start Up 10gen, das auf automatisiertes Infrastrukturmanagement abzielte. Aufgrund der Ablehnung durch Entwickler, die Kontrolle über ihre Technologie-Stacks zu verlieren, fokussierte sich 10gen auf die Datenbank und benannte sich in MongoDB, Inc. um. MongoDB wird als Open-Source-Projekt entwickelt, wobei die Community zur Mitarbeit ermutigt wird, das Geschäftsmodell umfasst die Unterstützung der Open-Source-Entwicklung und das Anbieten von Abonnementdiensten [18, S.5 f.].

MongoDB wurde zur schnellen Entwicklung von Webanwendungen und Internet-Infrastruktur designed. Es ist möglich, ein strukturiertes Dokument in der Datenbank zu speichern, ohne bei der Erstellung der Datenbank schon wissen zu müssen, wie die Datenstruktur aussehen wird oder neue Tabellen erstellen zu müssen, wenn sich die Daten ändern. Die Struktur wird durch den Code der Anwendung bestimmt. Außerdem ist es, wenn für die Speicherung der Datenmenge mehr als eine Datenbank benötigt wird, mit MongoDB deutlich einfacher und performanter diese zu skalieren, als bei einer relationalen Datenbank [18, S.4 ff.].

Intern speichert MongoDB die Daten als Binary JavaScript Object Notation (JSON), eine



## 2.3. TECHNOLOGIEN

Struktur, die JSON ähnelt, jedoch für die Speicherung vieler Dokumente optimiert ist. Nach einer Abfrage werden die Ergebnisse in eine leicht lesbare Datenstruktur übersetzt, das Dokumentenformat JSON, das aus Schlüsseln und Werten besteht (für Beispiel siehe Abbildung 2.5). Während relationale Datenbanken Tabellen verwenden, nutzt MongoDB stattdessen Kollektionen [18, S.7].

```
{
  "_id": {"_id"},
  "firstName": "Default",
  "lastName": "User",
  "email": "defaultuser@example.com",
  "password": "$2a$10$0pNHMV.0CDt/8frBXw7xuV9D0v4QMhUj8/L3HdzWL5lFb6UgCJnC",
  "_v": 0
}
```

Abbildung 2.5: Beispiel eines Dokuments in einer MongoDB

Insgesamt bietet MongoDB eine leistungsfähige, flexible und skalierbare Lösung für die Verwaltung komplexer und dynamischer Datenmodelle, was sie ideal für moderne Webanwendungen macht.

### 2.3.3 Node.js

Node.js ist eine serverseitige Laufzeitumgebung, die auf der V8-JavaScript-Engine von Google Chrome basiert und die Ausführung von JavaScript-Code außerhalb des Browsers ermöglicht. Entwickelt von Ryan Dahl im Jahr 2009, hat sich Node.js schnell als leistungsfähige Plattform für die Erstellung von serverseitigen Anwendungen etabliert [19, S.3 ff.]. Ein wesentliches Merkmal von Node.js ist sein ereignisgesteuertes, nicht-blockierendes I/O-Modell, das eine effiziente Verarbeitung gleichzeitiger Verbindungen ermöglicht [20]. Diese Architektur ist besonders für Anwendungen geeignet, die häufige I/O-Operationen ausführen, wie beispielsweise APIs oder kleine Webserver. Ein weiterer Vorteil von Node.js ist die Nutzung von JavaScript sowohl auf der Client- als auch auf der Serverseite, was zu einer einheitlichen Codebasis und vereinfachten Entwicklungsprozessen führt. Die Kombination von Angular und Node.js bietet daher eine schnelle und effiziente Datenverarbeitung sowie eine einheitliche Entwicklungsumgebung [21, S.5 ff.].

# 3 Entwicklung der User Experience mit Verwendung des Google Design Sprints

Die Entwicklung eines nutzerfreundlichen und effektiven UX-Design-Konzepts zur Abstimmung von Schichtwechseln wurde mithilfe des GDS realisiert. Ursprünglich wurde der GDS für einen Zeitraum von fünf Tagen entwickelt, um schnell Lösungen für komplexe Probleme zu finden und neue Ideen zu testen. In diesem Projekt wurde der GDS aufgrund der Durchführung durch eine Einzelperson angepasst.

Die Phasen 4 und 5 des GDS, die normalerweise die Entwicklung eines Prototypen und das Testen umfassen, wurden modifiziert. Stattdessen wurde ein umfangreiches Bedienungskonzept entwickelt, auf dessen Grundlage eine Nutzerumfrage durchgeführt wurde (siehe Kapitel 3.4). Die Ergebnisse dieser Umfrage wurden anschließend genutzt, um das Benutzerkonzept weiter zu verbessern und anschließend den Prototypen zu erstellen.

## 3.1 Phase 1: Verstehen und Definieren

In der ersten Phase werden die benötigten Informationen gesammelt und analysiert, um das Problem umfassend zu verstehen und klar zu definieren. Die Problemstellung und Zielsetzung wurden bereits in der Einleitung ausführlich behandelt, daher wird in diesem Kapitel direkt auf die praktische Anwendung der nächsten Phasen eingegangen.

### 3.2. PHASE 2 UND 3: SKIZZIEREN UND AUSWAHL DER IDEE

## 3.2 Phase 2 und 3: Skizzieren und Auswahl der Idee

In der Skizzieren Phase beginnt der Prozess der Designerstellung. Hierbei wurden sich zunächst die benötigten Seiten überlegt, um die benötigten Aufgaben und Anforderungen zu erfüllen. Dazu gehört eine Übersichtsseite auf der alle aktuellen Tauschangebote stehen und worüber der Nutzer zu allen weiteren Seiten gelangt, z.B. die zum Hinzufügen und zum Annehmen eines Tauschgebots, sowie eine Seite, auf der der Nutzer seine angenommenen Schichten sehen kann und individuelle Einstellungen vornehmen kann.

Die erste Version der Übersichtsseite war zunächst für einen Desktop ausgelegt (siehe Abbildung 3.1).

