

Bachelorarbeit

# Entwicklung eines positionsbezogenen Spiels als Web-App für Freiburg

Nico Gräf

Professur für Algorithmen und Datenstrukturen  
Albert-Ludwigs-Universität Freiburg

Begutachtet von  
Prof. Dr. Hannah Bast

Bearbeitungszeitraum  
04.03.2020 – 04.06.2020

# Eidesstattliche Erklärung

Hiermit erkläre ich, dass ich die nachfolgende Bachelorarbeit selbstständig angefertigt habe, keine anderen als die angegebenen Quellen oder Hilfsmittel verwendet habe und alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten Schriften entnommen wurden, als solche kenntlich gemacht habe. Darüber hinaus erkläre ich, dass diese Abschlussarbeit nicht, auch nicht auszugsweise, bereits für eine andere Prüfung angefertigt wurde.

*Nico Gräf*

# Zusammenfassung

Jedes Jahr werden innerhalb Deutschlands fast vier Millionen Umzüge über Gemeindegrenzen hinweg verzeichnet. Fehlende Verbundenheit und Ortskenntnisse, Einsamkeit und ein gemindertem Wohlbefinden gehören zu den Herausforderungen, denen Menschen nach einem Umzug begegnen. Befragungen und Studien weisen darauf hin, dass digitale Lösungen und Apps bei diesen Herausforderungen helfen könnten. In dieser Arbeit wurde prototypisch ein Spiel für Freiburg entwickelt, welches Wissen vermitteln, Ortskenntnisse verbessern, soziale Aktivitäten fördern und das allgemeine Wohlbefinden steigern soll. Dafür wurden über die Analyse von Studien Anforderungen an eine entsprechende App definiert. Anschließend wurde ein Konzept entworfen und die App designet. Konzept und Design wurden in die Entwicklung einer Web-App überführt und die implementierte App mit mehreren Testpersonen evaluiert. Die Evaluation hat gezeigt, dass die App auf spielerische Art Wissen und Ortskenntnisse vermitteln kann und dabei zur langfristigen Nutzung motiviert. Ebenfalls sind durch die Evaluation Schwachstellen im Konzept und dem Design aufgedeckt worden. Die soziale Aktivität der Testpersonen konnte durch die entwickelte App nicht gefördert werden. Anhand der Ergebnisse der Evaluation konnte eine Anforderungsliste für die weitere Entwicklung abgeleitet werden.

# Inhaltsverzeichnis

Eidesstattliche Erklärung .....	I
Zusammenfassung .....	II
Inhaltsverzeichnis .....	III
1 Einleitung .....	1
1.1 Ziel der Arbeit .....	3
1.2 Aufbau der Arbeit .....	3
2 Bestehende Lösungen .....	5
2.1 Stadt-Apps .....	5
2.2 Tourismus-Apps .....	7
2.3 Apps für Geflüchtete .....	8
3 Herausforderungen und Lösungen.....	10
3.1 Unzufriedenheit und gemindertem Wohlbefinden.....	10
3.2 Einsamkeit .....	11
3.3 Stress und Aufwand.....	12
4 Design der App.....	13
4.1 Das Konzept .....	13
4.2 Personas .....	15
4.3 Der Design-Prototyp.....	20
5 Entwicklung der App.....	28
5.1 Grundlagen .....	29
5.2 Aufbau und Funktionsweise der App .....	43
5.3 Die App aus Sicht der Nutzenden.....	51
5.4 Admin-Panel.....	60
5.5 Tests und Programmierstil.....	62
6 Evaluation.....	64

6.1	Usability-Test der App .....	65
6.2	Usability-Test des Admin-Panels .....	72
6.3	Umgesetzte Anforderungen.....	73
6.4	Praxistest.....	76
7	Ergebnis.....	78
8	Literatur .....	79

