



TECHNISCHE HOCHSCHULE NÜRNBERG
GEORG SIMON OHM

Fakultät Elektrotechnik Feinwerktechnik Informationstechnik

Projekt: Digital Dahoam

Gruppenprojektbericht im Studiengang Software Engineering

vorgelegt von

Daniel Reitberger, Andre Reif, Rebecca Vogler, Jan Scholz, Boas
Dünkel, Marc Jonas Roser

Betreuer: Johannes Walser

Vorgelegt am 15.04.2023

© 2023

Dieses Werk einschließlich seiner Teile ist **urheberrechtlich geschützt**. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Autors unzulässig und strafbar. Das gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen sowie die Einspeicherung und Verarbeitung in elektronischen Systemen.

Glossar

API Application Programming Interface.

BaaS Backend as a Service: Plattform, die die Bereitstellung von Backend-Funktionalität ermöglicht inklusive Datenbank, Authentifizierung, Datei-Upload, etc.

Bug Fehler in der Software.

Bugfix Behebung eines Fehlers in der Software.

CD Continuous Delivery / Continuous Deployment: Automatisches Ausrollen der neuen Funktionalität.

CI Continuous Integration: Frühes Integrieren kleiner Änderungen in den Hauptzweig (Git Branches).

Confluence Software zur Zusammenarbeit und Dokumentation.

Jira Software zum Projektmanagement.

jsx Javascript XML: Erweiterung von Javascript, die es ermöglicht, HTML-Elemente in Javascript zu definieren.

MUI Material-UI (Material Design Komponenten für React).

Node.js Javascript-Plattform, die es ermöglicht, Javascript außerhalb des Browsers auszuführen.

React Frontend-Framework basierend Node.js, das von Facebook entwickelt wird. Es ermöglicht die Entwicklung von Benutzeroberflächen für Webanwendungen.

tsx TypeScript XML: Erweiterung von TypeScript, die es ermöglicht, HTML-Elemente in TypeScript zu definieren.

UI Benutzeroberfläche, der Teil der Anwendung, der für den Nutzer sichtbar und nutzbar ist. Englisch: „User Interface“.

Usability Bedienbarkeit oder Nutzbarkeit einer Anwendung.

USP Unique Selling Point.

Wireframe Grobkonzept einer Benutzeroberfläche.

Inhaltsverzeichnis

1. Einleitung	1
2. Projekt	2
2.1. Projektbeschreibung	2
2.1.1. Ausgangssituation	2
2.1.2. Projektziel	3
2.1.3. Aufgaben	3
2.2. Team	4
3. Prozess	5
3.1. Scrum	5
3.2. Projektmanagement	5
3.3. Sprintübersicht	5
4. Analyse	6
4.1. Wettbewerbsanalyse	6
4.1.1. Ermittlung der Hauptwettbewerber	6
4.1.2. Konkurrenzprofil	7
4.1.3. Kurzdarstellung	8
4.1.4. Strategische Ausrichtung	14
4.1.5. Zusammenfassung	15
4.2. Anforderungsanalyse	16
4.3. Visuelle Grundstrukturen	16
4.3.1. Grundlagen Seitenstruktur	16
4.3.2. Grundlagen Layout	19
4.3.3. Rastersystem	19
4.3.4. Grundlagen Wireframes	21
4.3.5. Anwendung	22
4.4. Usabilityanalyse	25
5. Architektur	26
6. Technologien	27
6.1. Grundlage für Technologieentscheidung	27
6.2. TypeScript	28
6.2.1. Einsatz im Projekt	28
6.2.2. Grund für Technologieentscheidung	28
6.3. React	28
6.3.1. Allgemeines in Bezug auf die Implementierung mit React	28

6.3.2. Einsatz im Projekt	30
6.3.3. Grund für Technologieentscheidung	30
6.4. Material-UI	31
6.4.1. Einsatz im Projekt	31
6.4.2. Grund für Technologieentscheidung	31
6.5. Firebase	32
6.5.1. Firebase Authentication	32
6.5.2. Firebase Realtime Database	32
6.5.3. Firebase Cloud Firestore	32
6.5.4. Firebase Storage	33
6.5.5. Einsatz im Projekt	33
6.5.6. Grund für Technologieentscheidung	33
6.6. Github	33
6.6.1. Einsatz im Projekt	33
6.6.2. Grund für Technologieentscheidung	34
6.7. Vercel	34
6.7.1. Einsatz im Projekt	34
6.7.2. Grund für Technologieentscheidung	34
6.8. Jira	35
6.8.1. Einsatz im Projekt	35
6.8.2. Grund für Technologieentscheidung	35
7. Implementierung	36
7.1. Landing Page	36
7.2. Profil	37
7.2.1. Anmeldung & Registrierung	37
7.3. Authentifizierung	38
7.3.1. Verwendung des „Firebase-Authentication“-Services	38
7.3.2. Zugriff auf die Profilinformationen	39
7.3.3. Authentifizierungsprozess	40
7.3.4. Persönliches Accountmanagement	42
7.3.5. Profilseite & Profilbilder	42
7.3.6. Blockierte Nutzer	43
7.4. Beiträge	43
7.4.1. Beiträge erstellen	44
7.4.2. Alle Beiträge	45
7.4.3. Profilseite	46
7.4.4. Merkzettel	46
7.4.5. Marktplatz	47
7.4.6. Dashboard	47

7.5. Chat	48
7.5.1. Firebase Realtime Database für den Chat	48
7.5.2. Autorisierung der Nachrichten und Chaträume	48
7.5.3. Benutzung des Chats	49
7.5.4. Verknüpfung von Beiträgen mit dem Chatraum des Beitragautors	50
8. Testing	52
9. Fazit	53
A. Anhang	A
I. Bilder	A
II. Code	C
Abbildungsverzeichnis	E
Tabellenverzeichnis	F
Literaturverzeichnis	G

1. Einleitung

Dieser Projektbericht beschreibt die Entwicklung einer prototypischen Plattform für einen interaktiven Austausch innerhalb eines Ortes, einer Stadt oder einer Gemeinde. Er wurde im Rahmen des Masterstudiengangs *Software-Engineering* an der Technischen Hochschule Georg Simon Ohm Nürnberg erstellt.

Zu Beginn wird auf die Beschreibung des Projekts eingegangen, die Anforderungen an das Produkt werden definiert und die Aufgaben des Projekts werden aufgezeigt. Damit wird eine Grundlage gelegt, die für die weitere Arbeit im Projekt benötigt wird. Außerdem wird das Team in seinen verschiedenen Rollen vorgestellt.

Im nächsten Kapitel wird der Prozess der Entwicklung der App beschrieben. Dies geschah über eineinhalb Jahre hinweg und umfasst die Planung, die Konzeption und die Implementierung der Plattform. Mit der agilen Softwareentwicklungsmethode Scrum wurde ein iterativer Prozess durchgeführt, der die Entwicklung der Anwendung in mehreren Sprints abwickelte.

Im vierten Kapitel werden technische Aspekte von „Digital Dahoam“ beschrieben. Hierbei werden Technologien und Frameworks vorgestellt, die für die Entwicklung der Plattform verwendet wurden. Diese werden jeweils mit einem kurzen Überblick vorgestellt, der genaue Zweck bzw. Einsatz im Projekt erklärt und die Entscheidung für die Verwendung dieser Technologien wird erläutert.

Die Architektur der Plattform wird im fünften Kapitel genauer beschrieben und die einzelnen Komponenten vorgestellt. Außerdem wird die Kommunikation zwischen den Komponenten beschrieben.

Details zur Umsetzung der Website erscheinen in Kapitel 7. Hierbei wird die Umsetzung der einzelnen Funktionen der Plattform beschrieben. Zunächst wird die Umsetzung der Benutzeroberfläche beschrieben, danach werden die einzelnen Funktionen der Plattform vorgestellt. Außerdem wird durch Screenshots die Benutzeroberfläche der Plattform dargestellt.

Anschließend werden die Aktivitäten in den Sprints beschrieben. Dabei wird kurz über die einzelnen User Stories berichtet, die in den Sprints umgesetzt wurden. Außerdem wird die Arbeit im Team beschrieben und die Ergebnisse der einzelnen Sprints zusammengefasst.

Der Bericht wird abgeschlossen mit einem Fazit, in dem die Ergebnisse des Projekts zusammengefasst werden. Außerdem wird ein Ausblick auf mögliche Weiterentwicklungen der Plattform gegeben und eine Bewertung der Arbeit im Projekt abgegeben.

2. Projekt

Zu aller erst wird das Projekt vorgestellt und die Aufgabenstellung erläutert. Dabei wird auch auf die Ausgangssituation eingegangen und das Projektziel definiert. Dadurch wird der Leser in die Lage versetzt, die weitere Projektdokumentation besser zu verstehen. Anschließend wird das Team vorgestellt und die einzelnen Rollen und Aufgaben der Teammitglieder erläutert.

Die nachfolgende Projektbeschreibung stammt direkt aus dem Projektaushang und wurde nur leicht angepasst. Die Projektbeschreibung ist in der Projektarbeit nicht weiter ausführlich dargestellt, da sie nur eine kurze Einführung in das Projekt darstellt.

2.1. Projektbeschreibung

Das Projekt „Digital Home Town“ ist ein „smart city-Konzept“, welches zur digitalen Vernetzung verschiedener Generationen und Interessenten im Sozialraum dient. Lebendiger Austausch, voneinander lernen, kommunale Ressourcen effizient nutzen, als Stadt virtuell zusammenwachsen, darum geht es in diesem Projekt.

2.1.1. Ausgangssituation

Grundsätzlich geht es darum eine Plattform zu bauen, die Zielgruppen- und generationenübergreifend eine Stadt vernetzt und verschiedene Features anbietet. So können virtuelle Kursräume Informationen zur Verfügung stellen, durch Chattools Diskussionsforen eingerichtet werden und die physikalische Infrastruktur treffend verteilt werden, z. B. durch ein Buchungstool (Turnhallenbelegung, Sportheim). So können z. B. schulische Inhalte auch anderen Generationen zur Verfügung gestellt werden und umgekehrt zeithistorische Informationen für Schüler dargeboten werden sowie kommunale Ressourcen optimal genutzt und ausgelastet werden. Ferner entsteht ein digitaler Marktplatz für lokale Betriebe, um die Wirtschaftskraft in der Region zu stärken. Diese Beispiele stellen nur exemplarische Ansätze und Features dar. Denn die Plattform wächst in einer organischen Evolution mit den Anforderungen der nutzenden Gesellschaft.

Zwei besondere Anforderungen bringt dieses Projekt mit sich:

- Das fertige Produkt muss von Beginn an generationenübergreifend eine hohe Userakzeptanz erlangen, indem die Zugänglichkeit und Bedienbarkeit leicht, schnell und barrierefrei gewährleistet ist.
- Um ein solches Projekt langfristig zu realisieren, müssen Schnittstellen modular mitgedacht werden, damit auch sich verändernde Bedarfsstrukturen berücksichtigt werden können.

2.1.2. Projektziel

Ziel ist es in diesem Projekt eine prototypische Plattform zu entwickeln, die in ihrer Multi-funktionalität verschiedenste Anforderungen für die Nutzer erfüllt. Ob es nun ...

- ... ein Planungstool, ähnlich einem mit anderen Usern geteilten Kalender,
- ... ein Chat oder Blog-Tool zur virtuellen Interaktion und diskursiven Auseinandersetzung ist,
- ... virtuelle Kursräume und Datenarchive sind, die generationenübergreifendes Lernen ermöglichen,
- ... eine Tauschbörse oder einfach nur
- ... eine originäre Homepage ist,

mit allen Features wird das Zusammenleben in einer Kommune leichter und dem aktuellen Stand der Technik gerecht. Deshalb sollen diese sämtlich in diese eine Plattform einbezogen werden.

2.1.3. Aufgaben

Die zu bearbeitenden Aufgaben umfassen folgende Punkte:

- Einbindung des Kunden und Abstimmung zu spezifischen Anforderungen
- Anforderungsanalyse der zu priorisierenden Features
- Konzeption einer multimedialen Plattform
- Entwicklung eines Rechte- und Rollenkonzepts
- Entwicklung einzelner Features
- Umsetzung der Spezifikationen im Sinne der Entwicklung eines lauffähigen Prototyps.
- Konsequente Berücksichtigung der Usability für alle Adressaten
- Test der lauffähigen Funktionalitäten.

2.2. Team

Im Rahmen des Projekts „Digital Home Town“ arbeiteten folgende Personen zusammen:

- **Daniel Reitberger** (Scrum Master)
- **Andre Reif** (Product Owner, Entwickler)
- **Rebecca Vogler** (Product Owner, Tester)
- **Boas Dünkel** (Entwickler)
- **Jan Scholz** (Entwickler)
- **Jonas Roser** (Entwickler)

Dieses Team hat sich selbstständig für das Thema „Smart City“ entschieden und wird von Herrn Johannes Walser betreut. Dieser ist der Ansprechpartner für alle Fragen und Anliegen, die sich im Projekt ergeben. Dazu war er der „Kunde“ des Projekts, also derjenige, der die Anforderungen an das Produkt definiert hat.

Alle Teammitglieder haben sich in der Projektarbeit gegenseitig unterstützt und sich gegenseitig weiterentwickelt.

3. Prozess

3.1. Scrum

3.2. Projektmanagement

3.3. Sprintübersicht

4. Analyse

4.1. Wettbewerbsanalyse

Das Ermitteln und Priorisieren aller relevanten Kundenanforderungen ist ein kontinuierlicher und unerlässlicher Vorgang, um zum einen die Bedürfnisse der zukünftigen Nutzer erfüllen zu können und zum anderen, um das Produkt langfristig wettbewerbsfähig zu gestalten. Hierfür ist es essenziell, sich mit Wettbewerbern, ihren Produkten und vermeintlichen Strategien auseinandersetzen. Erst dies bildet die Grundlage die eigenen Chancen auf dem Markt zu bewerten und mögliche Alleinstellungsmerkmale (engl. unique selling point, USP) zu ermitteln, um sich von Wettbewerbern frühzeitig abzusetzen. Zusätzlich zeigt die Analyse, welche Features bereits auf dem Markt im Einsatz sind und vom User ggf. schon als grundlegende Features erwartet werden.

Die Durchführung der Wettbewerbsanalyse erfolgte in den folgenden Einzelschritten:

- Ermittlung der Hauptwettbewerber
- Erstellung eines Konkurrenzprofils
- Darstellung der einzelnen Wettbewerber
- Bewertung der strategischen Ausrichtung
- Zusammenfassung

4.1.1. Ermittlung der Hauptwettbewerber

Für die Ermittlung der bedeutendsten Mitbewerber ist es erforderlich die Hauptmerkmale und Ziele des eigenen Dienstes, für einen Abgleich mit den potenziellen Wettbewerbern, gut zu kennen. Hierbei ist zu berücksichtigen, dass es in der digitalen Welt selten Wettbewerber gibt, die 1:1 den gleichen Dienst anbieten bzw. die gleiche Strategie verfolgen. Jedoch können hierbei mittelfristig vollwertige Konkurrenzprodukte entstehen, die zum Zeitpunkt der Analyse nur wenige gemeinsame Parallelen oder Features aufgewiesen haben.

Digital Home Town (DHT) setzt in seiner langfristigen Ausrichtung auf die Stärkung des lokalen/ regionalen Raums mit seinen sozialen Verflechtungen. Dies soll geschehen durch die Förderung des digitalen Austauschs zwischen allen Menschen und Institutionen, die das örtliche Dasein widerspiegeln. Die Schwerpunkte von DHT sind wie folgt:

1. Das Vernetzen von Personen, die das gleiche Interesse oder ähnliche Bedürfnisse teilen, einen Austausch von bspw. Informationen, Waren usw. anstreben oder Unterstützung/ Hilfestellung suchen.

2. Die Stärkung lokaler Vereine und/ oder Dienstleistungen, damit diese ihre Angebote gezielter darstellen und leichter mit Interessierten in Kontakt treten können.
3. Die Unterstützung öffentlicher Einrichtungen, um einen leichteren Zugang u. a. zu amtlichen Informationen, Terminen usw. zu ermöglichen.

Zugleich sieht sich DHT allen Personen verpflichtet und strebt einen barrierefreien Zugang an.

Abgeleitet aus der Zielsetzung von DHT konnten eine Reihe von Web-Diensten ermittelt werden, die sich u. a. auf den lokalen Raum in Bezug auf Austausch, Vernetzung oder Zusammenführung privater Parteien konzentrieren. Als besonders etablierte und aktive Vertreter in mindestens einem der genannten Bereiche konnten zunächst

- Facebook,
- Nebenan.de,
- Spontacts,
- eBay Kleinanzeigen und
- Doctolib

ausfindig gemacht werden. Im weiteren Verlauf werden exemplarisch nur die Social-Networking-Vertreter weiter ausgeführt. Dies betrifft demnach nicht Doctolib, da es sich eher um eine spezialisierte Software für E-Health-Angebote handelt.

4.1.2. Konkurrenzprofil

Zur detaillierten Auseinandersetzung mit den ermittelten Wettbewerbern ist es erforderlich ein Profil jedes einzelnen zu erstellen. Dieses sogenannte Konkurrenzprofil gibt in der Gegenüberstellung mit dem eigenen Selbstprofil eine transparente Bewertungsbasis, mit dessen Hilfe wertvolle Erkenntnisse über die Stärken, Potenziale und strategischen Ziele abgeleitet werden können.

Für den Vergleich werden die folgenden Punkte gesondert betrachtet:

- Fokus des Dienstes und dessen Zielgruppe,
- Möglichkeiten der Selbstdarstellung und des Austausches,
- Benachrichtigungen über Änderungen und Neuigkeiten und
- Entdecken von Inhalten.