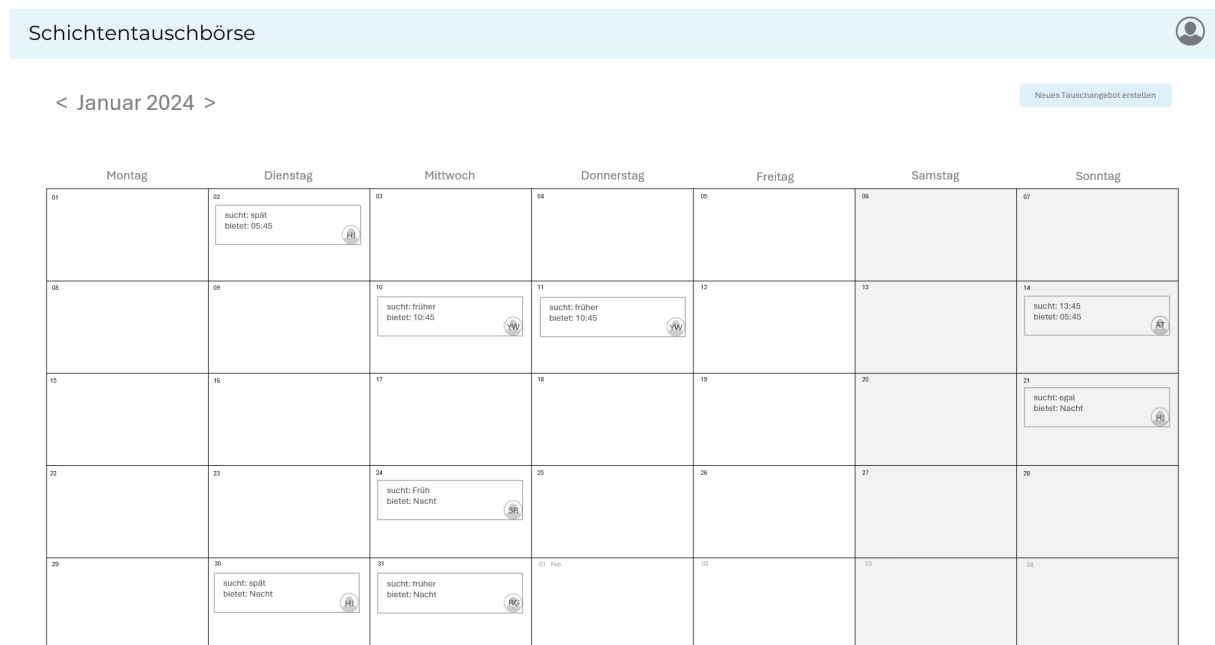


Abbildung 3.1: Erste Version der Übersichtsseite für den Desktop

Im oberen Bereich der Seite befindet sich der Header. Links darunter hat der Nutzer die Möglichkeit zwischen den Monaten zu navigieren. Rechts unter dem Header befindet sich „Neues Tauschangebot erstellen“ Button, mit dem Nutzer neue Tauschangebote erstellen können. Zentral auf der Seite ist der Kalender des aktuellen Monats groß dargestellt, die Wochentage sind darüber ausgeschrieben.

Bei der Übertragung des Designs in die Handy-Version wurden einige Schwachstellen identifiziert. Die Darstellung des Desktop-Designs auf dem Handy führt dazu, dass die

### 3.2. PHASE 2 UND 3: SKIZZIEREN UND AUSWAHL DER IDEE

Schrift aufgrund der kleinen Größe schlecht lesbar war (siehe Abbildung 3.2).

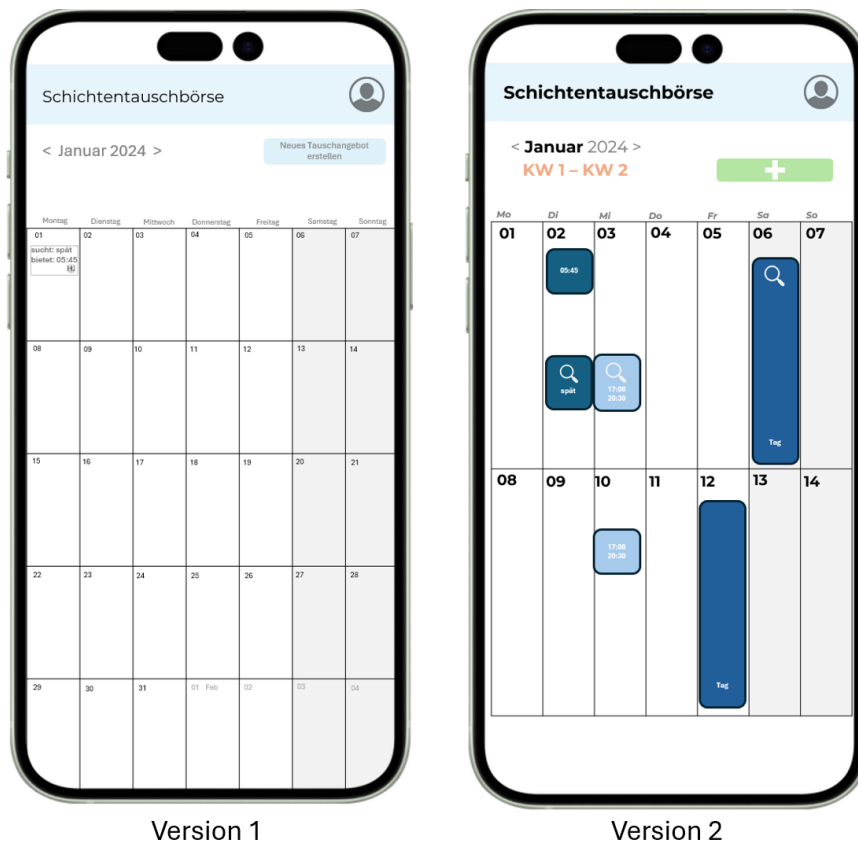


Abbildung 3.2: Erste und zweite Version der Übersichtsseite für das Handy

Da die meisten Nutzer die App wahrscheinlich auf dem Handy verwenden möchten, wurde das Design in Version 2 entsprechend angepasst.

