



Fakultät Elektrotechnik Feinwerktechnik Informationstechnik

Projekt: Digital Dahoam

Gruppenprojektbericht im Studiengang Software Engineering

vorgelegt von

Daniel Reitberger, André 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.

Akronyme

DH Digital Hometown.

FB Facebook.

HFW High-Fidelity-Wireframe.

LFW Low-Fidelity-Wireframe.

UI User Interface.

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
3.2.1. JIRA	5
3.2.2. Sprint Planning mit JIRA	9
3.3. Sprintübersicht	11
3.3.1. Sprintübersicht	11
4. Analyse	14
4.1. Wettbewerbsanalyse	14
4.1.1. Ermittlung der Hauptwettbewerber	14
4.1.2. Konkurrenzprofil	15
4.1.3. Kurzdarstellung	16
4.1.4. Strategische Ausrichtung	22
4.1.5. Zusammenfassung	22
4.2. Anforderungsanalyse	23
4.2.1. Value Proposition Canvas	24
4.2.2. Personas und Zielgruppe	25
4.2.3. Profile	25
4.2.4. Abgeleitete Features	28
4.3. Visuelle Grundstrukturen	31
4.3.1. Grundlagen Seitenstruktur	31
4.3.2. Grundlagen Layout	33
4.3.3. Rastersystem	33
4.3.4. Grundlagen Wireframes	35
4.3.5. Anwendung	37
4.4. Usabilityanalyse	40
5. Architektur	41

6. Technologien	42
6.1. Grundlage für Technologieentscheidung	42
6.2. TypeScript	43
6.2.1. Einsatz im Projekt	43
6.2.2. Grund für Technologieentscheidung	43
6.3. React	43
6.3.1. Allgemeines in Bezug auf die Implementierung mit React	43
6.3.2. Einsatz im Projekt	45
6.3.3. Grund für Technologieentscheidung	45
6.4. Material-UI	46
6.4.1. Einsatz im Projekt	46
6.4.2. Grund für Technologieentscheidung	46
6.5. Firebase	47
6.5.1. Firebase Authentication	47
6.5.2. Firebase Realtime Database	47
6.5.3. Firebase Cloud Firestore	47
6.5.4. Firebase Storage	48
6.5.5. Einsatz im Projekt	48
6.5.6. Grund für Technologieentscheidung	48
6.6. Github	48
6.6.1. Einsatz im Projekt	48
6.6.2. Grund für Technologieentscheidung	49
6.7. Vercel	49
6.7.1. Einsatz im Projekt	49
6.7.2. Grund für Technologieentscheidung	49
6.8. Jira	50
6.8.1. Einsatz im Projekt	50
6.8.2. Grund für Technologieentscheidung	50
7. Implementierung	51
7.1. Landing-Page	51
7.2. Profil	52
7.2.1. Anmeldung & Registrierung	53
7.3. Authentifizierung	54
7.3.1. Verwendung des „Firebase-Authentication“-Services	54
7.3.2. Zugriff auf die Profilinformationen	54
7.3.3. Authentifizierungsprozess	55
7.3.4. Persönliches Accountmanagement	58
7.3.5. Profilseite & Profilbilder	58
7.3.6. Blockierte Nutzer	59

7.4. Beiträge	59
7.4.1. Beiträge erstellen	60
7.4.2. Alle Beiträge	61
7.4.3. Profilseite	62
7.4.4. Merkzettel	62
7.4.5. Marktplatz	63
7.4.6. Dashboard	63
7.5. Chat	64
7.5.1. Firebase Realtime Database für den Chat	64
7.5.2. Autorisierung der Nachrichten und Chaträume	64
7.5.3. Benutzung des Chats	65
7.5.4. Verknüpfung von Beiträgen mit dem Chatraum des Beitragautors	66
8. Testing	68
8.1. Definition Testen	68
8.2. Aufgabengebiet als Tester	68
8.3. Testen im agilen Projekt	69
8.3.1. Wann in Scrum Testen	70
8.4. Test-Prozess	70
8.4.1. Entwicklertest	71
8.4.2. InkrementTest	72
8.4.3. Release-Test	73
8.5. Umsetzung im Projekt	75
8.5.1. Testen gegen User-Stories	75
8.5.2. Regressions-Test	77
8.5.3. Feld-Test	77
8.5.4. Exploratives-Testen	79
9. Fazit	81
A. Anhang	A
I. Bilder	A
II. Code	C
Abbildungsverzeichnis	E
Tabellenverzeichnis	G
Literaturverzeichnis	H

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)
- **André 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

In diesem Abschnitt wird der Bereich des Projektmanagements und der Projektplanung innerhalb des Digital Home Town Teams näher erläutert. Als Hilfsmittel wurde hierfür hauptsächlich die Software JIRA von Atlassian verwendet. Hierbei wird auch auf die einzelnen Tasktypen und deren Bedeutung eingegangen. Des Weiteren wurde ein eigener Arbeitsablauf für das Entwicklerteam definiert. Abschließend wird auf die Sprint-Planung und die spätere Verwendung einer sog. User Story Map eingegangen.

3.2.1. JIRA

JIRA ist ein Tool zur Unterstützung von Projektmanagement-Prozessen, das von vielen Unternehmen und Organisationen genutzt wird. Mit JIRA können Projekte geplant, verwaltet und kontrolliert werden, indem Aufgaben, Meilensteine und Prozesse visualisiert und koordiniert werden. Es bietet auch Funktionen zur Zusammenarbeit und Kommunikation innerhalb des Projektteams sowie zur Nachverfolgung von Fortschritten und Problemen. JIRA kann auch individuell angepasst werden, um den spezifischen Bedürfnissen eines Projekts gerecht zu werden. Im Zuge des Projekts wurden mehrere solcher Anpassungen vorgenommen sowie Teaminterne Regeln festgelegt. Diese sollen im Folgenden Abschnitt näher erläutert werden.

3.2.1.1. Tasks

Die Besonderheit an JIRA ist die Möglichkeit zur Aufteilung von mehreren Arbeiten in sogenannte Tasks. Ein Task hat normalerweise eine eindeutige Identifikationsnummer, eine kurze Beschreibung der Aufgabe, eine Zuweisung an einen Verantwortlichen, einen Status sowie ggf. zusätzliche Informationen wie z.B. Verknüpfungen mit anderen Tasks.

The screenshot shows a JIRA interface. On the left, there's a list of tasks in the backlog:

- DH-00 - Als Person möchte ich nach Beiträgen suchen können die mich interessieren um Kontakte zu knüpfen
- DH-01 - Durch Klick auf einen 'Entwickeln' Button im Header (Vgl. Dashboard/User) wird die Marketplace Seite (vgl. Marketplace) geöffnet
- Viele!
- Ergebnis
- Beiträge die zu den Filtereinstellungen passen werden angezeigt
- Über
- User kann Beitrag mit entsprechenden Filtern sehen.
- Anforderungen der Zukunft sind erfüllt.
- Funktion wurde getestet.

On the right, a detailed view of a task is shown:

Details

- Zugehörige Person: Jan
- Mit zuweisen: Keine
- Schwierigkeit: Keine
- Sprint: Keine
- Story point estimate: Keine
- Ungleichgewicht Schätzung: 0,00
- Zifferierung: Keine Ziel protokolliert
- Entwicklung:
 - Branch erstellen
 - Commit erstellen
- Autor: Andre Reif

At the bottom, it says: Erstellt 18. Mai 2022 um 16:13, Aktualisiert 17. Dezember 2022 um 14:59, Letzte Aktivität 17. Dezember 2022 um 14:59.

Abbildung (3.1) – Beispiel für einen Task im JIRA

Einzelne Tasks können in sog. Epics geclustert werden, um so die Aufgaben und langfristige Planung besser strukturieren zu können. Im Folgenden ist eine Übersicht der Epics innerhalb des Projekts sowie deren Fortschritt zu sehen.

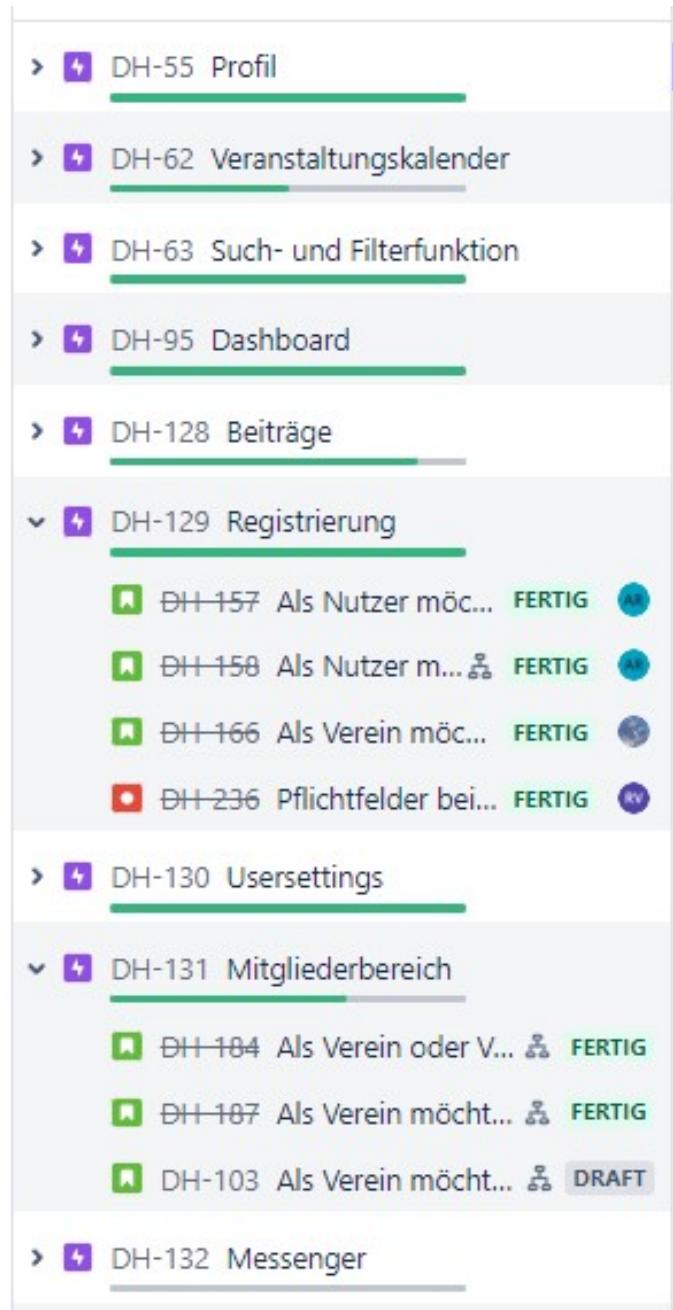


Abbildung (3.2) – Epic Übersicht des Projekts

Für die einzelnen Tasks wurden jeweils 4 verschiedene Typen unterschieden.

User-Story

Eine User Story beschreibt eine neue Nutzeranforderung, also ein Feature, dass umgesetzt werden muss das der Benutzer die entsprechende Tätigkeit ausführen kann. Diese werden per Definition nach dem folgenden Prinzip geschrieben:

Als Nutzer möchte ich <Auszuführende Tätigkeit> um <Vorteil oder Intention der ausgeführten Tätigkeit>.

Eine User Story hat den Folgenden Aufbau:

- **Titel** User Story nach Definition.
- **Wie?** Kurze Beschreibung wie die Funktion umgesetzt werden soll.
- **Ergebnis?** Beschreibung was passiert, wenn die Funktion ausgeführt wird.
- **DoD** Definition of Done, also eine Definition, wann der Task als erledigt gilt.

Sub-Task

Sub-Tasks werden in der Regel dazu verwendet, um besonders aufwendige User Stories und Tasks in kleinere Arbeitseinheiten zu unterteilen. Für Digital Home Town wurden die Sub-Tasks verwendet, um Task-spezifische Anforderungen zu dokumentieren und hervorzuheben.

Functional-Task

Für unterstützende Arbeiten, die nicht direkt mit der Entwicklung der Plattform in Verbindung stehen, wurden sog. Functional Tasks eingeführt. Diese sind in der Regel nicht näher definiert oder Beschrieben und sind Anhand des Titels nachvollziehbar.

Typ	Schlüssel	Zusammenfassung
☒	DH-194	Anpassungen an Wireframes müssen vorgenommen werden.
☒	DH-173	Anpassung der Passwort-Vergessen Email - Text/Design
☒	DH-170	Texte für Landing Page definieren
☒	DH-169	Begrifflichkeiten definieren
☒	DH-168	Usability Vergleich mit bestehenden Systemen
☒	DH-167	Umfrage Vorbereiten
☒	DH-98	Wireframe für Dashboard

Abbildung (3.3) – Beispiele für Functional Tasks

Bug

Ein Bug stellt einen gefundenen Fehler auf der Plattform da. Dieser spiegelt sowohl einen Fehlerbericht als auch einen neuen Arbeitsauftrag dar. Innerhalb des Entwicklerteams war es üblich, dass zumeist der Verursacher des Bugs auch für dessen Behebung verantwortlich war. Ein Bug durchläuft allerdings denselben Arbeitsablauf wie eine User-Story und nach den gleichen Standards getestet.



Abbildung (3.4) – Beispiel für ein Bug Ticket

Ein Bug besteht in der Regel aus:

- **Titel** Kurze Beschreibung des Fehlers
- **Beschreibung** Detaillierte Beschreibung des Fehlers und des Fehlerhergangs
- **Test Schritte** Schritt-für-Schritt Angaben über den Ablauf des Tests
- **Verhalten** Erwartetes und Tatsächliches Verhalten

3.2.1.2. Arbeitsablauf

In der folgenden Abbildung ist der Allgemeine Arbeitsablauf für ein einzelnes JIRA Ticket zu sehen.

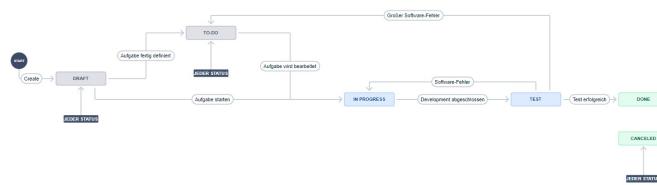


Abbildung (3.5) – Darstellung des Ablaufs eines einzelnen JIRA Tickets

Ein einzelner Task wurde zunächst mit dem Status „Draft“ erstellt. Dies wurde im Allgemeinen so interpretiert, dass das Ticket zwar angelegt wurde, jedoch nicht vom Team so

akzeptiert wurde, dass jeder die Aufgabe verstanden hat. Dies wurde bei der Sprint Planung bzw. dem Task Refinement während der Planung Besprochen, ggf. angepasst, und dann festgelegt.

Nachdem ein Task den Status „ToDo“ erhält, kann dieser in einem Sprint eingeplant und entsprechend bearbeitet werden. Der Status wechselt demnach zu „In Progress“.

Nach der Bearbeitung wird ein Ticket mit Hilfe des Flags „Test“ zum Testen freigegeben. Das Bedeutet der Code läuft in der Entwicklungsumgebung und kann von einem Tester bearbeitet werden. Ob im Fehlerfall ein Task zurück in den Status „ToDo“ oder „In Progress“ versetzt wurde, lag in der Schwere des Fehlers und im Ermessen des jeweiligen Testers. Falls der Test erfolgreich war, wurde der Task auf den Status „Done“ verschoben und war somit abgearbeitet.

Aufgrund sich ändernder Anforderungen wurde später der „Canceled“ Status eingeführt, um ggf. bereits begonnene Tasks abbrechen zu können.

Dieser Ablauf gilt in der Regel für jede User Story sowie jeden Bug. Die Stati „Draft“ und „Test“ hatten für Functional Tasks und Sub Tasks jedoch keine tiefergehende Bedeutung und konnten einfach übersprungen werden.

3.2.2. Sprint Planning mit JIRA

Die Sprints wurden in der Regel vorausgeplant. Dabei galt die Regel: der nächste Sprint steht am Ende des aktuellsten Sprints weitestgehend fest, der übernächste Sprint ist zu 50% geplant.

Während der Planungsmeetings wurden die geplanten Tasks und Bugs besprochen und auf deren Umsetzbarkeit und Priorität geprüft. Aufgrund der unterschiedlichen Kenntnisse und Fertigkeiten der Entwickler war es jedoch schwierig den Aufwand einzelner Tasks verallgemeinert für das gesamte Entwicklungsteam einzuschätzen. Zuzüglich dazu mussten Urlaubs- und Abwesenheitszeiten bei der Planung berücksichtigt werden, sodass die definierten Sprint Ziele auch erreicht werden konnten.

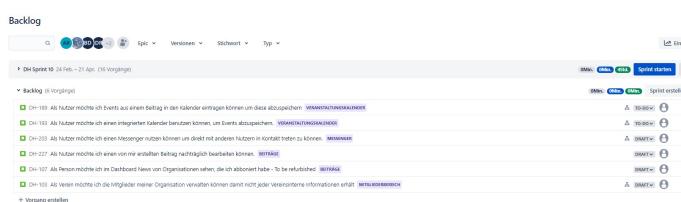


Abbildung (3.6) – Backlog Funktion im JIRA

Zur Planung der Sprints wurde die in der obigen Abbildung zu sehende Backlog Funktion verwendet. Hier sind zum sowohl die nicht geplanten Tasks zu sehen als auch die einem Sprint bereits zugeordnetem Task und können auch bei Bedarf entsprechend umgeplant werden.

So war es nicht unüblich, dass nachdem das Sprintziel bereits vor dem Ende des Sprints erreicht wurde, noch offene Bugs oder kleinere Functional Tasks dem Sprint hinzugefügt und abgearbeitet wurden. Zum Sprintende begonnene Arbeitsaufträge wurden grundsätzlich in den nachfolgenden Sprint verschoben und fertiggestellt.

3.2.2.1. User Story Map

Als weiteres Tool zur Projektplanung wurde der Ansatz des sogenannten User Story Mapping verwendet. Per Definition ist eine User Story Map eine Technik zur Visualisierung von Benutzeranforderungen und -funktionen in einer hierarchischen Struktur.

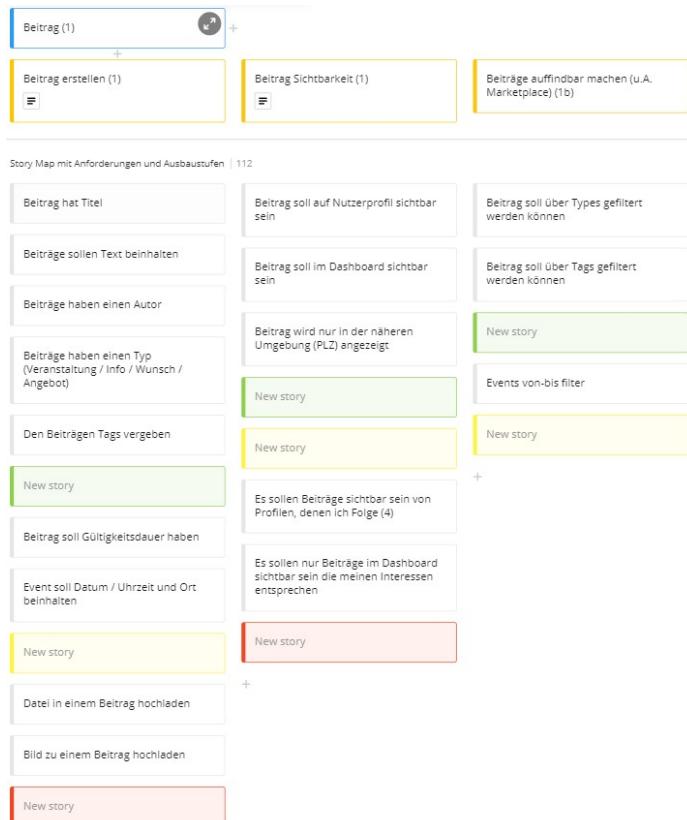


Abbildung (3.7) – Beiträge in der User Story Map

In der obigen Abbildung ist die User Story Map für einige Features der Beiträge zu finden. So gibt es zur User Story „Als Nutzer möchte ich einen Beitrag erstellen können um mich der Community mitteilen zu können“ verschiedene Anforderungen. Diese stehen unterhalb des jeweils gelben Blocks.

Die Anforderungen jeweils wurden dann in 3 Iterationsstufen Priorisiert:

- **Bis Grün** Minimum viable product, also die Minimalanforderungen die Umgesetzt werden sollen

- **Bis Gelb** Nice-to-Have, Anforderungen die umgesetzt werden sollen, aber nicht hoch Priorisiert werden
- **Bis Rot** Wird implementiert falls genügend Zeit dafür da ist

Aus dieser Map wurden dann die einzelnen User Story Tasks im JIRA angelegt und entsprechend in den Sprints verplant.

3.3. Sprintübersicht

3.3.1. Sprintübersicht

3.3.1.1. Sprint 0

Der „nullte“ Sprint stellt den Start des Projektes dar. Nachdem die Rollen innerhalb des Scrum Teams festgelegt und deren Aufgaben definiert wurden, wurden Kommunikations-, Management- und Dokumentationsregeln festgelegt. Es wurde eine Wettbewerbsrecherche und eine Ideensammlung ausgearbeitet, sowie die Vision für das Projekt überlegt.

3.3.1.2. Sprint 1

Im ersten Sprint wurde die Vision weiter ausformuliert und Ideen zur Umsetzung der Vision gesammelt. Parallel dazu hat sich das Entwicklerteam auf die Verwendung von Technologien geeinigt bzw. eingearbeitet werden. Daraus entstand ein erstes Architekturkonzept für die Plattform. Anhand der gesammelten Ideen konnten bereits erste User-Storys erstellt werden. Die Kundenanforderungen sowie die Zielgruppe wurden mit Hilfe des Value Proposition Canvas identifiziert und analysiert.