# 1 Einleitung

Deutschland ist ein Zuwanderungsland. Das zeigt die Entwicklung der Zu- und Fortzüge aus den letzten Jahren. [39] Inzwischen hat in Deutschland knapp jede vierte Person einen Migrationshintergrund. [38] Für Eingewanderte und Geflüchtete gibt es bereits einige Informationsangebote, die dabei helfen in Deutschland anzukommen. Beispielsweise bietet das Bundesamt für Migration und Flüchtlinge ein Online-Portal und eine Broschüre mit wichtigen Informationen zum Ankommen in Deutschland. [8, 9] Ausgelöst durch die enorme Zuwanderung in den Jahren 2015 bis 2017 wurden sogar spezielle Smartphone-Anwendungen (Apps) für Geflüchtete entwickelt. Die App „Moin Refugee“ war kostenlos für Android- und iOS-Smartphones erhältlich und wurde als mehrsprachiger Wegweiser für Geflüchtete in Schleswig-Holstein konzipiert. [32] Auch das Bundesamt für Migration und Flüchtlinge bietet eine App als Begleiter für die ersten Wochen in Deutschland. [6]

Aber auch innerhalb von Deutschland findet Migration in hohem Maße statt. Tatsächlich ist die Wanderungsrate innerhalb der Landesgrenzen höher als zwischen Deutschland und dem Ausland. Jedes Jahr werden hierzulande durchschnittlich fast vier Millionen Umzüge über Gemeindegrenzen hinweg verzeichnet. [39] Das bedeutet, dass allein in Deutschland jedes Jahr bis zu vier Millionen Mal eine neue Heimat gesucht und damit ein neues Leben begonnen wird. Bei vielen dieser Umzüge müssen die Menschen sich komplett neu orientieren, neue Freizeitaktivitäten und vor allem neue soziale Kontakte aufbauen.

Bei älteren Studien in Australien und der Türkei gaben mehr als 30 Prozent der Teilnehmenden an, mit ihrem Umzug nicht zufrieden zu sein, den Umzug zu bereuen oder die Rückkehr in Erwägung zu ziehen. [3, 41] Die Arbeiten von Knight und Gunatilaka (2010) und Liu et al. (2017) zeigen, dass Menschen in China, die von einer ländlichen Region in eine Stadt ziehen, durchschnittlich eine geringere Lebenszufriedenheit empfinden als Menschen, die in ihrer Heimat bleiben. [28, 31] Und auch eine aktuelle Studie aus Deutschland zeigt einen Unterschied in der allgemeinen Zufriedenheit zwischen Zugezogenen und Einheimischen. Hendriks, Ludwigs und Veenhoven (2016) konnten bei ihrer Studie mit Düsseldorfer Studierenden feststellen, dass Einheimische –

verglichen mit den zugezogenen Studierenden – im Allgemeinen glücklicher und zufriedener sind. 24 Als entscheidende Faktoren für die persönliche Zufriedenheit wurden in dieser Studie die verbrachte Zeit außer Haus, Bewegung im Allgemeinen und soziale Interaktionen beobachtet.

Laut den Daten des Sozio-ökonomischen Panels (SOEP) aus dem Jahr 2017 fühlen sich in Deutschland 9,5 Prozent der Menschen oft oder sehr oft einsam. [5, 30, zitiert nach: Einsamkeit in Deutschland] Während ein kurzweiliges Gefühl von Einsamkeit eher eine Frage des persönlichen Wohlbefindens ist, kann Einsamkeit über längere Zeit ein erhebliches Gesundheitsrisiko darstellen. [5, 26] Holt-Lunstad, Smith und Layton (2010) folgern aus Ihrer Meta-Studie „Social relationships and mortality risk“, dass soziale Beziehungen einen höheren (oder vergleichbaren) Einfluss auf die Lebenserwartung haben als Rauchen, Alkoholkonsum oder körperliche Inaktivität. [26]

Betrachtet man die Faktoren Gemeinsinn, Zivilengagement, Gesundheit und Lebenszufriedenheit des „OECD Better Life Index“, liegt Deutschland auf Platz 18 im internationalen Vergleich. [34] Bezüglich der Frage wie eng sich Menschen mit den Personen in Ihrer Wohngegend verbunden fühlen, belegt Deutschland laut der „Europäische Erhebung zur Lebensqualität 2016“ sogar den letzten Platz im europäischen Vergleich. [14]