Um in der Kalenderübersicht die Tauschanfragen gut lesbar und nutzerfreundlich darzustellen, werden statt fünf Wochen nur noch zwei Wochen angezeigt (siehe Version 2 in Abbildung 3.2). Aufgrund der schmalere Handybildschirme wird ein Teil der Wörter durch Piktogramme ersetzt, z.B. wurde beim „Neues Tauschangebot erstellen“ Button der Text durch ein Plus ersetzt. Dadurch wird Platz gespart und die Funktion wird klarer erkannt. Außerdem werden die Wochentage nicht mehr ausgeschrieben. Eine weitere Änderung ist, dass die Schichten in den gesuchten/gebotenen Zeit Blöcken am Tag abgebildet (siehe Version 2 in Abbildung 3.2).

Allerdings wurde festgestellt, dass der Kalender unübersichtlich wird, wenn mehrere Tauschangebote an einem Tag sind oder wenn welche zum gleichen Zeitpunkt angefragt werden.

### 3.2. PHASE 2 UND 3: SKIZZIEREN UND AUSWAHL DER IDEE

Deswegen wird eine weitere Version erstellt, in der die Schichten nicht mehr nach dem Zeitpunkt sortiert sind, sondern in einem Block organisiert, was eine bessere Übersicht ermöglicht (siehe Abbildung 3.3).



Abbildung 3.3: Dritte Version der Übersichtsseite für das Handy, im Light Mode und Dark Mode

Des weiteren wurde die Hauptfarbe von einem hellen Blau zu einem Orange gewechselt, welches wärmer und aktiver wirkt. Da der Dark Mode in den letzten Jahren stark an Popularität zugenommen hat [22] wird für die Anwendung auch ein Farbkonzept erstellt (siehe Dark Mode in Abbildung 3.3).

Für die dritte Version wurde sich letztendlich entschieden. Darauf aufbauend wurden die weiteren Seiten der Anwendung im Light Mode gestaltet.

## 3.3 Phase 4: Entwicklung des Bedienungskonzepts

Um den Nutzern eine bessere Vorstellung von der UX zu vermitteln, wurden zwei unterschiedliche Farbpaletten erstellt: eine für den Light Mode und eine für den Dark Mode (siehe Abbildung 3.4).

Aptos	255 255 255	Aptos	26 31 40
<b>Bold</b>	0 0 0	<b>Bold</b>	255 255 255
<i>Italic</i>	174 174 174	<i>Italic</i>	215 140 61
Light	235 197 158	Light	174 174 174
Regular	215 140 61	Regular	132 151 176
SemiBold	107 70 30	SemiBold	39 48 63

Abbildung 3.4: Farbpalette für den Light Mode und Dark Mode

Die Farbpalette für den Light Mode besteht aus verschiedenen Orangetönen und einem Braun. Die Hintergrundfarbe des Dark Modes ist ein dunkles Blau, Orange wird nur noch als Akzentfarbe verwendet, ansonsten wird ein helleres Blau genutzt. Die Schriftart „Aptos“ wurde aufgrund ihrer Lesbarkeit und Ästhetik ausgewählt.

Ein zentrales Element des UX Designs war die Entwicklung eines klaren und intuitiven Bedienungskonzepts. Ziel war es, eine benutzerfreundliche Navigation zu schaffen, die es den Nutzern ermöglicht, alle Funktionen der Anwendung schnell und einfach zu erreichen.

Nachdem sich der Nutzer registriert und anschließend erfolgreich angemeldet hat, gelangt er auf die zentrale Übersichtsseite, wie in Abbildung 3.5 dargestellt. Diese ermöglicht den Zugang zu allen anderen Seiten der Anwendung. Über die User-Kennung im Header gelangt der Nutzer zu einer Einstellungsseite. Dort kann er seine persönlichen Daten ändern und die bereits getauschten Schichten einsehen. Durch Klicken auf den Namen der Anwendung „Schichtentauschbörse“ im Header gelangt er jederzeit zurück zur Übersicht.

Möchte ein Nutzer eine neue Tauschanfrage erstellen, ist das über den Button direkt unter dem Header möglich. Dabei wird eine neue Seite aufgerufen, auf der die gebotene und gesuchte Schicht aus einem Dropdown-Menü ausgewählt werden können. Alternativ kann der Nutzer auch eine bestimmte Uhrzeit eingeben, falls eine spezifische Zeit gewünscht ist. Wenn der Nutzer seine Anfrage doch nicht stellen möchte, kann er den Vorgang abbrechen