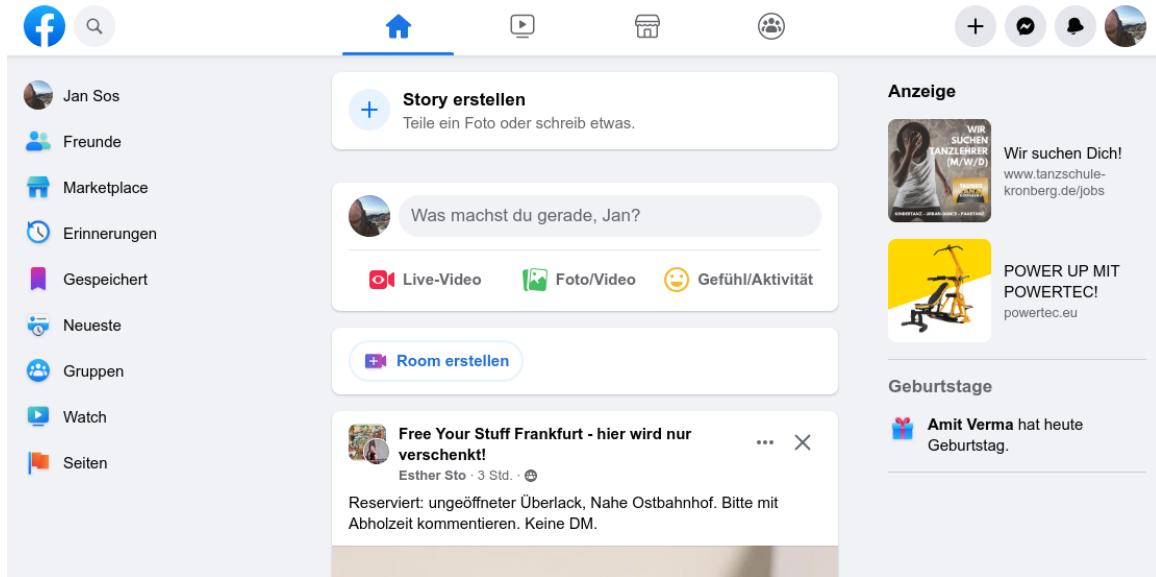


Abbildung (4.1) – Startseite von Facebook

4.1.3. Kurzdarstellung

4.1.3.1. Facebook

Facebook ist ein etablierter und weltweit verbreiteter Dienst, der es ermöglicht sich mit Menschen aus der ganzen Welt zu vernetzen, miteinander in Kontakt zu bleiben und Inhalte aus dem eigenen Leben mit anderen zu teilen. In den frühen Jahren von Facebook waren insbesondere Studenten die anvisierte Zielgruppe, welche über die Jahre jedoch breiter geworden und mittlerweile sehr heterogen ist.

Die Nutzung von Facebook kann je nach Bedürfnissen des Einzelnen sehr verschieden sein. Hier bietet Facebook neben reinen Darstellungs- und Austauschmöglichkeiten eine Bandbreite an Anwendungen an, wie u. a. integrierte Video-, Games- und Dating-Portale und ein eigenes Bezahlungssystem, auf welche hier im weiteren Verlauf nicht weiter eingegangen wird.

Selbstdarstellung Zur Darstellung einer Organisation, einer öffentlichen oder privaten Person, bietet Facebook je nach Typ ein eigenes Profil an, auf welchem u. a. Bilder, persönliche Informationen und Vorlieben, wie z. B. Musik öffentlich dargestellt werden können. Inwieweit diese Inhalte einsehbar sind, ist individuell einstellbar. Zusätzlich zu den statischen Informationen beinhaltet das Profil noch eine Pinnwand, auf welcher ältere und aktuelle Beiträge listenmäßig aufgeführt sind.

Formen des Austausches Für den interaktiven Austausch und das Teilen von Momenten dienen Beiträge (Posts). Diese können von jedem verfasst und auf der eigenen sowie fremden Pinnwand angebracht werden. Diverse Ereignisse, wie bspw. das Hochladen von

neuen Bildern, generieren systematisch einen eigenen Beitrag, um Bekannte, Follower etc. über eine Neuigkeit auf dem Profil zu informieren. Die Beiträge können wiederum von Anderen kommentiert oder mittels eines Icons bewertet werden. Zudem können sie neben einfachem Text auch Bilder oder Videos enthalten und lassen sich mittels Tags (Orte, GPS, Veranstaltungen, Personen, ...) genauer spezifizieren.

Für den Austausch mit mehreren Personen bietet Facebook Gruppen an. Dort können sich Anwender über gemeinsame Themen austauschen. Gruppen haben wie Profile einen statischen Teil, in dem die Admins im Freitext eine Kurzbeschreibung der Gruppe verfassen und veröffentlichen können. Darüber hinaus verfügen Gruppen auch über eine Pinnwand, über die die Mitglieder miteinander im Austausch stehen. Gruppenmitglieder können darüber hinaus auch Events erstellen, an welchen die Mitglieder teilnehmen können. Gruppen werden sehr häufig und für verschiedene Themen verwendet. Sie dienen besonders im städtischen Raum bspw. zum Verschenken von ungenutzten Dingen (Free your Stuff), zum Finden von neuen Kontakten in einer neuen Stadt (New in Munich) oder das Finden von Menschen in der Nähe mit ähnlichen Hobbies (z. B. Wandern). Auch der überregionale/ internationale Austausch über diverse Themen in unterschiedlichen Bereichen ist weit verbreitet.

Der Austausch materieller Gegenstände findet neben den einzelnen Gruppen hauptsächlich auf einem digitalen Marktplatz statt. Die geschalteten Anzeigen enthalten eine Überschrift, freie Beschreibungen, Bilder, Angaben über den Zustand, Preisvorstellung, Ort und den Herausgeber der Anzeige. Bei der Erstellung dieser muss der Verfasser die Anzeige einer vordefinierten Kategorie zuweisen, um die Auffindbarkeit zu erleichtern. Für den Interessenten besteht neben der direkten Kontaktaufnahme die Möglichkeit sich die Anzeige zu merken, sie mit einem Kontakt zu teilen oder einen Alarm zu erstellen, sobald ähnliche Produkte angeboten werden.

Zur Pflege von Kontakten ist es in Facebook möglich, Personen bei gegenseitigem Einverständnis in eine Kontaktliste aufzunehmen. Mithilfe dieser sogenannten Freundesliste können Inhalte selektiv verteilt werden. Für Personen, die nicht in der Kontaktliste enthalten sind, sind diese Inhalte nicht sichtbar. Darüber hinaus werden befreundete Personen über das Dashboard gezielter informiert, sobald ein Kontakt aus der Liste eine Aktivität durchgeführt hat.

Für den direkten und nicht öffentlichen Austausch wird dem Anwender ein Chat angeboten, in welchem 1:1- und Gruppenchats stattfinden können. Der Nachrichtenaustausch findet hierbei in Echtzeit statt. Die Nachrichten können analog, wie Beiträge, die gleichen Inhalte aufweisen (Bilder, Links etc.). In 1:1-Gesprächen ist es darüber hinaus auch möglich, Sprach- und Videoanrufe zu tätigen.

Neuigkeiten Um bei der Vielzahl der Beiträge und Reaktionen von Freunden oder in Gruppen den Überblick zu behalten, werden auf einem Dashboard Neuigkeiten in Form

einer endlosen und unsortierten Liste angezeigt. Darüber hinaus beinhaltet das Dashboard kommerzielle Werbung und Beiträge von noch nicht abonnierten Gruppen, die für den Nutzer von Interesse sein könnten.

Neben dem Dashboard gibt es zusätzlich noch eine Notifications-Seite, welche einen gezielten Fokus aufs Wesentliche ermöglicht. Notifications sind kurze Benachrichtigungen, die gesammelt in einer Liste abgelegt werden. Sie beinhalten u. a. Geburtstage der Kontakte, bevorstehende angemeldete Events, neue Beiträge aus abonnierten Gruppen sowie Reaktionen auf eigene oder kommentierte Beiträge.

Recherche Zur Findung von Personen, Gruppen usw. existiert eine Suchfunktion, mit deren Hilfe alle FB-Inhalte durchsucht werden können. Mit dem Suchfilter können auch die Art der Inhalte, Verfasser, Gruppen, Zeitraum u. v. m. genauer spezifiziert werden.

Das Sichern von Beiträgen und öffentlichen Profilen kann ebenso vorgenommen werden. Das Speichern erfolgt direkt am Objekt. Gespeicherte Beiträge und öffentliche Profile können im Anschluss auf einer separaten Seite je nach Typ aufgelistet und dort auch verwaltet werden.

4.1.3.2. Nebenan.de

Nebenan.de ist ein im Jahr 2015 in Betrieb genommenes Portal, welches wie Facebook seinen Mehrwert im Vernetzen von Menschen sieht. Anders als Facebook fokussiert es sich dabei ausschließlich auf die unmittelbare Nachbarschaft des Anwenders.

Die Zielgruppe besteht daher aus allen in der Nachbarschaft vorkommenden Interessengruppen, wie z. B. Bewohner, Vereine, Firmen und Organisationen. Die Nachbarschaftsgrenze wird durch die eigene Adresse vom System definiert und kann nicht individuell angepasst werden. Ein Austausch über die Nachbarschaftsgrenze hinaus ist nicht möglich, was den Fokus auf die Stärkung und Belebung der Nachbarschaft deutlich hervorhebt.

Selbstdarstellung Für die Darstellung existieren in Nebenan.de zwei Arten von Profilen. Die erste Profilart gilt Einzelnutzern. Hierbei können sich Anwohner über ein Kurzprofil mit Foto, einem Freitext sowie ihren Interessen und Angeboten vorstellen. Zur Auswahl der Interessen und Angebote stehen fest definierte Vorschläge zur Verfügung. Angebote beschreiben neben den Interessen, was der Anwohner bereit ist, in seine Nachbarschaft einzubringen. Dies könnte bspw. Pakete annehmen, Blumen gießen, Fahrrad reparieren oder Gesellschaft leisten sein. Neben der eigenen Darstellung zeigt das Profil die Aktivität des Anwohners an. Dies beinhaltet bspw. die Auflistung von beigetretenen nebenan-Gruppen und die Anzahl der erhaltenen virtuellen Dankeschöns. Die zweite Profilart wird von Organisationen und Läden/ Services genutzt, die vor Ort angesiedelt sind und zum alltäglichen Leben in der Nachbarschaft beitragen. Diese Profile sind im Vergleich zu den Anwohnerprofilen mit deutlich

The screenshot shows the homepage of nebenan.de. At the top, there's a navigation bar with the logo 'nebenan.de', a 'nebenan.de fördern' button, user profile icons for 'jan', and a search bar. A sidebar on the right lists various sections: Startseite (highlighted in green), Veranstaltungen, Marktplatz, Läden & Services, Gruppen, Postfach, Deine Nachbarn, Deine Nachbarschaft, and Nachbarn einladen. Below the sidebar, there are links for Ausloggen, Organisationen, Hilfe, Magazin, Team, Presse, Kontakt, Impressum, Datenschutz, and Nutzungsbedingungen.

Bornheim Ost & Umgebung
Aktive Nachbarn: 5226

Suche • Dinge

Wer hat Guiness-Gläser? 😊
von Vera C. aus Bornheim Süd vor 16 Stunden

Für eine Party Ende März suche ich Guiness-Gläser. Hat jemand welche, die ich ausleihen dürfte? Würde mich seeehr freuen! (bearbeitet: vor 3 Stunden)

Abbildung (4.2) – Startseite von nebenan.de

mehr Informationen ausgestattet. Jedes dieser Profile muss beim Anlegen mit einer Kategorie versehen werden. Dies kann im Beispiel eines Geschäfts z. B. Restaurant, Reisen oder Sport und im Fall einer Organisation Non-Profit, Political Party, Nachbarschafts-Initiative usw. sein. Neben Name, Adresse, Kontaktdaten und Öffnungszeiten können auch zukünftige Ereignisse/ Events, Gesuche, Bekanntmachungen, das Verkaufsangebot und weitere Informationen hinterlegt werden. Die Anwohner können darüber hinaus ihre eigenen Erfahrungen mit der Organisation oder dem Laden sowie Empfehlungen auf deren öffentlichen Profilen kundtun.

Formen des Austausches Als zentrales Mittel für den Austausch wird bei nebenan.de der Beitrag genutzt. Dieser kann nach dem Veröffentlichen von allen Anwohnern kommentiert und positiv bewertet werden. Ein Beitrag wird bei der Erstellung mit einer Kategorie versehen. Anhand dieser wird festgelegt, ob es sich bei dem Beitrag um etwas Allgemeines, ein Gesuch, ein Angebot, eine Empfehlung oder ein Event handelt. Jede Kategorie wird des Weiteren zur besseren Einordnung noch in eine oder mehrere aufeinander aufbauende Unterkategorien unterteilt. Je nach Art werden die Beiträge neben der öffentlichen Wand zusätzlich an diverse Seiten wie z. B. den Event-Feed im Marktplatz der Nachbarschaft angeheftet. Die Beiträge bestehen wie bei Facebook klassisch aus einem Titel und einem Freitext und können optional mit Bildern versehen werden. Zusätzlich können je nach Art noch weitere Felder wie z. B. Datum und Ort bei Veranstaltungen vorhanden sein.

Die Beiträge des Marktplatzes werden neben der Kategorie Angebot noch in die Unterkategorien Help, Give, Rent, Trade usw. unterteilt. Der Inhalt eines Verkaufsangebots kann des Weiteren noch mit einer Angebotskategorie, wie Food, Baby & Children, Pets usw. , versehen werden. Alle Angebote werden gesammelt in einer Liste dargestellt und können nach ihrer Angebotskategorie gefiltert oder mit der globalen nebenan.de-Suchfunktion nach Wörtern durchsucht werden.

Für den Austausch von Personen mit gleichen Interessen können in nebenan.de Gruppen erstellt werden, deren Zweck durch einen Gruppennamen und einen Freitext genauer ausgeführt werden kann. In den Gruppen können verschiedene Arten von Beiträgen (Mitteilung, Suche, Angebot, Veranstaltung, ...) abgesetzt werden. Diese werden an der Gruppenpinnwand angezeigt und können von allen Gruppenmitglieder kommentiert werden. Die Gruppenfunktion findet jedoch auf nebenan.de nur eine geringe Beachtung.

Eine Chatfunktion für den direkten Austausch mit einzelnen Nutzern ist ebenso integriert. Neben dem reinen Text und Emoticons können mit ihr zusätzlich noch Fotos und Empfehlungen versandt werden.

Neuigkeiten Zur Benachrichtigung von neuen Beiträgen gibt es das Dashboard. Hier werden alle Arten von Beiträgen mit Ausnahme der Angebotsbeiträge (Give) nach der letzten Änderung sortiert angezeigt. Der Austausch mit der Nachbarschaft passiert maßgeblich über das Dashboard, wo reinkommende Beiträge entdeckt und beantwortet werden können. Für den besseren Überblick werden durch Benachrichtigungen alle Beiträge, bei denen man selbst aktiv teilgenommen hat und sich neue Ereignisse ergeben haben, nochmal separat aufgelistet.. Darüber hinaus werden auf dem Benachrichtigungs-Feed alle Events angezeigt, die zukünftig in der Nachbarschaft stattfinden.

Recherche Für die Suche von Beiträgen jeglicher Art steht die nebenan-Suche zur Verfügung. Über diese können anhand eines Freitextes die Beiträge der Nachbarschaft durchforscht werden. Beiträge, die dem User besonders zusagen oder für ihn wichtige Informationen enthalten, können als Bookmark gespeichert werden und auf der Bookmark-Seite unter Feed oder Marktplatz eingruppiert eingesehen und wieder gelöscht werden.

4.1.3.3. Spontacts

Gegenüber Facebook und Nebenan.de legt Spontacts sein Augenmerk ausschließlich auf das Verabreden/ Treffen für Freizeitaktivitäten von Menschen, die in der gleichen Region wohnen. Spontacts richtet sich an Menschen, die Interesse an bestimmten Aktivitäten haben und dazu neue Kontakte suchen.

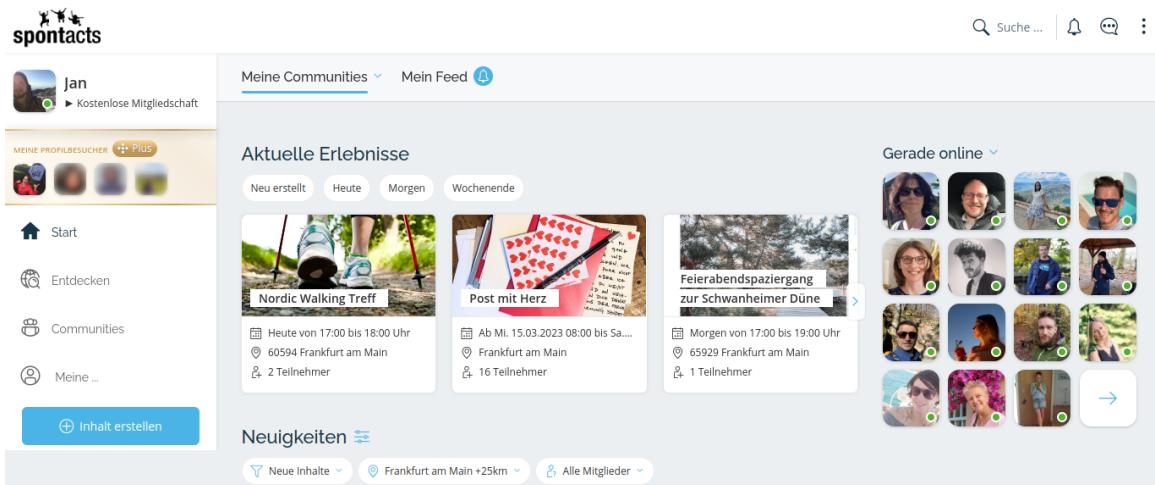


Abbildung (4.3) – Startseite von Spontacts

Der Anwender kann hierfür an erstellten Veranstaltungen teilnehmen oder selbst welche erstellen. Im Gegensatz zu den bereits vorgestellten Diensten besitzt Spontacts einen kostenlosen sowie kostenpflichtigen Account. Ein weiterer Unterschied liegt darin, dass regelmäßig Angebote für die Nutzer von bei Spontacts Angestellten bereitgestellt werden und es sich dadurch nicht um reinen User-Content handelt.