Auslöser für Einsamkeit sind unter anderem der Umzug in eine fremde Stadt und der Beginn eines Studiums oder einer Ausbildung, mit dem oftmals auch ein Umzug einhergeht. [5, 25, 36]

Aufgrund der Schwierigkeiten, die mit einem Umzug verbunden sind, bedarf es Anwendungen, die auf die Bedürfnisse und Probleme der Menschen nach einem Umzug abgestimmt sind. Besonders empfohlen werden Angebote, die nach einem Umzug in den Bereichen Sozialleben und Freizeitgestaltung unterstützen. [15, 24, 26, 36]

Da die Nutzerzahlen von Smartphones in Deutschland steigen, könnten entsprechende Apps eine Lösung darstellen. [4] Bei einer aktuellen Bitkom-Studie sagten 87 Prozent der Smartphone-Nutzenden, dass das Smartphone ihnen eine große Erleichterung im Alltag bietet. Rund 65 Prozent sind der Meinung, dass sie ihr Smartphone anderen Menschen

näherbringt. [4] Gleichzeitig gibt es deutlichen Digitalisierungsbedarf. Die Mehrheit der Bevölkerung hält ihre Heimat für nicht digital, können sich aber vorstellen in einer Stadt oder Gemeinde mit vielen digitalen Angeboten zu leben. [7]

Wie erfolgreich Smartphone-Anwendungen Menschen dazu motivieren können, sich mehr in ihrer Stadt zu bewegen, zeigt das positionsbezogene Spiel Pokémon Go. Es wurde weltweit mehr als eine Milliarde Mal heruntergeladen und wird von Millionen Menschen aktiv auf der ganzen Welt gespielt. [1, 42] Bei positionsbezogenen Spielen wie Pokémon Go müssen sich die Spielenden in der echten Welt bewegen, um im Spiel weiter zu kommen. Viele Spielelemente sind an reale Orte geknüpft. Bei Pokémon Go können sich Spielende vor Ort auch gegenseitig unterstützen oder gegeneinander antreten. Im Juli 2019 fand im Dortmunder Westfalenpark ein mehrtägiges Pokémon Go Fest statt. Rund 85 tausend Menschen aus der ganzen Welt trafen sich hier zum gemeinsamen Spielen. [13] Dieses Beispiel zeigt wie positionsbezogene Smartphone-Anwendungen mit spielerischen Elementen dafür sorgen können, dass sich Menschen mehr in ihrer Stadt bewegen und darüber mit anderen Menschen in Kontakt kommen.

## **1.1 Ziel der Arbeit**

Jedes Jahr ziehen mehr als 20 tausend Menschen nach Freiburg. [40] Im Rahmen dieser Bachelorarbeit soll ein positionsbezogenes Spiel für die Stadt Freiburg als Smartphone-Anwendung entwickelt werden. Die Anwendung (im weiteren Verlauf als App bezeichnet) soll spielerisch Wissen über die Geschichte und Kultur von Freiburg vermitteln, dazu motivieren, die lokalen Sehenswürdigkeiten aufzusuchen und Möglichkeiten schaffen, mit anderen Menschen in Kontakt zu kommen. Die App soll so konzipiert werden, dass Menschen nach ihrem Umzug nach Freiburg dabei unterstützt werden die (noch) fremde Stadt besser kennenzulernen und zu erkunden.

## **1.2 Aufbau der Arbeit**

Diese Arbeit ist in sieben Kapitel unterteilt. Kapitel zwei beschäftigt sich mit den bestehenden Lösungen. Kapitel drei betrachtet die Herausforderungen, denen Menschen nach einem Umzug begegnen, sowie mögliche Lösungen, mit denen man diesen Herausforderungen begegnen kann. Im vierten Kapitel werden daraus resultierenden



Anforderungen an die zu entwickelnde App abgeleitet und der Designprozess erläutert. Das fünfte Kapitel beschreibt die Implementierung als Web-App. Im sechsten Kapitel wird die entwickelte App evaluiert und im letzten, dem siebten Kapitel werden die Ergebnisse diskutiert.