### 3.4. PHASE 5: NUTZERUMFRAGE

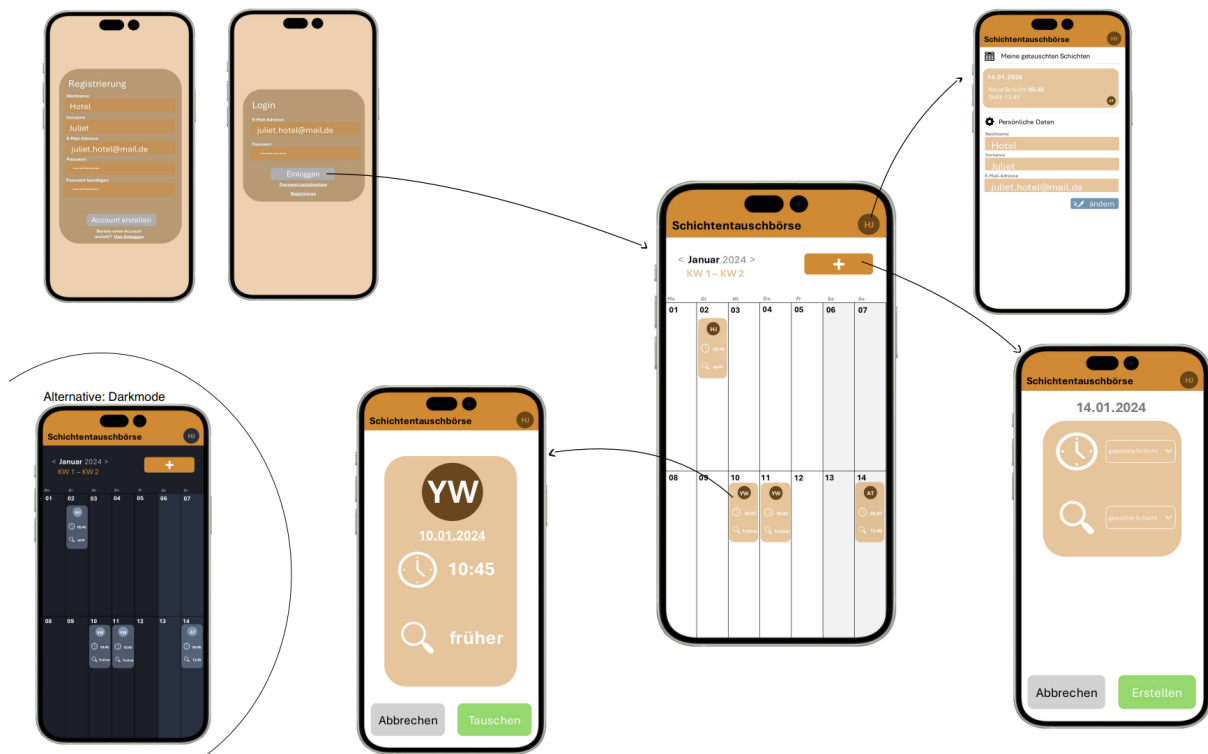


Abbildung 3.5: Bedienungskonzept der Anwendung

und gelangt zurück zur Übersichtsseite. Mit dem Drücken des „Erstellen“-Buttons wird die neue Tauschanfrage finalisiert und im Kalender für alle Nutzer sichtbar.

Klickt ein Nutzer im Kalender auf eine Anfrage, kann er das Annehmen der Tauschanfrage entweder über den Button „Abbrechen“ beenden oder über den Button „Tauschen“ die Schicht tauschen. Nach dem Annehmen verschwindet die Tauschanfrage aus dem Kalender und wird in den Einstellungen unter „Meine getauschten Schichten“ angezeigt.

Zusätzlich wurde eine alternative Dark Mode-Version der Übersichtsseite erstellt, um unterschiedliche Präferenzen der Nutzer einzubinden.

## 3.4 Phase 5: Nutzerumfrage

Die Farbpalette und die Übersicht über das Bedienungskonzept, für detailliertere Informationen siehe Abschnitt 3.3, bilden die Grundlage für die Nutzerumfrage. Diese wurde als zentraler Bestandteil der Evaluierung des UX Designs der Webanwendung durchge-

### 3.4. PHASE 5: NUTZERUMFRAGE

führt. Ziel der Umfrage ist es, Feedback von den Nutzern zu sammeln, um die Benutzerfreundlichkeit und Zufriedenheit zu messen und mögliche Verbesserungspotenziale zu identifizieren.

Die Umfrage richtete sich an die zukünftigen Endnutzer der Schichttauschenanwendung innerhalb der Firma. Für detailliertere Informationen zum Fragebogen der Nutzerumfrage, siehe Anhang 6.

#### 3.4.1 Methodik und Durchführung der Nutzerumfrage

Die Umfrage besteht aus einem quantitativen und einem qualitativen Abschnitt. Der quantitative Teil der Umfrage umfasst Fragen, die die Intuitivität der Funktionen, die Ansprechbarkeit des visuellen Designs, die Übersichtlichkeit des Bildschirmaufbaus, die Ästhetik im Light Mode und Dark Mode sowie die Präferenz für die Nutzung der Anwendung über Web, Handy oder beides bewertet haben.

Im qualitativen Teil der Umfrage wurden die Teilnehmer durch offene Fragen dazu angeregt, ihre Verbesserungsvorschläge und Anregungen zur Umsetzung zu geben. Es wurde nach den fehlenden Funktionen oder Informationen sowie den wichtigsten Aspekten für die Nutzer gefragt.

Die Umfrage wurde online über ein Microsoft Forms Dokument durchgeführt, was eine einfache Verteilung und Teilnahme ermöglicht hat.

#### 3.4.2 Auswertung und Analyse der Nutzerumfrage

An der Umfrage haben fünf von sechs zukünftigen Nutzern der Schichttauschenanwendung teilgenommen. Die Ergebnisse bieten einen Einblick in die Benutzerfreundlichkeit, das visuelle Design und die Funktionalität der Anwendung.

Die Teilnehmer bewerteten die Intuitivität der Funktionen der Anwendung durchschnittlich mit 9,0 von 10 Punkten. Dies deutet darauf hin, dass die meisten Nutzer die Bedienung der Anwendung als sehr intuitiv empfanden. Dennoch wurden spezifische Verbesserungswünsche geäußert, es wurde der Wunsch geäußert, den Prozess des Ringtausches zu erleichtern und klare Kommunikationsmöglichkeiten bei der Angabe von Schichten zu schaffen, die für den Tausch angeboten werden können, insbesondere wenn ein Nutzer nur



### 3.4. PHASE 5: NUTZERUMFRAGE

nach einer späteren Schicht sucht.

Die Umfrageergebnisse der wichtigsten Aspekte für den Nutzer zeigen, dass diesen vor allem die Übersichtlichkeit der Anwendung am wichtigsten ist, was fünfmal genannt wurde. Weitere bedeutende Aspekte umfassen die Effizienz der Anwendung, die Organisation bei mehreren Tauschwünschen an einem Tag und die Bestätigung des Tauschs mit möglichst wenigen Schritten. Außerdem wurde der Wunsch nach helleren Farben für einen besseren Kontrastmodus und ein schnelles Erfassen der gesuchten Schichten geäußert.