Selbstdarstellung Die Benutzer von Spontacts können, wie auch in vergleichbaren Anwendungen, zur Präsentation ihrer Person ein Profil erstellen. Das Grundprofil setzt sich aus Name, Profilbild, Alter und Wohnort zusammen und kann mit weiteren Informationen wie bspw. präferierte Kontakte - für Freizeit, Sport, Reisen, Tanzen und Dating - oder mit zusätzlichen Profilabschnitten (Hobbies, Interessen/ Vorlieben usw.) angereichert werden. In einem weiteren Abschnitt können vergangene und zukünftige Aktivitäten des Users sowie seine Gruppenmitgliedschaften eingesehen werden.

Formen des Austausches Zur Interaktion mit anderen Benutzern dienen Beiträge. Diese Beiträge werden je nach Art als einfacher Beitrag, als Aktivität oder als Frage/ Diskussion abgesetzt und anschließend in Communities, Gruppen oder ggf. in Foren angezeigt und können wie gewohnt kommentiert, geliked und geteilt werden.

Die Communities beschreiben als eine Obergruppe einen Themenbereich (Ausgehen & Party, Essen & Trinken, Natur & Umwelt usw.), zu welchem der Beitrag zugeordnet wird. Beim Erstellen einer Gruppe muss hierfür immer eine passende Community ausgewählt werden. Anders als bei den anderen vorgestellten Diensten, stellen die Gruppen auf Spontacts das zentrale Feature der Anwendung dar. Diese ähneln vom Funktionsumfang den FB-Gruppen und sind lediglich in ihrem Aufbau anders strukturiert. In ihnen kann sich der Anwender über das Gruppenprofil mit Kurzbeschreibung, Mitgliederanzahl usw. informieren, über eine

Pinnwand alle Beiträge der Mitglieder durchstöbern sowie die geplanten Veranstaltungen der Gruppe einsehen.

Für Auseinandersetzungen über die Beiträge hinaus kann mit jedem Anwender eine private Unterhaltung über den Chat geführt werden. Mit dem Chat können Textnachrichten ohne Emoticons, Bilder etc. versendet werden.

Recherche Zum Entdecken von Beiträgen, Veranstaltungen, Gruppen, Mitgliedern u. v. m. bietet eine Suchfunktion je nach zu suchendem Typ weitere Filterfunktionen. Der Anwender kann neben den allgemeinen Einstellungen wie Suchbegriff, Umkreis, Erstelldatum und Community die Anfrage noch mit objektspezifischen Kriterien wie Datum, Alter und ähnliches verfeinern.

Die während der Nutzung des Portals oder während eines Treffens entdeckten Mitglieder können in die Kontaktliste oder Merkliste aufgenommen werden. Die Merkliste ist eine private Liste von Personen, die der Anwender als merkenswert empfand. Die Kontaktliste ist hingegen öffentlich und zur Aufnahme muss eine Anfrage an die Person versendet und bestätigt werden. Darüber hinaus kann man Mitgliedern folgen, um ihre Aktivitäten besser verfolgen zu können. Wer wem folgt kann jeder Benutzer im jeweiligen Profil einsehen.

Neigkeiten Um innerhalb der Plattform auf dem Laufenden zu bleiben, gibt es wie bei den anderen Portalen ein Dashboard und einen Benachrichtigungs-Feed. Im Dashboard werden in der Rubrik „Neigkeiten“ alle anstehenden Veranstaltungen und aktuellen Inhalten aus den gewählten Communities dargestellt. Unter der Rubrik „Mein Feed“ im Dashboard können hingegen alle Veränderungen in beigetretenen Gruppen eingesehen werden. Hierzu zählen insbesondere neue Mitglieder und neue Veranstaltungen. Der Benachrichtigungs-Feed enthält neue Kontaktanfragen, Beiträge aus den Gruppen und Plattform News.

4.1.4. Strategische Ausrichtung

Die drei untersuchten Hauptwettbewerber weisen als soziale Netzwerke insbesondere in ihren Features, wie das Vorhandensein von Profil, Chat, Gruppen usw., viele Gemeinsamkeiten auf. In ihrer strategischen Ausrichtung unterscheiden sie sich jedoch teils stark voneinander. Facebook ist eine sehr aktive Plattform. Sie unterstützt ganz allgemein das Vernetzen von Menschen mit beliebigen Interessen, Themen und Bedürfnissen. Es fokussiert sich dabei nicht auf einen geografischen Raum oder schränkt die Anwender in irgendeiner Form ein, die einen Austausch verhindern würde. Ein Ausbau oder eine intensive Weiterentwicklung an der Plattform konnte in der vergangenen Zeit nicht wahrgenommen werden bzw. dieser fand nur im Hintergrund statt. Lang- und mittelfristig ist hierbei keine Veränderung zu erwarten, da sich der Konzern hinter Facebook maßgeblich der Entwicklung von zukünftigen Produkten im 3D-Umfeld gewidmet hat.

Nebenan.de ist in seiner bisherigen Ausrichtung der größte Konkurrent für DHT. Die Plattform bietet die Möglichkeit sich innerhalb der eigenen Nachbarschaft zu verschiedenen Themen auszutauschen und zu vernetzen. Der Fokus liegt hierbei auf der Stärkung des Miteinanders in einem lokalen Raum. Die große Herausforderung, die vor Nebenan.de liegt, ist maßgeblich der Aufbau von lokalen Communities, in welchen sich alle gesellschaftlichen Schichten und Altersklassen angesprochen fühlen. Ein weiteres Vorhaben könnte zugleich auch der Ausbau der Plattform sein, um zum einen diese intuitiver zu gestalten sowie zum anderen einen zielgerichteteren Austausch zu ermöglichen.

Spontacts fokussiert sich hingegen sehr stark auf das Vernetzen von Menschen zur Durchführung gemeinsamer Aktivitäten im privaten Bereich. Die Plattform hat eine rege Community. Die Anwender sind in der Lage beliebige Inhalte zu erstellen und an allen Veranstaltungen innerhalb Spontacts teilzunehmen. Harte lokale Einschränkungen bzgl. der Sichtbarkeit von Inhalten existieren wie auch bei nebenan.de nicht. Die Inhalte der Seite werden darüber hinaus noch durch lokal ansässige Moderatoren betreut. Die Moderatoren erstellen ergänzend weitere Veranstaltungsangebote und führen diese zugleich auch durch. Strategisch versucht Spontacts zunehmend kostenpflichtige Accounts einzuführen, über welche zusätzliche Funktionen, wie bspw. bessere Filterfunktionen, den Anwendern zur Verfügung stehen. Ein weiterer Schritt, um sich weiter auf dem Markt zu etablieren, wäre es die bisherige Zielgruppe - alleinstehend und ca. 30-50 Jahre - zunehmend weiter anzusprechen.

4.1.5. Zusammenfassung

Mithilfe der durchgeführten Analyse der Wettbewerber konnte verdeutlicht werden, dass die untersuchten Dienste alle verschiedene Motivationen sowie Zielgruppen verfolgen. Zugleich verfolgt keiner den Ansatz sich, wie DHT, gezielt auf die Region/ Gemeinde zu konzentrieren, um dort die Vernetzung und den Austausch zu fördern. DHT schließt hierbei die Lücke, die sich zwischen FB als weltweiter Akteur und nebenan.de mit reinem Nachbarschaftsfokus auftut. Absetzen tut sich DHT neben dem örtlichen Bezug auch durch die Einbindung von Vereinen und öffentlichen Organen, die die Plattform zur Kommunikation mit den Anwohnern nutzen können.

Ein weiterer Aspekt, der im direkten Vergleich deutlich wird, ist, dass die Bedienung und das Zurechtfinden auf den Plattformen sehr unterschiedlich ausfallen. Teils sind die Inhalte sehr versteckt, teils sind die zugrunde liegenden Konzepte nicht selbsterklärend gestaltet. Durch den Anspruch von DHT für alle von jung bis alt (generationsübergreifend) verständlich zu sein, ist es essenziell für den Erfolg von DHT, eine Plattform zu entwerfen, die in der Handhabung einfach und intuitiv ist.

Die Analyse zeigt am Beispiel von nebenan.de und Spontacts auch, dass der Aufbau und das Etablieren eines sozialen Netzwerkes viele Herausforderungen mit sich bringen. Als Vorreiter ist Facebook sehr weit in der Gesellschaft verbreitet und ermöglicht durch sein breites

Angebot an Features diese für verschiedene Bedürfnisse zu nutzen. Dies stellt insbesondere neuere Plattformen - wie Nebenan.de und Spontacts - im Aufbau einer Community vor große Herausforderungen, da sie als Newcomer ohne Community und Content mit meist den gleichen FB-Features den Anwender überzeugen müssen, sodass ihre Plattform einen spürbaren Mehrwert gegenüber FB liefert. Dies ist besonders bei Plattformen schwierig, bei denen die Inhalte ausschließlich von den Nutzern erstellt werden. Der Aufbau einer solchen Community ist als besonders zäh anzusehen, da neue Nutzer sich aus Mangel an aktuellen Inhalten schnell wieder von der Plattform abwenden. Das kontinuierliche Vorhandensein von neuen Inhalten ist ein essenzieller Bestandteil, um ein Etablieren zu ermöglichen. Spontacts setzt hierfür gezielt seine Moderatoren ein, um ein regelmäßiges Angebot für seine Nutzer zur Verfügung zu stellen. Alternativ kann die Bindung an ein Portal auch durch Angebote von Dritten erfolgen. Dies kann bspw. durch regelmäßige Informationen von Vereinen, einer Stadtverwaltung oder einer Zeitung stattfinden, wie für DHT angedacht, oder durch zusätzliche Funktionen, die den Nutzer bei bestimmten Dingen unterstützen, wie z. B. ein Buchungsportal oder Ähnliches.

4.2. Anforderungsanalyse

4.3. Visuelle Grundstrukturen

Ein wichtiger Faktor für den Erfolg des Scrum-Teams ist es, dass jeder im Team das gleiche Verständnis über das zu entwickelnde Produkt hat. Die Sicherstellung dessen ist von hoher Bedeutung, um vermeidbaren Zeitfressern, wie lange Diskussionen, Missverständnisse, Nacharbeit und eine ggf. daraus hervorgehende Demotivation im Team entgegenzuwirken. Darüber hinaus entstehen durch eine Visualisierung des Produktes Ideen und Lösungsansätze für Abläufe und Features, die frühzeitig im Entwicklungsprozess angesprochen, diskutiert und eingearbeitet werden können, ohne einen bemerkbaren Mehraufwand zu generieren. Wireframes eignen sich für diese Aufgabe sehr gut, da sich diese schnell erstellen und abändern lassen und zugleich jedem Teammitglied einen ersten Entwurf des Produktes aufzeigen. Ein weiterer Vorteil besteht darin, dass bereits frühzeitig dem Kunden auf einfache Weise die ersten Designkonzepte vorgestellt werden können und dieser sein Feedback in einer frühen Entwicklungsphase mit einfließen lassen kann.

Das Erstellen von Wireframes beinhaltet zugleich, sich frühzeitig mit dem Aufbau und der Konzeption der Website auseinanderzusetzen. Aus diesem Grunde wird im weiteren Verlauf auch auf die Themen Seitenstruktur und Layout eingegangen.

4.3.1. Grundlagen Seitenstruktur

Die Seitenstruktur einer Website beschreibt die Art und Weise, wie die Inhalte einer Anwendung präsentiert, organisiert und verlinkt sind.

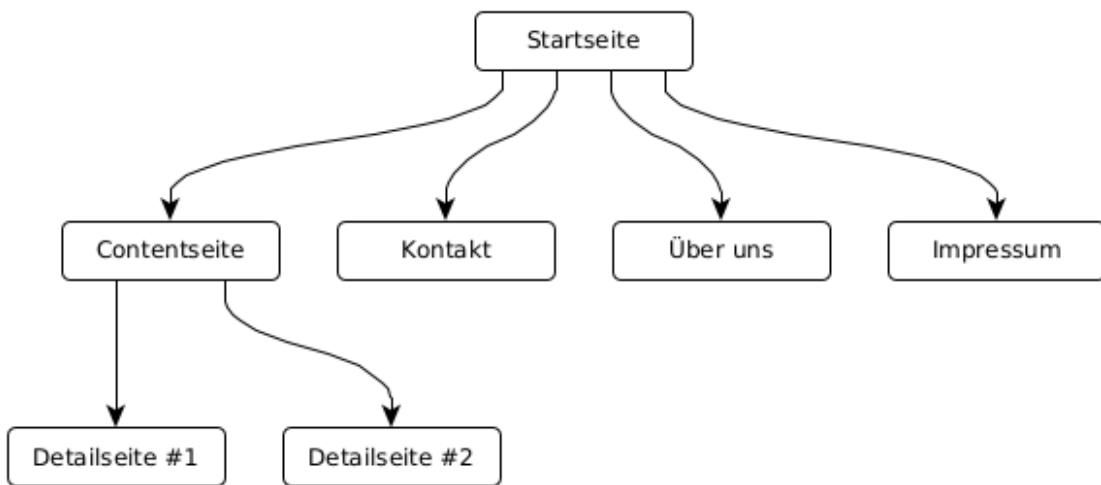


Abbildung (4.4) – Seitenstruktur einer Website

Die darzustellenden Informationen müssen daher nach Inhalt aufgeteilt und thematisch auf einzelnen Seiten organisiert werden. Jede Seite der Website besitzt somit ein klares Ziel, worüber sie informieren soll. Die einzelnen Seiten unterscheiden sich daher bzgl. ihres Ziels und Inhaltes. Jedoch existieren auf jeder Website eine Reihe von sogenannten Kernseiten, die oft aufzufinden sind. Klassische Seitentypen sind bspw. Startseite, Kontaktseite, Landingpage, Content- sowie Detailseiten.

Die Startseite, auch als Homepage bekannt, beschreibt die Seite, die den Startpunkt bzw. Ursprung einer Website darstellt. Von ihr aus kann über Verlinkungen zu allen Unterseiten navigiert werden. Die Landingpage hingegen ist die erste Seite, die ein User wahrnimmt, wenn er von einem externen Link auf die Website geführt wird und eine Session beginnt. Die Landingpage wird oft für das Marketing hergeholt, um den Besucher für den Dienst zu begeistern und weitere Schritte anzuregen. Die Contentseite hingegen gibt einen Überblick über die angebotenen Inhalte und verweist auf die zugehörige Detailseite, auf der die Inhalte detailliert aufgeführt sind.

Wie aus der Beschreibung der Seitentypen hervorgeht, werden die einzelnen Seiten zu unterschiedlichen Zeitpunkten oder in einer bestimmten Reihenfolge aufgerufen. Dies lässt sich auch als eine Art Hierarchie verstehen, die die einzelnen Seiten zueinander haben. Ob es sich hierbei um eine flache oder tiefe Hierarchie handelt, ist von der Navigationsstruktur abhängig. Allgemein werden häufig, wegen der besseren Orientierung, flache Seitenhierarchien empfohlen.

Die einzelnen Seiten einer Website unterscheiden sich jedoch i. d. R. nicht vollständig voneinander. Sie folgen alle einem gleichbleibenden Aufbau. Dieses Grundgerüst einer Seite besteht oftmals aus einer Kopf- und Fußleiste und einem Inhaltsbereich. Die Kopf- und Fußleiste ist üblicherweise auf allen Seiten gleich ausgestaltet.

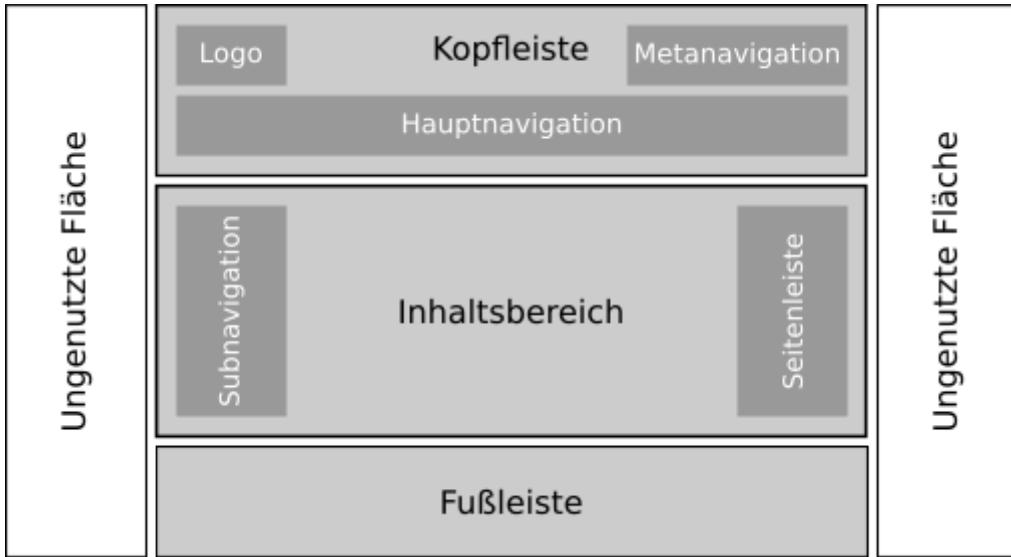


Abbildung (4.5) – Elemente einer Website

Der Kopfbereich einer Seite befindet sich im obersten Teil der Seite und umfasst das Logo, die Haupt- und Metanavigation. Der Kopfbereich wird häufig auch als Header bezeichnet. Das Logo im Kopfbereich stellt ein wichtiges Erkennungs- und Differenzierungsmerkmal der Website dar und zielt auf das Erzeugen von Assoziationen mit der Website ab. Die Hauptnavigation gibt eine Übersicht über die verfügbaren Inhalte und stellt diese strukturiert dar. Die Hauptnavigation erweist sich für die Navigation als wichtigstes Element und wird häufig auffällig im oberen Bereich platziert. Eine weitere Navigationsleiste ist die Metanavigation, welche ergänzende Serviceinhalte wie z. B. Accounteinstellungen der Seite verfügbar macht. Die Inhalte dieser Leiste haben keinen Bezug zu den Hauptthemen und werden daher gesondert aufgeführt.