## 2 Bestehende Lösungen

Diese Arbeit konzentriert sich auf Smartphone-basierte Lösungen (Apps). Aus diesem Grund werden hier klassische Angebote wie Willkommenspakete, Stadtführer, sowie Ratgeber oder E-Government-Services nicht betrachtet. Im Folgenden werden bestehende Lösungen aus drei Anwendungsgebieten vorgestellt und bezüglich der Problemstellung dieser Arbeit bewertet.

### 2.1 Stadt-Apps

Stadt-Apps bieten allgemeine Informationen und Dienstleistungen, die in erster Linie von der Stadt selbst angeboten werden. Die App „Mein Jena“ versorgt Nutzende (Personen, die diese App nutzen) beispielsweise mit lokalen Nachrichten, einem Veranstaltungskalender, Wohnungsangeboten, ÖPNV-Plänen und der Möglichkeit Tickets für den Bus oder das Schwimmbad zu kaufen. [37] Die Lösungen der Firma „cm city media“ gehen noch einen Schritt weiter. Hier können Städte auch Informationen zu Sehenswürdigkeiten, sowie Rad- und Wanderwegen bereitstellen. Auch können die Städte ein Verzeichnis über die lokale Gastronomie und die lokalen Vereine führen. [10] Die meisten Stadt-Apps bieten die gleichen Informationen wie Ratgeber und Broschüren. Darüber hinaus können sie die Nutzenden durch Suchfunktion, Menüführung und modernem User Interface schneller und einfacher helfen, beispielsweise die gesuchte Einrichtung zu finden und eine Wegbeschreibung abzurufen. Zudem bieten Funktionen wie die lokalen Nachrichten und der Veranstaltungskalender den Nutzenden die Möglichkeit sich über regionale Themen und das lokale Sozialleben zu informieren und helfen somit teilweise bei der Freizeitgestaltung. Allerdings erfordern diese Apps immer noch eine hohe Eigeninitiative und Eigenmotivation. Es gibt keinen gesonderten Bereich und keine in sinnvoller Weise aufbereiteten Informationen für Zugezogene. Eher führt das tägliche Lesen über das lokale Geschehen und die eigene, oft mit erhöhtem Aufwand verbundene, Auseinandersetzung mit den weiteren Funktionen langfristig zu einem Wissenszuwachs und verbesserter Ortskenntnis. Eine vorgegebene Struktur, die den Nutzenden den Einstieg erleichtert und ihn beim Erkunden und Kennenlernen der neuen

Heimat begleitet, wäre hier wünschenswert. Ein weiterer sehr entscheidender Nachteil ist die fehlende Funktion, über die App mit anderen Menschen in Kontakt zu kommen.