Das visuelle Design der Anwendung wurde im Durchschnitt mit 7,2 Punkten bewertet.

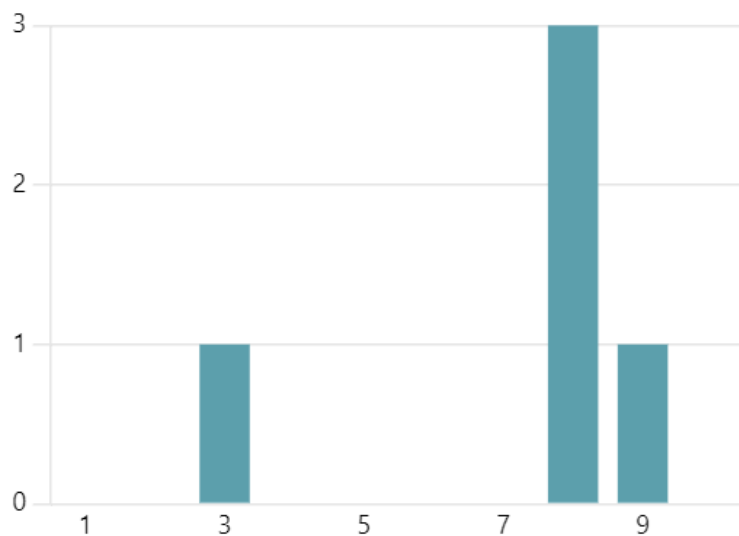


Abbildung 3.6: Bewertung des visuellen Designs der Anwendung

Wie in Abbildung 3.6 dargestellt, reichten die Bewertungen von 3 bis 9 Punkten, was auf eine unterschiedliche Wahrnehmung und verschiedene Präferenzen der Nutzer hinweist.

Die Übersichtlichkeit des Bildschirmaufbaus erhielt eine sehr hohe durchschnittliche Bewertung von 9,4 Punkten. Dies zeigt, dass die Mehrheit der Nutzer den Bildschirmaufbau als sehr klar und strukturiert empfand.

Die allgemeine Ästhetik der Anwendung wurde sowohl im Light Mode als auch im Dark Mode bewertet. Der Light Mode erhielt eine durchschnittliche Bewertung von 8,0 Punkten, während der Dark Mode eine durchschnittliche Bewertung von 6,8 Punkten erhielt. Zwei

### 3.4. PHASE 5: NUTZERUMFRAGE

Personen gaben an, dass ihnen der Light Mode deutlich besser gefiel (3 bzw. 4 Punkte Unterschied), während zwei andere Personen den Dark Mode bevorzugten (2 und 1 Punkt Unterschied). Eine Person bewertete beide Modi gleich gut.

Die Mehrheit der Nutzer (vier von fünf) gab an, hauptsächlich die mobile App nutzen zu wollen, während eine Person sowohl die Web-Anwendung als auch die mobile App gleich häufig nutzen würde. Dies bestätigt die beim Entwurf der Anwendung getroffene Vermutung.

Die Nutzer äußerten mehrere Vorschläge für zusätzliche Informationen und Funktionen, die das Tauschen der Schichten vereinfachen könnten. Dazu gehören:

- Eine Anzeige, ob an einem Dienst ein Trainee beteiligt ist.
- Die Möglichkeit, Anfragen als dringend zu kennzeichnen.
- Ein Feld für Details zu den Anfragen.
- Eine Priorisierung bei mehreren Personen, die die gleiche Schicht suchen.
- Eine Verknüpfung mit dem eigenen Kalender, um Tauschmöglichkeiten schnell zu erkennen.
- Die Anzeige aller möglichen Tauschpartner und der zu tauschenden Dienste.
- Ideal wäre eine Anbindung an den Kalender oder den tatsächlichen Dienstplan, um das Hin- und Herspringen zwischen Anwendungen zu vermeiden.

Alle Teilnehmer gaben an, dass die Anwendung ihren Anforderungen entspricht, abgesehen von den Anmerkungen bei den vorherigen Fragen. Dies zeigt, dass die Anwendung bereits viele Bedürfnisse der Nutzer erfüllt, jedoch weiterhin etwas Raum für Verbesserungen und die Integration zusätzlicher Informationen besteht.

## 4 Weiterentwicklung des Benutzerkonzepts auf Basis der Ergebnisse der Nutzerumfrage

Basierend auf den Ergebnissen der Nutzerumfrage wird das Benutzerkonzept weiterentwickelt (siehe Abbildung 4.1). Mehrere Änderungen sind eingebunden, um die Benutzerfreundlichkeit zu verbessern.

Nutzer haben nun in den Einstellungen die Möglichkeit, Tauschangebote in ihren persönlichen Kalender zu exportieren. Dies erleichtert die Planung und Übersicht ihrer Schichtwechsel.

Außerdem werden Nutzer beim Hinzufügen einer neuen Schicht nach zusätzlichen Details gefragt. Insbesondere wird erfragt, ob ein Trainee bei der Schicht anwesend sein wird oder ob die Anfrage dringend ist. Dringende Anfragen sind visuell hervorgehoben, indem sie rot hinterlegt sind. Dies gilt sowohl im normalen als auch im Dark Mode, wie auf den Übersichtsseiten dargestellt.

Wenn Nutzer im Dropdown-Menü nur „später“ oder „früher“ auswählen, müssen sie einen Kommentar im Kommentarfeld hinzufügen. In anderen Fällen bleibt es den Nutzern überlassen, ob sie zusätzliche Details einfügen möchten.

Unten rechts auf der Abbildung 4.1 ist ein Beispiel für eine Tauschanfrage ohne besondere Merkmale zu sehen. Links neben der Übersichtsseite wird eine Tauschanfrage dargestellt, bei der ein Trainee bei der Schicht anwesend ist und die gesuchte Schicht früher stattfinden soll, mit geschriebenen Details im Kommentarfeld.