3.3.1.3. Sprint 2

Das Ergebnis des zweiten Sprints umfasste bereits ein bestehendes Grundgerüst der Website. Die Registrierung sowie eine Seite in der eine Feature Übersicht zu sehen war wurden bereits umgesetzt. Über Personas wurde das Kundensegment weiter spezifiziert und daraus Features abgeleitet. Zeitgleich wurden als elementare Elemente der Plattform Private und Vereinsprofile spezifiziert.

3.3.1.4. Sprint 3

Während des Sprint 3 wurde die Struktur und Oberflächenarchitektur der Website geplant, wobei der Schwerpunkt auf dem Basisfeature „Profil“ lag. Die zentrale Frage lautete, wie ein Benutzerprofil in unserem Netzwerk aussehen sollte und welche Inhalte es enthalten sollte. Das Ergebnis war eine Liste aller Anforderungen, die für die Erstellung eines Benutzerprofils erforderlich sind. Es wurde eine Übersicht über alle Features erstellt. Die Funktionen, die

sich im vorherigen Sprint überlegt wurden, wurden kategorisiert und priorisiert. Auch die Profile von Vereinen und deren Besonderheiten wurden in Betracht gezogen.

3.3.1.5. Sprint 4

Im vierten Sprint wurden die zuvor besprochene Oberflächenarchitektur und Seitenstruktur in Wireframes umgesetzt. Dies ermöglichte es, erste Designentscheidungen zu treffen und die Inhalte und Funktionen unserer Website genauer zu diskutieren. Durch die in Sprint 0 angesprochene Wettbewerbsanalyse konnten wir das Alleinstellungsmerkmal für unsere Plattform „Digital Dahoam“ identifizieren: Das Tagsystem. Nun konnten Anforderungen für das Dashboard und eine Filterfunktion über Tags festgelegt werden. Gleichzeitig wurde das Anlegen von Benutzerprofilen entwicklungsseitig umgesetzt.

3.3.1.6. Sprint 5

Als weitere Basisfunktion vom Digital Home Town wurde die Chatfunktion rudimentär umgesetzt. Im Sprint wurde ebenfalls an der Weiterentwicklung des Layouts gearbeitet und erste Aspekte der Benutzerfreundlichkeit in die Entwicklung einbezogen. Durch den Ansatz des User-Story-Mapping wurde eine Übersicht als Grundlage für die Projektplanung erarbeitet. Zusätzlich wurden weitere Funktionen für Vereine definiert und priorisiert.

3.3.1.7. Sprint 6

Im Sprint 6 wurde einerseits die Beitragsfunktion implementiert. Andererseits wurden Texte und Bilder für die Landingpage definiert, um sowohl interessierte Privatpersonen als auch Vereine über die Inhalte und Möglichkeiten innerhalb von Digital Dahoam zu informieren. Darüber hinaus wurde die Benutzerfreundlichkeit mit bestehenden Plattformen abgeglichen. Wireframes für das Profil und das Dashboard wurden weiter ausgearbeitet. Ein Konzept für einen Feldtest unserer bestehenden Plattform wurde ebenfalls erarbeitet.

3.3.1.8. Sprint 7

Während des siebten Sprints wurden grundlegende Accounteinstellungen implementiert und der Registrierungsprozess weiter verbessert. Die im vorherigen Sprint entwickelten Beiträge wurden interaktiver gestaltet. Um Feedback zum bestehenden System zu erhalten und Features besser priorisieren zu können, wurde eine Usability-Umfrage erstellt und ausgewertet. Darüber hinaus wurden Layouts für zuvor nur funktional umgesetzte Funktionen definiert.

3.3.1.9. Sprint 8

In diesem Sprint wurden das Merkzettel-Feature sowie der Marktplatz implementiert. Beste hende Funktionen wurden zudem weiter optimiert und getestet. Zur Befüllung des Netzwerks wurden Dummy-Datensätze erstellt, die später im Netzwerk eingespielt werden können. In

einem Abstimmungstermin mit dem Kunden wurde der aktuelle Stand von Digital Dahoam vorgestellt und weitere Umsetzungen abgeleitet.

3.3.1.10. Sprint 9

Im neunten und letzten Sprint wurden noch offene Bugs behoben und bereits begonnene Features final umgesetzt. Die Ergebnisse wurden dann im Zuge einer Abschlusspräsentation dem Kunden präsentiert.

4. Analyse

4.1. Wettbewerbsanalyse

Um die Bedürfnisse zukünftiger Nutzer zu erfüllen und das Produkt langfristig wettbewerbsfähig zu halten, ist es unerlässlich, kontinuierlich alle relevanten Kundenanforderungen zu ermitteln und zu priorisieren. Zusätzlich ist es wichtig, sich mit Wettbewerbern, ihren Produkten und Strategien auseinanderzusetzen, um die eigenen Marktchancen zu bewerten. Eine Analyse gibt zudem Aufschluss darüber, welche Features bereits als grundlegend erwartet werden und welche Möglichkeiten es gibt, um sich von anderen Anbietern zu differenzieren.

Die Wettbewerbsanalyse wurde in mehreren Schritten durchgeführt:

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

4.1.1. Ermittlung der Hauptwettbewerber

Für die Identifikation der bedeutendsten Mitbewerber ist es wichtig, die Hauptmerkmale und Ziele des eigenen Dienstes zu kennen und mit potenziellen Wettbewerbern abzugleichen. Dabei sollte berücksichtigt werden, dass es in der digitalen Welt nicht immer Wettbewerber gibt, die exakt den gleichen Dienst anbieten oder dieselbe Strategie verfolgen. Dennoch können mittelfristig vollwertige Konkurrenzprodukte entstehen, die zum Zeitpunkt der Analyse nur wenige gemeinsame Parallelen oder Features aufweisen.

Die langfristige Ausrichtung von Digital Hometown (DH) besteht darin, den lokalen/ regionalen Raum und seine sozialen Verflechtungen zu stärken, indem der digitale Austausch zwischen allen Personen und Institutionen gefördert wird, die das örtliche Dasein widerspiegeln. Die Schwerpunkte von DH sind wie folgt:

1. Das Vernetzen von Personen, die gleiche Interessen oder ähnliche Bedürfnisse haben und einen Austausch von Informationen, Waren und Dienstleistungen anstreben oder Unterstützung und Hilfestellung suchen bzw. anbieten.
2. Die Stärkung lokaler Vereine und Dienstleistungen, damit diese ihre Angebote gezielter darstellen und leichter mit Interessenten in Kontakt treten können.

3. Die Unterstützung öffentlicher Einrichtungen, um einen einfacheren Zugang zu amtlichen Informationen, Terminen und anderen Ressourcen zu ermöglichen.

Zugleich verpflichtet sich DH dazu, allen Nutzern einen barrierefreien Zugang zu ermöglichen.

Basierend auf den Zielen von DH wurden verschiedene Webdienste identifiziert, die sich auf den lokalen Raum und den Austausch, die Vernetzung oder die Zusammenführung von Privatpersonen konzentrieren. Zunächst wurden als etablierte und aktive Vertreter in mindestens einem der genannten Bereiche identifiziert:

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

Im Folgenden werden beispielhaft nur die Vertreter aus dem Bereich Social Networking näher betrachtet, zu denen Facebook, Nebenan.de und Spontacts zählen.

4.1.2. Konkurrenzprofil

Für eine eingehende Analyse der ermittelten Wettbewerber ist es notwendig, individuelle Konkurrenzprofile für jeden Marktteilnehmer zu erstellen. Diese Gegenüberstellung des eigenen Selbstprofils bietet eine transparente Bewertungsbasis, die wertvolle Erkenntnisse über Stärken, Potenziale und strategische Ziele liefert.

Folgende Punkte werden bei der Konkurrenzanalyse gesondert betrachtet:

- Fokus des Dienstes und Zielgruppe
- Möglichkeiten der Selbstdarstellung und des Austauschs
- Benachrichtigungen über Änderungen und Neuigkeiten
- Entdecken von Inhalten

4.1.3. Kurzdarstellung

4.1.3.1. Facebook

Facebook (FB)¹ ist ein weltweit bekannter und etablierter Dienst, der es Nutzern ermöglicht, sich mit Menschen aus aller Welt zu vernetzen, in Kontakt zu bleiben und Inhalte aus ihrem Leben mit anderen zu teilen. Ursprünglich richtete sich der Fokus von FB auf Studenten, aber im Laufe der Jahre hat sich die Zielgruppe verbreitert und ist heute sehr heterogen.

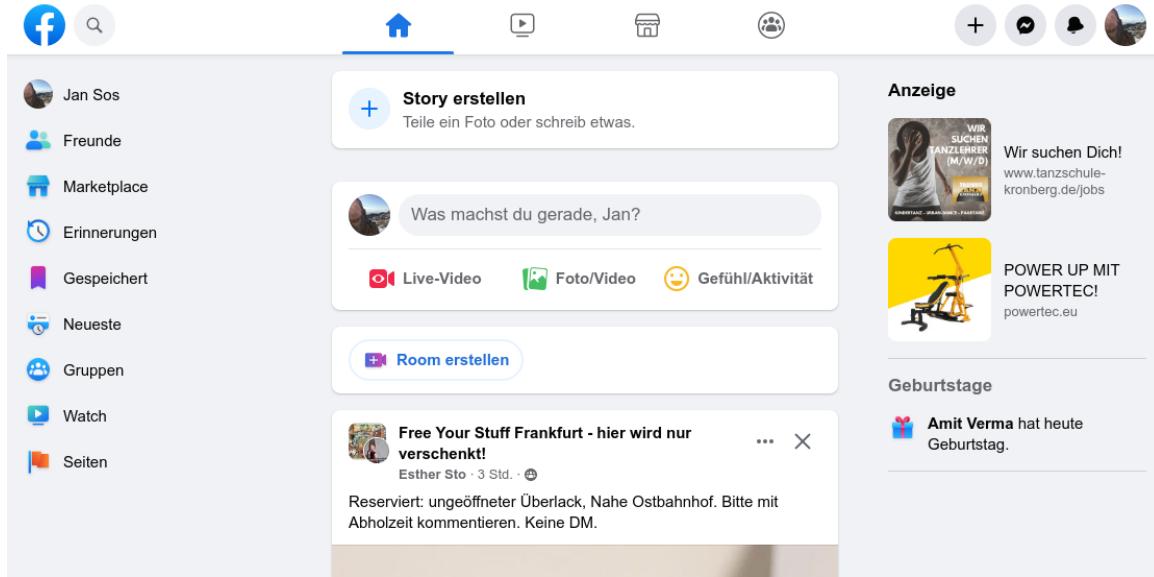


Abbildung (4.1) – Startseite von Startseite von FB

Die Nutzung von FB ist sehr vielfältig und hängt von den Bedürfnissen des einzelnen Nutzers ab. Neben reinen Darstellungs- und Austauschmöglichkeiten bietet FB eine breite Palette von Anwendungen an, wie z.B. integrierte Video-, Game- und Datingportale sowie ein eigenes BezahlSystem. Auf diese speziellen Anwendungen wird im weiteren Verlauf nicht weiter eingegangen.

Selbstdarstellung Für die Selbstdarstellung von Organisationen oder Personen bietet FB ein eigenes Profil an, auf dem Bilder, persönliche Informationen und Vorlieben wie Musik öffentlich dargestellt werden können. Die Sichtbarkeit dieser Inhalte kann individuell angepasst werden. Neben den statischen Informationen enthält das Profil auch eine Pinnwand, auf der ältere und aktuelle Beiträge listenmäßig aufgeführt werden.

Formen des Austausches Beiträge ermöglichen den interaktiven Austausch und das Teilen von Momenten. Jeder Nutzer kann sie verfassen und auf seiner eigenen sowie auf fremden

¹vgl. Facebook 2023 [1]

Pinnwänden veröffentlichen. Ereignisse wie das Hochladen von neuen Bildern generieren automatisch einen Beitrag, um Bekannte, Follower und andere Personen über Neuigkeiten im Profil zu informieren. Beiträge lassen sich neben einfachem Text auch mit Bildern oder Videos sowie mit Tags (Orte, GPS, Veranstaltungen, Personen, etc.) spezifizieren. Andere Nutzer können Beiträge kommentieren oder bewerten.

FB-Gruppen ermöglichen den Austausch mit mehreren Personen zu gemeinsamen Themen. Ähnlich wie Profile haben Gruppen einen statischen Teil, in dem Administratoren eine Kurzbeschreibung der Gruppe veröffentlichen. Eine Pinnwand ermöglicht den Mitgliedern die Kommunikation untereinander. Durch das Erstellen von Events können Gruppenmitglieder gemeinsame Treffen oder Aktivitäten planen.

Gruppen werden sehr häufig und für verschiedene Themen genutzt. Insbesondere im städtischen Raum dienen sie zum Beispiel dem Verschenken von ungenutzten Dingen, dem Knüpfen neuer Kontakte in einer neuen Stadt oder dem Treffen von Menschen in der Nähe mit ähnlichen Hobbies. Auch der Austausch über diverse Themen in unterschiedlichen Bereichen auf regionaler, nationaler und internationaler Ebene ist weit verbreitet.

Der Austausch materieller Gegenstände findet hauptsächlich auf einem digitalen Marktplatz statt. Die geschalteten Anzeigen enthalten eine Überschrift, eine freie Beschreibung, Bilder, Angaben zum Zustand, zur Preisvorstellung, zum Ort und zum Herausgeber der Anzeige. Der Verfasser muss bei der Erstellung der Anzeige eine vordefinierte Kategorie auswählen, um die Auffindbarkeit zu erleichtern. Neben der direkten Kontaktaufnahme besteht für Interessenten die Möglichkeit, die Anzeige zu merken, mit einem Kontakt zu teilen oder einen Alarm zu erstellen, sobald ähnliche Produkte angeboten werden.

Auf FB können Kontakte gepflegt werden, indem man sich gegenseitig in eine Freundesliste aufnimmt. Über diese Liste können Inhalte selektiv verteilt werden, so dass sie nur für *Freunde* sichtbar sind. Zusätzlich werden *Freunde* über das Dashboard gezielt informiert, wenn ein *Freund* einen neuen Beitrag erstellt hat.

Für den direkten und privaten Austausch bietet FB einen Chat an, in dem sowohl 1:1- als auch Gruppenchats möglich sind. Die Kommunikation erfolgt in Echtzeit und die Nachrichten können wie Beiträge Bilder, Links und andere Inhalte enthalten. In 1:1-Gesprächen besteht auch die Möglichkeit, Sprach- und Videoanrufe zu tätigen.

Neuigkeiten Um bei der Vielzahl der Beiträge und Reaktionen von Freunden oder Gruppen den Überblick zu behalten, bietet FB ein Dashboard an, auf dem Neuigkeiten in Form einer endlosen und unsortierten Liste angezeigt werden. Das Dashboard enthält auch kommerzielle Werbung und Beiträge von Gruppen, die der Nutzer noch nicht abonniert hat, aber von Interesse sein könnten.

Neben dem Dashboard gibt es auch eine Notifications-Seite, die einen gezielten Fokus auf das Wesentliche ermöglicht. Hier werden kurze Benachrichtigungen wie Geburtstage von *Freunden*, bevorstehende Events, neue Beiträge aus abonnierten Gruppen sowie Reaktionen auf eigene oder kommentierte Beiträge angezeigt.

Recherche FB bietet eine Suchfunktion, die es Nutzern ermöglicht, nach Personen, Gruppen und anderen Inhalten zu suchen. Die Suchergebnisse können durch verschiedene Filter wie Art des Inhalts, Verfasser, Gruppen, Zeitraum und mehr genauer spezifiziert werden.

Zusätzlich können Nutzer Beiträge und öffentliche Profile als *Bookmarks* sichern. Die *Bookmarks* können dann auf einer dafür vorgesehenen Seite je nach Typ aufgelistet und verwaltet werden.

4.1.3.2. Nebenan.de

Nebenan.de² ist eine Plattform, die im Jahr 2015 gestartet wurde und wie FB darauf abzielt, Menschen zu vernetzen (Abb. 4.2). Der Unterschied zu FB besteht darin, dass sich Nebenan.de ausschließlich auf die unmittelbare Nachbarschaft des Nutzers konzentriert.

The screenshot shows the homepage of Nebenan.de. At the top, there's a navigation bar with the logo 'nebenan.de', a button 'nebenan.de fördern', and social sharing icons for Jan, a profile picture, and a bell. A search bar contains the placeholder 'Suchbegriff eingeben...' with a magnifying glass icon. On the left, a sidebar lists categories: 'Startseite' (highlighted in green), 'Veranstaltungen', 'Marktplatz', 'Läden & Services', 'Gruppen', 'Postfach', 'Deine Nachbarn', 'Deine Nachbarschaft', and 'Nachbarn einladen'. Below the sidebar, there's a link to 'Ausloggen' and other site links like 'Organisationen', 'Hilfe', 'Magazin', 'Team', 'Presse', 'Kontakt', 'Impressum', 'Datenschutz', and 'Nutzungsbedingungen'. The main content area features a large search bar with the placeholder 'Was möchtest du mitteilen?'. Below it, a section for 'Bornheim Ost & Umgebung' shows 'Aktive Nachbarn: 5226'. A post from a user named Vera C. is displayed, asking for 'Guinness-Gläser?' with a sad face emoji. The post includes the text: 'Für eine Party Ende März suche ich Guinness-Gläser. Hat jemand welche, die ich ausleihen dürfte? Würde mich sehr freuen (bearbeitet: vor 3 Stunden)'. Below the post is a blurred image of a Guinness pint.

Abbildung (4.2) – Startseite von nebenan.de

Die Zielgruppe umfasst alle Interessengruppen, die in der Nachbarschaft des Benutzers vertreten sind, wie z.B. Anwohner, Vereine, Firmen und Organisationen. Die Nachbarschafts-

²Vgl. nebenan.de 2023 [2]

grenze wird vom System anhand der eigenen Adresse des Nutzers definiert und kann nicht individuell angepasst werden.

Selbstdarstellung Für die Darstellung existieren auf Nebenan.de zwei Arten von Profilen. Die erste Profilart gilt für Einzelnutzer. Hierbei können sich Anwohner über ein Kurzprofil mit Foto, einem Freitext sowie ihren Interessen und Angeboten vorstellen. Zur Auswahl der Interessen und Angebote gibt es fest definierte Vorschläge, um zu beschreiben, was der Anwohner in die Nachbarschaft einbringen kann. Dazu zählen beispielsweise das *Annehmen von Paketen*, das *Blumengießen*, die *Reparatur von Fahrrädern* oder das *Gesellschaft leisten*. Das Profil zeigt auch die Aktivitäten des Anwohners an, wie beispielsweise beigetretene nebenan-Gruppen und die Anzahl der erhaltenen virtuellen Dankeschöns.

Organisationen und Läden/Services, die vor Ort angesiedelt sind und zum alltäglichen Leben in der Nachbarschaft beitragen, nutzen die zweite Profilart. Diese Profile enthalten im Vergleich zu den Anwohnerprofilen deutlich mehr Informationen und müssen beim Anlegen einer Kategorie zugeordnet werden, wie z.B. *Restaurant*, *Reisen*, *Sport*, *Political Party* oder *Nachbarschaftsinitiative*. Neben grundlegenden Informationen wie Name, Adresse, Kontaktdataen und Öffnungszeiten können auch zukünftige Veranstaltungen, Gesuche, Bekanntmachungen, das Verkaufsangebot und weitere Informationen hinterlegt werden. Anwohner können auf den öffentlichen Profilen ihre eigenen Erfahrungen mit der Organisation oder dem Laden teilen und Empfehlungen aussprechen.

Formen des Austausches Nebenan.de nutzt den Beitrag als zentrales Mittel für den Austausch zwischen Anwohnern. Jeder Beitrag kann von allen Anwohnern kommentiert und positiv bewertet werden. Beiträge werden bei der Erstellung einer Kategorie zugeordnet, um zu kennzeichnen, ob es sich um ein *allgemeines Thema*, ein *Gesuch*, ein *Angebot*, eine *Empfehlung* oder eine *Veranstaltung* handelt. Jede Kategorie wird in mehrere aufeinander aufbauende Unterkategorien unterteilt, um die Beiträge besser einordnen zu können. Abhängig von der Art des Beitrags werden diese entweder auf dem öffentlichen Dashboard, dem Event-Feed oder dem Marktplatz angezeigt. Ein Beitrag besteht aus einem Titel, einem Freitext und optionalen Bildern. Bei Veranstaltungen können weitere Felder wie Datum und Ort hinzugefügt werden.

Die Beiträge auf dem Marktplatz werden neben der Hauptkategorie *Angebot* noch weiter in die Unterkategorien wie *Hilfe*, *Schenken*, *Verleihen* oder *Tauschen* unterteilt. Verkaufsangebote können zudem weiter spezifiziert werden, indem sie einer Angebotskategorie wie *Essen*, *Baby & Kinder* oder *Haustiere* zugeordnet werden. Alle Angebote sind in einer Liste verfügbar und können nach Kategorie gefiltert oder mithilfe der globalen Suchfunktion durchsucht werden.

Um den Austausch unter Menschen mit gemeinsamen Interessen zu erleichtern, bietet nebenan.de die Möglichkeit, Gruppen zu erstellen. Dabei kann der Zweck der Gruppe durch einen aussagekräftigen Namen und eine Beschreibung genauer erläutert werden. In diesen Gruppen können Mitglieder verschiedene Arten von Beiträgen wie Mitteilungen, Suchanfragen, Angeboten oder Veranstaltungshinweisen veröffentlichen. Diese Beiträge erscheinen auf der Gruppenpinnwand und können von allen Gruppenmitgliedern kommentiert werden.

Neben der Gruppenfunktion gibt es auch eine Chatfunktion für den direkten Austausch mit einzelnen Nutzern. Mit dieser Funktion können Nutzer neben reinem Text und Emojis auch Fotos und Empfehlungen verschicken.

Neuigkeiten Für die Anzeige von neuen Beiträgen steht ein Nachbarschafts-Dashboard zur Verfügung. Hier werden alle Arten von Beiträgen nach dem Datum der letzten Änderung sortiert angezeigt. Der Austausch mit der Nachbarschaft findet hauptsächlich über das Dashboard statt, wo neue Beiträge entdeckt und beantwortet werden können.