Der Inhaltsbereich ist direkt unter dem Kopfbereich angeordnet und umfasst die zu vermittelnden Inhalte. Der Inhaltsbereich weist je nach Bedarf entweder nur die reinen Inhalte auf oder ggf. noch eine Subnavigation, die meist links angeordnet ist, sowie eine Seitenleiste (engl. Sidebar) für weiterführende Inhalte, die auf der rechten Seite untergebracht werden. Die zentralen Inhalte werden zwischen den Leisten, im sogenannten Inhaltsbereich, und meist nach Wichtigkeit absteigend sortiert aufgelistet.

Die untere Begrenzung der Internetseite bildet die Fußleiste (engl. Footer). Die Fußleiste beinhaltet meist Basisinformationen der Seite, ergänzende Inhalte oder auch ggf. weitere Navigationsmöglichkeiten.

Umschlossen werden die Seitenbereiche von einem umgebenden Block, der die ungenutzte Fläche der Website darstellt.



Abbildung (4.6) – Beispiel eines Rastersystems

4.3.2. Grundlagen Layout

4.3.3. Rastersystem

Für die Überführung der ersten Skizzen in ein stimmiges Layout eignet sich die zur Hilfenahme eines Rastersystems. Ein Rastersystem stellt ein Netz mit Zeilen und Spalten dar, an welchen die Inhalte ausgerichtet und letztlich im Rastersystem platziert werden. Der Vorteil besteht darin, dass die Inhaltselemente und Einzelseiten in eine gleiche Struktur gebracht werden und die Seiten zunehmend abgestimmter und einheitlicher wirken. Das Raster wird lediglich für die Gestaltung herangezogen und sollte möglichst unauffällig oder dezent für den Endnutzer wirken,

Die Ausgangsbasis eines Rastersystems ist zu meist eine Leinwand mit definierten Abmessungen. Die Fläche wird in Spalten (engl. columns) unterteilt und ggf. wird zusätzlich noch zwischen den Spalten ein gleichbleibender Freiraum (engl. gutter) angelegt. Je höher die Spaltenanzahl gewählt wird, desto größer wird der gestalterische Spielraum, wobei der Nutzen des Rasters ab einer gewissen Anzahl zunehmend verschwindet. In einem weiteren Schritt kann eine horizontale Unterteilung vorgenommen werden. Als Grundlage wird hier das sogenannte Baseline Grid verwendet, welches sich aus der Schriftgröße und dem Zeilenabstand zusammensetzt. Die einzelnen Spalten und Zeilen können des Weiteren noch in Bereiche zusammengefasst werden, um ein modulares Rastersystem zu erstellen. Die Website-Inhalte werden im Nachgang den Bereich zugeordnet und am Raster ausgerichtet.

Layouttypen

Als es nur eine kleine Vielfalt an Endgeräten mit verschiedenen Auflösungen gab, wurde oft mit einem statischen Layout gearbeitet, welches für eine weit verbreitete Auflösung optimiert

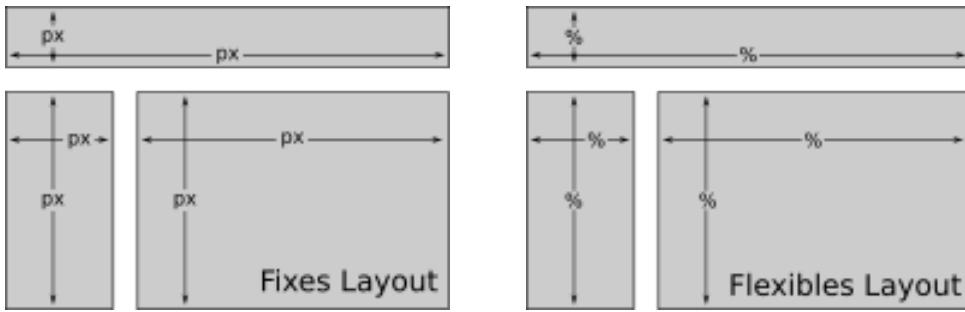


Abbildung (4.7) – Aufbau eines fixen und flexiblen Layouts

war. Heute lässt sich dieser Ansatz wegen der unüberschaubaren Menge an Geräten, die sich zudem stark in ihren Auflösungen unterscheiden, kaum noch heranziehen. Insbesondere schwer ist die Findung einer passenden Auflösung, die für einen Großteil der Geräte als ansprechend erscheint. Heutzutage geht man hingegen nicht mehr von einer festen Breite bei der Layouterstellung aus, sondern erstellt Layouts, die sich je nach Auflösung individuell dem Gerät anpassen.

Während dieser Entwicklung sind verschiedene Layouttypen entstanden, die je nach Anwendungsfall zurate gezogen werden.

Das fixe Layout beschreibt einen Layouttyp, der mit festen Pixelwerten eine fixe Breite definiert. Bei einer passenden Auflösung werden alle Inhalte korrekt angezeigt. Je größer hingegen die Auflösung wird, desto größer wird der umgebende Block der Seite und viel ungenutzter Platz entsteht. Bei einer zu kleinen Auflösung tauchen horizontale Scrollbalken auf und die Inhalte erscheinen abgeschnitten bzw. werden nur noch unvollständig angezeigt.

Das Gegenstück zum fixen Layout stellt das flexible Layout dar. Dieses passt sich allen Veränderungen unmittelbar an und behält dadurch zu jeder Zeit alle vorgegebenen Größenverhältnisse. Definiert wird es im Gegensatz zu den festen Pixelwerten mit relativen prozentualen Werten, wodurch es sich unterschiedlichen Varianten einer Website leicht anpassen kann. Der reine Layouttyp kommt jedoch eher selten zum Einsatz. Vielmehr lässt sich häufig eine Mischform aus fixem und flexiblem Layout vorfinden.

Eine weitere Layoutform ist das elastische Layout. Das elastische Layout verfolgt den Ansatz, dass sich die Inhalte einer Seite anpassen. Dieser Layouttyp ist besonders für Inhalte geeignet, die die vollständige Bildschirmbreite ausfüllen, was bspw. bei einer Produktpräsentation mit großformatigen Bildern und Videos der Fall ist. Die Inhalte müssen hierfür in der Lage sein sich automatisch flexibel anzupassen. Daher ist es für diese Form von Vorteil, wenn es eher wenige Inhalte zum Vorstellen gibt.

Auf Basis des flexiblen Layouts setzt das responsive Layout auf und erweitert die Möglichkeiten situationsgerecht ein passendes Layout für das Endgerät zur Verfügung zu stellen.

Das responsive Layout besitzt als Erweiterung sogenannte Media-Queries, welche es ermöglichen beim Über- oder Unterschreiten fester Schwellwerte eine Veränderung der Ansicht zu starten und die Inhalte bspw. neu anzuordnen.

4.3.4. Grundlagen Wireframes

4.3.4.1. Definition/ Inhalte

Als Wireframe wird die schematische Darstellung von Inhalten und Elementen der Seitenoberfläche verstanden. Wireframes dienen insbesondere zur Konzeptionierung in der Planungsphase, um einerseits einen groben Entwurf für die Verteilung, Anordnung und Gestaltung von den Seitenelementen zu erhalten sowie zum anderen die Beziehungen zwischen den Seiten herzustellen. Die Darstellung des Seitenlayouts ist zumeist eine skizzenhafte, in schwarz-weiß/ grau gehaltene Abbildung. Die einzelnen Bestandteile der Seite werden dabei durch einfache geometrische Formen verdeutlicht. Das Darstellen von Design, Farben, Schrift und Bilder ist kein Bestandteil der Methode. Ein fertiges Wireframe gibt dem Betrachter final Aufschluss über die Platzierung der Informationsinhalte, die Struktur und Navigation der Seite und die Interaktionselemente (Interface), mit welchen der Nutzer interagiert. Die ausgearbeiteten Wireframes stellen weiter fort die Grundlage für die visuelle und funktionale Detaillierung des Produktes dar. Anschließende Schritte können u. a. das Erstellen von Mockups oder Prototypen sein.

4.3.4.2. Arten

Wireframes werden allgemein in Low-Fidelity- und High-Fidelity-Wireframes unterschieden. Low-Fidelity-Wireframes (LFW) stellen das klassische Verständnis von Wireframes dar, bei denen der Fokus allein auf dem funktionalen Design liegt. Die Seitenschemas werden mit einfachsten Formen und ohne konkrete Inhalte erstellt. Die High-Fidelity-Wireframes (HFW) hingegen stellen die nächste Entwicklungsstufe für die Ausarbeitung des Designs dar. In ihr kommen zunehmend mehr und mehr Designkomponenten wie Farben, Typografie, Abstände, Icons, Text, Bilder und Grafiken zum Einsatz. In den Seiten werden zunehmend auch mehr reale Textlängen und Größenverhältnisse der Elemente und Inhalte mit einbezogen.

4.3.4.3. Abgrenzung

Häufig werden im Zusammenhang mit Wireframes noch weitere visuelle Methoden in Verbindung gebracht. Weit verbreitet sind hier insbesondere Mockups und Prototypen. Diese Methoden verwenden jeweils Wireframes als Grundlage und zielen auf die zunehmende reale Veranschaulichung des zukünftigen Produktes.

Unter Mockups wird das Nachbilden eines Produktes oder auch ein maßstabsgerechtes Modell verstanden. Im Vordergrund der Methode steht das visuell-interaktive Design. Hierzu wird das Konzept der Wireframes übernommen und mit den Elementen der Benutzeroberflächen erweitert. Funktionale oder animierte Elemente sind nicht Bestandteil der Methode

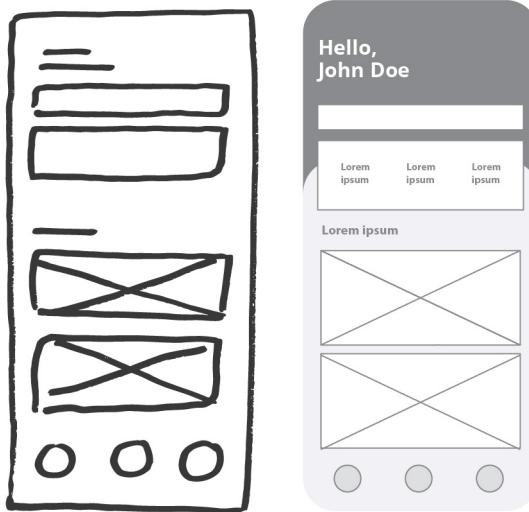


Abbildung (4.8) – Beispielhafte Darstellung von LFW und HFW

und werden nicht für einen Mockup aufgegriffen. Mockups werden im natürlichen Umfeld des Produktes dargestellt - zum Beispiel auf einem Gerätebildschirm. Das Ziel besteht darin das Produkt so echt wie möglich visuell nachzubilden, um dem Kunden ein reales Gefühl über die Erscheinung seines Produktes zu vermitteln.

Als Prototyp wird ein vereinfachtes VersuchsmodeLL des geplanten Produktes verstanden. Der Prototyp baut auf die Ergebnisse eines Mockups auf und erweitert dieses mit funktionalen Elementen, um die Interaktion eines Users mit dem Dienst simulieren zu können. Ein klassisches Beispiel für einen Prototypen ist der Klick-Dummy. Ein Klick-Dummy ist ein teilweise interaktionsfähiges Demo einer Bedienoberfläche, welches alle relevanten Merkmale eines Produktes widerspiegelt und u. a. zur Vorstellung oder für Testläufe genutzt werden kann.

4.3.5. Anwendung

4.3.5.1. Integration

Die Wireframes wurden innerhalb des Projektverlaufes kontinuierlich für die Ausgestaltung und Kommunikation der User-Stories während des Backlog Refinement verwendet. Hierfür wurden in einer frühen Phase die jeweiligen User-Stories vom PO erläutert und ein erster Wireframe-Entwurf konzipiert. Erkenntnisse, die bereits während der Konzipierung entstanden, wurden an den PO übermittelt und nach Absprache noch während des Sprints direkt in die Wireframes aufgenommen. Im Refinement dienten sie einerseits dem PO zur Vorstellung und Erklärung der User-Stories und andererseits zur Präzisierung und Detaillierung der Story im Team. Die daraus neu gewonnenen Informationen wurden vom PO ins Backlog aufgenommen und erneut zum nächsten Refinement in die Wireframes eingearbeitet.

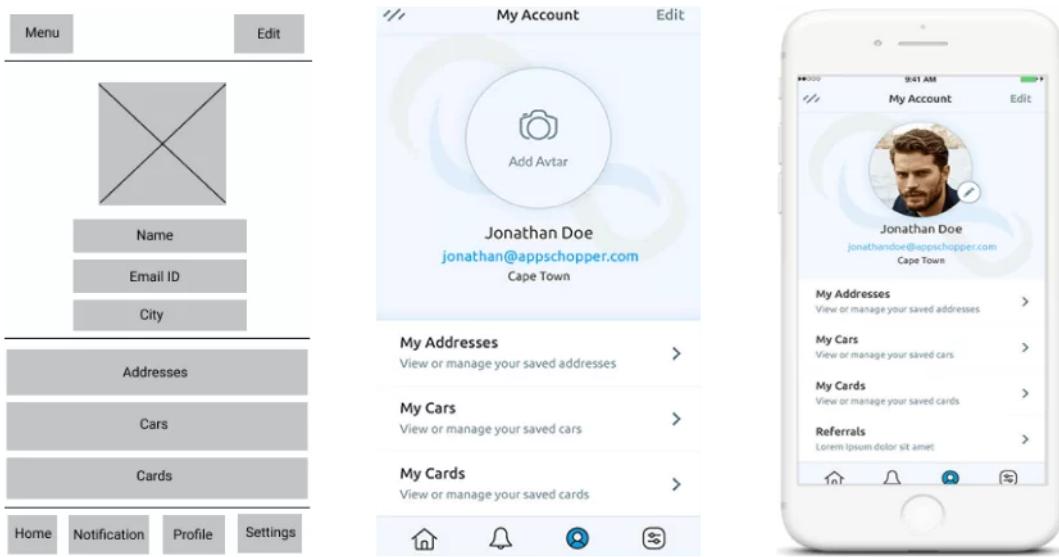


Abbildung (4.9) – Interaktionsschritte an einem Beispiel: Wireframe - Mockup - Prototyp

Neben der Definition und Veranschaulichung der jeweiligen Features wurden die Wireframes auch im Entwicklungsprozess herangezogen. In diesem wurden sie als gestalterischer Entwurf berücksichtigt und so weit wie es dem Entwickler möglich war, implementiert.

4.3.5.2. Auswahl Tool

Für die Erstellung von Wireframes stehen eine Vielzahl von verschiedenen Softwarelösungen zur Verfügung. Grundsätzlich lassen sich diese in desktop- und webbasierte Anwendungen unterscheiden. Für die Erstellung der Wireframes war es essenziell, dass diese leicht erstellt und angepasst werden können, ggf. auch paralleles Arbeiten möglich ist und dass das Teilen der aktuellen Entwürfe ohne zusätzlichen Aufwand geschieht. Anhand der Basisanforderungen konzentrierte sich die Auswahl zunehmend auf rein webbasierte Lösungen. Als Anbieter kristallisierte sich im weiteren Rechercheverlauf zunehmend Figma als ein passender Dienst heraus.

Figma ist eine Onlineanwendung, die sich auf die Erstellung von Wireframes, Mocks und Prototypen spezialisiert und sich in diesem Bereich etabliert hat. Gerade durch die Etablierung des Dienstes lässt sich schließen, dass es als Werkzeug ausgereift ist und die Erstellung einfacher Wireframes durchgängig unterstützt. Für das Erstellen der Wireframes wurde sich auf die grundlegenden Funktionen von Figma beschränkt. Der verfolgte Ansatz während der Ausarbeitung mit Figma lag hierbei besonders auf der Beschränkung der wesentlichen Elemente, um die Kerninhalte der Features deutlich hervorzuheben. Eingesetzt wurden hierbei u. a. zur Gestaltung die Elemente Rechtecke und Text sowie zur Orientierung und Ausrichtung der Inhaltselemente das Raster und die Gruppierfunktion.

Anforderungen an die GUI

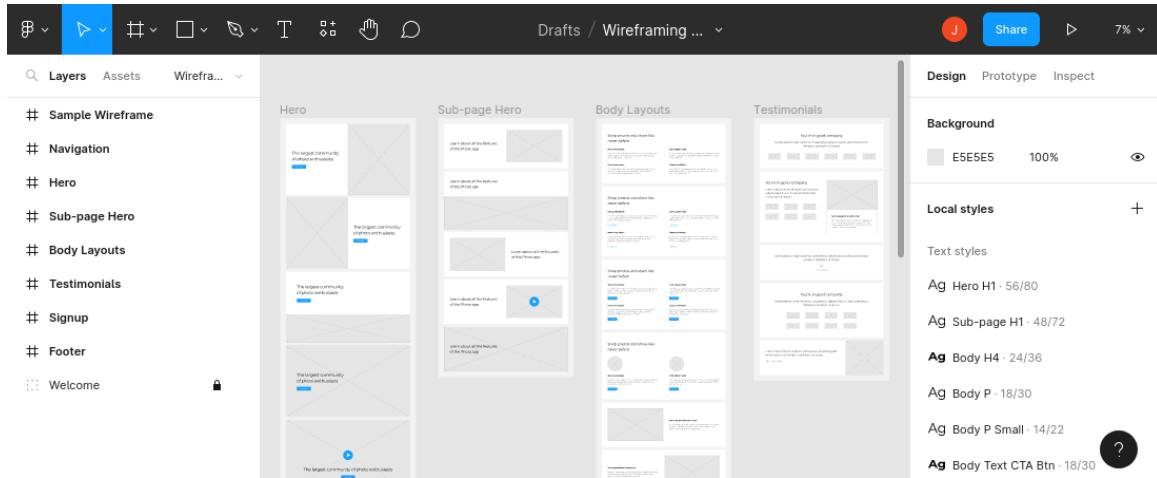


Abbildung (4.10) – Figmas-Arbeitsumgebung

Für die Ausarbeitung der Wireframes wurden vorab einige Anforderungen festgelegt, welche bei der Erstellung beachtet werden mussten. Die Anforderungen bezogen sich zum einen auf die Gestaltung der Oberfläche und zum anderen auf die technologiekonforme Gestaltung der Wireframes.