Weitere Vorschläge für zusätzliche Informationen aus der Nutzerumfrage wurden protokolliert, jedoch bislang nicht eingebunden.



Abbildung 4.1: Weiterentwickeltes Benutzerkonzept (Seite zum Registrieren und zum Einloggen sind gleich geblieben und sind deswegen nicht mit abgebildet)

# 5 Entwicklung eines Prototypen des Benutzerkonzepts mit dem Angular-Framework

Der Prototyp ist auf Basis des vorher entwickelten Benutzerkonzepts entwickelt. Für eine genaue Beschreibung des Benutzerkonzepts siehe Abschnitt 3.3.

## 5.1 Backend

Das Backend ist mit Node.js und MongoDB entwickelt. Hierbei wird der Express-Server von Node.js und die Mongoose-Bibliothek von MongoDB verwendet, um eine effiziente und skalierbare Datenverwaltung zu gewährleisten.

Die Kernfunktionalitäten des Backends umfassen die Benutzerregistrierung und -verwaltung. In der Datei `server.js` werden die Endpunkte für die Registrierung und Anmeldung der Nutzer definiert. Die Datei `userModel.js` enthält das Schema für die Benutzer, welches die Felder Vorname, Nachname, E-Mail-Adresse und Passwort umfasst. Das Passwort wird in der `userModel.js`-Datei mithilfe der Bibliothek `bcryptjs` verschlüsselt, bevor es in der Datenbank gespeichert wird. Dies stellt sicher, dass Passwörter nicht im Klartext in der Datenbank gespeichert werden. Datei `server.js` ist ebenfalls für die Überprüfung zuständig, ob ein Nutzer existiert und ob seine Anmeldedaten korrekt sind. Dies ermöglicht eine sichere Authentifizierung der Nutzer.

Die Registrierung eines neuen Nutzers erfolgt über die Methode `app.post('/api/register', async (req, res) => { ... })`. Der Nutzer gibt seine Daten ein, und bei korrekter Eingabe wird ein neues Benutzerobjekt basierend auf diesen Eingaben erstellt. Über die Methode `app.post('/api/login', async (req, res) => { ... })` wird beim Anmelde-

## 5.2. FRONTEND

versuch eines Nutzer abgeglichen, ob die eingegebenen E-Mail-Adresse und das Passwort mit den in der Datenbank gespeicherten Daten übereinstimmen. Zunächst wird überprüft, ob ein Nutzer mit der angegebenen E-Mail-Adresse existiert. Falls nicht, wird eine Fehlermeldung „Benutzer nicht gefunden“ zurückgegeben. Anschließend wird das eingegebene Passwort mit dem in der Datenbank gehashten Passwort verglichen. Ist das Passwort falsch, erhält der Nutzer die Fehlermeldung „Falsches Passwort“. Bei korrekter Eingabe der Anmeldedaten wird die Nachricht „Login erfolgreich“ zurückgegeben.

## 5.2 Frontend

Das Frontend der Anwendung ist mit dem Angular Framework entwickelt. Alle Seiten der Anwendung sind responsiv gestaltet, sodass die Nutzer die Anwendung sowohl auf dem Handy als auch auf dem Laptop angenehm nutzen können. Für den Dark Mode werden derzeit die Systemeinstellungen des Geräts verwendet.

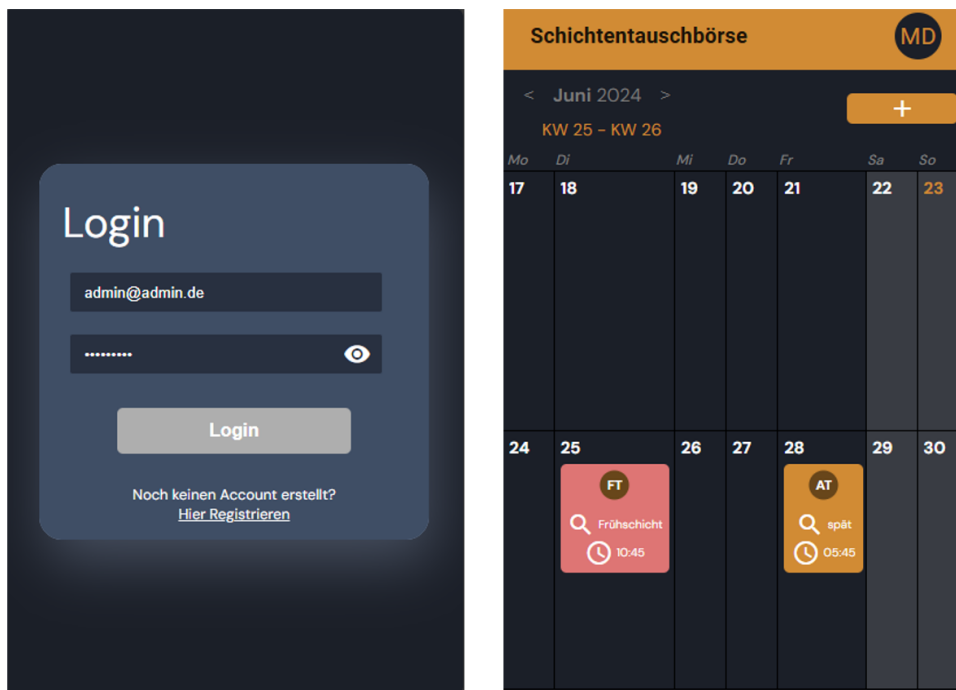


Abbildung 5.1: Login- und Übersichtsseite im Dark Mode auf einem 400x600px großen Bildschirm

Wenn ein Nutzer die Anwendung zum ersten Mal benutzt, muss er sich zunächst registrieren. Die Eingabefelder für die Registrierung werden validiert, um sicherzustellen,

## 5.2. FRONTEND

dass beispielsweise die E-Mail-Adresse im richtigen Format eingegeben wird. Mit der `PasswordStrengthValidator()` Funktion wird überprüft, ob das Passwort sicher genug ist. Da der Nutzer das Passwort zur Überprüfung zweimal eingeben muss, wird mithilfe der `passwordMatchValidator()` Funktion sichergestellt, dass beide Passwörter übereinstimmen. Falls dies nicht der Fall ist, wird eine Fehlermeldung angezeigt. Über das Augensymbol im Eingabefeld kann der Nutzer sich jedoch seine Passwörter anzeigen lassen (siehe Login-Seite in Abbildung 5.1).