Zusätzlich werden alle Beiträge, an denen man aktiv teilgenommen hat und bei denen neue Ereignisse aufgetreten sind, nochmals separat in einem Benachrichtungs-Feed aufgelistet, um einen besseren Überblick zu gewährleisten. Darüber hinaus werden im Benachrichtigungs-Feed alle zukünftigen Veranstaltungen in der Nachbarschaft angezeigt.

Recherche Die nebenan-Suche bietet eine Möglichkeit zur Suche von Beiträgen jeglicher Art. Interessante Beiträge oder solche, die wichtige Informationen für den Benutzer enthalten, können als Lesezeichen gespeichert werden. Diese Lesezeichen können je nach Art des Beitrages unter *Feed* oder *Marktplatz* eingesehen und bei Bedarf gelöscht werden.

4.1.3.3. Spontacts

Spontacts³ konzentriert sich im Gegensatz zu FB und Nebenan.de ausschließlich auf die Organisation von Freizeitaktivitäten für Menschen in der gleichen Region (vgl. Abb. 4.3). Die Plattform richtet sich insbesondere an Personen, die Gleichgesinnte für bestimmte Aktivitäten suchen und dabei neue Kontakte knüpfen möchten.

Selbstdarstellung Das Nutzerprofil auf Spontacts ähnelt dem anderer vergleichbarer Anwendungen und beinhaltet grundlegende Informationen wie den Namen, ein Profilbild, das Alter und den Wohnort. Es können jedoch auch weitere Angaben wie präferierte Kontakte für *Freizeit*, *Sport*, *Reisen*, *Tanzen* und *Dating* oder zusätzliche Profilabschnitte (Hobbies, Interessen/Vorlieben usw.) hinzugefügt werden. Ein weiterer Abschnitt zeigt vergangene und zukünftige Aktivitäten des Nutzers sowie dessen Gruppenmitgliedschaften an.

³Vgl. Spontacts 2023 [3]

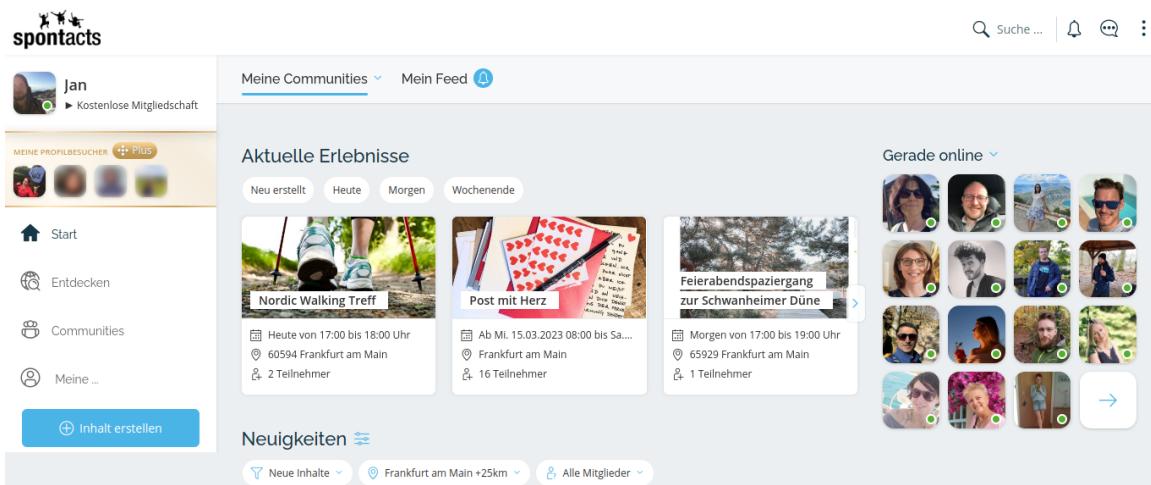


Abbildung (4.3) – Startseite von Spontacts

Formen des Austausches Zur Interaktion mit anderen Benutzern können Beiträge in Spontacts erstellt werden. Diese Beiträge können je nach Art als einfacher Beitrag, als Aktivität oder als Frage/Diskussion verfasst werden und werden dann in Communities, Gruppen oder gegebenenfalls in Foren angezeigt. Wie gewohnt können sie kommentiert, geliked und geteilt werden.

Die Communities sind eine Art Oberkategorie, die einen bestimmten Themenbereich (*Ausgehen & Party, Essen & Trinken, Natur & Umwelt usw.*) beschreiben und jedem Beitrag zugeordnet werden. Beim Erstellen einer Gruppe muss daher immer eine passende Community ausgewählt werden. Im Gegensatz zu den anderen vorgestellten Diensten sind Gruppen bei Spontacts das zentrale Feature der Anwendung. Sie ähneln in ihrem Funktionsumfang den Gruppen auf FB.

Über die Beiträge hinaus können die Nutzer auch über den Chat eine private Unterhaltung führen. In diesem Chat können ausschließlich Textnachrichten ausgetauscht werden.

Recherche Um Beiträge, Veranstaltungen, Gruppen, Mitglieder und mehr zu entdecken, bietet Spontacts eine Suchfunktion mit spezifischen Filteroptionen je nach dem gesuchten Typ. Der Nutzer kann neben den allgemeinen Einstellungen wie Suchbegriff, Umkreis und Erstelltdatum auch objektspezifische Kriterien wie Veranstaltungsdatum, Alter und ähnliches hinzufügen, um die Suche zu verfeinern.

Mitglieder können in die Kontakt- oder Merkliste aufgenommen werden, wenn sie für den Nutzer von Interesse sind. Die Merkliste ist eine private Liste, in die jedes Mitglied aufgenommen werden. Die Kontaktliste hingegen ist öffentlich und erfordert eine Bestätigung der Anfrage durch die betreffende Person.

Zusätzlich besteht die Möglichkeit, anderen Mitgliedern zu folgen, um ihre Aktivitäten besser verfolgen zu können. Wer wem folgt, kann in jedem Benutzerprofil eingesehen werden.

Neuigkeiten Um innerhalb der Plattform auf dem Laufenden zu bleiben, gibt es wie bei anderen Portalen ein Dashboard für Gruppen/Communities sowie einen Benachrichtigungs-Feed. Im Dashboard werden in der Rubrik *Neuigkeiten* alle anstehenden Veranstaltungen und aktuellen Inhalte der Communities dargestellt. Unter der Rubrik *Mein Feed* können hingegen alle Veränderungen in beigetretenen Gruppen eingesehen werden, wie beispielsweise neue Mitglieder und Veranstaltungen.

Der Benachrichtigungs-Feed enthält Informationen über neue Kontaktanfragen, Beiträge aus den Gruppen und Neuigkeiten über die Plattform.

4.1.4. Strategische Ausrichtung

Die drei Hauptwettbewerber weisen viele Gemeinsamkeiten auf, unterscheiden sich jedoch in ihrer strategischen Ausrichtung voneinander.

FB ist eine sehr aktive Plattform, die sich darauf konzentriert, Menschen mit beliebigen Interessen, Themen und Bedürfnissen zu vernetzen, ohne dass der Nutzer dabei auf einen bestimmten geografischen Raum beschränkt ist. In letzter Zeit konnte keine intensive Weiterentwicklung der Plattform beobachtet werden, da sich das Unternehmen hauptsächlich der Entwicklung von zukünftigen Produkten im 3D-Umfeld widmet. Es ist daher lang- und mittelfristig keine Veränderung zu erwarten.

Nebenan.de ist der größte Konkurrent von DH. Die Plattform ermöglicht es den Nutzern, sich in ihrer Nachbarschaft auszutauschen und zu vernetzen, wobei der Fokus auf der Stärkung der Gemeinschaft liegt. Eine große Herausforderung für Nebenan.de besteht darin, eine lebendige Community aufzubauen, die alle gesellschaftlichen Schichten und Altersklassen ansprechen.

Im Gegensatz dazu konzentriert sich Spontacts stark darauf, Menschen im privaten Bereich für gemeinsame Aktivitäten zu vernetzen. Die Plattform hat eine aktive Community, in der Nutzer Inhalte erstellen und an allen Veranstaltungen teilnehmen können, ohne harte lokale Einschränkungen wie bei nebenan.de. Darüber hinaus werden die Inhalte der Seite von lokal ansässigen Moderatoren betreut, die zusätzliche Veranstaltungen organisieren und durchführen. Strategisch setzt Spontacts zunehmend auf kostenpflichtige Accounts, die den Nutzern zusätzliche Funktionen wie verbesserte Filterfunktionen bieten sollen.

4.1.5. Zusammenfassung

Nach der Analyse der Konkurrenz wurde deutlich, dass die untersuchten Dienste ähnliche Funktionen wie Profile, Gruppen und Chats bereitstellen. Die Unterschiede zwischen den

Features liegen eher in Nuancen, die von den jeweiligen Betreibern individuell gestaltet werden.

Die Plattformen unterscheiden sich jedoch in Bezug auf ihre Motivationen, Zielgruppen und geografischen Nutzerbereiche stark voneinander. Während einige Plattformen sich auf bestimmte Nachbarschaften beschränken, decken andere globale Communities ab.

Im Gegensatz dazu konzentriert sich DH gezielt auf die Region/Gemeinde. Dadurch füllt DH die Lücke zwischen FB als weltweitem Akteur und nebenan.de mit Fokus auf die Nachbarschaft.

Ein weiterer Aspekt, der sich bei einem direkten Vergleich deutlich zeigt, ist die unterschiedliche Benutzerfreundlichkeit und Nutzerführung auf den verschiedenen Plattformen. Teils sind Inhalte schwer auffindbar und die zugrunde liegenden Konzepte sind nicht immer selbsterklärend. Um sicherzustellen, dass DH für alle Altersgruppen verständlich ist, ist es unerlässlich, eine Plattform zu entwickeln, die einfach strukturiert und intuitiv zu bedienen ist.

Die Analyse zeigt anhand von nebenan.de und Spontacts, dass der Aufbau und die Etablierung eines sozialen Netzwerks viele Herausforderungen mit sich bringen. Als Vorreiter hat sich FB in der Gesellschaft bereits stark etabliert und bietet mit einer breiten Palette von Features vielseitige Nutzungsmöglichkeiten für unterschiedliche Bedürfnisse.

Insbesondere für neue Plattformen stellt der Aufbau einer Community eine große Herausforderung dar. Als Newcomer müssen sie ohne bereits vorhandene Community und Content meist mit ähnlichen Funktionen wie Facebook die Nutzer überzeugen, dass ihre Plattform einen spürbaren Mehrwert im Vergleich zu Facebook bietet. Gerade bei Plattformen, bei denen die Nutzer ausschließlich die Inhalte erstellen, gestaltet sich der Aufbau einer Community als besonders zäh, da neue Nutzer aufgrund mangelnder Inhalte schnell wieder abwandern können.

Daher ist es essentiell, dass kontinuierlich neue Inhalte zur Verfügung stehen, um sich zu etablieren. Spontacts setzt hierfür gezielt Moderatoren ein, um seinen Nutzern regelmäßig Angebote zu präsentieren. Alternativ kann die Bindung an ein Portal auch durch Angebote von Dritten erfolgen, wie zum Beispiel durch regelmäßige Informationen von Vereinen, Stadtverwaltungen oder Zeitungen, wie es bei DH geplant ist. Zusätzliche Funktionen, wie ein Buchungsportal, können den Nutzern auch bei bestimmten Aktivitäten unterstützen.

4.2. Anforderungsanalyse

Die Anforderungsanalyse ist ein entscheidender Schritt im Software-Entwicklungsprozess, der dazu beiträgt, die Bedürfnisse der Nutzer zu identifizieren und zu verstehen. Um jedoch die Bedürfnisse der Nutzer vollständig zu verstehen, ist es notwendig, eine detaillierte Analyse der Geschäftsanforderungen durchzuführen. Das Value Proposition Canvas bietet dabei eine effektive Methode zur Identifizierung der Kundenbedürfnisse und zur Entwicklung von

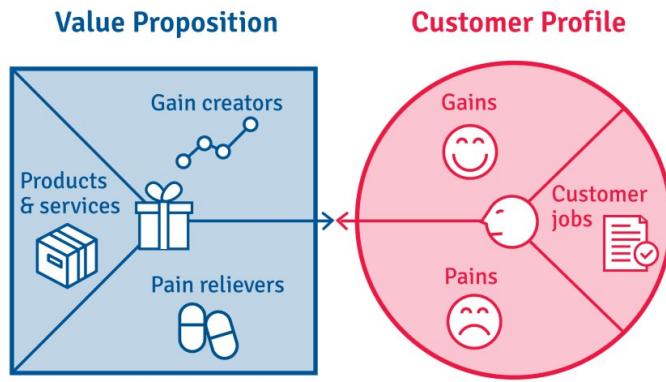


Abbildung (4.4) – Value Proposition Canvas

wertvollen Produkten und Services. Dieses Kapitel befasst sich mit der Anforderungsanalyse mit dem Hilfsmittel des Value Proposition Canvas.

4.2.1. Value Proposition Canvas

Die Anforderungsanalyse nach dem Value Proposition Canvas ist ein Prozess, bei dem die Bedürfnisse und Anforderungen der Kunden identifiziert und dokumentiert werden, um sicherzustellen, dass das Unternehmen Produkte oder Dienstleistungen anbietet, die einen hohen Nutzen für die Kunden bieten.

Der Value Proposition Canvas besteht aus zwei Hauptbereichen:

1. Der Kundenprofilbereich: In diesem Bereich identifizieren Sie die Bedürfnisse und Anforderungen Ihrer Zielgruppe und erstellen ein detailliertes Profil der Kundenperspektive.
2. Der Value-Map-Bereich: In diesem Bereich legen Sie fest, wie Ihr Produkt oder Ihre Dienstleistung den Kundenbedarf erfüllt und welche Vorteile es im Vergleich zu anderen Lösungen bietet.

Vorgehensweise

Um eine erfolgreiche Analyse nach dem Value Proposition Canvas durchzuführen, sollten folgende Schritte beachtet werden:

1. Kundenprofilbereich
 - Identifizierung der Bedürfnisse und Herausforderungen der Zielgruppe
 - Definierung der Kundenperspektive und Erstellen eines Profils für die Zielgruppe
2. Value-Map-Bereich

- Definition der Produkte oder Dienstleistungen, die angeboten werden sollen
- Festlegung der Vorteile, die das Produkt gegenüber anderen Konkurrenten bieten könnte
- Sicherstellung, dass die Bedürfnisse der Zielgruppe erfüllt werden

4.2.2. Personas und Zielgruppe

Zu Beginn des Projekts wurde bereits die initiale Zielgruppe definiert. Da eine Social Media Plattform grundsätzlich ein potentiell sehr breites Spektrum an Nutzern besitzt, wurde die Entscheidung der Zielgruppe anhand dem bestmöglichen erreichen vieler Menschen definiert: Junge Erwachsene und Vereine - unter Betrachtung eines eher ländlicheren Landkreises.

Junge Erwachsene sind in der Regel affin und offener für den Umgang mit Technik, bei Erfolg der Plattform könnten diese wiederum über persönliche Kontakte und Werbung weitere Nutzer auf die Plattform ziehen. Vereine besitzen in der Regel bereits ein Netzwerk und bieten gleichzeitig die Möglichkeit, Menschen mit ähnlichen Interessen miteinander zu verknüpfen.

4.2.3. Profile

Um die Zielgruppen abilden zu können, wurden insgesamt 4 Profile definiert, diese wurden nach dem Value Proposition Canvas analysiert und daraus jeweils Features abgeleitet. Diese sollen im Folgenden beschrieben werden:

Musikverein

Beim Musikverein wird angenommen, es handelt sich um einen existierenden Verein mit einer funktionierenden Vorstandsschaft und einer lebhaften Mitgliedschaft. Dieser hat nach dem Value Proposition Design die folgenden Eigenschaften:

Tabelle (4.1) – Musikverein Customer Profile

Jobs	Pains	Gains
Öffentlichkeitsarbeit	-Werbung schalten ist teuer -Zeitaufwand	-Kostenlose Werbung -Neue Mitglieder -Höherer Bekanntheitsgrad in der Gemeinde
Austausch mit anderen Mitgliedern	-Abstimmung mit Mitgliedern zur Terminfindung -Planung von Events und Veranstaltungen	-Management des Vereins über ein Tool -Bessere Effizienz

Tabelle (4.2) – Musikverein Value Proposition

Products & Services	Pain Reliever	Gain Creator
Öffentliches Profil	-Kostenlose Anmeldung -Schnelle Registrierung	-lokale Präsenz auf Gemeindeebene
Teilen von Events und Beiträgen	-Öffentliche Beiträge	-Über Tags können Interessierte Präziser an Informationen gelangen
Gruppenchats	-Anlegen von Gruppenunterhaltungen	-Einteilung in Arbeitsgruppen für verschiedene Veranstaltungen -Alles auf einer Plattform

Alleinstehender Erwachsener Mann, frisch zugezogen

Bei dieser Person handelt es sich um einen jungen alleinstehenden Mann, der erst wenige Wochen im Ort wohnt um eine neue Arbeit anzutreten. Dieser kennt nur das nötigste, ist zwar technikaffin, aber handwerklich nicht sonderlich begabt.

Tabelle (4.3) – Alleinstehender Mann Customer Profile

Jobs	Pains	Gains
Suche nach Freizeitgestaltung und sozialen Kontakten	-Kennt sich nicht aus -Hat keine Bezugspersonen	-Recherche über auf ihn zugeschnittene Interessen -Finden von gleichgesinnten -„digitale“ Erkundung der Umgebung
Wohnung vervollständigen	-Fehlendes Werkzeug -Wenig handwerkliches Geschick	-Neue Inspirationen -Hilfe beim Aufbau -Lernen neuer Fähigkeiten

Tabelle (4.4) – Alleinstehender Mann Value Proposition

Products & Services	Pain Reliever	Gain Creator
Marktplatz & Entdecken	-Gezielte Suche nach Interessen -Suche nach (Hilfs-)Angeboten	-Kennenlernen neuer Menschen mit ähnlichen Interessen
Events und Beiträge	-Teilen von Hilfegesuchen -Teilen der eigenen Interessen -Veranstaltungen in der Nähe finden	-Größere Reichweite -Kennenlernen neuer Menschen mit ähnlichen Interessen

Alleinstehende Frau mittleren Alters

Diese Frau ist alleinstehend, geschieden und Vollzeit berufstätig. Sie hat 2 jugendliche Kinder welche bei ihr wohnen und ist auf der Suche nach einem neuen Lebenspartner. Sie ist Mitglied in einem Gartenbauverein in der Gemeinde.

Tabelle (4.5) – Alleinstehende Frau Customer Profile

Jobs	Pains	Gains
Menschen mit gleichen Interessen finden	-voll berufstätig, sehr wenig Zeit	- Finden von gleichgesinnten
Interessen der Kinder vertreten	-Finden von Nachhilfe -Bilden von Fahrgemeinschaften	-Zeitersparnis
Aktiv das Vereinsleben mitgestalten	-keine Einheitliche Plattform auf der Kommuniziert wird	-Effizientere Kommunikation und Planung von Events

Tabelle (4.6) – Alleinstehende Frau Value Proposition

Products & Services	Pain Reliever	Gain Creator
Marktplatz & Entdecken	-Finden von Hilfsangeboten in der Nachbarschaft	-Finden und Kennenlernen von Menschen mit ähnlichen Interessen
Events und Beiträge	-Gezielte Suche von Hilfe in der Nachbarschaft	-Über Tags können Interessierte Präziser an Informationen gelangen
Gruppenchats	-Anlegen von Gruppenunterhaltungen	-Einteilung in Arbeitsgruppen für verschiedene Veranstaltungen -Alles auf einer Plattform

Junges Pärchen

Mit diesem Profil soll ein junges Pärchen abgebildet werden. Diese sind beide Mitte 20 und leben in ihrer ersten gemeinsamen Wohnung. Beide sind voll berufstätig und daher zeitlich sehr eingeschränkt. Beide studieren und würden ihr wissen gerne mit anderen teilen.

Tabelle (4.7) – Junges Pärchen Customer Profile

Jobs	Pains	Gains
Wohnung vervollständigen	-neue Möbel sind oftmals sehr teuer -Fehlendes Werkzeug	-Effizientere Kommunikation und Planung von Events
Wissen teilen	-Interessenten finden, die von dem Wissen profitieren können	-Erfahrungsaustausch mit anderen Personen

Tabelle (4.8) – Junges Pärchen Value Proposition

Products & Services	Pain Reliever	Gain Creator
Marktplatz & Entdecken	-Finden von Angeboten in der Nachbarschaft	-Keine langen Fahrwege, da sich alles in der Nähe abspielt
Events und Beiträge	-Schreiben von Angeboten für Nachhilfe oder ähnliches	-Interessierte können gezielt danach suchen

4.2.4. Abgeleitete Features

Zusammen mit den Zielen aus der Aufgabenbeschreibung sowie den Erkenntnissen aus der Kundenanalyse des vorangegangenen Kapitels wurden folgende Features abgeleitet.

4.2.4.1. Persönliche und Vereinsprofile

Im Vordergrund einer Social Media Plattform steht die Darstellung der eigenen Person. Daher war die Grundlage für die Plattform die Implementierung von Profilen. Die Anforderungen für das Profil waren, sich kurz und knapp selbst beschreiben zu können. Die Einfachheit der Bedienung und damit für potentielle Profiländerungen sollte gegeben sein. Mit den Nutzerprofilen sollte nur das Interesse an der Person oder dem Verein geweckt werden, um im nächsten Schritt einen direkten Kontakt über die Chatfunktion zu ermöglichen. Ein Schlüsselement der Plattform ist das sogenannte Tag-System, ähnlich dem bekannten Hash-Tag von Netzwerken wie bspw. Twitter. Über die Tags kann ein Nutzer seine Hobbys

und Interessenfelder kurz und knapp angeben. Zusätzlich zum Tag sollte auch die Möglichkeit gegeben sein, sich selbst über einen Freitext beschreiben zu können. Die Option ein Profilbild hochladen zu können ist ebenfalls eines der Basis-Anforderungen die das System haben muss.