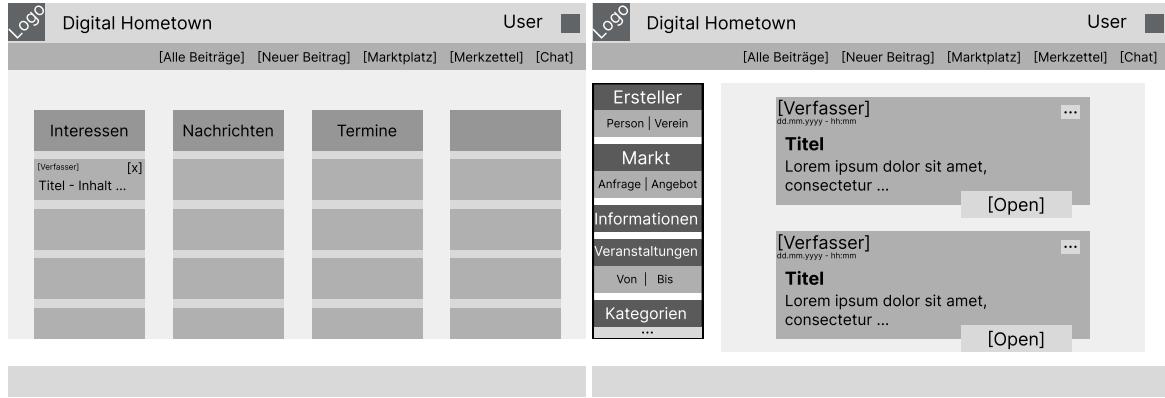
Eine wichtige Anforderung an die Plattform war es, die UI generationsübergreifend zu konzipieren. Aus diesem Ansatz heraus ließen sich die folgenden Ansprüche an die UI ableiten:

- Einfache und intuitive Seitengestaltung, um die Inhalte, den Funktionsumfang und die Bedienung schnell selbstständig erfassen zu können,
- Reduzierung der Inhalte auf das Wesentliche, um Verwirrungen/ Ablenkungen zu vermeiden,
- Flache Hierarchien, um direkte Zugriffe auf die gewünschten Inhalte zu ermöglichen.

Von der technologischen Seite her war es erforderlich, dass die erstellten Konzeptentwürfe sich auch mit den ausgewählten Web-Technologien umsetzen lassen. Die Entwickler sollten in die Lage versetzt werden die Entwürfe entweder mittels Eigenentwicklungen oder durch das Heranziehen von Bibliotheken umzusetzen, ohne auf größere Herausforderungen zu stoßen. Das Ziel war es darüber hinaus einen hohen Grad an Wiederverwendbarkeit der Komponenten zu erreichen.

Pages

Als geeigneter Wireframetyp wurde ein Hybrid aus LFW und HFW gewählt. Die erstellten Wireframes umfassten alle layouttypischen Bereiche wie Kopf-, Inhalts- und Fußbereich, deren Inhalte in Feldern vereinfacht symbolisiert wurden. Die einzelnen Felder beinhalteten Texte, Dummy-Blöcke bspw. für Bilder und Interaktionselemente (Buttons, Dialoge usw.). Die Wireframes wurden sehr schlicht in Graustufen gehalten und ohne die Berücksichtigung von Designelementen (Farbe, Typografie usw.) erstellt. Für die regelmäßige Erstellung und

**Abbildung (4.11)** – Startseite (DHT)**Abbildung (4.12)** – Marktplatz (DHT)

Anordnung der Komponenten hat sich nach einigen Entwürfen die Leinwandbreite von 1160 Pixel mit einem Raster von 27×37 Pixel als vorteilhaft ergeben.

Wireframes wurden für die Seiten/ Dialoge

- Landingpage,
- Registrierung,
- Dashboard,
- Chat,
- Profil,
- Accounteinstellungen,
- Merkzettel und
- Marktplatz

erstellt.

Die Art und Detailtiefe der Ausgestaltung zeigen exemplarisch die folgenden Wireframes.

4.4. Usabilityanalyse

5. Architektur

Auf Grundlage der Technologie Entscheidungen wurde folgende Architektur für das Projekt entwickelt. Diese ist in folgender Abbildung dargestellt und wird in diesem Kapitel noch genauer erklärt:

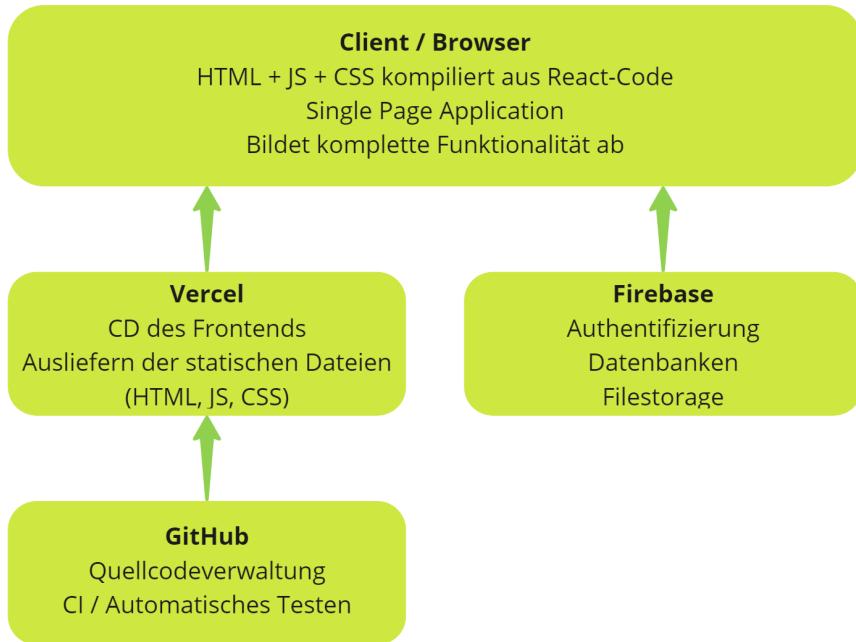


Abbildung (5.1) – Übersicht der technischen Softwarearchitektur

Das Frontend wird in React entwickelt. Dieses enthält die Single Page Application (SPA) für die Benutzeroberfläche. Die gesamte Funktionalität wird in diesem Frontend implementiert. Um im Team bei der Entwicklung effizient zusammenzuarbeiten, wird als Quellcodeverwaltungstool Git bzw. Github verwendet. Wird eine Änderung am Quellcode veröffentlicht, werden automatisch die CI / CD Prozesse gestartet. Sind die Tests erfolgreich, wird die neue Version über Vercel gehostet.

Wird der entsprechende Internetlink im Browser aufgerufen, erhält der anfragende Client (Browser) die statischen Dateien für die Benutzeroberfläche. Der Browser führt das darin enthaltene Javascript aus, das ein dynamisches Laden der Inhalte der Datenbank (Firebase) ermöglicht.

6. Technologien

Für die Entwicklung eines Softwaresystems ist insbesondere bei der Implementierung entscheidend, welche Technologien verwendet werden. Aus diesem Grund widmet sich diese Kapitel den technischen Aspekten der Plattform beschrieben. Dabei werden die Technologien und Frameworks vorgestellt, die für die Entwicklung der Plattform verwendet wurden.

Die Plattform wurde mit den folgenden Technologien und Frameworks entwickelt:

- **TypeScript** als Programmiersprache
- **React** als Frontend-Framework
- **MUI** als UI-Framework
- **Firebase** als Backend as a Service (BaaS)Plattform
- **Vercel** als Hosting-Plattform
- **Github** als Versionsverwaltungs-Plattform
- **Jira** als Projektmanagement-Plattform

Diese werden im Folgenden kurz vorgestellt.

6.1. Grundlage für Technologieentscheidung

Die einzige aus der Aufgabenstellung ersichtliche Vorgabe ist, dass das Softwareprodukt „Digital Hometown“ eine Plattform für den Austausch bieten soll. Die Wahl der Technologie ist uns hierbei offengelassen. Was daraus jedoch hervorgeht ist, dass es sich um eine für möglichst viele Nutzer verwendbare Web- oder Mobilanwendung handeln soll. Insbesondere bei der Wahl einer Webanwendung, die dem Nutzer jeglichen Installationsaufwand erspart, ist die Hemmschwelle sehr gering, ein Softwareprodukt auszuprobieren.

Da, wie beschrieben, für eine solche Plattform des sozialen Austauschs eine ausreichend große Nutzerzahl entscheidend ist, fiel die Entscheidung schnell auf eine Webanwendung. Obwohl beim aktuellen Stand der Plattform „Digital Dahoam“ nicht in erster Linie auf die Benutzbarkeit auf mobilen Endgeräten gelegt wurde, sei an dieser Stelle erwähnt, dass sich Webanwendungen mit etwas mehr Aufwand sehr gut auch für mobile Geräte wie Smartphones oder Tablets entwickeln lassen. Der Fachbegriff hierfür ist die Umsetzung einer „Progressive Web-App“.

6.2. TypeScript

TypeScript ist eine Erweiterung von Javascript, die statische Typisierung und Klassen hinzufügt. Dadurch wird die Entwicklung von Software vereinfacht, da die Typisierung die Lesbarkeit des Codes verbessert und die Klassen die Wiederverwendbarkeit von Code ermöglichen. Die Programmiersprache wird von Microsoft entwickelt und ist Open Source.¹

6.2.1. Einsatz im Projekt

TypeScript wurde im Projekt verwendet, um die Entwicklung der Plattform zu vereinfachen. Der komplette Code der Website wurde mit TypeScript, HTML und CSS geschrieben, wobei TypeScript hierbei die Hauptrolle spielt.

6.2.2. Grund für Technologieentscheidung

Da TypeScript in großen Teilen der Javawelt inzwischen als de facto Standard ist um vor allem große Anwendungen sicher und effizient zu entwickeln, wurde diese Technologie für die Entwicklung der Plattform verwendet. React (6.3), das größte Frontend-Framework der Welt, wird inzwischen auch in TypeScript entwickelt.²

6.3. React

React ist ein Open-Source Frontend-Framework, das von Facebook entwickelt wird. Es ermöglicht die Entwicklung von Benutzeroberflächen für Webanwendungen. Dabei wird die Benutzeroberfläche in einzelne Komponenten aufgeteilt, die unabhängig voneinander entwickelt werden können. Diese Komponenten werden in einer `.jsx` Datei definiert, die eine Kombination aus Javascript und HTML ist. Die Komponenten werden in einer React Anwendung in einer `.jsx` Datei eingebunden. In einer neueren Version, ist es auch möglich mit TypeScript zu arbeiten. Die neue Dateierweiterung für diese TypeScript ist `.tsx`. Diese Dateien werden dann in eine Javascript Datei kompiliert, die von einem Browser ausgeführt werden kann.³

6.3.1. Allgemeines in Bezug auf die Implementierung mit React

React basiert auf dem Model-View-Controller Design Pattern. Der Browser Document Object Model (DOM) fungiert dabei als die View-Komponente. Die Model-Komponente, ist der Virtual DOM, das vom Controller (React) manipuliert wird.

¹Vgl. TypeScript 2022 [1]

²Vgl. TypeScript 2023 [2]

³Vgl. React 2022 [3]

6.3.1.1. Einrichten und Starten der React-Anwendung

Für die Entwicklung von React-Anwendungen eignen sich alle moderne Entwicklungsumgebungen. Im Projektteam wurde sich auf den weitverbreiteten Texteditor „Visual Studio Code“ geeinigt. Neben einer Entwicklungsumgebung wird Node.js benötigt, um Javascript-Code auf der Entwicklungsmaschine ausführen zu können. Die notwendigen Abhängigkeiten werden mit dem Paketmanager „yarn“ installiert. Mit dem in der Datei package.json definierten Alias yarn dev startet der lokale Node.js Entwicklungsserver automatisch, nachdem alle benötigten Pakete installiert wurden. Handelt es sich um eine lauffähige Version, wird automatisch im Browser die Startansicht der entwickelten React-Anwendung geöffnet.

6.3.1.2. React Components

React besteht aus Komponenten, die automatisch neu gerendert werden, wenn sich die Parameter der Komponente ändern. Komponenten können als Functional- bzw. als Class-Komponenten implementiert werden. Während die Verwendung von Class-Komponenten in älteren React-Versionen üblich war, wird in den aktuellen Versionen meist die funktionelle Implementierung verwendet.

```
1 function Component(props: {name: string}) {
2   return <div>Hallo {props.name}!</div>
3 }
```

Beispiel einer React-Komponente

6.3.1.3. React Hooks

Die Komponenten bilden die Basis jeder React-Anwendung. Durch die React-Hooks wird es einer Komponente ermöglicht, dynamische Bestandteile und einen Zustand zu besitzen. Es gibt mehrere Hooks für verschiedene Anwendungsfälle und es lassen sich auch eigene Hooks definieren. Eines der wichtigsten React-Hooks ist useState. Durch useState wird es ermöglicht, eine Variable über ein oder mehrere Komponenten hinweg zu benutzen und manipulieren. Die Verwendung eines useState Hooks wird in Code 2 gezeigt. Hier wird auch ein weiterer wichtiger Hook aufgeführt. Der useEffect Hook ermöglicht es, auf die Änderung eines Zustands zu reagieren.

```
1 function Component() {
2   // count ist der aktuelle Wert
3   // setCount ist die Funktion, um den Wert zu ändern
4   const [count, setCount] = React.useState<number>(0)
5
6   React.useEffect(() => {
7     // wird ausgeführt, wenn count sich ändert
8     console.log("count changed")
9   }, [count])
10}
```

```

11  return (
12    <div>
13      <p>{count} mal geklickt.</p>
14      <button onClick={() => setCount(count + 1)}></button>
15      Button
16    </div>
17  )
18 }

```

Beispiel einer React-Komponente mit Hooks

Es lassen sich beliebig viele Komponenten verschachteln. Das Durchreichen der Parameter wird bei größeren Projekten sehr aufwändig – insbesondere in Bezug auf die Wartbarkeit. Um dies zu entschärfen, gibt es weitere Konzepte wie der React Context, der im Folgenden beschrieben wird.

6.3.1.4. React Context

Der React Context ermöglicht es einen Zustand über mehrere Komponenten hinweg zu benutzen, ohne ihn mittels Parameter an alle Unterkomponenten durchzurütteln. Man kann den React Context mit einer globalen Variable vergleichen. Ein typischer Anwendungsfall für den React Context ist das Verwenden von Authentifizierungsdaten wie der Name über die gesamte Anwendung hinweg.

6.3.2. Einsatz im Projekt

React wurde verwendet, um die gesamte Website aufzubauen. Sie bildet alles ab, was der Benutzer sieht und mit der Plattform interagiert.

6.3.3. Grund für Technologieentscheidung

React wurde für die Entwicklung der Plattform verwendet, da es das meistgenutzte Frontend-Framework der Welt ist. Außerdem gab es ein großes Interesse der verschiedenen Entwickler, sich in dieses Framework einzuarbeiten.

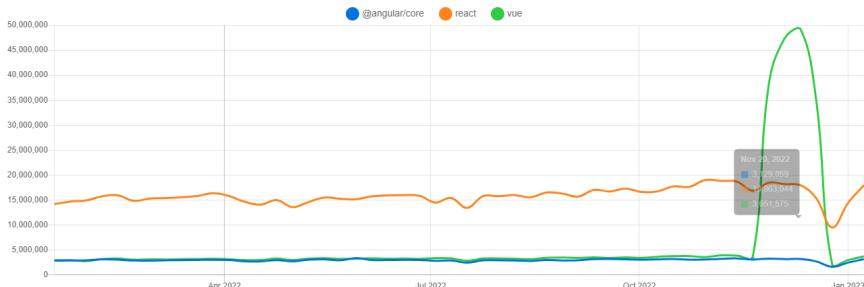


Abbildung (6.1) – Vergleich Downloadzahlen verschiedener Frontend Frameworks [4]

6.4. Material-UI

Material-UI (kurz MUI) ist eine Bibliothek, die es ermöglicht, Komponenten aus dem Material Design zu verwenden. Diese Komponenten sind konfigurierbar und können so an die eigenen Bedürfnisse angepasst werden. Trotzdem entsprechen Sie alle einer einheitlichen Designsprache, die von Google entwickelt wurde.⁴

MUI bietet Komponenten für folgende Bereiche an:

- **Navigation** – Komponenten für die Navigation.
- **Inputs** – Komponenten für die Eingabe von Daten.
- **Layout** – Komponenten für das Layout der Website.
- **Data Display** – Komponenten für die Anzeige von Daten.
- **Feedback** – Komponenten für die Rückmeldung an den Benutzer.
- **Surfaces** – Komponenten für Oberflächen.
- **Utils** – Komponenten für die Unterstützung.

Diese Bibliothek wird von Material-UI SAS. entwickelt und ist Open-Source auf Github verfügbar. MUI bietet eine direkte Integration mit React.⁵

6.4.1. Einsatz im Projekt

Da React nur sehr rudimentäre Komponenten für die Benutzeroberfläche bereitstellt, wurde MUI verwendet, um die Benutzeroberfläche zu gestalten. Es bietet eine große Auswahl an Komponenten, die direkt in React verwendet werden können.

6.4.2. Grund für Technologieentscheidung

MUI vereinfachte die Gestaltung der Benutzeroberfläche, da es eine große Auswahl an Komponenten bietet, die direkt in React verwendet werden können. Aus diesem Grund konnten die Entwickler schnell mit der Gestaltung der Benutzeroberfläche beginnen.