Der Service `user.service.ts` kommuniziert für Login und Registrierung über HTTP-Anfragen mit dem Express-Server im Backend. Nach erfolgreicher Registrierung wird der Nutzer zum Login-Formular weitergeleitet, wo er sich mit seinem erstellten Account anmelden kann. Wenn die Anmeldedaten korrekt sind, wird der Nutzer zur Übersichtsseite weitergeleitet, von der aus alle anderen Seiten erreichbar sind.

Die Felder, in denen ein Tauschangebot eingetragen ist, sind etwas breiter, damit die Informationen besser zu erkennen sind. Der heutige Tag ist farbig hervorgehoben, wie in Abbildung 5.1 zu sehen ist.

Zusätzlich kann sich der Nutzer in den Einstellungen auch wieder ausloggen. Hierfür wird die Funktion `logout()` aus der `auth.service.ts`-Datei verwendet. Eine Funktion zum Exportieren der Tauschangebote in den Kalender wird im Frontend bereits angezeigt, ist jedoch im Backend noch nicht implementiert.

Wenn der Nutzer auf der Übersichtsseite auf den Button mit dem Plus klickt, kann er eine neue Tauschanfrage erstellen, die Eingabefelder dafür verwenden Angular Material. Der Nutzer kann das Datum aus einem Kalender auswählen oder selbst eingeben. Die angebotene und gesuchte Schicht kann aus einem Dropdown-Menü ausgewählt werden. Wenn der Nutzer eine benutzerdefinierte Uhrzeit eingeben möchte, kann er "Benutzerdefinierte Uhrzeit auswählen". Dann erscheint ein neues Eingabefeld, in dem er diese Uhrzeit eintragen kann. Durch Angular Material wird sichergestellt, dass die Uhrzeit im richtigen Format eingegeben wird. Eine Anfrage kann nur abgeschickt werden, wenn mindestens das Datum sowie die angebotene und gesuchte Schicht eingetragen sind.

Durch Klicken auf ein Tauschangebot im Kalender (siehe Abbildung 5.1) öffnet sich eine Seite mit detaillierteren Informationen, je nachdem, welche Informationen für die Schicht relevant sind und vom Nutzer eingetragen wurden.

## 6 Fazit und Ausblick

Die manuelle Methode zur Verwaltung von Schichtwechseln über WhatsApp hat Herausforderungen und Ineffizienzen aufgezeigt. Die ständige Aktualisierung der Liste führt zu einer unübersichtlichen Informationsflut, wodurch Änderungen oder neue Einträge leicht übersehen werden können.

Durch die Anwendung des GDS konnte eine effizientere und nutzerfreundlichere Lösung zur Abstimmung von Schichtwechseln entwickelt werden. In diesem Projekt wurde der GDS an die Durchführung durch eine Einzelperson angepasst, wobei die Phasen 4 und 5 modifiziert wurden, um ein umfangreiches Bedienungskonzept zu entwickeln und anschließend daran eine Nutzerumfrage durchzuführen.

Die Nutzerumfrage, die als zentraler Bestandteil der Evaluierung des UX-Designs diente, zeigte, dass die Anwendung insgesamt positiv bewertet wurde. Die Nutzer empfanden die Funktionen als sehr intuitiv und die Übersichtlichkeit der Anwendung wurde besonders hoch bewertet. Das visuelle Design erhielt positive, aber gemischte Bewertungen, dabei schnitt der Light Mode im Durchschnitt besser ab als der Dark Mode. Die Mehrheit der Nutzer bevorzugte die mobile Anwendung gegenüber der Web-Anwendung. Basierend auf den Ergebnissen der Nutzerumfrage wurden einige Verbesserungsvorschläge in das Benutzerkonzept eingebaut. Darauf aufbauend wurden Prototypen mit Angular entwickelt.

Im Rahmen der Prototypenentwicklung konnten grundlegende Funktionen wie das Anlegen und Einloggen von Nutzern implementiert werden, wobei die Daten in einer MongoDB-Datenbank gespeichert werden. Der aktuelle Entwicklungsstand beschränkt sich auf die Bereitstellung dieser Funktionen im Frontend, während der Prototyp derzeit nur lokal über den Localhost verfügbar ist. Um die Anwendung in der Praxis nutzbar zu machen und einen ortsunabhängigen Zugriff zu ermöglichen, ist eine Weiterentwicklung erforderlich, bei der das Projekt beispielsweise mithilfe von Docker allgemein zugänglich gemacht



wird.

Basierend auf den Ergebnissen der Nutzerumfrage und eigenen Überlegungen wurden verschiedene Verbesserungsvorschläge erarbeitet, um die Nutzerfreundlichkeit und Funktionalität der Anwendung weiter zu optimieren. Dazu gehört die Implementierung von Benachrichtigungen, die Nutzer darüber informieren, wenn ihre erstellte Schicht getauscht wurde oder ein neues Tauschangebot vorliegt. Zudem soll die Möglichkeit bestehen, diese Benachrichtigungen in den Einstellungen zu deaktivieren.

Eine Integration des Dienstplans soll sicherstellen, dass nur diejenigen Nutzer ein Tauschangebot annehmen können, die tatsächlich in der gesuchten Schicht eingetragen sind. Bisher wird darauf vertraut, dass die Nutzer diese Regel eigenständig einhalten. Mit dieser Funktion könnten alle potenziellen Tauschpartner angezeigt werden, was die Übersicht und Organisation erheblich erleichtert. Außerdem wäre es für die Nutzer von Vorteil, eine Ringtausch-Funktion zu implementieren, die den Austausch von Schichten zwischen mehreren Personen vereinfacht.