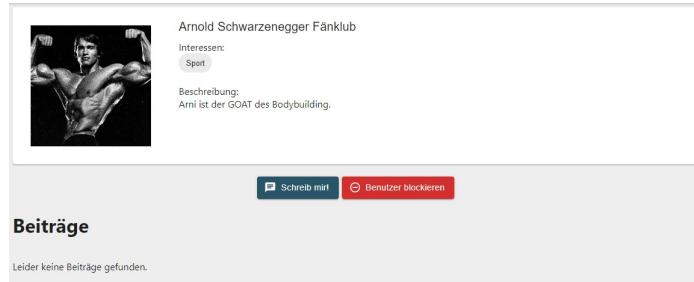


Abbildung (4.5) – Beispielprofil

In der obigen Abbildung ist ein Beispielhaftes Profil zu sehen. Es kann kurz und knapp auf das Thema der Organisation, in diesem Fall ein Fanclub über Arnold Schwarzenegger, abgeleitet werden. Des Weiteren sind auf dem Profil auch die erstellten Beiträge zu finden, ebenso wie eine Möglichkeit direkten Kontakt aufnehmen zu können.

4.2.4.2. Chats und Chaträume

Ebenso obligatorisch wie die Profile ist auch die Chatfunktion, die zur Kontaktaufnahme unabdingbar ist. Auch hier stand wieder die Einfachheit im Fokus. Dies kann dadurch begründet werden, dass die Plattform nur zum Kennenlernen und Vernetzen der Menschen gedacht ist. Ebenfalls soll die Möglichkeit bestehen, Gruppenchats zu erstellen. Dadurch kann man sowohl den Privatpersonen als auch Vereinen eine Möglichkeit geben sich in Gruppen zu organisieren.



Abbildung (4.6) – Chats in Digital Dahoam

4.2.4.3. Beiträge, Veranstaltungen und Events

Ebenfalls wichtig zur Kommunikation und Selbstdarstellung ist das Erstellen von Beiträgen.

Aufgrund der Anforderungen wurden hierfür 4 Kategorien an Beiträgen herauskristallisiert:

- **Angebot** Man bietet etwas an, wie bspw. Hilfeleistung bei handwerklichen Tätigkeiten
- **Anfrage** Hilfegesuch zu einem bestimmten Thema
- **Information** Allgemeine Mitteilung
- **Veranstaltung** Teilen einer bevorstehenden Veranstaltung

Durch diese Kategorien können grundlegend die meisten Bedürfnisse zufriedengestellt werden. Es gibt sowohl die Möglichkeit, sich auszutauschen, als auch die Möglichkeit aktiv bzw. passiv nach Hilfe in der Nachbarschaft zu suchen bzw. diese anzubieten. Vereine können ihre Events mit der Öffentlichkeit teilen.

4.2.4.4. Marktplatz und Entdecken

Aufbauend auf den Beiträgen soll der Nutzer die Möglichkeit haben, gezielt nach eben jenen suchen zu können. Dies soll über das Explorationstool der Website ermöglicht werden. Die Anforderungen hierfür waren, dass grundsätzlich nach allem und jedem gesucht bzw. gefiltert werden kann, anhand der zugewiesenen Tags. Dadurch kann dem Nutzer ermöglicht werden, auch wirklich nur die Beiträge zu sehen, die ihn auch interessieren. Des Weiteren sollten basierend auf demselben Prinzip (Tags) auch Personen gefunden werden können. Es wurde hierbei bewusst auf eine Klarnamensuche verzichtet, mit dem Ziel so möglichst viele neue Bekanntschaften zu ermöglichen.

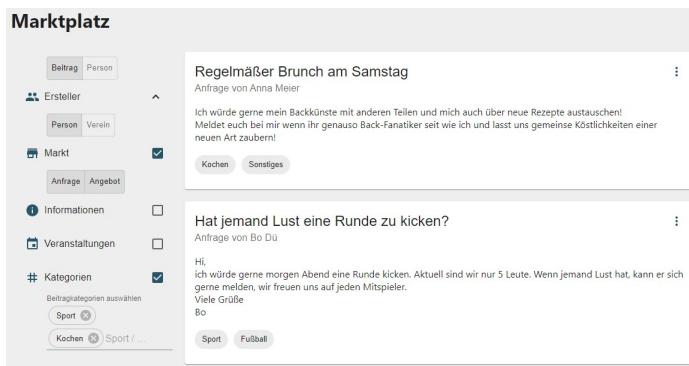


Abbildung (4.7) – Marktplatz von Digital Dahoam

4.2.4.5. Persönlicher Merkzettel

Als weiteres Feature sollte ein sog. Merkzettel implementiert werden. Dieser stellte sich als unabdingbar notwendig heraus für den Fall, dass ein Nutzer eine Veranstaltung sieht an der er interessiert ist und für später merken möchte. Aus dieser Anforderung wurde das allgemeine Feature des Merkzettels implementiert, sodass grundsätzlich die Möglichkeit

gegeben ist, sich alle Beiträge merken zu können. Ähnlich wie beim Marktplatz kann auch hier nach den verschiedenen Beitragstypen gefiltert werden.



Abbildung (4.8) – Merkzettel

4.3. Visuelle Grundstrukturen

Im Verlauf der konzeptionellen und inhaltlichen Ausarbeitung von DH wurde deutlich, dass es einen zunehmenden Bedarf an visuellen Darstellungen der Konzepte gibt. Insbesondere bei der Gestaltung und Entwicklung von Benutzerschnittstellen gab es Defizite, die durch die angewandten Methoden nicht ausgeglichen werden konnten. Eine Lösung hierfür waren Wireframes, die in der Softwareentwicklung sehr beliebt sind.

Wireframes sind besonders geeignet, um Weboberflächen zu visualisieren, da sie schnell erstellt und angepasst werden können und jedem ein gutes Verständnis vermitteln können. Ein weiterer Vorteil besteht darin, dass Kunden zeitnah die ersten Designkonzepte präsentiert werden können und ihr Feedback in frühen Entwicklungsphasen einfließen kann.

Die Erstellung von Wireframes erfordert gleichzeitig eine Auseinandersetzung mit dem Aufbau und der Konzeption der Website. Folglich werden in diesem Kapitel zunächst die Grundlagen für Seitenstrukturen (4.3.1) und Layouts (4.3.2) erarbeitet, bevor Wireframes (4.3.4) näher eingegangen werden.

4.3.1. Grundlagen Seitenstruktur

Bei der Erstellung einer Website ist es daher wichtig, die Seitenstruktur im Vorfeld zu planen und zu entwerfen. Eine klare Seitenstruktur ermöglicht es den Nutzern, sich auf der Website zurechtzufinden und schnell die gewünschten Informationen zu finden.

Abbildung 4.9 zeigt beispielhaft die Seitenstruktur einer Website, bei der die Informationen thematisch aufgeteilt und auf einzelnen Seiten organisiert sind. Jede Seite hat ein klares Ziel und informiert über bestimmte Inhalte. Darüber hinaus gibt es auf jeder Website eine Reihe von Kernseiten, die oft zu finden sind, wie beispielsweise die Startseite, Kontaktseite, Landingpage, Content- sowie Detailseiten.

Die Startseite einer Website, auch als Homepage bekannt, bildet den Startpunkt und die zentrale Anlaufstelle für den Nutzer. Von hier aus können alle Unterseiten über entsprechende Verlinkungen erreicht werden.

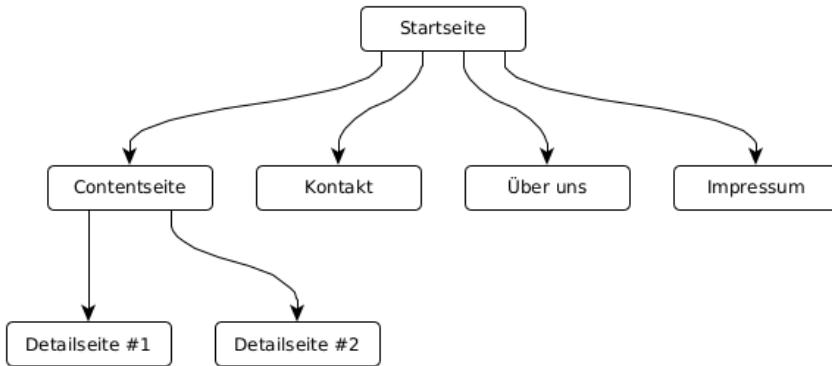


Abbildung (4.9) – Seitenstruktur einer Website

Die Landingpage ist die erste Seite, die ein Nutzer sieht, wenn er über einen externen Link auf die Website gelangt und eine Sitzung startet. Sie wird häufig im Rahmen von Marketingmaßnahmen eingesetzt, um den Nutzer von den Angeboten der Website zu überzeugen und zu weiteren Schritten zu animieren.

Die Contentseite bietet einen Überblick über die verfügbaren Inhalte und verlinkt auf die zugehörige Detailseite, auf der die Inhalte ausführlich dargestellt werden. Es ist wichtig, dass die Contentseite übersichtlich gestaltet ist und dem Nutzer einen schnellen Zugang zu den relevanten Informationen bietet.

Wie aus der Beschreibung der verschiedenen Seitentypen hervorgeht, werden diese in unterschiedlicher Reihenfolge oder zu unterschiedlichen Zeitpunkten aufgerufen. Dies führt zu einer gewissen Hierarchie zwischen den Seiten. Ob diese Hierarchie flach oder tief ist, hängt von der Navigationsstruktur ab. Im Allgemeinen wird empfohlen, flache Seitenhierarchien zu verwenden, um eine bessere Orientierung für den Nutzer zu gewährleisten.

Die einzelnen Seiten einer Website unterscheiden sich in der Regel nicht vollständig von einander, sondern folgen einem gleichbleibenden Aufbau. Dieses Grundgerüst einer Seite besteht oft aus einer Kopf- und Fußleiste sowie einem Inhaltsbereich (siehe Abb. 4.10). Die Kopf- und Fußleiste sind üblicherweise auf allen Seiten gleich gestaltet, um eine konsistente Nutzererfahrung zu gewährleisten.

Der Kopfbereich einer Webseite ist ein wichtiger Bestandteil des Layouts und befindet sich im obersten Teil der Seite. Er umfasst üblicherweise das Logo, die Haupt- und Metanavigation und wird oft auch als Header bezeichnet.

Das Logo dient als visuelles Erkennungsmerkmal und soll das Unternehmen oder die Marke repräsentieren. Die Hauptnavigation ermöglicht eine strukturierte Übersicht über die verfügbaren Inhalte und ist entscheidend für eine gute Navigation auf der Webseite. Daher wird sie oft auffällig im oberen Bereich platziert. Die Metanavigation bietet ergänzende Serviceinhalte wie zum Beispiel Account-Einstellungen und ist getrennt von den Hauptthemen

aufgeführt.

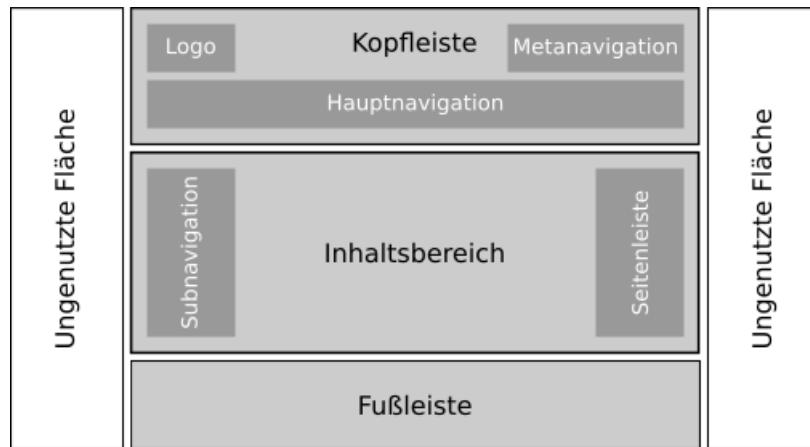


Abbildung (4.10) – Grundgerüst einer Website

Der Inhaltsbereich befindet sich direkt unter dem Kopfbereich und enthält die zu vermittelnden Inhalte. Je nach Bedarf kann er entweder nur die reinen Inhalte oder auch eine Subnavigation auf der linken Seite sowie eine Seitenleiste für weiterführende Inhalte auf der rechten Seite enthalten. Die zentralen Inhalte werden zwischen den Leisten im Inhaltsbereich platziert und in der Regel nach Wichtigkeit absteigend sortiert dargestellt.

Die untere Begrenzung einer Webseite wird üblicherweise durch die Fußleiste gebildet. Hier finden sich häufig Basisinformationen zur Seite, ergänzende Inhalte sowie gegebenenfalls weitere Navigationsmöglichkeiten.

Die Seitenbereiche werden dabei von einem umgebenden Block eingerahmt, der die ungenutzte Fläche der Website darstellt.

4.3.2. Grundlagen Layout

4.3.3. Rastersystem

Um die ersten Skizzen in ein stimmiges Layout zu überführen, kann ein Rastersystem verwendet werden. Dabei handelt es sich um ein Netz mit Zeilen und Spalten, an denen die Inhalte ausgerichtet und schließlich im Rastersystem (siehe Abbildung 4.11) platziert werden. Der Vorteil besteht darin, dass die Inhaltselemente und Einzelseiten in eine einheitliche Struktur gebracht werden und die Seiten dadurch zunehmend abgestimmter und harmonischer wirken. Das Raster wird lediglich für die Gestaltung verwendet und sollte für den Endnutzer möglichst unauffällig oder dezent wirken.

Die Basis eines Rastersystems ist in der Regel eine Leinwand mit festen Abmessungen. Diese Fläche wird in Spalten aufgeteilt und gegebenenfalls mit gleichbleibendem Freiraum zwischen den Spalten versehen. Je mehr Spalten man wählt, desto größer wird der gestalterische Spielraum, aber auch der Nutzen des Rastersystems nimmt ab einer gewissen Spaltenanzahl

ab. Im nächsten Schritt kann eine horizontale Unterteilung vorgenommen werden, wobei das sogenannte Baseline Grid als Grundlage dient. Das Baseline Grid setzt sich aus der Schriftgröße und dem Zeilenabstand zusammen. Die einzelnen Spalten und Zeilen können weiterhin in Bereiche unterteilt werden, um ein modulares Rastersystem zu schaffen. Abschließend werden die Inhalte der Website den entsprechenden Bereichen zugeordnet und am Raster ausgerichtet. Das Ziel ist, ein stimmiges Layout zu erreichen und den visuellen Eindruck der Website zu verbessern.



Abbildung (4.11) – Beispiel eines Rastersystems [4]

Layouttypen

Früher wurde aufgrund der begrenzten Anzahl von Endgeräten mit unterschiedlichen Auflösungen oft ein statisches Layout verwendet, das für eine gängige Auflösung optimiert war. Heute ist dieser Ansatz aufgrund der Vielzahl an Geräten mit sehr unterschiedlichen Auflösungen nicht mehr praktikabel. Es ist besonders schwierig, eine passende Auflösung zu finden, die für die meisten Geräte geeignet ist. Deshalb geht man heutzutage nicht mehr von einer festen Breite bei der Layouterstellung aus, sondern erstellt Layouts, die sich individuell an die Auflösung des Geräts anpassen.

Das fixe Layout ist ein Layouttyp, der eine feste Breite mit festen Pixelwerten definiert. Wenn die Auflösung des Bildschirms zum Layout passt, werden alle Inhalte korrekt dargestellt. Wenn jedoch die Auflösung zu hoch ist, wird der umgebende Block der Seite unnötig groß und es entsteht viel ungenutzter Platz. Wenn die Auflösung zu niedrig ist, erscheinen horizontale Scrollbalken und die Inhalte werden abgeschnitten oder unvollständig angezeigt.

Das flexible Layout bildet das Gegenstück zum fixen Layout und zeichnet sich durch seine Fähigkeit aus, sich unmittelbar allen Veränderungen anzupassen und dabei alle vorgegebe-

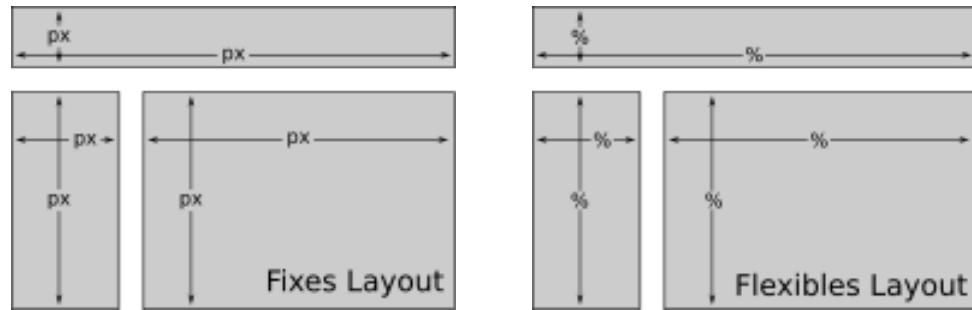


Abbildung (4.12) – Aufbau eines fixen und flexiblen Layouts

nen Größenverhältnisse beizubehalten. Im Gegensatz zum fixen Layout wird es mit relativen prozentualen Werten definiert, wodurch es sich leicht verschiedenen Varianten einer Website anpassen kann. Obwohl der reine Layouttyp selten zum Einsatz kommt, findet man häufig eine Mischform aus fixem und flexiblem Layout.

Eine weitere Layoutform ist das elastische Layout, bei dem sich die Inhalte einer Seite flexibel anpassen können. Dieser Layouttyp eignet sich besonders für Inhalte, die die volle Breite des Bildschirms ausfüllen, wie z.B. bei einer Produktpräsentation mit großformatigen Bildern und Videos. Es ist von Vorteil, wenn es nicht zu viele Inhalte gibt, da sich diese automatisch anpassen müssen, um optimal dargestellt zu werden.

Das flexible Layout bildet die Grundlage für das responsive Layout, welches die Möglichkeiten erweitert, ein passendes Layout für jedes Endgerät situationsgerecht bereitzustellen. Als Erweiterung des flexiblen Layouts verfügt das responsive Layout über sogenannte Media-Queries, die es ermöglichen, bei Überschreiten oder Unterschreiten bestimmter Schwellenwerte eine Anpassung der Ansicht zu initiieren und die Inhalte beispielsweise neu anzurufen.

4.3.4. Grundlagen Wireframes

4.3.4.1. Definition/ Inhalte

Wireframes sind schematische Darstellungen von Inhalten und Elementen der Seitenoberfläche und dienen der Konzeptionierung in der Planungsphase. Sie ermöglichen einen groben Entwurf für die Verteilung, Anordnung und Gestaltung der Seitenelemente sowie das Herstellen von Beziehungen zwischen den Seiten. Typischerweise sind Wireframes skizzähafte, schwarz-weiße oder graue Abbildungen, bei denen die einzelnen Bestandteile der Seite durch einfache geometrische Formen verdeutlicht werden. Design, Farben, Schrift und Bilder sind hierbei nicht Teil der Methode. Das fertige Wireframe gibt einen Überblick über die Platzierung der Informationsinhalte, die Struktur und Navigation der Seite sowie die Interaktionselemente, mit denen der Nutzer interagiert. Ausgearbeitete Wireframes bilden die Grundlage für die visuelle und funktionale Detaillierung des Produkts. Anschließende

Schritte können beispielsweise das Erstellen von Mockups oder Prototypen sein, welche als weitere visuelle Methoden in Kapitel 4.3.4.3 erläutert werden.

4.3.4.2. Arten

Wireframes werden allgemein in Low-Fidelity- und High-Fidelity-Wireframes unterschieden. Bei den Low-Fidelity-Wireframe (LFW) liegt der Fokus allein auf dem funktionalen Design. Die Seitenschemas werden mit einfachen Formen und ohne konkrete Inhalte erstellt. Die High-Fidelity-Wireframe (HFW) stellen hingegen die nächste Entwicklungsstufe für die Ausarbeitung des Designs dar. Hier kommen zunehmend mehr Designkomponenten wie Farben, Typographie, Abstände, Icons, Text, Bilder und Grafiken zum Einsatz. Dabei werden auch reale Textlängen und Größenverhältnisse der Elemente und Inhalte berücksichtigt.

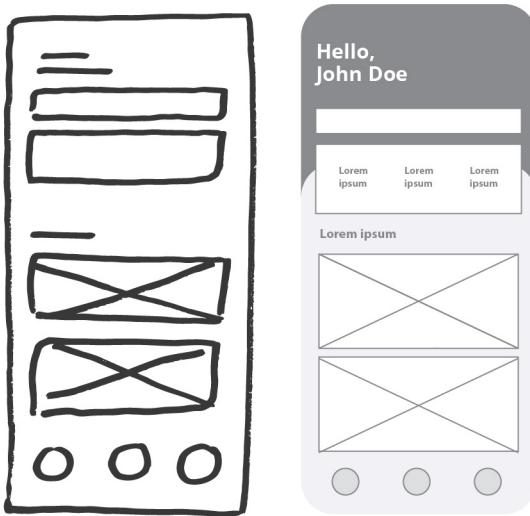


Abbildung (4.13) – Beispielhafte Darstellung von LFW und HFW [5]

4.3.4.3. Abgrenzung

Wireframes werden oft in Verbindung mit anderen visuellen Methoden genutzt, wie zum Beispiel Mockups und Prototypen. Diese Methoden bauen auf Wireframes auf und haben zum Ziel, das zukünftige Produkt realitätsnah darzustellen.

Mockups sind Nachbildungen oder maßstabsgerechte Modelle des Produkts und legen den Fokus auf das visuell-interaktive Design. Dabei werden die Wireframes um Elemente der Benutzeroberfläche ergänzt, jedoch sind keine funktionalen oder animierten Elemente enthalten. Mockups werden oft in der natürlichen Umgebung des Produkts, z.B. auf einem Gerätebildschirm, dargestellt, um dem Kunden ein realitätsnahes Gefühl des Erscheinungsbilds zu vermitteln.