⁴Vgl. Google 2021 [5]

⁵Vgl. Material-UI 2022 [6]

6.5. Firebase

Firebase ist eine BaaS Plattform, die die Bereitstellung von Backend-Funktionalität ermöglicht inklusive Datenbank, Authentifizierung, Datei-Upload, etc. Dabei wird die Funktionalität in einzelne Module aufgeteilt, die unabhängig voneinander verwendet werden können. Die Plattform wird von Google entwickelt⁶ und besteht aus mehreren Komponenten, die nun kurz vorgestellt werden.

6.5.1. Firebase Authentication

Firebase Authentication ist ein Modul von Firebase, das die Authentifizierung von Nutzern ermöglicht. Dabei werden verschiedene Authentifizierungsmethoden unterstützt, wie z. B. E-Mail und Passwort, Google, Facebook, etc. Außerdem ist es möglich eigene Authentifizierungsmethoden zu implementieren, sowie Nutzer über Telefonnummern zu authentifizieren. Dabei können Nutzer auch in mehreren Geräten gleichzeitig eingeloggt sein.

Bis auf die Authentifizierungsmethode über Telefonnummern, die nur in den USA verfügbar ist, werden alle Authentifizierungsmethoden kostenlos angeboten.⁷

6.5.2. Firebase Realtime Database

Firebase Realtime Database ist ein Modul von Firebase, das die Bereitstellung einer Datenbank ermöglicht. Diese Datenbank ist eine NoSQL Datenbank, ähnlich wie MongoDB. Hier werden die Daten nicht relational in Dokumenten gespeichert, sondern in einer Baumstruktur. Updates in der Datenbank werden in Echtzeit an alle Nutzer gesendet, die sich mit der Datenbank verbinden.⁸

6.5.3. Firebase Cloud Firestore

Firebase Cloud Firestore ist ein Modul von Firebase, das die Bereitstellung einer Datenbank ermöglicht. Dabei wird die Datenbank in einzelne Dokumente aufgeteilt, die in einer Baumstruktur organisiert sind. Firestore ist die Weiterentwicklung der Realtime Database und bietet einige Vorteile gegenüber dieser. So ist die Datenbank in mehrere Regionen aufgeteilt, was die Verfügbarkeit erhöht. Außerdem ist es möglich, die Datenbank in mehrere Projekte aufzuteilen, was die Sicherheit erhöht. Abfragen in der Datenbank können mit Indexen optimiert werden, was die Performance verbessert.⁹

⁶Vgl. Firebase 2022 [7]

⁷Vgl. Firebase Authentication 2022 [8]

⁸Vgl. Firebase Realtime Database 2022 [9]

⁹Vgl. Firebase Cloud Firestore 2022 [10]

6.5.4. Firebase Storage

Firebase Storage ist ein Modul von Firebase, das die Bereitstellung von Datei-Upload ermöglicht. Dabei können Dateien in einem Bucket gespeichert werden, das in einzelne Ordner aufgeteilt ist. Der Datei-Upload kann über die Firebase Konsole oder über die Firebase SDK's erfolgen. Diese Funktion ist vergleichbar mit AWS S3 Buckets.¹⁰

6.5.5. Einsatz im Projekt

Firebase wurde im Projekt für die Bereitstellung der Datenbank und der Authentifizierung verwendet. Außerdem speichert es die Bilder, die von den Nutzern hochgeladen werden.

6.5.6. Grund für Technologieentscheidung

Durch Firebase konnte die Entwicklung der Backend-Funktionalität beschleunigt werden, da die Entwickler sich nicht um die Bereitstellung dieser Funktionalität kümmern mussten.

6.6. Github

Github ist eine Plattform, die es ermöglicht, Softwareprojekte zu verwalten. Diese Plattform wird von Github Inc. entwickelt. Github bietet eine direkte Integration mit Git. Die wichtigsten Funktionen von Github sind in Tabelle 6.1 aufgeführt.¹¹

Funktion	Beschreibung
Versionsverwaltung	Github bietet Versionsverwaltung auf Grundlage von Git an.
Projektmanagement	Anforderungen können als Issues angelegt und verwaltet werden.
Dokumentation	Mithilfe von Markdown Wikis.
Teamarbeit	In Form von Kommentaren, Reviews, etc.
Hosting	Bereitstellung statischer Seiten.
CI/CD	Automatisierte Tests und Deployment.

Tabelle (6.1) – Funktionen von Github

6.6.1. Einsatz im Projekt

Github wurde im Projekt für die Versionsverwaltung und CI verwendet. Die Versionsverwaltung wurde durch die Integration mit Git ermöglicht. Außerdem wurden Github Actions benutzt, welches automatisierte Tests und andere Sanity-Checks ermöglicht.

¹⁰Vgl. Firebase Storage 2022 [11]

¹¹Vgl. Github 2022 [12]

6.6.2. Grund für Technologieentscheidung

Github wurde im Projekt eingesetzt, da es eine gute Integration mit Git bietet und somit die Versionsverwaltung vereinfacht. Außerdem wurde durch die verfügbare CI Funktionalität die Qualität des Codes verbessert und stetig getestet werden. Die Entwicklungsgeschwindigkeit wurde dadurch vereinfacht.

6.7. Vercel

Vercel ist eine Hosting-Plattform, die es ermöglicht, statische Webseiten zu hosten. Diese Plattform wird von Vercel Inc. entwickelt. Vercel bietet eine direkte Zuordnung von Github zu Vercel und erstellt neue Deployments und Builds, sobald ein neuer Commit in Github verfügbar ist. Dies funktioniert auch mit mehreren Branches und Pull Requests. Jedes Deployment wird mit einer eigenen URL versehen, sodass mehrere Versionen der gleichen Webseite gleichzeitig verfügbar sind. So können Pull Requests getestet werden, bevor sie in den Master Branch gemerged werden.¹²

6.7.1. Einsatz im Projekt

Vercel wurde im Projekt für die Bereitstellung der Webanwendung verwendet. Sobald bei Github ein neuer Commit verfügbar ist, wird automatisch ein neues Deployment erstellt. Dies geschah für den main-Branch auf der Domain <https://dahoam.rosen.dev> und für den dev-Branch auf der Domain <https://dev.dahoam.rosen.dev>. Jeder andere Branch bekam sein eigenes Deployment auf einer eigenen Domain, welche von Vercel erstellt wurde.

6.7.2. Grund für Technologieentscheidung

Vercel wurde verwendet, um die Bereitstellung der Webanwendung zu vereinfachen. Nach Errichtung des Github Repositories und initialer Projekterstellung wurde die Integration mit Vercel in wenigen Klicks aktiviert und hostet seitdem kostenlos die Webanwendung.

¹²Vgl. Vercel 2022 [13]

6.8. Jira

Jira ist eine Plattform, die es ermöglicht, Softwareprojekte zu verwalten. Diese Plattform wird von Atlassian entwickelt.

Die wichtigsten Funktionen von Jira sind in Tabelle 6.2 aufgeführt.¹³

Funktion	Beschreibung
Projektmanagement	Anforderungen können als Issues angelegt und verwaltet werden.
Dokumentation	Mithilfe von Markdown Wikis.
Teamaarbeit	In Form von Kommentaren, Reviews, etc.

Tabelle (6.2) – Funktionen von Jira

6.8.1. Einsatz im Projekt

In Jira wurden die einzelnen User Stories verwaltet und bearbeitet, sowie in Sprints eingeplant. Durch ein Kanbanboard wurden die einzelnen User Stories in den einzelnen Sprints angezeigt. Jira bietet eine direkte Integration mit Github.

6.8.2. Grund für Technologieentscheidung

Jira wurde verwendet, um die Verwaltung der User Stories zu vereinfachen. Durch die direkte Integration mit Github wurden die einzelnen User Stories mit den dazugehörigen Commits verknüpft. Dadurch konnte die Entwicklung der einzelnen User Stories nachvollzogen werden. Außerdem ist es ein sehr simples Tool, welches für die Verwaltung von User Stories sehr gut geeignet ist.

¹³Vgl. Atlassian 2023 [14]

7. Implementierung

In diesem Kapitel wird die Implementierung der Anwendung beschrieben.

Da der komplette Code und viele Screenshots den Rahmen dieses Kapitels sprengen würden, wird in den einzelnen Kapiteln nur ein Auszug gezeigt. Den kompletten Code findet man direkt im Github Repository unter <https://github.com/Jonasdero/digital-hometown-frontend>.

7.1. Landing Page

Bei dem Aufruf der Plattform „Digital Dahoam“ wird der Nutzer auf die Landing Page (s. Abbildung 7.1) weitergeleitet. Diese ist abhängig davon, ob der Nutzer eingeloggt ist oder nicht.

Auf der Landing Page wird die Vision der Plattform vorgestellt. Bei dem Projekt „Digital Dahoam“ geht es darum, dass sich Menschen in der Nachbarschaft vernetzen können. Die Plattform soll zudem die Menschen dazu motivieren, anderen zu helfen, bzw. selbst um Hilfe zu bitten. Insbesondere für neu zugezogene Menschen soll die Plattform eine Möglichkeit sein, die Umgebung zu entdecken und interessante Vereine kennenzulernen.

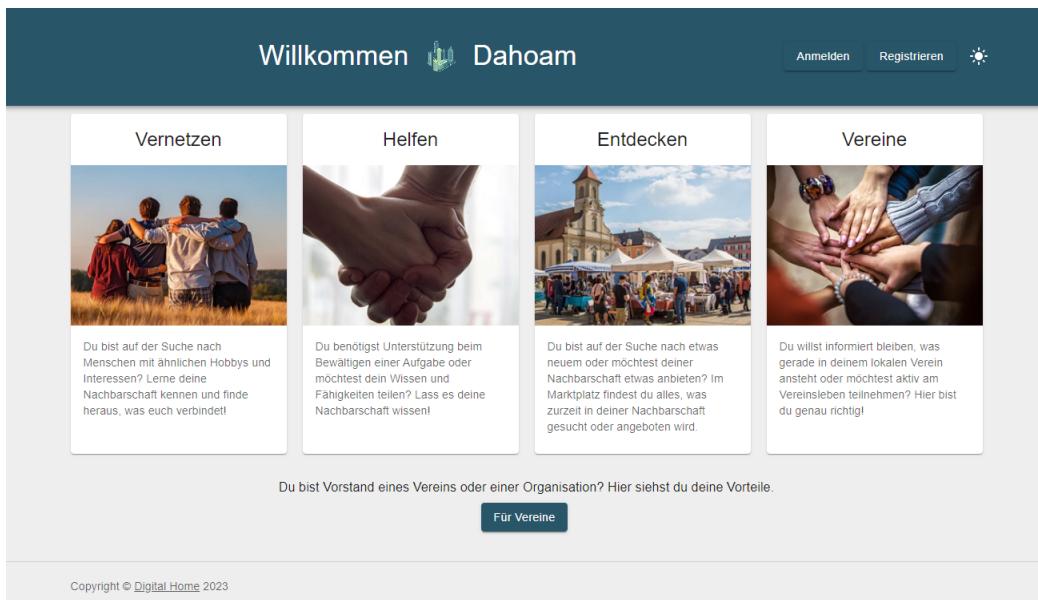


Abbildung (7.1) – Bildschirmaufnahme der Startseite für nicht eingeloggte Nutzer

Will sich ein Benutzer bspw. ein Vereinsvorstand mit seinem Verein registrieren, kann er über den Button „Für Vereine“ zur Landing Page für Vereine navigieren. Über die Landing Page für Vereine kann der Vorstand seinen Verein bei der Plattform registrieren oder sich einloggen (s. ??). Der Inhalt der Landing Page beschreibt, welche Vorteile die Plattform für Vereine bietet.

7.2. Profil

Die Profile eines Nutzers sind in der Anwendung sehr wichtig. Jeder Nutzer verwaltet sein eigenes Profil, welches er mit anderen Nutzern teilt. Dazu gehören Informationen wie Name, E-Mail-Adresse, Geburtsdatum, Geschlecht und ein Profilbild. Damit andere Nutzer die Informationen des Profils sehen können, muss das Profil öffentlich sein. Zudem sollen Benutzer der Anwendung durch Interessen, Profilbilder und persönliche Beschreibungen voneinander unterscheidbar sein und sich so besser vernetzen können. Um sich mit anderen Nutzern zu verbinden, ist es wichtig, dass diese Informationen in der Anwendung gespeichert werden.

Außerdem soll die soziale Interaktion zwischen Nutzer ermöglicht werden. Dazu gehören Funktionen wie das Folgen von anderen Nutzern oder auch das Schreiben von Nachrichten zwischen zwei Nutzern oder in Gruppen. Dies soll direkt vom Profil eines Nutzers aus möglich sein.

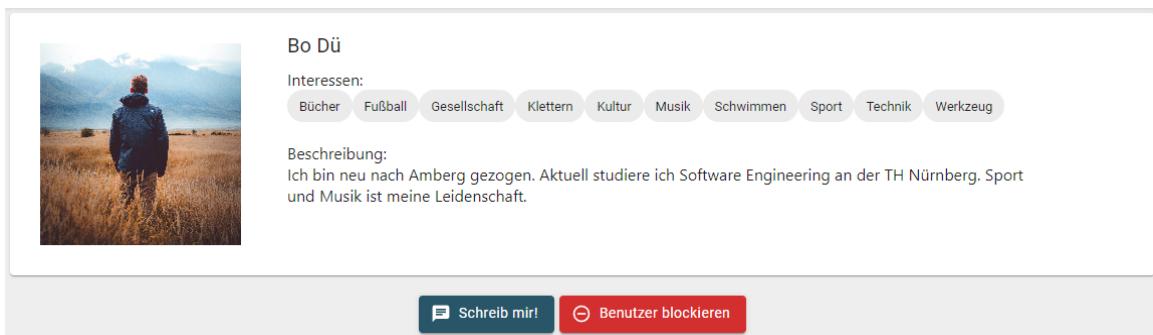


Abbildung (7.2) – Übersicht eines Benutzerprofils

Hier sieht man die verschiedenen Informationen, die ein Nutzer in seinem Profil angeben kann. In den nächsten Seiten wird die Implementierung dieser Funktionen beschrieben. Diese wird aufgeteilt in die verschiedenen Bereiche: *Anmeldung & Registrierung, persönliches Accountmanagement, Profilseite & Profilbilder* und *blockierte Nutzer*.

7.2.1. Anmeldung & Registrierung

Den Nutzern wird die Möglichkeit gegeben, sich mit einer E-Mail-Adresse und einem Passwort anzumelden oder eine direkte Anmeldung über OAuth mit Google zu nutzen. Dies wird beides durch die Firebase Authentication API ermöglicht.

Sobald ein Nutzer sich registriert hat, wird von Firebase intern ein Benutzerprofil erstellt, welches wichtige Nutzermetadaten und Authentifizierungsinformationen enthält. Außerdem wird bei der Anmeldung über Google das Profilbild mit in Firebase gespeichert. Die Anmeldemaske sieht wie folgt aus:

Um diese Daten in der Anwendung zu speichern, wird ein eigener Benutzerdatensatz angelegt, sobald ein Nutzer registriert wurde und Daten aus dem internen Benutzerprofil von

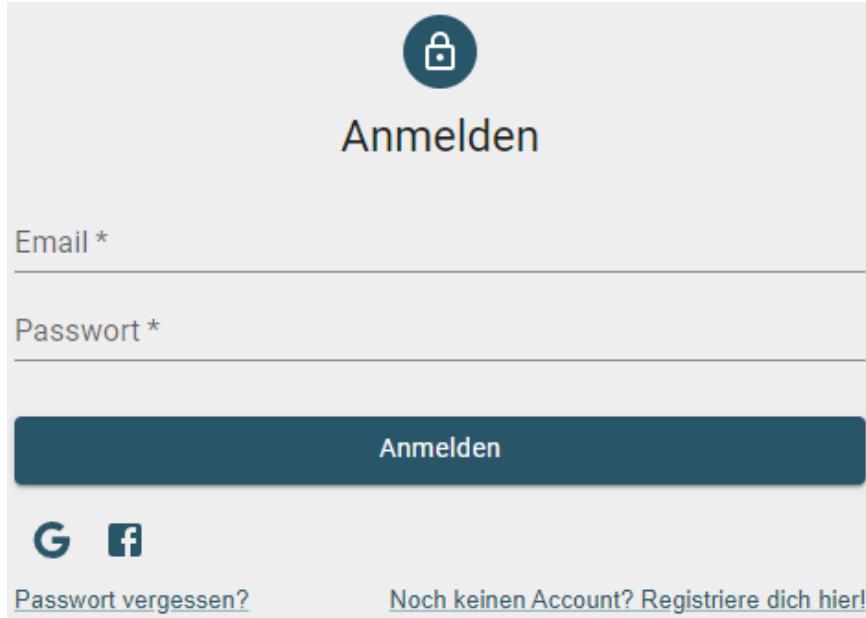


Abbildung (7.3) – Anmeldemaske

Firebase mit übertragen. Hierbei wird unterschieden, ob der aktuelle Benutzer ein normaler Nutzer oder ein Verein ist. Diese werden in unterschiedlichen „Collections“ gespeichert. Bei der Anmeldung wird dann geprüft, welchen Typ der Benutzer hat und die Daten entsprechend geladen. Dadurch kann mit einer Datenstruktur gearbeitet werden, die für beide Typen geeignet ist. Andere Ansichten basieren dann häufig auf der Unterscheidung zwischen Nutzern und Vereinen. Die Struktur des Datensatzes kann man in Abbildung A.1 sehen.

7.3. Authentifizierung

Essentiell für ein soziales Netzwerk ist es, dass die Benutzer sich mit einem Profil registrieren können. Wie das im Detail funktioniert und wie die Authentifizierung implementiert wurde, wird in diesem Kapitel beschrieben.