Ein weiterer Aspekt betrifft die Anpassung der Farbmodi. Bisher orientieren sich die Farbmodi der Anwendung an den Systempräferenzen der Nutzer. Dafür kann eine Funktion in den Einstellungen eingebaut werden, die es den Nutzern ermöglicht, direkt zwischen einem Dark Mode und einem Light Mode zu wählen.

# Literatur

- [1] International Labour Organization. *Working Time and the Future of Work*. Nov. 2018. URL: <https://www.ilo.org/publications/working-time-and-future-work> (besucht am 16.06.2024).
- [2] Jana Noack und José Díaz. *Das Design Sprint Handbuch*. 1. Aufl. dpunkt.verlag, 2019. ISBN: 978-3-86490-656-5.
- [3] *Google Design Sprint: a beginner guide*. URL: <https://medium.com/productschool/google-design-sprint-a-beginner-guide-835301d2d301> (besucht am 26.05.2024).
- [4] Marina Carulli Monica Bordegoni und Elena Spadoni. *Prototyping User eXperience in eXtended Reality*. Springer Cham, 2023. ISBN: 978-3-031-39683-0.
- [5] *ISO 9241-210:2019(en) Ergonomics of human-system interaction — Part 210: Human-centred design for interactive systems*. URL: <https://www.iso.org/obp/ui#iso:std:iso:9241:-210:ed-2:v1:en> (besucht am 28.05.2024).
- [6] Elvis Canziba. *Hands-On UX Design for Developers*. Packt, 2018. ISBN: 9781788626699.
- [7] Demis Basso Aurora Berni Yuri Borgianni und Claus-Christian Carbon. „Fundamentals and issues of user experience in the process of designing consumer products“. In: *Design Science* (2023), S. 1–36.
- [8] Jakob Nielsen. *Usability Engineering*. Morgan Kaufmann Publishers Inc., 1994. ISBN: 978-0-08-052029-2.
- [9] Adi Katz Noam Tractinsky und D. Ikar. „What is beautiful is usable“. In: *Interacting with Computers* (2000), S. 127–145.
- [10] *Interaction Design (IxD)*. URL: <https://www.interaction-design.org/literature/topics/interaction-design> (besucht am 28.05.2024).

## Literatur

- [11] Marc Hassenzahl. „The Interplay of Beauty, Goodness, and Usability in Interactive Products“. In: *Human-Computer Interaction* (2004), S. 319–349.
- [12] Marc Hassenzahl und Noam Tractinsky. „User experience - A research agenda“. In: *Behaviour and Information Technology* (2006), S. 91–97.
- [13] *Eye-tracking: a comprehensive guide*. URL: [https : / / couseux . com / eye - tracking/](https://couseux.com/eye-tracking/) (besucht am 28.05.2024).
- [14] Jennifer Romano Bergstrom und Andrew Schall. *Eye Tracking in User Experience Design*. Elsevier Science, 2014. ISBN: 978-0-12-408138-3.
- [15] *Introduction to Angular concepts*. URL: [https : / / angular . io / guide / architecture](https://angular.io/guide/architecture) (besucht am 20.05.2024).
- [16] Anton Moiseev Yakov Fain. *Angular Development with TypeScript*. Manning, 2018. ISBN: 9781638355250.
- [17] *Common Routing Tasks*. URL: <https://angular.dev/guide/routing/common-router-tasks> (besucht am 18.06.2024).
- [18] Kyle Banker u. a. *MongoDB in Action: Covers MongoDB version 3.0*. 2nd. Manning Publications Co., 2016. ISBN: 1617291609.
- [19] TJ Holowaychuk Mike Cantelon Marc Harter und Nathan Rajlich. *Node.js in Action*. Manning Publications, 2014.
- [20] Stefan Tilkov und Steve Vinoski. „Node.js: Using JavaScript to Build High-Performance Network Programs“. In: *IEEE Internet Computing* 14.6 (2010), S. 80–83.
- [21] Mario Casciaro und Luciano Mammino. *Node.js Design Patterns: Design and implement production-grade Node.js applications using proven patterns and techniques*. 3rd. Packt Publishing, 2020. ISBN: 978-1839214110.
- [22] Felix Kristallovich Henriette Eisfeld. „The Rise of Dark Mode - A qualitative study of an emerging user interface design trend“. In: (2020).

# Anhang

## Fragebogen der Nutzerumfrage zur Bewertung der UX

\* Erforderlich

1. Wie intuitiv finden Sie die Funktionen der Anwendung? \*

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ☐ 8 ☐ 9 ☐ 10

nicht intuitiv/schwer  
zu erkennen

intuitiv

2. Welche Funktion ist nicht intuitiv bzw. könnte besser umgesetzt werden?

☐

3. Welche Elemente sind Ihnen bei der Anwendung am wichtigsten/unwichtigsten? \*

☐

## Literatur

4. Wie ansprechend finden Sie das visuelle Design? \*

1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	----

nicht ansprechend

ansprechend

5. Sieht der Bildschirmaufbau für Sie übersichtlich aus? \*

1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	----

unübersichtlich

übersichtlich

6. Wie beurteilen Sie die allgemeine Ästhetik der Anwendung - Light Mode? \*

1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	----

nicht ästhetisch

ästhetisch

7. Wie beurteilen Sie die allgemeine Ästhetik der Anwendung - Dark Mode? \*

1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	----

nicht ästhetisch

ästhetisch

## Literatur

8. Welche Art der Benutzeroberfläche würden Sie hauptsächlich nutzen? \*

- ☐ Web-Anwendung
- ☐ mobile App
- ☐ beide ungefähr gleich oft

9. Entspricht die Anwendung allen Ihren Anforderungen? \*

10. Welche Informationen/Funktionen wären noch nützlich, um das Tauschen der Schichten zu vereinfachen? \*

11. Was würden Sie gerne verändern? \*