Ein Prototyp ist ein vereinfachtes Versuchsmodell des geplanten Produkts, das auf den Ergebnissen eines Mockups aufbaut. Er wird um funktionale Elemente erweitert, um die

Interaktion eines Benutzers mit dem Dienst simulieren zu können. Ein Beispiel für einen klassischen Prototypen ist ein Klick-Dummy. Dieser bietet eine teilweise interaktive Demo einer Bedienoberfläche, die alle relevanten Merkmale des Produkts widerspiegelt. Er kann für Vorstellungen oder Testläufe genutzt werden.

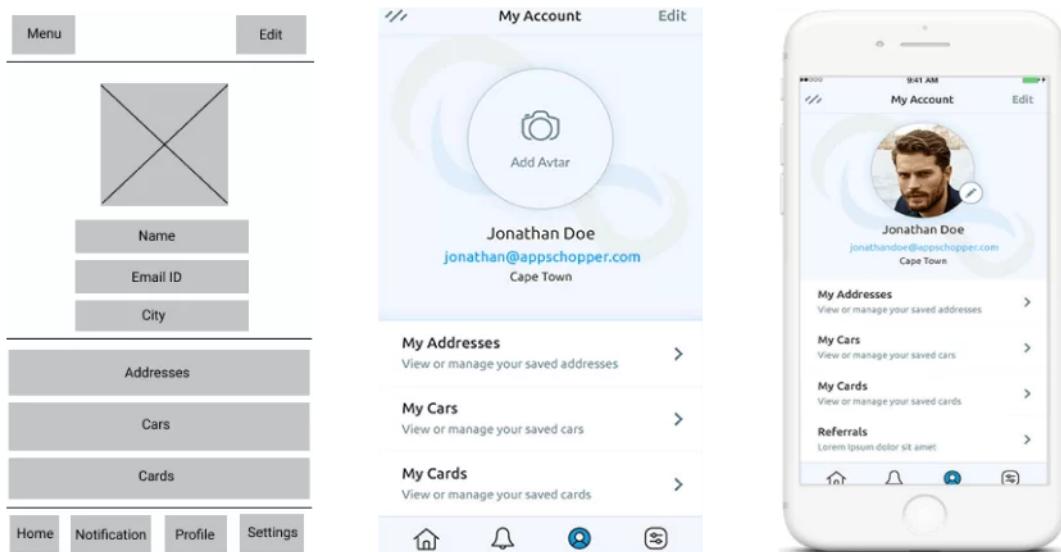


Abbildung (4.14) – Interationsschritte: Wireframe - Mockup - Prototyp [6]

4.3.5. Anwendung

4.3.5.1. Integration

Die Wireframes wurden kontinuierlich im Projektverlauf genutzt, um die Ausgestaltung und Kommunikation der User-Stories während des Backlog Refinement zu unterstützen. Zunächst wurden in einer frühen Phase die jeweiligen User-Stories vom Product Owner erläutert und ein erster Entwurf der Wireframes erstellt. Erkenntnisse, die während der Konzeptionierung entstanden sind, wurden mit dem Product Owner besprochen und direkt in den Wireframes umgesetzt. Während des Refinement-Prozesses dienten die Wireframes einerseits dem Product Owner zur Vorstellung und Erklärung der User-Stories und andererseits zur Präzisierung und Detaillierung der Story im Team. Die daraus gewonnenen Informationen wurden vom Product Owner in das Backlog aufgenommen und erneut in die Wireframes integriert, um sie im nächsten Refinement-Prozess zu berücksichtigen.

Die Wireframes wurden nicht nur genutzt, um die Features zu definieren und zu veranschaulichen, sondern auch während des Entwicklungsprozesses. Hier dienten sie als gestalterische Grundlage und wurden soweit wie möglich vom Entwickler implementiert.

4.3.5.2. Toolauswahl

Es gibt eine Vielzahl von Softwarelösungen, die für die Erstellung von Wireframes genutzt werden können. Diese lassen sich grob in desktop- und webbasierte Anwendungen unterteilen. Für uns war es bei der Wahl der Softwarelösung entscheidend, dass die Wireframes einfach erstellt und angepasst werden können, dass ein paralleles Arbeiten möglich ist und dass die aktuellen Entwürfe ohne zusätzlichen Aufwand geteilt werden können. Aufgrund dieser Anforderungen haben wir uns auf webbasierte Lösungen konzentriert und sind bei unserer Recherche auf Figma⁴ gestoßen, das sich als passender Dienst herausgestellt hat.

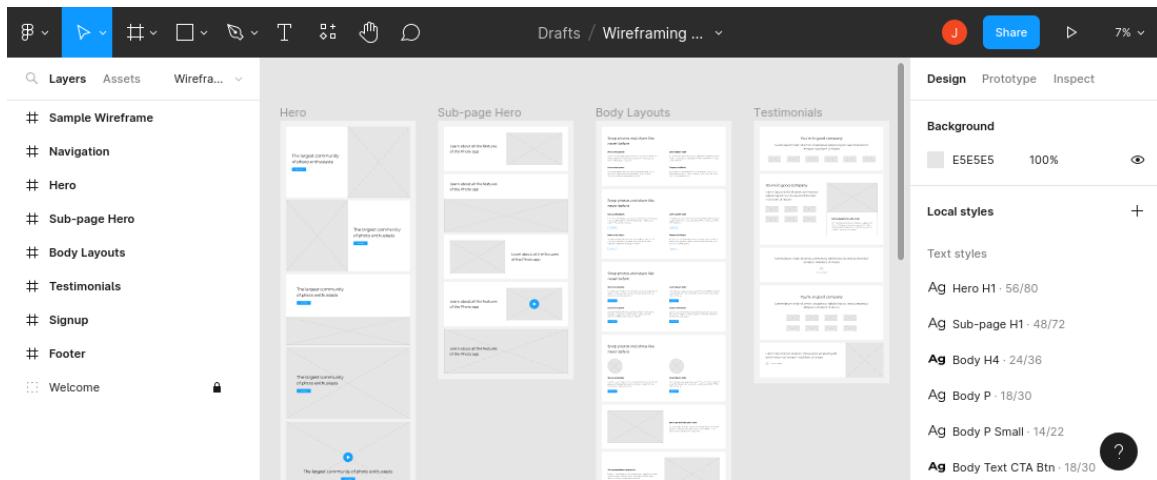


Abbildung (4.15) – Figmas-Arbeitsumgebung

Figma hat sich als etablierte Onlineanwendung auf die Erstellung von Wireframes, Mockups und Prototypen spezialisiert. Als Werkzeug ist es ausgereift und unterstützt die Erstellung einfacher Wireframes durchgängig. Bei der Erstellung der Wireframes wurde sich auf die grundlegenden Funktionen von Figma beschränkt und besonders darauf geachtet, die wesentlichen Elemente zu betonen, um die Kerninhalte der Features hervorzuheben. Dazu wurden Rechtecke und Textelemente zur Gestaltung sowie Raster und Gruppierungs-Funktionen zur Orientierung und Ausrichtung der Inhaltselemente eingesetzt.

4.3.5.3. Anforderungen an die Benutzerschnittstelle

Für die Erstellung der Wireframes wurden im Vorfeld bestimmte Anforderungen festgelegt, die bei der Gestaltung berücksichtigt werden mussten. Diese Anforderungen betrafen sowohl die Gestaltung der Oberfläche als auch die technologiekonforme Umsetzung der Wireframes.

Eine wichtige Anforderung war es, die Benutzerschnittstelle – auf engl. User Interface (UI) – generationenübergreifend zu gestalten. Daraus ergaben sich folgende Anforderungen an die UI:

⁴Siehe <https://www.figma.com/>

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

Um die Konzeptentwürfe erfolgreich umzusetzen, war es entscheidend, bei der Erstellung die verwendeten Webtechnologien zu berücksichtigen. So sollten die Umsetzungen der Entwürfe für die Entwickler ohne größere Schwierigkeiten erfolgen können. Ein weiteres Ziel bestand darin, einen hohen Grad an Wiederverwendbarkeit der Komponenten zu erreichen.

4.3.5.4. Ausarbeitung der Seiten

Als geeigneter Wireframetyp wurde ein Hybrid aus LFW und HFW gewählt. Die erstellten Wireframes umfassten alle typischen Layoutbereiche wie Kopf-, Inhalts- und Fußbereich, deren Inhalte in Feldern vereinfacht symbolisiert wurden. Die einzelnen Felder beinhalteten Texte, Dummy-Blöcke wie beispielsweise für Bilder sowie die zugehörigen Interaktionselemente wie Buttons und Dialoge. Die Wireframes wurden in schlichtem Graustufen gehalten und ohne Berücksichtigung von Designelementen wie Farbe und Typografie erstellt. Für die regelmäßige Erstellung und Anordnung der Komponenten hat sich nach einigen Entwürfen die Leinwandbreite von 1160 Pixel mit einem Raster von 27x37 Pixel als vorteilhaft erwiesen.

Wireframes wurden für folgende Seiten/Dialoge erstellt:

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

erstellt.

Die Art und Detailtiefe der Ausgestaltung zeigen exemplarisch die Wireframes in Abbildung 4.16 und 4.17.

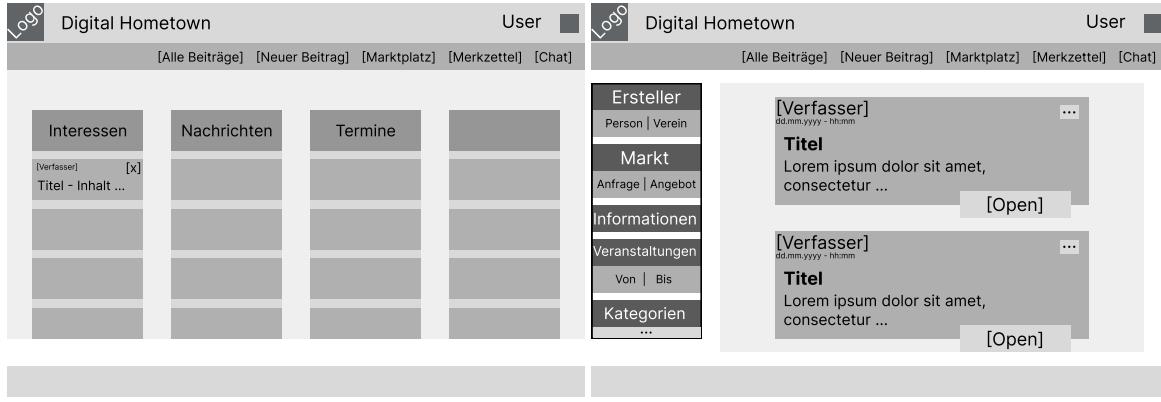


Abbildung (4.16) – Startseite (DH)

Abbildung (4.17) – Markt (DH)

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:

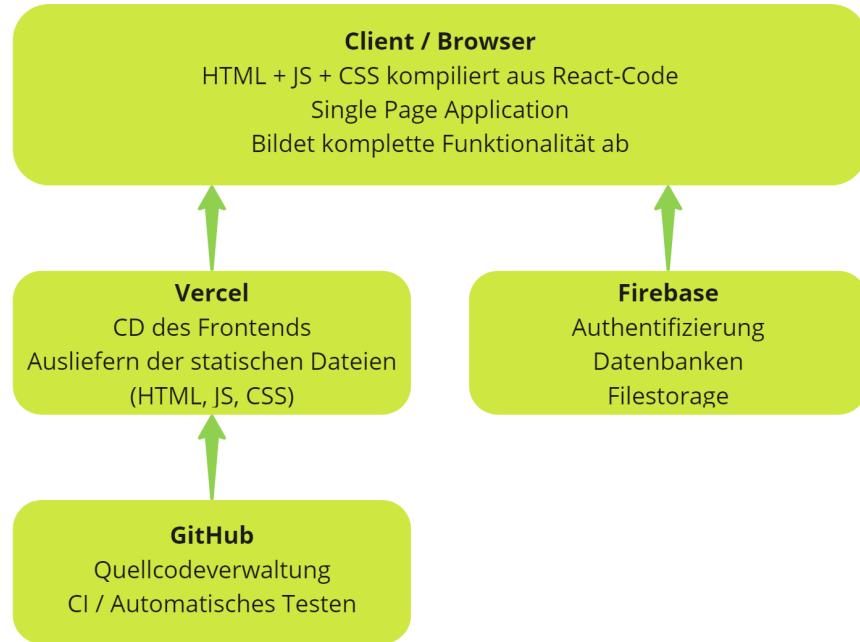


Abbildung (5.1) – Übersicht der technischen Softwarearchitektur

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 ein 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 [7]

²Vgl. TypeScript 2023 [8]

³Vgl. React 2022 [9]

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.

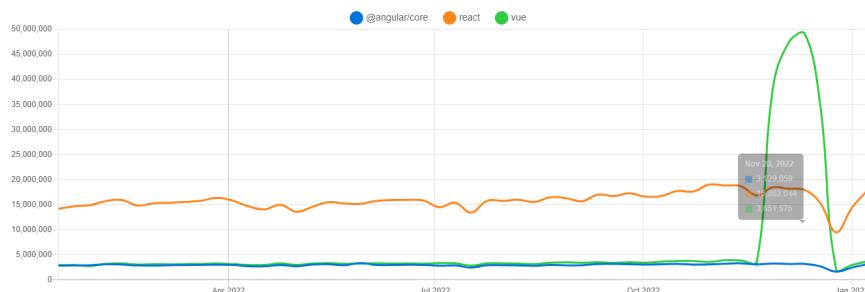


Abbildung (6.1) – Vergleich Downloadzahlen verschiedener Frontend Frameworks [10]

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 [11]

⁵Vgl. Material-UI 2022 [12]

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 [13]

⁷Vgl. Firebase Authentication 2022 [14]

⁸Vgl. Firebase Realtime Database 2022 [15]

⁹Vgl. Firebase Cloud Firestore 2022 [16]

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 [17]

¹¹Vgl. Github 2022 [18]

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 zu 6.6 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 [19]

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.
Teamarbeit	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 [20]

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.

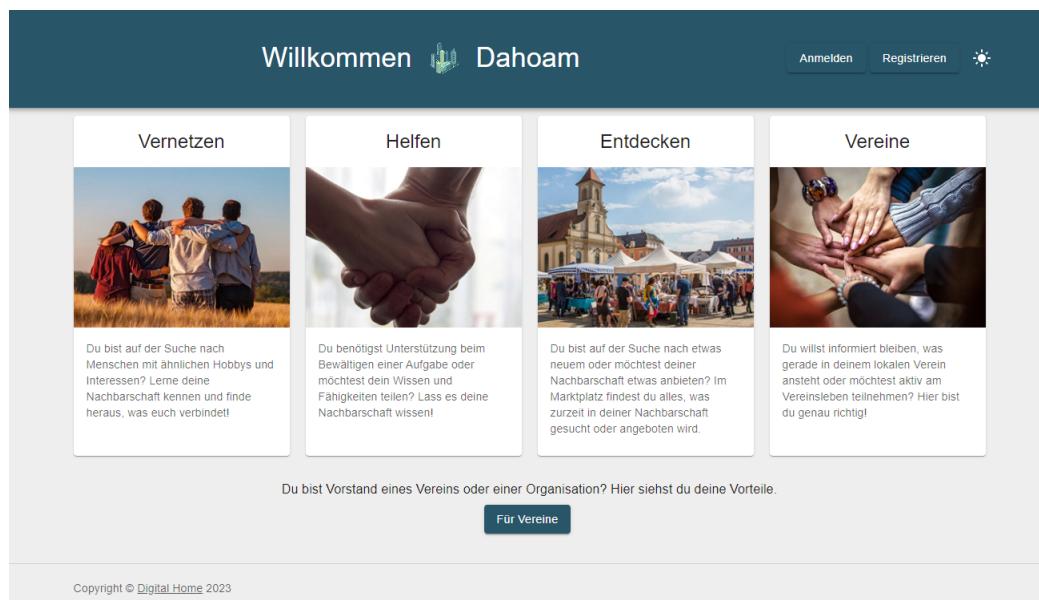


Abbildung (7.1) – Bildschirmaufnahme der Startseite für nicht eingeloggte Nutzer

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. Abbildung 7.2). Der Inhalt der Landing Page beschreibt, welche Vorteile die Plattform für Vereine bietet.

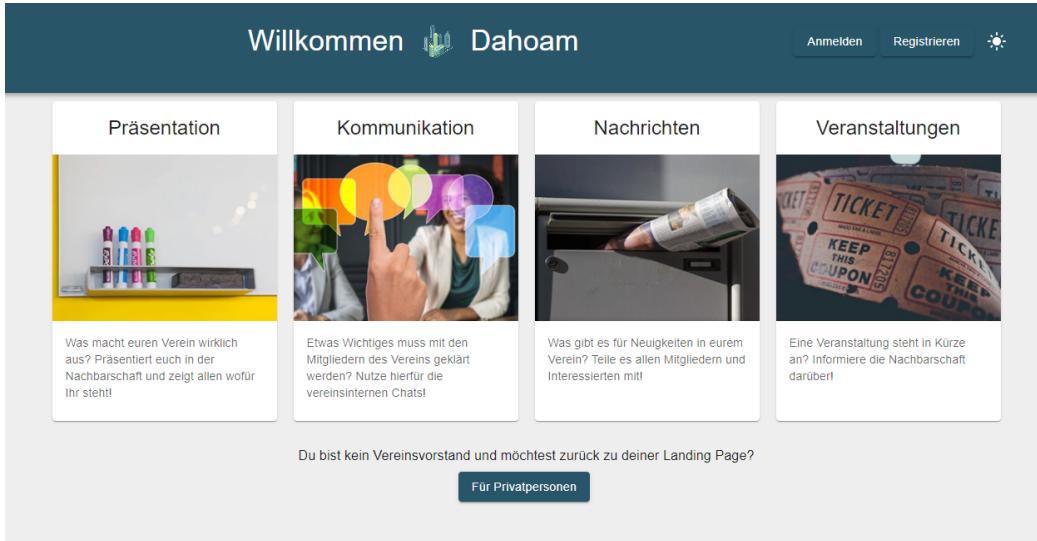


Abbildung (7.2) – Bildschirmaufnahme der Startseite für nicht eingeloggte Vereine

Wenn der Benutzer eingeloggt ist, wird er auf das Dashboard weitergeleitet, das aktuelle Beiträge und Veranstaltungen anzeigt.

Da die Landing-Page eine statische Seite darstellt, wird hier nicht weitere auf die Implementierung eingegangen.

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.

Hier sieht man die verschiedenen Informationen, die ein Nutzer in seinem Profil angeben kann. In den nächsten Seiten wird die Implementierung dieser Funktionen beschrieben.

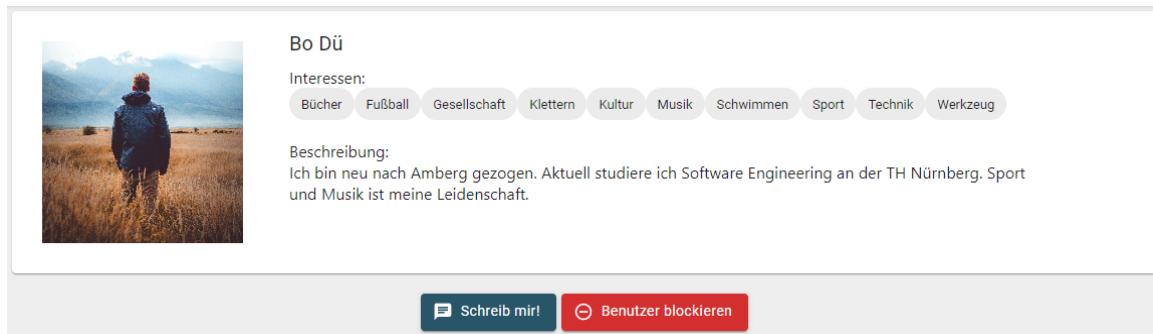


Abbildung (7.3) – Übersicht eines Benutzerprofils

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:

Abbildung (7.4) – Anmeldemaske

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 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 zudem 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 7.5). Fehlerhafte Eingaben der E-Mail-Adresse, sowie die redundante Eingabe des Passworts werden hierbei validiert.

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 7.6).

The registration form is titled "Registrieren". It contains four input fields: "Benutzername *", "Email *", "Passwort *", and "Passwort (Wiederholung) *". Below the fields is a large blue "Registrieren" button. At the bottom left is a link "Passwort vergessen?", and at the bottom right is a link "Schon einen Account? Melde dich hier an!".

Abbildung (7.5) – Registrierungsseite

The profile editing form has a header "Vervollständige dein Profil". It includes fields for "Anzeigename *", "Email *", "Geburtstag", and "Postleitzahl". Below the fields are two buttons: "Speichern" (green) and "Account löschen" (red). A success message box at the bottom left says "Hallo Test1234, du bist nun registriert." and includes a close button "x". The footer shows "Copyright © Digital Home 2023".

Abbildung (7.6) – 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 in Unterabschnitt 7.3.5 beschrieben.

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 7.7). Dies steigert die Usability der Plattform, da der Nutzer nicht mehr die E-Mail-Adresse und das Passwort eingeben muss.

Wenn der Nutzer sich mit einer E-Mail-Adresse registriert hat, kann er sich auf folgender Seite anmelden (s. Abbildung 7.8).