7.3.1. Verwendung des „Firebase-Authentication“-Services

Die Implementierung der Authentifizierung wird mit dem Backend-as-a-Service-Anbieter Firebase durchgeführt. Firebase bietet eine Schnittstelle, die es ermöglicht, Profile anzulegen sowie den Anmelde- und Registrierungsprozess mit wenigen Zeilen Code zu implementieren.

Durch die verschiedenen Authentifizierungsmethoden, die Firebase anbietet, kann der Nutzer sich mit E-Mail und Passwort, Google oder Facebook anmelden. Die Authentifizierungsmethoden können einfach in der Firebase-Konsole aktiviert werden.

Es wird zu dem die Möglichkeit geboten, dass Nutzer, die ihre Zugangsdaten vergessen haben, eine E-Mail mit einem Link zum Zurücksetzen des Passworts erhalten.

Die Beschriebenen Funktionen decken also einen standardmäßigen Anmelde- und Registrierungsprozess ab.

7.3.2. Zugriff auf die Profilinformationen

Wie bei der React-Einführung beschrieben, ist es aufwendig, Informationen wie die des Profils über die gesamte Anwendung hinweg durchzureichen. Dafür wurde auch bereits die Möglichkeit des React Contexts vorgestellt. Da dieses Konzept auch für die Profilinformationen eingesetzt wird hier nochmal ein Beispiel der Implementierung gezeigt.

Das in Listing 7.1 stellt das Interface des Contexts dar, das für alle Components befüllt wird. Die Profilinformationen sind im Attribute `currentUser` gespeichert.

```

1 interface AuthContextI {
2   currentUser: User | Club | undefined | null
3   setCurrentUser: React.Dispatch<React.SetStateAction<User | Club |
4     undefined | null>>
5   logOut: () => void
6   logIn: (email: string, password: string) => void
7   signUpWithEmail: (email: string, password: string, displayName: string,
8     isOrg: boolean) => Promise<void>
9   signUpOAuth: (providerName: "google" | "facebook", isOrg: boolean) => void
10  // usw.
11 }

```

Listing (7.1) – Auszug aus dem Interface des Authentifizierungscontexts

Um den Vorteil des Contexts für die Codequalität hervorzuheben wird in Listing 7.2 dargestellt, wie einfach es möglich ist, die zentral befüllten Attribute in einer beliebigen React-Komponente zu verwenden. Hierbei wird das Prinzip der Higher-Order-Components (HOC) genutzt, das zuvor implementiert wurde. Es ist lediglich notwendig, die implementierte Komponente mit dem HOC `withAuth` zu umschließen. Dieses HOC stellt der jeweiligen Komponente die Attribute des Contexts als Props zur Verfügung.

```

1 function SignOut({ logOut }: AuthContextI) {
2   useEffect(() => logOut(), [logOut])
3   return <Navigate to="/" />
4 }
5
6 export default withAuth(SignOut)

```

Listing (7.2) – Verwendung des Authentifizierungscontext-HOCs

Das Component `SignOut` bekommt hier beispielsweise durch das HOC automatisch die Funktion `logOut` als Prop übergeben, die dann zum Ausloggen des Benutzers verwendet wird.

7.3.3. Authentifizierungsprozess

Im folgenden wird nun anhand von Bildschirmaufnahmen gezeigt, wie der Authentifizierungsprozess abläuft.

Beim Klick auf den Registrieren-Button der Landing-Page wird man auf eine Seite weitergeleitet, über die Benutzername, Email und Passwort eingegeben werden können (s. Abbildung 7.4). Fehlerhafte Eingaben der E-Mail-Adresse, sowie die redundante Eingabe des Passworts werden hierbei validiert.

Abbildung (7.4) – Registrierungsseite

Nach der erfolgreichen Registrierung des Nutzers, wird er zunächst auf eine Seite geführt, auf der er seine Profilinformationen um das Geburtsdatum und die Postleitzahl ergänzen kann (s. Abbildung 7.5).

Abbildung (7.5) – Ergänzen der Profilinformationen beim ersten Anmelden

Nach der erfolgreichen Registrierung wird der Nutzer auf sein Profil weitergeleitet, wo er u. a. Interessen hinterlegen kann, um von Nutzern gefunden zu werden, welche die gleichen Interessen haben. Details hierzu sind dem Projektbericht von Jonas Roser zu entnehmen.

Eine besonders einfache Möglichkeit, sich bei der Plattform zu registrieren, bzw. anzumelden ist die Authentifizierung über Google. Beim Klick auf das Google-Icon öffnet sich direkt der Google-OAuth-Dialog (s. Abbildung 7.6). Dies steigert die Usability der Plattform, da der Nutzer nicht mehr die E-Mail-Adresse und das Passwort eingeben muss.

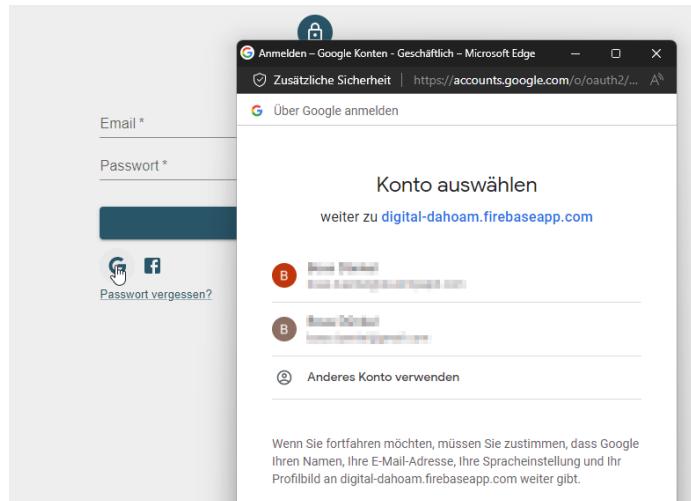


Abbildung (7.6) – Authentifizierungs-Dialog beim Anmelden mit Google

Wenn der Nutzer sich mit einer E-Mail-Adresse registriert hat, kann er sich auf folgender Seite anmelden (s. Abbildung 7.7).

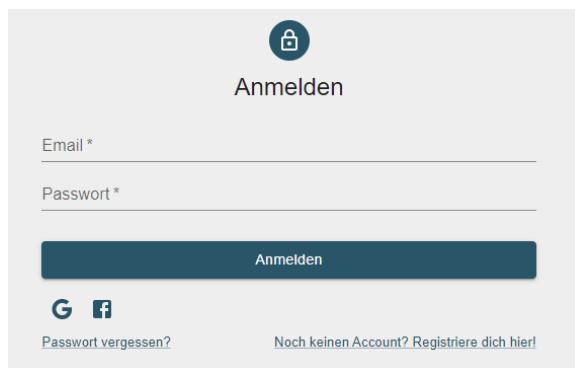


Abbildung (7.7) – Anmeldeseite

7.3.4. Persönliches Accountmanagement

Um die Daten des Benutzerprofils zu verwalten, gibt es eine Seite, auf der der Nutzer seine persönlichen Daten ändern kann. Diese werden dann in der Anwendung aktualisiert und in der Datenbank gespeichert. Er hat hier die Möglichkeit seinen Namen, seine E-Mail, sein Geburtsdatum und seine Postleitzahl zu ändern. Hier kann außerdem der komplette Account gelöscht werden, um die Daten des Nutzers zu löschen.

Vervollständige dein Profil

Anzeigename *

Email *

jonas.rosen@gmail.com

Geburtstag

28.10.2022



Postleitzahl

90342

Speichern

Account löschen

Zeige meine Profilseite

Abbildung (7.8) – Benutzereinstellungen

7.3.5. Profilseite & Profilbilder

Auf der Profilseite des Nutzers kann er seine persönlichen Daten einsehen und bearbeiten. Hier können Interessen und eine Beschreibung hinzugefügt werden. Außerdem kann er sein Profilbild ändern, indem er auf das Profilbild oder auf den Knopf mit der Kamera klickt. Dies wird dann im Firebase Storage gespeichert und in der Datenbank verlinkt.

Das Ganze ist so aufgebaut, dass Nutzer zwischen einer Vorschau, also der Sicht, die auch andere Nutzer von seinem Profil sehen und der Bearbeitungssicht unterscheiden können.

Jonas

Interessen:

Interessen auswählen

Fußball Sport Technik Sport / Werkzeug / ...

Beschreibung:

Ich bin Jonas, ein Entwickler dieses Projekts.
In meiner Freizeit mache ich gerne Sport und interessiere mich allgemein für Technik und technologische Fortschritte.

Vorschau

Abbildung (7.9) – Persönliches Profil

7.3.6. Blockierte Nutzer

Um die Privatsphäre der Nutzer zu schützen, können diese andere Nutzer blockieren. Blockierte Nutzer können dann nicht mehr auf das Profil des Blockierenden zugreifen und auch keine Nachrichten mehr schreiben.

Über die Profilseite kann ein Nutzer einen anderen Nutzer blockieren.

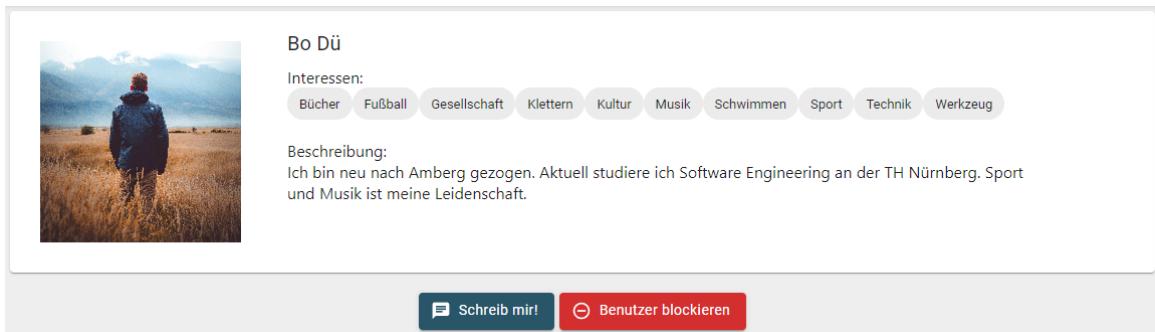


Abbildung (7.10) – Blockieren eines Nutzers

Blockierte Nutzer kann man dann in der Anwendung unter dem Menüpunkt *Blockiert* einsehen. Dort wird jeder Nutzer oder Verein aufgeführt, welcher blockiert ist. Mit einem Klick auf das „X“ wird der Block aufgehoben.

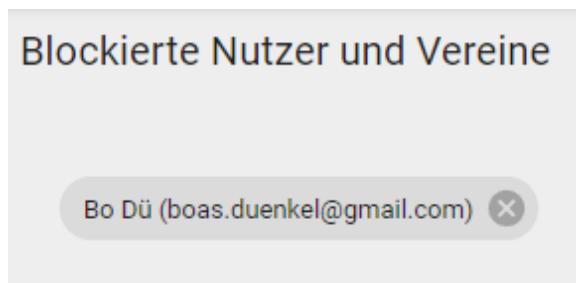


Abbildung (7.11) – Blockierte Nutzer

7.4. Beiträge

Beiträge sind ein anderes wichtiges Feature von „Digital Dahoam“. Hiermit können Nutzer Informationen austauschen, die für andere Nutzer interessant sein könnten. Beiträge können von allen Nutzern erstellt werden, die sich registriert haben. Es gibt folgende Typen von Beiträgen:

- **Anfrage:** Hier können Nutzer eine Anfrage stellen, die dann von anderen Nutzern beantwortet werden kann. Anfragen können auch benutzt werden, wenn bestimmte Gegenstände im Haushalt fehlen, z. B. ein bestimmtes Werkzeug oder ein bestimmtes Lebensmittel.

- **Angebot:** Hier können Nutzer ein Angebot erstellen, das dann von anderen Nutzern angenommen werden kann. Dies kann wie Ebay-Kleinanzeigen benutzt werden, um überflüssige Gegenstände zu verkaufen.
 - **Information:** Hier können Nutzer Informationen teilen, die für andere Nutzer interessant sein könnten.
 - **Veranstaltung:** Hier können Nutzer Veranstaltungen erstellen, die dann von anderen Nutzern besucht werden können.

7.4.1. Beiträge erstellen

Beiträge können mit einem Klick auf den Button *Beiträge erstellen* erstellt werden. Hier öffnet sich ein Pop-up, welchem der Nutzer den Titel, die Beschreibung, den Beitragstyp und die Kategorie des Beitrags eingeben kann. Außerdem werden für jeden Beitrag das Startdatum, also ab wann der Beitrag gültig ist und angezeigt wird, und das Enddatum, also bis wann der Beitrag gültig ist und angezeigt wird, festgelegt. Für Veranstaltungen gibt es zusätzlich noch den Ort und das Datum. Sobald auf *absenden* geklickt wird, wird der Beitrag in der Datenbank gespeichert und auf der Startseite angezeigt.

Erstelle einen Beitrag

Gib deinem Beitrag einen Titel *

Beitragstyp auswählen *

Gültigkeit ab
26.01.2023 

Gültigkeit bis
26.01.2024 

Schreibe deinen Beitrag *

Beitragskategorien auswählen

AbbruchAbsende

Abbildung (7.12) – Beiträge erstellen

7.4.2. Alle Beiträge

Unter *Alle Beiträge* findet man alle Beiträge, die ein valides Gültigkeitsdatum haben. Diese werden mit Titel, Beschreibung, Kategorie und Beitragstyp angezeigt. Für Veranstaltungen gibt es zusätzlich noch den Ort und das Datum.



Abbildung (7.13) – Beitrag

Ein Problem, was es während der Implementierung zu lösen galt, war die richtige Filterung der Beiträge. Die hier verwendete Sortierfunktion (siehe II) wurde dann für die restlichen Listen verwendet und die gefilterten Beiträge dort nur noch verfeinert. Diese musste folgendes erfüllen:

- Eigene Beiträge werden immer angezeigt.
- Beiträge, die nicht mehr gültig sind, werden nicht angezeigt.
- Beiträge von blockierten Nutzern werden nicht angezeigt.
- Beiträge müssen nach Datum sortiert werden.

Mit dem Klick auf die 3 kleinen Punkte in der rechten oberen Ecke öffnet man das Beitragsmenü. Hier gibt es folgende Punkte:

- **Beitrag zum Merkzettel:** Hinzufügen eines Inhalts zum Merkzettel.
- **Details:** Anzeigen der Details des Beitrags. (siehe A.2)
- **Zum Autor:** Direkter Link zum Profil des Autors.
- **Nachricht an Autor:** Direkter Link zur Nachrichtenfunktion, um dem Autor eine Nachricht zu schicken.

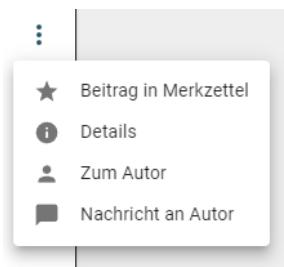


Abbildung (7.14) – Beitragsmenü

7.4.3. Profilseite

Auf der Profilseite kann der Nutzer seine Beiträge anzeigen. Hier werden auch abgelaufene Beiträge angezeigt und grau hinterlegt.

The screenshot shows a user profile for 'Jonas'. It includes a photo of a smiling man, a section for interests ('Interessen') with 'Fußball', 'Sport', and 'Technik' selected, and a bio describing him as a developer of the project. A green 'Vorschau' (Preview) button is visible. Below the profile is a section titled 'Beiträge' (Posts) with an update from 'Jonas' about post functionality.

Updates zu Posts
Information von Jonas
Ich habe die Post Funktionalität upgedated.
Nun werden die Datumswerte besser gespeichert und alles wird neu geladen, sobald Posts erstellt oder gelöscht werden.

Abbildung (7.15) – Beiträge des Nutzers

7.4.4. Merkzettel

Beiträge, die für den Merkzettel markiert sind erscheinen dort, und können dort auch wieder entfernt werden. Außerdem werden sie nach Beitragstyp gruppiert.

The screenshot shows a 'Merkzettel' (Note-taking) interface. On the left is a sidebar with filters: 'Alle', 'Anfrage', 'Angebot', 'Information', and 'Veranstaltung'. The main panel displays a note titled 'Brandwache 24.01' about a fire station shift at the Feuerwehr Amberg on 1/24/2023, categorized under 'Technik'.

Abbildung (7.16) – Merkzettel

7.4.5. Marktplatz

Auf dem Marktplatz können Personen und Beiträge durchsucht werden. Durch den Filter auf Kategorien, Informationen oder Veranstaltungen findet man schnell den richtigen Beitrag.

The screenshot shows the 'Marktplatz' (Marketplace) interface. On the left, there is a sidebar with filters: 'Beitrag' (selected), 'Person', 'Ersteller' (Creator), 'Markt', 'Informationen', 'Veranstaltungen', and 'Kategorien'. The main area displays two posts:

- Geräteinventur 2023**
Information von Freiwillige Feuerwehr Amberg
Hallo zusammen,
bis Ende Januar findet unsere Geräteinventur statt.
Werkzeug
- Brandwache 24.01**
Veranstaltung von Freiwillige Feuerwehr Amberg
Datum: 1/24/2023
Ort: Feuerwehr Amberg
Brandwache der Feuerwehr Amberg
Technik

Abbildung (7.17) – Marktplatz

7.4.6. Dashboard

Auf der Startseite werden aktuelle Beiträge aus der Nähe angezeigt, sowie die Veranstaltungen, die der Nutzer auf dem Merkzettel hat.

The screenshot shows the 'Dashboard' interface. It has two main sections:

- Beiträge aus deiner Nähe**
Geräteinventur 2023
Information von Freiwillige Feuerwehr Amberg
Hallo zusammen,
bis Ende Januar findet unsere Geräteinventur statt.
Werkzeug
- Anstehende Veranstaltungen**
Brandwache 24.01
Veranstaltung von Freiwillige Feuerwehr Amberg
Datum: 1/24/2023
Ort: Feuerwehr Amberg
Brandwache der Feuerwehr Amberg
Technik

Abbildung (7.18) – Dashboard

7.5. Chat

Ein wichtiges Tool für das soziale Netzwerk „Digital Hometown“ ist der Chat, da hier die Personen in Kontakt treten und sich austauschen können.