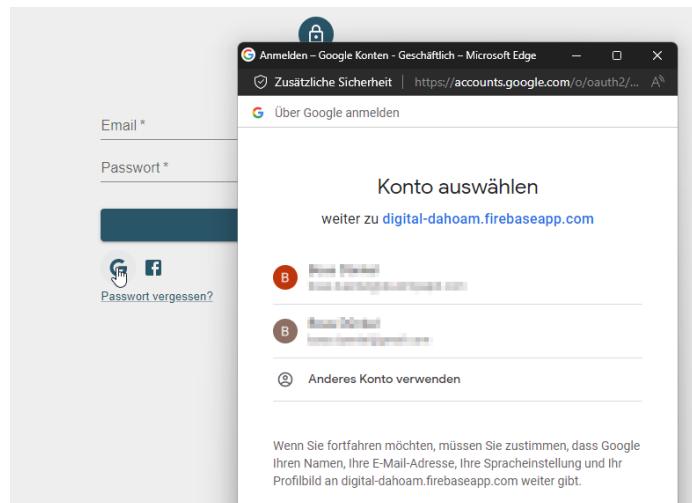


Abbildung (7.7) – Authentifizierungs-Dialog beim Anmelden mit Google

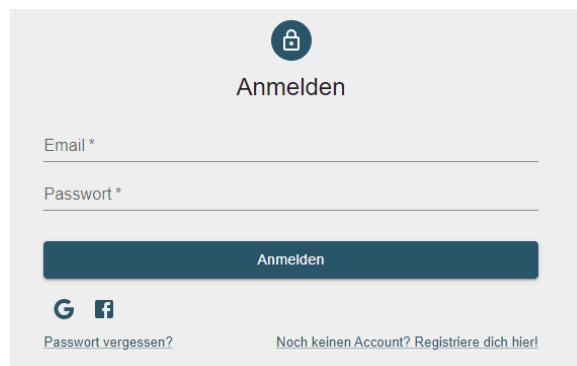


Abbildung (7.8) – Anmeldeseite

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 *

Jonas

Email *

jonas.rosen@gmail.com

Geburtstag

28.10.2022

Postleitzahl

90342

Abbildung (7.9) – 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.



Abbildung (7.10) – 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.

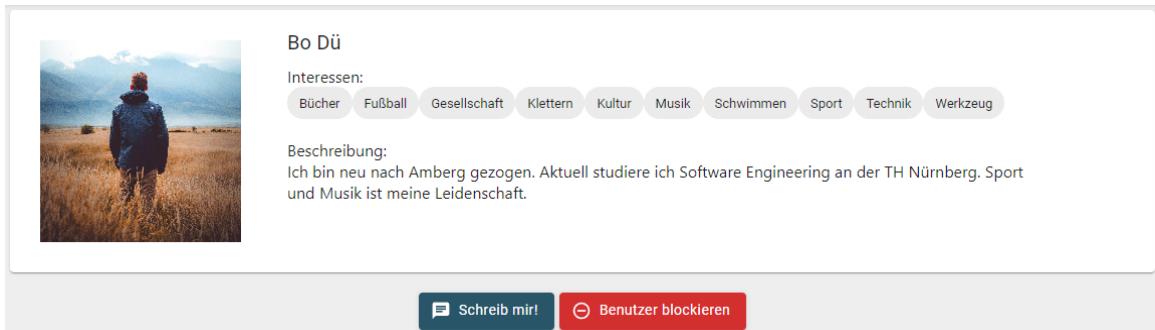


Abbildung (7.11) – Blockieren eines Nutzers

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.

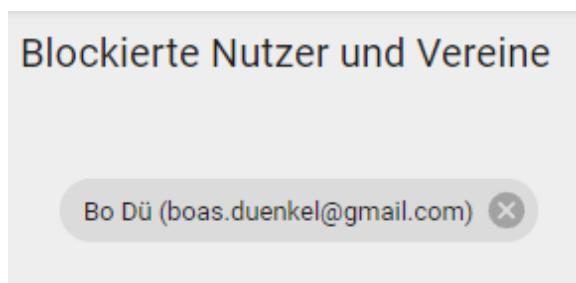


Abbildung (7.12) – Blockierte Nutzer

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 (7.13) – 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 (7.14) – 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.

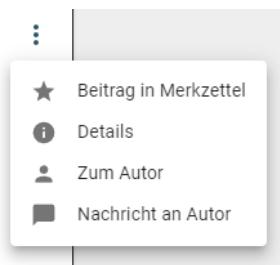


Abbildung (7.15) – Beitragsmenü

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 profile picture of a smiling man, a section for interests ('Interessen') with options like Fußball, Sport, and Technik, and a bio ('Beschreibung') stating he is a developer and enjoys sports and technology. Below the profile is a 'Beiträge' (Posts) section with a message about post functionality updates. A green 'Vorschau' (Preview) button is visible at the bottom.

Abbildung (7.16) – 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' (Notes) interface. On the left is a sidebar with categories: 'Alle', 'Anfrage', 'Angebot', 'Information', and 'Veranstaltung'. The main panel displays a note titled 'Brandwache 24.01' about a fire station event in Amberg, dated 1/24/2023, with a 'Technik' tag.

Abbildung (7.17) – 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 (7.18) – 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 (7.19) – 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 7.20) verwendet werden. Bei dem Design wurde darauf geachtet, dass die einzelnen Hauptwurzelemente keine zu große Verschachtelung aufweisen, um eine gute Performanz zu gewährleisten.



Abbildung (7.20) – 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       }
8     }
9   }
10 }
```

```

6     ".write": "root.child('rooms/' + $roomUid + '/members').hasChild(
7       auth.uid)"
8   },
9   "messages": {
10    ".indexOn": "sendAt"
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 7.21 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.

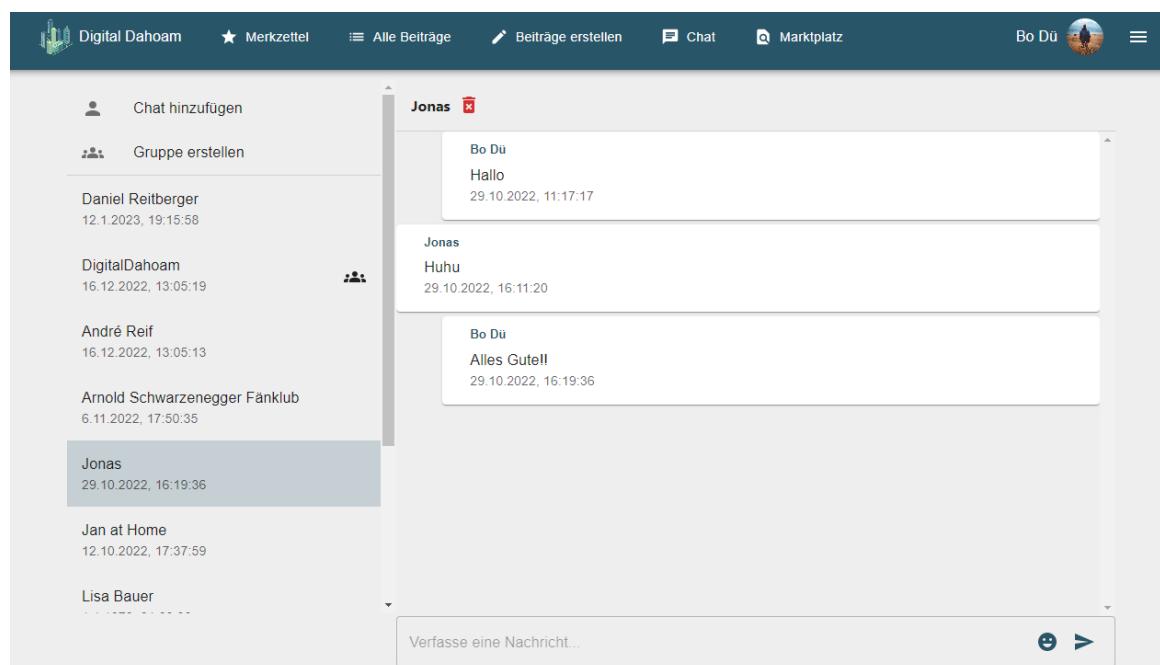


Abbildung (7.21) – Ansicht des Chats in der Anwendung

Über den Button „Chat hinzufügen“ wird die in Abbildung 7.22 dargestellte Ansicht geöffnet, wo alle Personen der Plattform angezeigt werden. Hier kann ein neuer Chat mit einer Einzelperson erstellt werden.

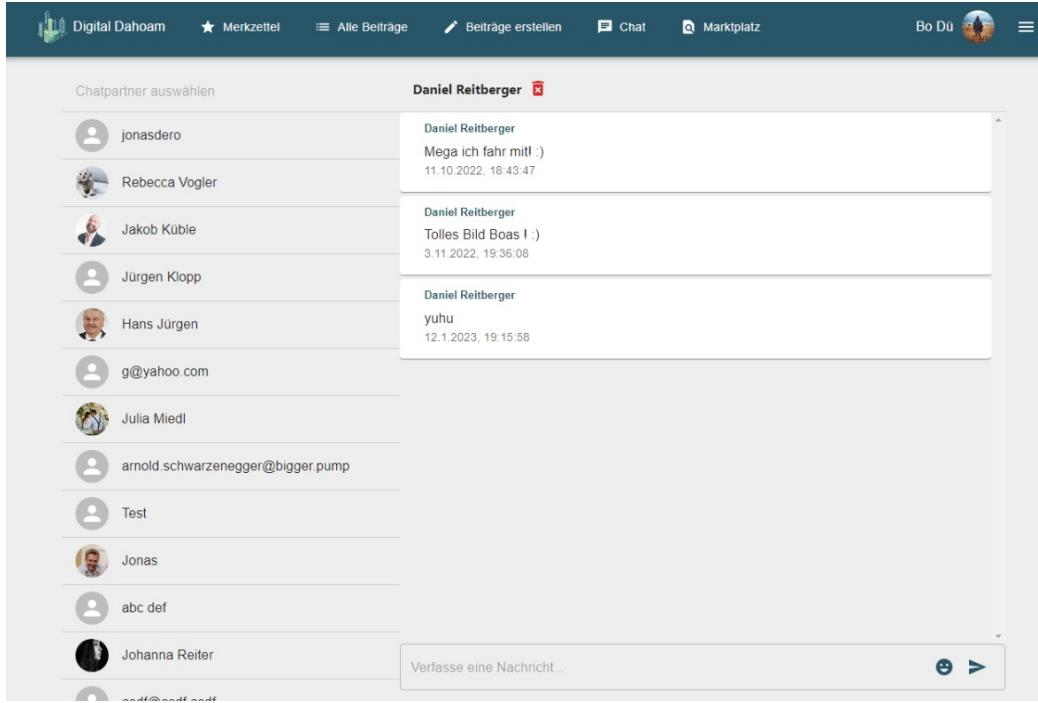


Abbildung (7.22) – Hinzufügen von Chats mit Einzelpersonen

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 dieser ausgewählt, lässt sich über einen Button die in Abbildung 7.23 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 7.24 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.

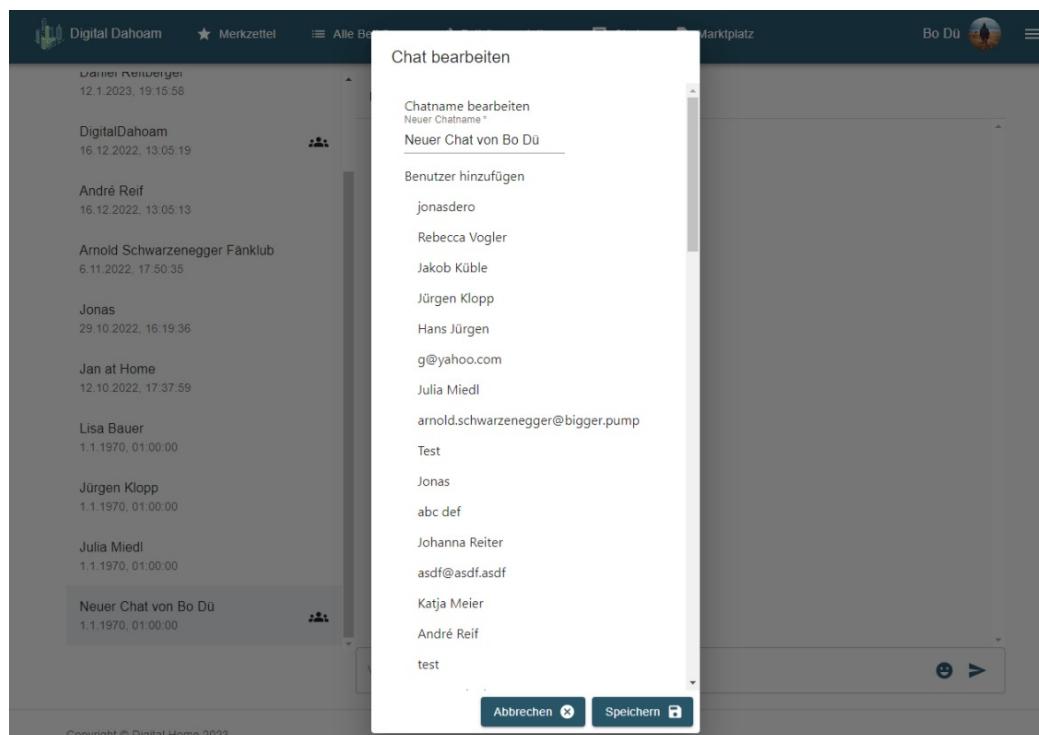


Abbildung (7.23) – Chatgruppe umbenennen und Mitglieder hinzufügen



Abbildung (7.24) – Möglichkeit von einem Beitrag direkt zum Chatraum des Beitragautors zu gelangen

8. Testing

Diese Kapitel beschäftigt sich mit dem Prozess des Testens in diesem Projekt. Dabei soll zunächst etwas Theorie vorgestellt, in der klar gemacht werden soll, was Testen eigentlich ist, wie der Test Prozess im Projekt nach SCRUM gelebt wurde und welche Aufgaben dem Tester zufallen. Abgeschlossen wird dieses Kapitel dem beispielhaften Aufzeigen der Umsetzung der definierten Testmethoden in diesem Projekt.

8.1. Definition Testen

Softwaresysteme sind ein wesentlicher Bestandteil des Lebens: von Fachanwendungen (z.B. im Bankwesen) bis hin zu Verbraucherprodukten (z.B. Autos). Die meisten Menschen haben bereits Erfahrungen mit Software gemacht, die nicht wie erwartet funktioniert hat. Software, die nicht korrekt arbeitet, kann zu vielfältigen Problemen führen, u.a. zu Geld-, Zeit- oder Imageverlust, sogar bis hin zu Verletzungen oder Tod. Softwaretesten ist ein Mittel, die Qualität von Software zu beurteilen und das Risiko einer Fehlerwirkung im Betrieb zu reduzieren. Es ist eine gängige Fehleinschätzung, dass Testen ausschließlich darin besteht, Tests auszuführen, d.h. im Sinne von: die Software auszuführen und die Ergebnisse zu prüfen. Softwaretesten ist ein Prozess, der viele unterschiedliche Aktivitäten umfasst. Testen beinhaltet also auch die Prüfung von Arbeitsergebnissen wie Anforderungen, User-Stories und Quellcode im Rahmen von Reviews. Eine weitere gängige Fehleinschätzung ist, dass sich Testen ausschließlich auf die Verifizierung von Anforderungen, User-Stories oder anderen Spezifikationen konzentriert. Auch wenn das Testen es erfordert, zu prüfen, ob das System spezifische Anforderungen erfüllt, so umfasst es auch die Validierung, also die Prüfung, ob das System in seiner Einsatzumgebung die Bedürfnisse von Benutzern und anderen Stakeholdern erfüllen wird.

8.2. Aufgabengebiet als Tester

Das Testen ist untrennbar mit der Entwicklung verwoben. Teammitglieder mit Testfokus decken mehrere Rollen ab: zum einen, die des Testers (operative Aufgaben), zum anderen die des Testmanagers (Strategische Aufgaben) und schließlich auch die des Qualitätsmanagers.

Jedes agile Team besteht aus zumindest ein bis zwei ausgebildete Tester. Weiterhin stehen dem Team ergänzend noch einige Support-Teams oder auch einzelne Fachexperten zur Verfügung, die für Spezialthemen herangezogen werden können (s. Abbildung 8.1).

Vereinfacht kann die Teamzusammenstellung wie folgt dargestellt werden (Abbildung 8.2).

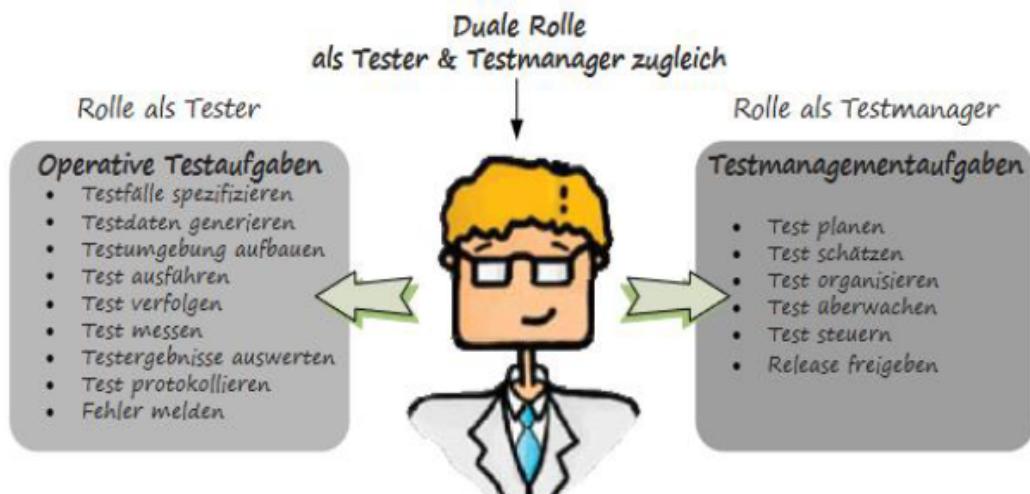


Abbildung (8.1) – Aufgabengebiete des Testers

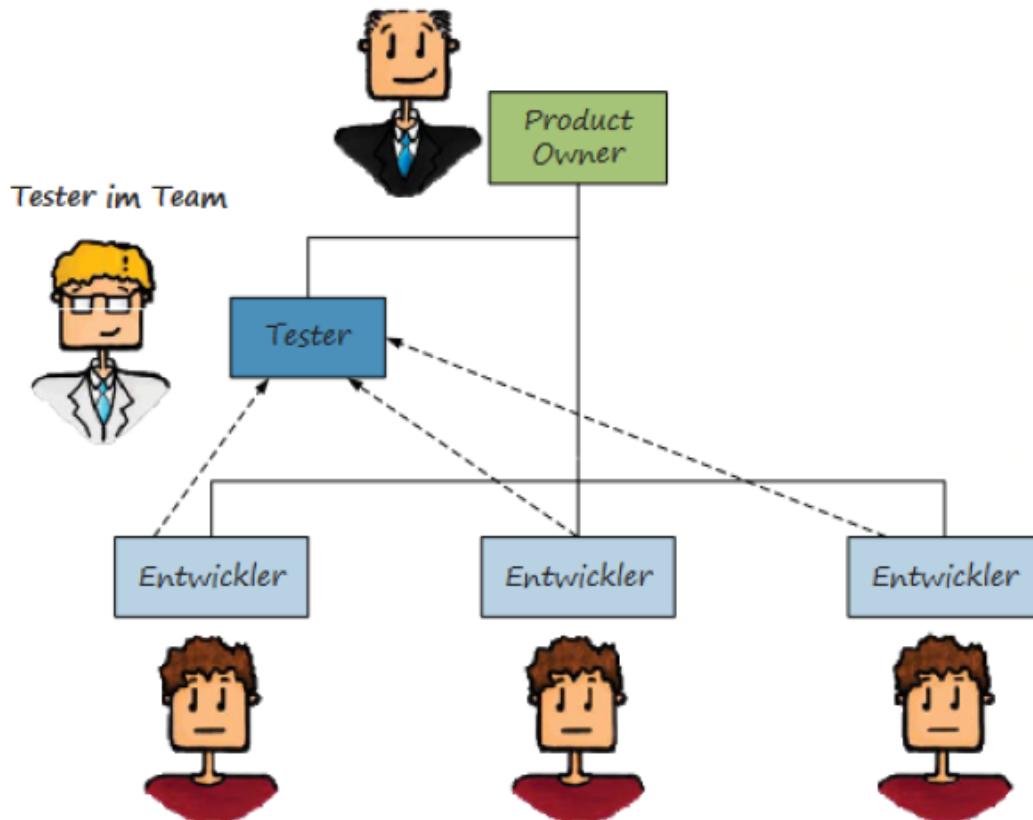


Abbildung (8.2) – Teamzusammenstellung und Position der Tester

8.3. Testen im agilen Projekt

Das Testen hat auch in SCRUM einen hohen Stellenwert. Die Rolle des Testers sollte ein fester Bestandteil eines cross-funktionalen Teams, die Qualitätssicherung automatisch in jedem Entwicklungszyklus integriert sein. Das agile Testen im SCRUM muss so gestaltet

sein, dass es die Ziele der agilen Entwicklung unterstützt. Das heißt, es muss auf hohe Kundenzufriedenheit, hohe Produktivität und hohe Entwicklungsgeschwindigkeit ausgerichtet sein. Es muss schnell auf Änderungen reagieren können und selbstorganisiert sein. Als agiles Testen wird das Testen von Software im Rahmen eines agilen Entwicklungsprojekts bezeichnet. Testen in agilen Entwicklungsprojekten bedarf dabei vor allem eines Fokus auf die Unterstützung des Entwicklungsteams.