Auf technischer Ebene sind für einen Chat mehrere Technologien erforderlich, wodurch die Umsetzung nicht trivial ist. Neben den Laden der Bereits gesendeten Nachrichten und dem Absenden von Nachrichten, ist es für eine gute Usability notwendig, dass neue Nachrichten sofort bei allen Chatteilnehmern sichtbar sind. Diese Anforderung kann in effizienter Weise durch die Verwendung von bidirektionalen WebSocket-Verbindungen ermöglicht werden.

7.5.1. Firebase Realtime Database für den Chat

Für den Chat wird die Firebase Realtime Database (Realtime DB) verwendet. Dadurch wird die Implementierung deutlich vereinfacht, da die entsprechende JavaScript Bibliothek einfach in die React Anwendung integriert werden kann. Eine Echtzeitsynchronisation zwischen den Clients (Browseranwendungen der User) und der Datenbank kann mit geringem Aufwand umgesetzt werden. Die Realtime DB ist eine NoSQL Datenbank. Die Firebase DB wird in diesem Projekt ausschließlich für den Chat verwendet und hat dabei zwei Hauptwurzelemente (s. Abbildung 7.19) verwendet werden. Bei dem Design wurde darauf geachtet, dass die einzelnen Hauptwurzelemente keine zu große Verschachtelung aufweisen, um eine gute Performanz zu gewährleisten.



Abbildung (7.19) – Hauptwurzelemente der Realtime DB

7.5.2. Autorisierung der Nachrichten und Chaträume

Eine wichtige Anforderung für die Implementierung einer Chatfunktionalität ist es, dass Nachrichten nur von Mitgliedern des jeweiligen Chatraums gelesen werden können. Dies wird mit den Realtime DB Regeln umgesetzt. Ein Auszug hiervon ist in Listing 7.3 dargestellt. Kurz zusammengefasst ermöglicht diese Konfiguration, dass nur Nachrichten von Benutzern gelesen und geschrieben werden können, die ein Mitglied des Chatraumes sind.

```

1  {
2      "rules": {
3          "messages": {
4              "$roomUid": {
5                  ".read": "root.child('rooms/' + $roomUid + '/members').hasChild(auth
6                      .uid)",
7                  ".write": "root.child('rooms/' + $roomUid + '/members').hasChild(
8                      auth.uid)"
9              }
10         }
11     }
12 }
```

```

7     },
8     "messages": {
9       ".indexOn": "sendAt"
10    }
11  },
12  ...
13 }
14 }
```

Listing (7.3) – Realtime DB Regeln für Chatnachrichten

7.5.3. Benutzung des Chats

Im folgenden wird anhand von Screenshots die Benutzung des Chats beschrieben. Abbildung 7.20 zeigt die Standardansicht des Chats. Auf der linken Seite sind die Chaträume aufgelistet. Die Chaträume sind in zwei Kategorien unterteilt. Zum einen gibt es die Chaträume, die mit Einzelpersonen erstellt wurden und zum anderen die Chaträume mit mehreren Personen (Gruppenchats). In der Liste der Chaträume werden Gruppenchats mit einem Gruppenicon hervorgehoben.

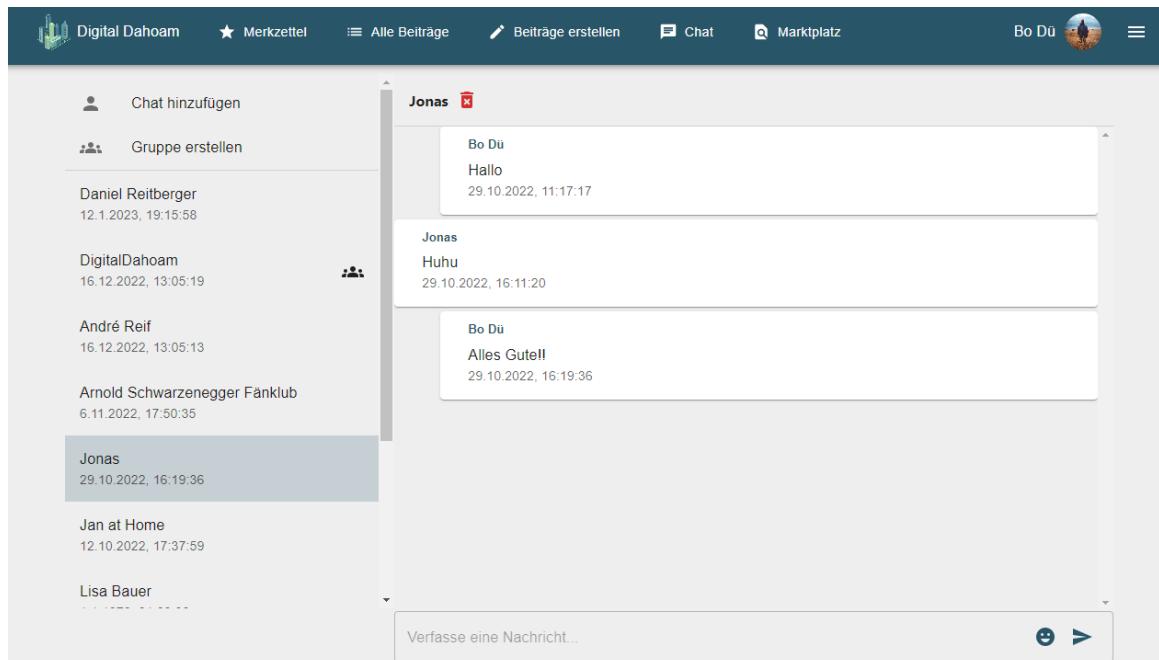


Abbildung (7.20) – Ansicht des Chats in der Anwendung

Über den Button „Chat hinzufügen“ wird die in Abbildung 7.21 dargestellte Ansicht geöffnet, wo alle Personen der Plattform angezeigt werden. Hier kann ein neuer Chat mit einer Einzelperson erstellt werden.

Das Hinzufügen von Gruppenchats erfolgt über den Button „Gruppe hinzufügen“. Der Ablauf sieht hier wie folgt aus. Zuerst wird automatisch ein neuer Chatraum erstellt. Wird

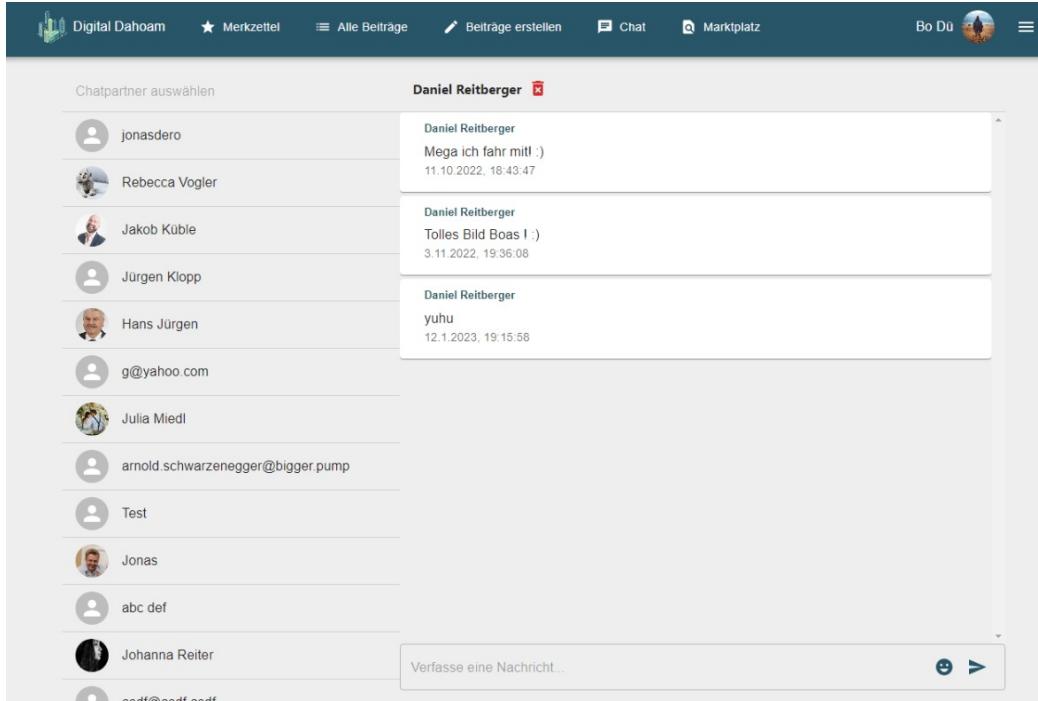


Abbildung (7.21) – Hinzufügen von Chats mit Einzelpersonen

dieser ausgewählt, lässt sich über einen Button die in Abbildung 7.22 dargestellte Ansicht öffnen. Hier können die Mitglieder des Chatraumes hinzugefügt werden, indem die entsprechenden Personen aus der Liste ausgewählt werden. Es besteht zudem die Möglichkeit, der Chatgruppe einen anderen Namen zu geben.

7.5.4. Verknüpfung von Beiträgen mit dem Chatraum des Beitragautors

Die Motivation des in diesem Absatz beschriebenen Features wird mit folgendem Fallbeispiel erläutert.

Angenommen, ein Benutzer meldet sich an, da er an Fußball interessiert ist und gerne mit Menschen aus seiner Nachbarschaft gerne zum Fußball spielen treffen möchte. Er sieht nun einen Beitrag, der genau diesem Bedürfnis entspricht. Wie in Abbildung 7.23 dargestellt, kann er nun über den Button „Nachricht an Autor“ direkt zum Chatraum des Beitragautors gelangen. Dort kann er den Beitragautor anschreiben und sich mit ihm zum Fußball spielen treffen.

Im Hintergrund muss dafür der gemeinsame Chatraum ermittelt werden. Falls dieser noch nicht existiert, muss er erstellt werden. Wird dann der Nutzer auf die Chatseite weitergeleitet, öffnet sich automatisch der entsprechende Chatraum.

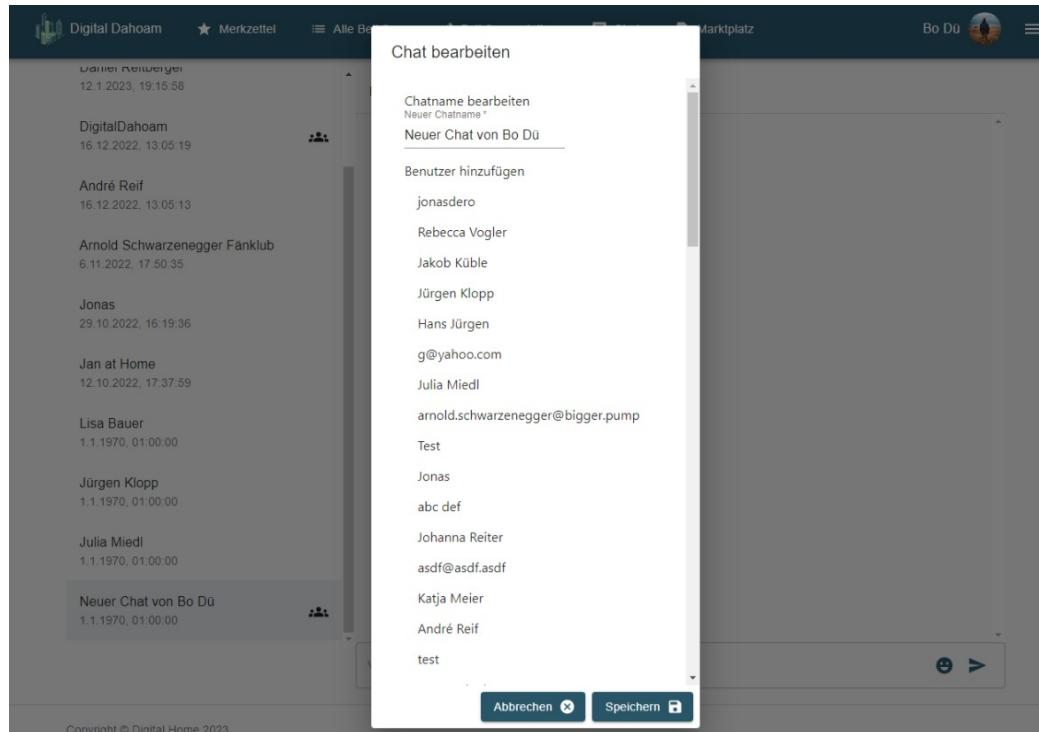


Abbildung (7.22) – Chatgruppe umbenennen und Mitglieder hinzufügen



Abbildung (7.23) – Möglichkeit von einem Beitrag direkt zum Chatraum des Beitragautors zu gelangen

8. Testing

9. Fazit

A. Anhang

I. Bilder

```
blocked: []  
dateOfBirth: 1666990800000  
desc: "Ich bin Jonas, ein Entwickler dieses Projekts. In meiner Freizeit mache ich  
gerne Sport und interessiere mich allgemein für Technik und  
technologische Fortschritte."  
displayName: "Jonas"  
email: "jonas.rosen@gmail.com"  
favoritePosts: ["mFCoElDAPOTxuJCCbJCE", "...]  
friends  
id: "DCNu11SgrTYrBWu1bHxPjhYvIIQ2"  
interests: ["Fußball", "Sport", "Tech..."]  
isOrg: false  
photoURL: "https://firebasestorage.googleapis.com/v0/b/digital-  
dahoam.appspot.com/o/images%2FDCNu11SgrTYrBWu1bHxPjhYvIIQ2?  
alt=media&token=a90ec139-31e7-4e65-bc97-a8b150ecc187"  
postCode: 90342
```

Abbildung (A.1) – Benutzerdatensatz in Firestore



Abbildung (A.2) – Beitragsdetails

II. Code

```
1 const filtered = posts
2 .filter((post) => !currentUser?.blocked?.includes(post.author.id))
3 .filter((post) => {
4     // check validity
5     if (currentUser?.id === post.author.id) {
6         return true
7     }
8
9     if (!(post.validityStart && post.validityEnd)) {
10        // no validity set
11        return true
12    }
13    console.log(
14        moment(post.validityStart).toDate().getTime(),
15        moment(post.validityStart).toDate().getTime() <= new Date().getTime(),
16    )
17    if (
18        moment(post.validityStart).toDate().getTime() >= new Date().getTime() &&
19        moment(post.validityEnd).toDate().getTime() <= new Date().getTime()
20    ) {
21        return true
22    } else {
23        return false
24    }
25 })
```

Filterfunktion der Beiträge

Abbildungsverzeichnis

4.1.	Startseite von Facebook	8
4.2.	Startseite von nebenan.de	11
4.3.	Startseite von Spontacts	13
4.4.	Seitenstruktur einer Website	17
4.5.	Elemente einer Website	18
4.6.	Beispiel eines Rastersystems	19
4.7.	Aufbau eines fixen und flexiblen Layouts	20
4.8.	Beispielhafte Darstellung von LFW und HFW	22
4.9.	Interactionsschritte an einem Beispiel: Wireframe - Mockup - Prototyp	23
4.10.	Figmas-Arbeitsumgebung	24
4.11.	Startseite (DHT)	25
4.12.	Marktplatz (DHT)	25
5.1.	Übersicht der technischen Softwarearchitektur	26
6.1.	Vergleich Downloadzahlen verschiedener Frontend Frameworks [4]	30
7.2.	Übersicht eines Benutzerprofils	37
7.3.	Anmeldemaske	38
7.8.	Benutzereinstellungen	42
7.9.	Persönliches Profil	42
7.10.	Blockieren eines Nutzers	43
7.11.	Blockierte Nutzer	43
7.12.	Beiträge erstellen	44
7.13.	Beitrag	45
7.14.	Beitragsmenü	45
7.15.	Beiträge des Nutzers	46
7.16.	Merkzettel	46
7.17.	Marktplatz	47
7.18.	Dashboard	47
A.1.	Benutzerdatensatz in Firestore	A
A.2.	Beitragsdetails	B

Tabellenverzeichnis

6.1. Funktionen von Github	33
6.2. Funktionen von Jira	35

Literaturverzeichnis

- [1] Typescript, “JavaScript With Syntax For Types.” 2022. [Online]. Available: <https://www.typescriptlang.org/>
- [2] ——, “Documentation - React,” Jan. 2023. [Online]. Available: <https://www.typescriptlang.org/docs/handbook/react.html>
- [3] React, “React – A JavaScript library for building user interfaces,” 2022. [Online]. Available: <https://reactjs.org/>
- [4] NPM, “@angular/core vs react vs vue | npm trends,” Jan. 2023. [Online]. Available: <https://npmtrends.com/@angular/core-vs-react-vs-vue>
- [5] Google, “Design - Material Design,” May 2021. [Online]. Available: <https://m2.material.io/design>
- [6] MUI, “MUI: The React component library you always wanted,” 2022. [Online]. Available: <https://mui.com/>
- [7] G. Firebase, “Firebase,” 2022. [Online]. Available: <https://firebase.google.com/>
- [8] G. Authentication, “Firebase Authentication,” 2022. [Online]. Available: <https://firebase.google.com/docs/auth?hl=de>
- [9] F. Realtime Database, “Firebase Realtime Database,” 2022. [Online]. Available: <https://firebase.google.com/docs/database?hl=de>
- [10] G. Firestore, “Firestore,” 2022. [Online]. Available: <https://firebase.google.com/docs/firestore?hl=de>
- [11] G. CloudStorage, “Cloud-Speicher für Firebase | Cloud Storage for Firebase,” 2022. [Online]. Available: <https://firebase.google.com/docs/storage?hl=de>
- [12] Github, “Build software better, together,” 2022. [Online]. Available: <https://github.com>
- [13] Vercel, “Develop. Preview. Ship. For the best frontend teams – Vercel,” 2022. [Online]. Available: <https://vercel.com/>
- [14] Atlassian, “DH Board - Agile board - Jira,” Jan. 2023. [Online]. Available: <https://digitaldahoam.atlassian.net/jira/software/projects/DH/boards/1>