8.3.1. Wann in Scrum Testen

Es gilt der Grundsatz von Realisierung und Test non-stop. Während des Sprints muss dafür gesorgt werden, dass die „Tester“ vom ersten Tag an ins Team integriert sind und parallel mit (bzw. zusammen mit) den „Entwicklern“ an der Realisierung der Stories arbeiten. Schlüsselfertigkeiten hierfür sind:

- Die Beteiligung der Tester an der Sprint-Planung und den Weekly Scrum Meetings
- Die zeitliche Verschränkung von Entwicklungs- und Testaktivitäten. Hierzu sind Prinzipien wie „Test Driven Development“ bzw. „Test First“ und Pairing zwischen Entwicklern und Testern geeignet
- Die Kombination von systematischen Tests und explorativen Tests. Letztere liefern besonders schnell Feedback und finden Fehler, die von systematischen Tests ggf. nicht gefunden werden können. Diese wurden von mir als Tester durchgeführt.
- Die transparente Repräsentation von Tests in der Planung und Statusverfolgung durch geeignete Darstellung auf dem Taskboard, dem zentralen Planungsmittel des agilen Teams

8.4. Test-Prozess

Der fundamentale Testprozess (FTP) ist einer der am verbreitetsten Testprozesse und mittlerweile internationaler ISTQB-Standard. Der Prozess beinhaltet verschiedene Testaktivitäten, welche während des gesamten Testprozesses abgedeckt sein sollten. Diese Aktivitäten sind die Testplanung, der Testentwurf, die anschließende Testimplementierung sowie deren Ausführung, eine anschließende Testauswertung sowie ein abschließender Testabschluss. Jeder der im Folgenden erklärten Testtypen (Entwicklertest, Inkrement Test und Release-Test) soll all diese Testaktivitäten strukturiert in unterschiedlicher Intensität anwenden. Die Testplanung findet während des Sprint-Plannings und des Daily Scrums (weekly), der Testentwurf, die Testimplementierung und -ausführung während des Sprints, die Testauswertung während des Sprint-Reviews sowie der Testabschluss während der Sprint-Retrospektive stattfinden sollen. Für die einzelnen Testaktivitäten in den jeweiligen Testtypen sollten die

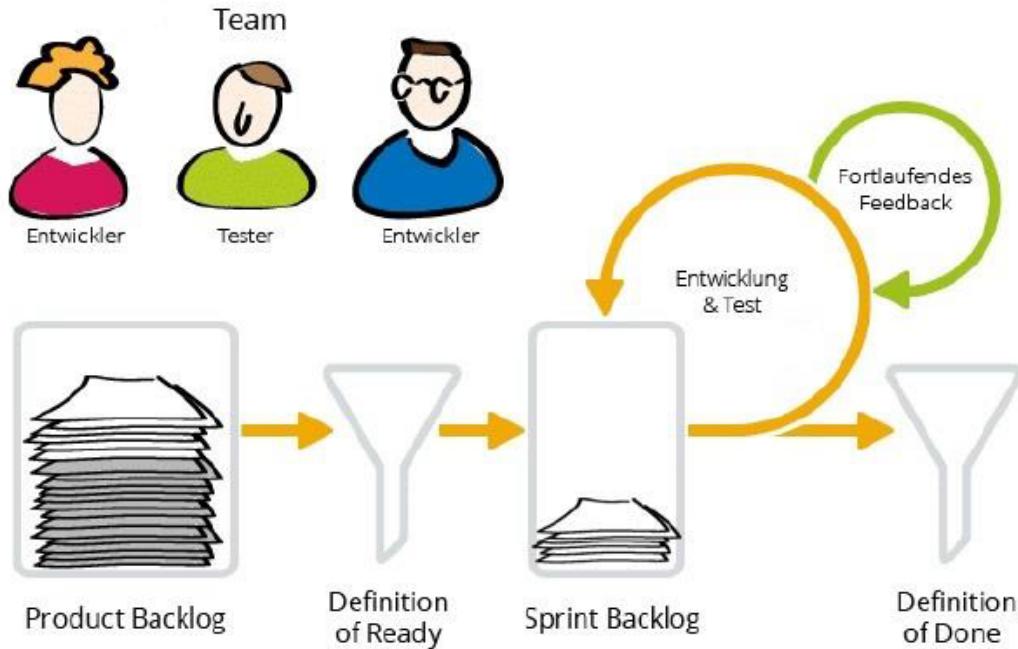


Abbildung (8.3) – Wann in Scrum Testen

folgenden Aspekte genutzt werden: eine Testbasis, gegen welche getestet wird (z.B. Anforderungen im Backlog), die Testobjekte, welche getestet werden (z. B. ein Inkrement), die Teststufen und Testarten, welche durchgeführt werden (z. B. Systemtest als Teststufe und Regressionstest als Testart) sowie die verschiedenen Testwerkzeuge, welche das Testen unterstützen. (s. Abbildung 8.4)

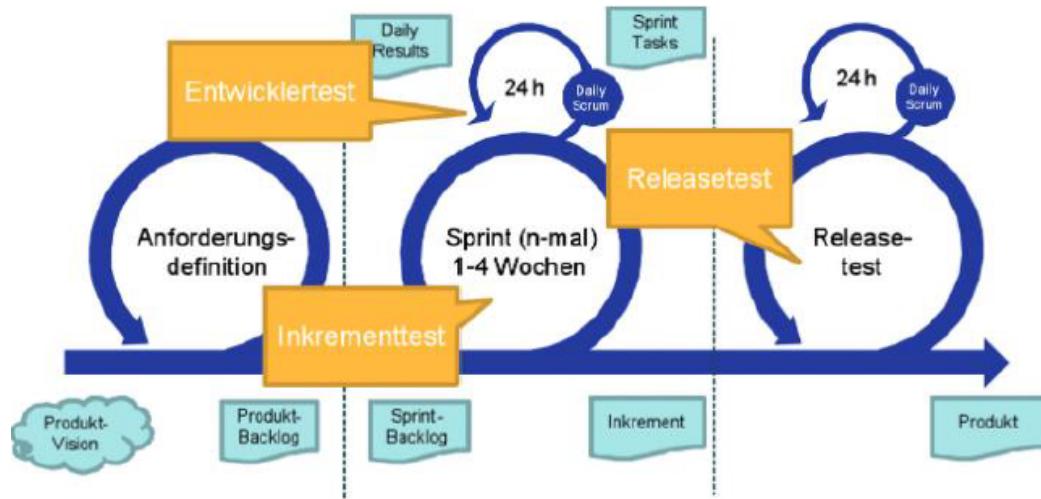


Abbildung (8.4) – Testprozess

8.4.1. Entwicklertest

Der Entwicklertest findet während der täglichen (in diesem Projekt wöchentlichen) Entwicklungsarbeit statt. Das Test Ziel ist die Validierung der täglichen (wöchentlichen) Ent-

wicklungsergebnisse gegen die Sprint-Tasks. Jeder Entwickler kümmert sich um seine aktuellen Sprint-Tasks, welche hier als Testbasis dienen, aus der die Testfälle abgeleitet werden müssen. Das Daily Scrum findet wie üblich, es werden allerdings zusätzlich auch die Testaufgaben und Testfälle besprochen. Diese Tests (s. Abbildung 8.5) werden vom Entwickler übernommen.

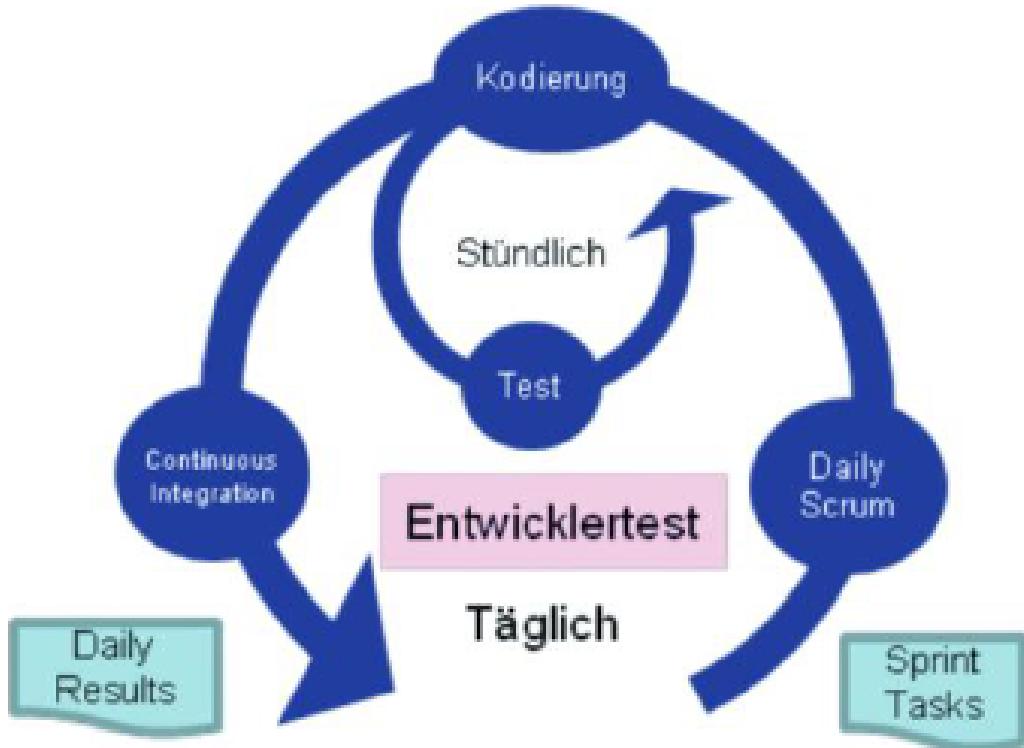


Abbildung (8.5) – Entwicklertest

8.4.2. InkrementTest

Der Entwicklertest findet im täglichen Rahmen statt, wohingegen der Inkrement-Test erst gegen Ende eines Sprints durchgeführt wird, wenn das Inkrement fertig wird. Obwohl Aktivitäten im Entwicklertest eine kontinuierliche Qualitätsüberwachung des Inkrements ermöglichen, ist am Ende eines Sprints ein dedizierter Inkrement-Test notwendig, um folgende Testziele zu gewährleisten.

- Die korrekte Funktionsweise des im Sprint entwickelten Inkrements und des korrekten Zusammenspiels mit früher entwickelten Inkrementen soll gezeigt werden.
- Es soll sichergestellt werden, dass frühere Inkremeante nicht verändert wurden. Der Test kann je nach Sprintlänge nach einer Woche oder auch erst nach vier Wochen durchgeführt werden.

Gegen Ende des Sprints ist ein Regressionstest sehr wichtig ist, damit die alte Funktionalität immer noch richtig läuft, nachdem neue Funktionalität hinzugefügt wurde. Die Testbasis sind im Gegensatz zum Entwickler-test nicht nur die einzelnen Sprint-Tasks, sondern das gesamte Sprint-Backlog, welches das zu liefernde Inkrement, hier auch das Testobjekt, definiert. Die Definition of Done besagt, wann ein Task wirklich fertig ist, wozu u. a. auch das Testen gehört. Im abschließenden Review Meeting, wird diese „Definition of Done“ vom Team zusammen mit dem Product Owner ausgewertet, ob sie für die Entwicklung und das Testen des Inkrements erfüllt wurde. Zur Testdurchführung gehören im Inkrement-Test als Teststufen neben dem Entwickler-test insbesondere auch der Systemtest, der prüft, ob das entsprechende Inkrement als ein Ganzes funktioniert und so vom Benutzer akzeptiert wird. Als Testarten gehören zu diesem Test-Typ nicht nur die funktionalen Tests und Regressions-tests, sondern insbesondere auch die nicht-funktionalen Tests, wie beispielsweise Performanz Tests, Usability-Tests usw.

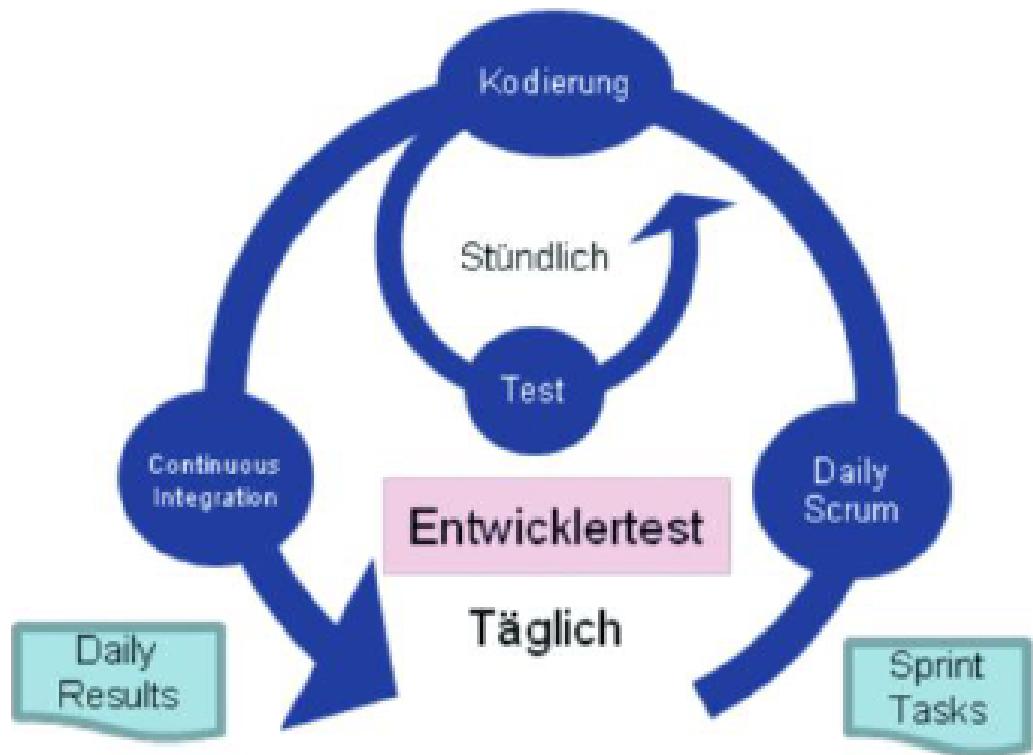


Abbildung (8.6) – Inkrement-Test

8.4.3. Release-Test

Der Release-Test wird das Produkt in seiner Gesamtheit getestet. Der Release-Test findet abschließend nach den Inkrementtests in Form eines dedizierten Sprints statt. Häufig wird eine Software inpotenziell fertigen Inkrementen entwickelt, welche aber anschließend zu einem gesamten Release zusammengefasst wird. Am Anfang wird geplant, nach wie vielen

Sprints ein Release üblicherweise fertig sein soll. Der Release-Test kann nun ebenfalls als ein Sprint aufgebaut werden und ist damit weiter Scrum-konform. Die Testbasis ist hier nun das gesamte Product Backlog, da hier nicht nur ein Inkrement nach einem Sprint, sondern das gesamte Produkt das Testobjekt ist. Anstatt eines Sprint-Backlogs kann analog ein sogenanntes Test-Backlog erstellt werden, welches die gesamten Release-Tests enthält.

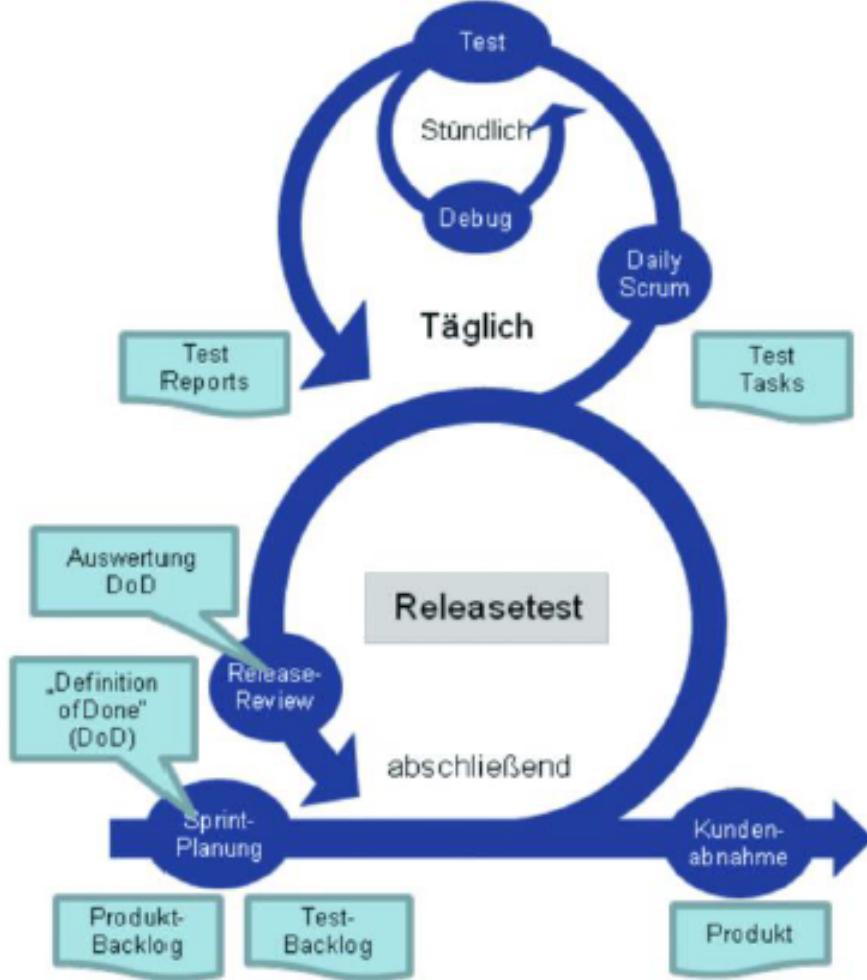


Abbildung (8.7) – Release-Test

Als Teststufen enthalten diese Release-Tests Systemintegrationstests, Inbetriebnahmetests und insbesondere Akzeptanztests. Im Release-Test können die Testfälle aus den Inkrement-Tests wiederverwendet werden, sodass für den Testentwurf kein erheblicher Zusatzaufwand entsteht. Als Teststart ist hier neben den funktionalen und nicht-funktionalen Tests auch das sogenannte End-To-End-Testing zu finden, welches über das gesamte System und Produkt im Zusammenspiel aller Komponenten stattfindet. Für die Durchführung der Tests wird das Test-Backlog analog zum Sprint-Backlog in Test-Tasks aufgeteilt, welche täglich entwickelt und durchgeführt werden. Weiterhin findet ein Daily Scrum statt um die Arbeit zu besprechen. Auf stündlicher Basis findet neben den Tests auch das Debugging statt, um Fehler zu

finden und zu beheben. Am Ende des Tages steht ein Test-Report an. Am Ende des Sprints findet nicht nur eine Auswertung der speziellen Definition of Done bezüglich des Releases statt, sondern ebenfalls die finale Kundenabnahme des Produktes.

8.5. Umsetzung im Projekt

Hier soll nun genauer erläutert werden, welche der Testmethoden in dem Projekt genutzt wurden, um möglichst hohe fehlerfrei Qualität unsere Plattform zu gewährleisten.

8.5.1. Testen gegen User-Stories

Beim Abtesten der User-Stories wurde das Verfahren des Black-Box Testens angewandt. Bei einem Black-Box-Test werden die Testfälle ausschließlich aus der Spezifikation (User-Stories) des zu testenden Objekts abgeleitet, ohne dabei dessen innere Struktur, also Architektur und Code, zu berücksichtigen (- diese werden als „Black Box“ behandelt). Es wird also nur das von außen sichtbare Verhalten des Testobjektes beobachtet.

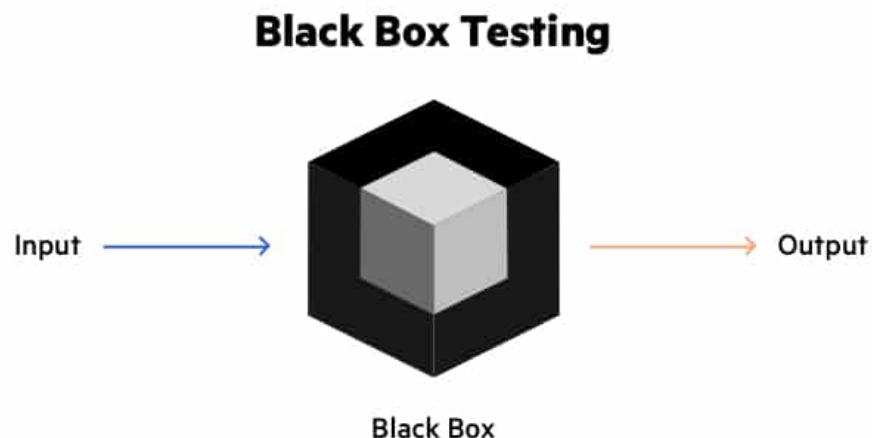


Abbildung (8.8) – Black-Box-Testing

Dabei wurde als Akzeptanzkriterium die Definition of Done der einzelnen User Stories genutzt. Dabei wurde der Fokus auf das positive Testing und negative Testing gelegt. Zusätzlich wurden aber auch einige Grenzwert-Testfälle spezifiziert, wo dies nötig war, und durchgeführt.

Positive Testing:

Positivtests sind eine Art von Softwaretests, bei denen davon ausgegangen wird, dass alles so abläuft wie erwartet. Es wird mit der Annahme durchgeführt, dass nur gültige und relevante Dinge auftreten werden.

Negative Testing:

Negativtests sind eine Art von Softwaretests, die durchgeführt werden, um das System auf unerwartete Bedingungen zu prüfen. Negative Tests spielen eine wichtige Rolle bei der Entwicklung leistungsfähiger Software. Dabei wird geprüft, wie sich die Software unter solchen unerwarteten Bedingungen verhält.

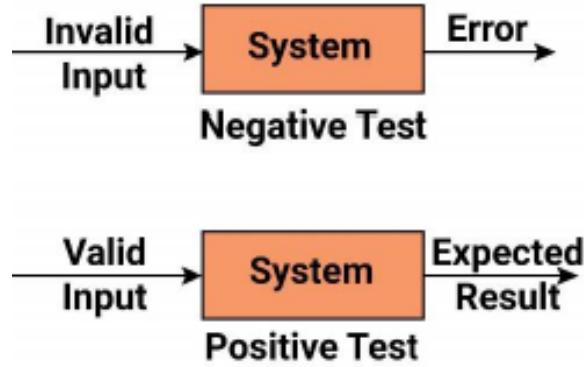


Abbildung (8.9) – Negative vs. Positive Testing

Grenzwert-Testing:

Das Testen von Grenzwerten testet extreme Grenzen der Eingangswerte. Unter extreme Grenzen fallen Enden wie Start-End-, Lower-Upper-, Maximum-Minimum-, Just Inside-Just Outside-Werte. Das Testen wird als 'Testen von Grenzwerten' bezeichnet.

Als Nutzer möchte ich meinem Beitrag mit Daten und Gültigkeitsdauer versehen um immer aktuelle Beiträge zu verfassen

Untergeordnete Vorgänge		Sortieren nach	...	+
DH-179	Requirement: Beitrag hat Gültigkeitsdauer	=	RV	FERTIG
DH-180	Requirement: Event hat Datum, Uhrzeit und Ort	=	RV	FERTIG

Abbildung (8.10) – Beispiel User-Story für Test

Die Umsetzung soll nun im Folgenden anhand eines Beispiels genauer erklärt werden. In diesem Beispiel wird der Fokus nur auf die Angabe des Gültigkeitsdatums gelegt (DH-179).

ID	User-Story ID	Test-Steps	Erwartetes Verhalten	Tatsächliches Verhalten	Type	Test-Result	Priority	Regression Priority	Bug-ID
26	DH-179	1. Erstelle einen Beitrag 2. Vergib dem Beitrag ein Gültigkeitsdatum (2 Wochen in der Zukunft von heute an) 3. Bestätige, dass Beitrag gepostet wurde 4. Überprüfe nach Ablauf des angegebenen Gültigkeitsdatums (z.B. nach 3 Wochen), dass der Beitrag nicht mehr angezeigt wird.	1. Beitragserstellung möglich 2. Eingabe des Gültigkeitsdatums möglich 3. Beitrag wurde gepostet 4. Beitrag wird nicht mehr angezeigt	1. Beitragserstellung möglich 2. Eingabe des Gültigkeitsdatums möglich 3. Beitrag wurde gepostet 4. Beitrag wird nicht mehr angezeigt	Positiv Testing	PASS	High	High	DH-237
27	DH-179	1. Erstelle einen Beitrag 2. Vergib dem Beitrag ein Gültigkeitsdatum 2 Wochen in der Zukunft 3. Bestätige, dass Beitrag gepostet wurde 4. Überprüfe vor dem Ablauf des angegebenen Gültigkeitsdatums (z.B. 1 Woche vorher), dass der Beitrag weiterhin angezeigt wird.	1. Beitragserstellung möglich 2. Eingabe des Gültigkeitsdatums möglich 3. Beitrag wurde gepostet 4. Beitrag wird angezeigt	1. Beitragserstellung möglich 2. Eingabe des Gültigkeitsdatums möglich 3. Beitrag wurde gepostet 4. Beitrag wird angezeigt	Negative Testing	PASS	High	High	/
28	DH-179	1. Erstelle einen Beitrag 2. Versuche Gültigkeitsdatum in der Vergangenheit anzugeben	1. Beitragserstellung möglich 2. Vergabe des ungültigen Datums nicht möglich	1. Beitragserstellung möglich 2. Vergabe des ungültigen Datums nicht möglich	Negative Testing	PASS	Medium	Medium	/
29	DH-179	1. Erstelle einen Beitrag 2. Versuche ein ungültiges Gültigkeitsdatum anzugeben (z.B. 37.04.2025)	1. Beitragserstellung möglich 2. Vergabe des ungültigen Datums nicht möglich	1. Beitragserstellung möglich 2. Vergabe des ungültigen Datums nicht möglich	Negative Testing	PASS	Low	Low	/

Abbildung (8.11) – Umsetzung der Testfaelle

8.5.2. Regressions-Test

Wenn eine Software durch die Einführung neuer oder geänderter Funktionen an Funktionalität verliert, spricht man von einem Rückschritt zu einem weniger entwickelten Zustand. Selbst geringfügige Änderungen an der Software oder am ursprünglichen Code können zu erheblichen Fehlern wie Abstürzen, Störungen und teilweisem oder vollständigem Verlust der Funktionalität führen. Regressionstests dienen dazu, diese Fehler zu erkennen und die Anwendung wieder zu stabilisieren. Sowohl funktionale als auch nicht-funktionale Testverfahren bewerten die Auswirkungen neuer Funktionen auf den bestehenden Code. In diesem Projekt wurde das folgendermaßen gelebt: Die Testfälle gegen User-Stories (4.5.1) wurden priorisiert. Anschließend wurden 1-2 Testfälle ausgewählt zu jeder User-Story, welche nach einer Optimierung oder Änderung des dazugehörigen Feature jedes Mal mit ausgeführten wurden. Das bedeutet zum Beispiel, dass zunächst die Funktion des Beitrags erstellen möglich war. Anschließend wurde die Funktion des Gültigkeitsdatums hinzugefügt. Danach wurde überprüft, dass alle vorherigen Funktionen weiterhin die Definition of Done erfüllen und alles weiterhin fehlerfrei funktioniert. Eine vollständige Wiederholung aller Testfälle wurde im letzten Sprint vor dem Release vorgenommen (Release Test), um sicherzustellen, dass bei der Übergabe zum Kunden alles fehlerfrei funktioniert.

8.5.3. Feld-Test

Ein Feldtest ist die Erprobung einer produktionsfähigen Vorabversion einer Software durch eine repräsentative Anzahl an von Anwendern. Das Ziel eines Feldtests ist das Erkennen von noch nicht komplett spezifizierten Einsatzumgebungen und Bedingungen sowie dem Prüfen auf Akzeptanz des Marktes. Dazu wurde eine bereite Anzahl an potentiellem Nutzer ins Visier genommen. Da die Plattform Generationenübergreifend sein soll, wurden daher Testpersonen mit unterschiedlichem Alter, unterschiedlichem Background Wissen und unterschiedlichen Interessen ausgewählt. Für die Durchführung des Feld-Testes wurde eine Guideline erstellt und der Tester hat sich zusammen mit der Testperson an einen Rechner gesetzt und die aktuelle Plattform ausprobiert. Der Tester hat dabei die Testperson nur

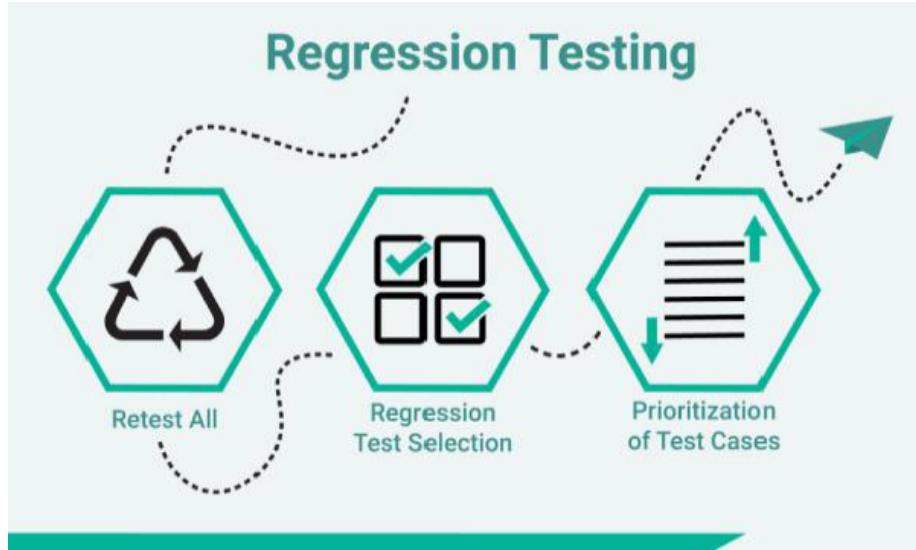


Abbildung (8.12) – Regressions-Test

gebeten gewisse Aktionen durchzuführen und hat die Testperson gebeten, dabei unentwegt Rückmeldung darüber zu geben, welche Gefühle er dabei empfindet und wie ihm die bestehenden Features gefallen. Abgeschlossen wurde der Test dadurch, dass die Personen ein kurzes Feedback geben sollten. Der Test wurde insgesamt drei Mal durchgeführt mit jeweils 6 Personen, und das Ergebnis wurde anschließend im Team besprochen, wo etwaige Änderungen vorgenommen werden müssen beziehungsweise welche Features eventuell neu priorisiert werden müssen.



Abbildung (8.13) – Feld-Test

Teilnehmer:

- P. Vogler (Alter: 53, weiblich, Bankkauffrau)
- N. Vogler (Alter: 55, männlich, Berufsschulehrer)
- D. Bandac (Alter: 26, männlich, Requirements-Ingenieur)
- D. Blana (Alter: 27, weiblich, Zahntechnikerin)
- T. Scholl (Alter: 36, männlich, System-Tester)
- K. Warmuth (Alter: 42, weiblich, Büro-Angestellte)

8.5.4. Exploratives-Testen

Bei dieser Methode stehen nicht Testpläne im Mittelpunkt, sondern die Tester „entdecken“ die Anwendung schrittweise selbst, indem sie diese benutzen. Parallel dazu erstellen die ersten Testfälle und optimieren sie iterativ in dem Maße, in dem mehr über die Anwendung gelernt wird. Gefragt sind hierbei vor allen Dingen Intelligenz, Intuition und Erfahrung. Im Gegensatz zu den klassischen Testmethoden – die dadurch keineswegs obsolet werden – fokussiert sich der explorative Ansatz auf die grundsätzliche Benutzbarkeit der Anwendung. Denn Schwächen in der Darstellung, umständliche Arbeitsabläufe oder logische Brüche, werden oft erst im Nutzererlebnis offenbar.

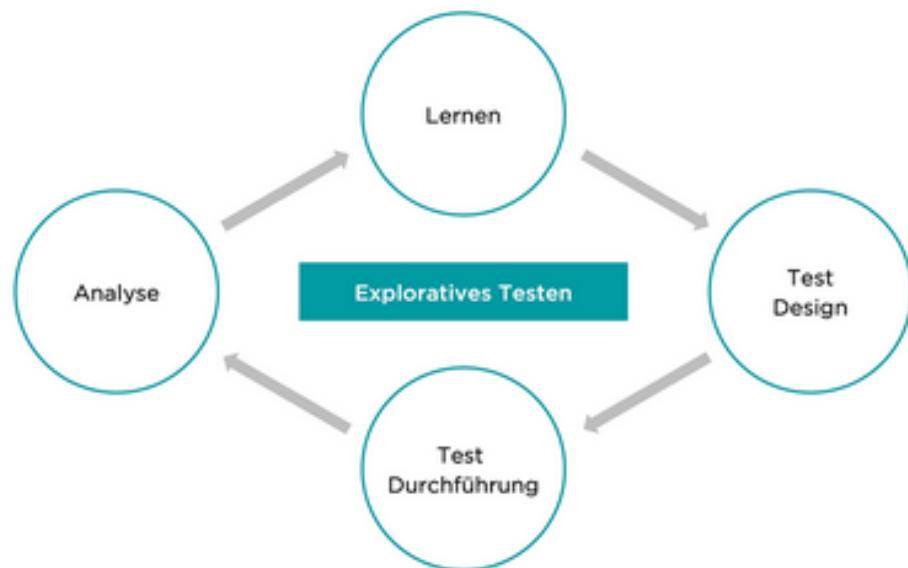


Abbildung (8.14) – Exploratives Testen

Hierbei spezialisierte vor allem Ende des jeden Sprints fokussiert. Dabei wurde überprüft, welche neuen Funktionen implementiert wurden. Anschließend wurden diese in den Fokus genommen und mit den bisherigen Funktionen zusammen ausprobiert. Dabei werden auch bereits bestehende Testfälle optimiert und neue Testfälle hinzugefügt (s. Dazu 4.5.1). Hinzu kommt, dass dabei auch die Usability der einzelnen Funktionen überprüft wurden um die

Qualität unserer Plattform aus der Sicht des Endbenutzers zu bewerten. Bei Auffälligkeiten wurden im Weekly Meeting Rücksprache mit dem Team gehalten, um etwaige Änderungen am Design zu erörtern und in die Wireframes wenn nötig einzubinden. Bei Fehlern wurde ein Fehlerbericht erstellt (Bug-Bericht), und dem jeweiligen Entwickler in Jira zugewiesen.

9. Fazit

Abschließend lässt sich zum erreichten Ergebnis festhalten, dass wir eine Plattform geschaffen haben, die dem gesetzten Projektziel entspricht. Die Plattform bietet mit ihren einzelnen Features die Möglichkeit der Vernetzung von Nachbarn. Hierbei sind vor allem die generallistisch zu verwendenden Beiträge zu nennen. Mit den Beiträgen erreichen Menschen oder Vereine die nötige Reichweite, um wichtige Informationen preiszugeben. Des weiteren ist es durch die Möglichkeit etwas anzubieten (Geben) bzw. Anfragen zu stellen (Nehmen) möglich, dass Menschen in der Nachbarschaft in Kontakt kommen bzw. sich über die Plattform kennenlernen können.

Die Benutzbarkeit der Plattform ist durch die intuitive und übersichtliche Benutzeroberfläche gegeben. Die Benutzeroberfläche ist hierbei in erster Linie für Desktop-Geräte optimiert, kann jedoch aufgrund der ausgewählten Technologie mit begrenztem Aufwand für mobile Geräte angepasst werden.

Durch die Verwendung von Firebase, das von Google betrieben wird, sowie von Vercel zum Hosting der Plattform, ist eine sehr hohe Verfügbarkeit der Plattform gegeben.

Auch wenn es Mangels der Zeit nicht möglich war, alle Features in einer optimalen Qualität zu implementieren, haben wir es in der knappen Zeit geschafft, eine funktionierende Plattform zu entwickeln, die in iterativen Schritten immer weitere Features dazugewonnen hat. Die Aufgabenstellung an sich hätte sicherlich eine Vielzahl weiterer Features ergeben, umso wichtiger war unsere Diskussion, welche Features für die Plattform im ersten Schritt wirklich notwendig sind. Ich finde, unsere Plattform entspricht diesem Prinzip und kann sich deshalb sehen lassen.

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

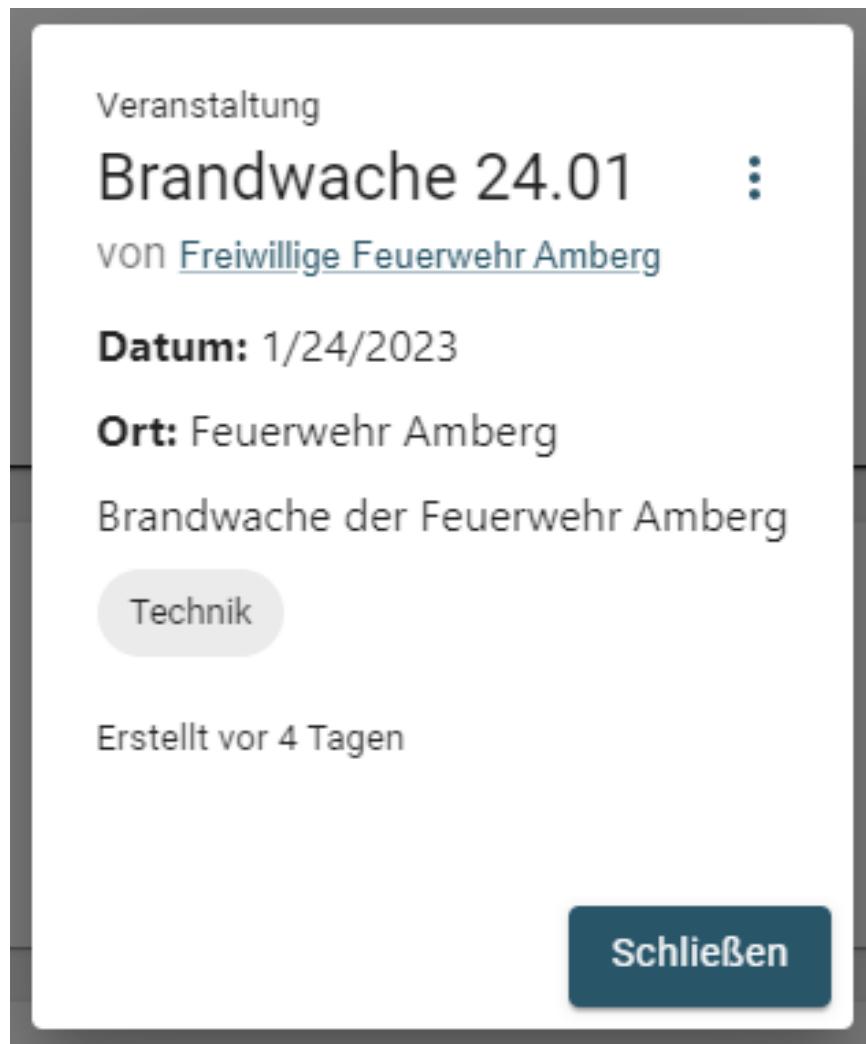


Abbildung (A.2) – Beitragsdetails

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

3.1.	Beispiel für einen Task im JIRA	5
3.2.	Epic Übersicht des Projekts	6
3.3.	Beispiele für Functional Tasks	7
3.4.	Beispiel für ein Bug Ticket	8
3.5.	Darstellung des Ablaufs eines einzelnen JIRA Tickets	8
3.6.	Backlog Funktion im JIRA	9
3.7.	Beiträge in der User Story Map	10
4.1.	Startseite von FB	16
4.2.	Startseite von nebenan.de	18
4.3.	Startseite von Spontacts	21
4.4.	Value Proposition Canvas	24
4.5.	Beispielprofil	29
4.6.	Chats in Digital Dahoam	29
4.7.	Marktplatz von Digital Dahoam	30
4.8.	Merkzettel	31
4.9.	Seitenstruktur einer Website	32
4.10.	Grundgerüst einer Website	33
4.11.	Beispiel eines Rastersystems	34
4.12.	Aufbau eines fixen und flexiblen Layouts	35
4.13.	Beispielhafte Darstellung von LFW und HFW	36
4.14.	Interactionsschritte an einem Beispiel: Wireframe - Mockup - Prototyp	37
4.15.	Figmas-Arbeitsumgebung	38
4.16.	Startseite (DH)	40
4.17.	Marktplatz (DH)	40
5.1.	Übersicht der technischen Softwarearchitektur	41
6.1.	Vergleich Downloadzahlen verschiedener Frontend Frameworks [10]	45
7.3.	Übersicht eines Benutzerprofils	53
7.4.	Anmeldemaske	53
7.9.	Benutzereinstellungen	58
7.10.	Persönliches Profil	58
7.11.	Blockieren eines Nutzers	59
7.12.	Blockierte Nutzer	59
7.13.	Beiträge erstellen	60
7.14.	Beitrag	61
7.15.	Beitragsmenü	61

7.16. Beiträge des Nutzers	62
7.17. Merkzettel	62
7.18. Marktplatz	63
7.19. Dashboard	63
A.1. Benutzerdatensatz in Firestore	A
A.2. Beitragsdetails	B

Tabellenverzeichnis

4.1. Musikverein Customer Profile	25
4.2. Musikverein Value Proposition	26
4.3. Alleinstehender Mann Customer Profile	26
4.4. Alleinstehender Mann Value Proposition	26
4.5. Alleinstehende Frau Customer Profile	27
4.6. Alleinstehende Frau Value Proposition	27
4.7. Junges Pärchen Customer Profile	28
4.8. Junges Pärchen Value Proposition	28
6.1. Funktionen von Github	48
6.2. Funktionen von Jira	50

Literaturverzeichnis

- [1] Facebook. Meta Platforms. [Online]. Available: <https://www.facebook.com/>
- [2] nebenan.de. Good Hood GmbH. [Online]. Available: <https://nebenan.de/>
- [3] Spontacts. SYNEXIT GmbH. [Online]. Available: <https://spontacts.com/>
- [4] M. Shapiro. (2016) Grid system guide pngs. [Online]. Available: <http://matthiasshapiro.com/DisneyShorts/grid-system-guide-pngs>
- [5] Y. Vasileva. (2022) Ux wireframes and how they help us during the design process. [Online]. Available: <https://www.resolutesoftware.com/news/ux-wireframes/>
- [6] AppsChopper. (2021) A quick guide on mobile app wireframe vs mockup vs prototype. [Online]. Available: <https://www.appschopper.com/blog/quick-guide-on-mobile-app-wireframe-vs-mockup-vs-prototype/>
- [7] Typescript, “JavaScript With Syntax For Types.” 2022. [Online]. Available: <https://www.typescriptlang.org/>
- [8] ——, “Documentation - React,” Jan. 2023. [Online]. Available: <https://www.typescriptlang.org/docs/handbook/react.html>
- [9] React, “React – A JavaScript library for building user interfaces,” 2022. [Online]. Available: <https://reactjs.org/>
- [10] NPM, “@angular/core vs react vs vue | npm trends,” Jan. 2023. [Online]. Available: <https://npmtrends.com/@angular/core-vs-react-vs-vue>
- [11] Google, “Design - Material Design,” May 2021. [Online]. Available: <https://m2.material.io/design>
- [12] MUI, “MUI: The React component library you always wanted,” 2022. [Online]. Available: <https://mui.com/>
- [13] G. Firebase, “Firebase,” 2022. [Online]. Available: <https://firebase.google.com/>
- [14] G. Authentication, “Firebase Authentication,” 2022. [Online]. Available: <https://firebase.google.com/docs/auth?hl=de>
- [15] F. Realtime Database, “Firebase Realtime Database,” 2022. [Online]. Available: <https://firebase.google.com/docs/database?hl=de>
- [16] G. Firestore, “Firestore,” 2022. [Online]. Available: <https://firebase.google.com/docs/firestore?hl=de>

- [17] G. CloudStorage, “Cloud-Speicher für Firebase | Cloud Storage for Firebase,” 2022. [Online]. Available: <https://firebase.google.com/docs/storage?hl=de>
- [18] Github, “Build software better, together,” 2022. [Online]. Available: <https://github.com>
- [19] Vercel, “Develop. Preview. Ship. For the best frontend teams – Vercel,” 2022. [Online]. Available: <https://vercel.com/>
- [20] Atlassian, “DH Board - Agile board - Jira,” Jan. 2023. [Online]. Available: <https://digitaldahoam.atlassian.net/jira/software/projects/DH/boards/1>