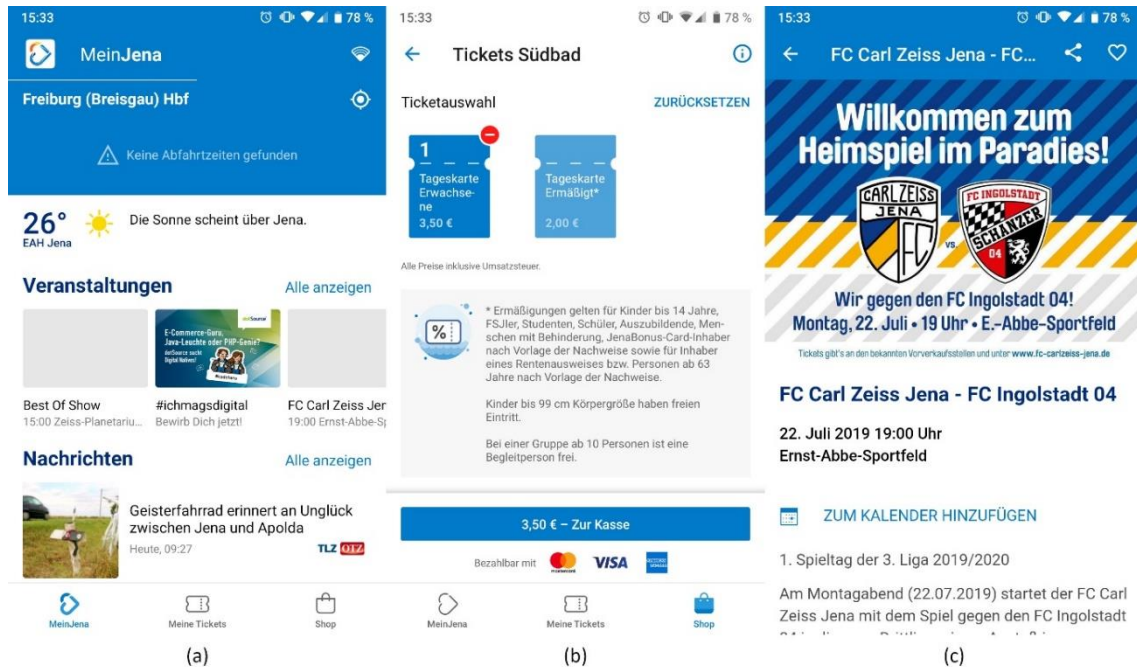


Abbildung 1: Die App „Mein Jena“. (a) Hauptbildschirm mit Übersicht zu allen Funktionen. (b) Ticketkauf für das Schwimmbad. (c) Veranstaltungseintrag. Quelle: Eigene Screenshots

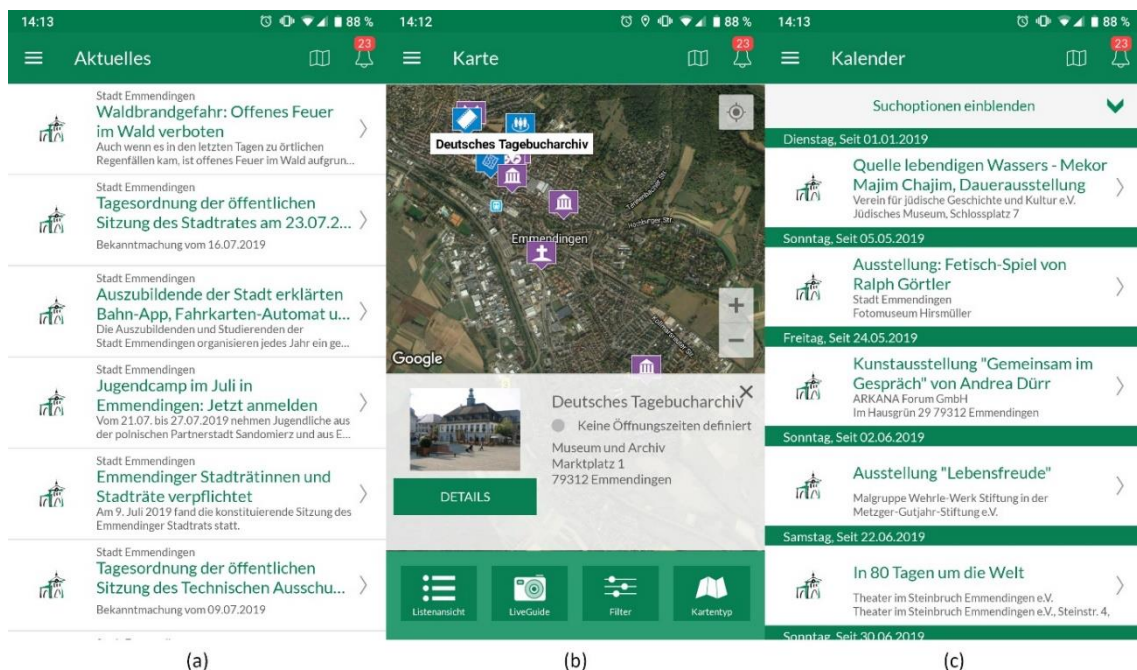


Abbildung 2: Stadt-App der Stadt Emmendingen. (a) Lokale Nachrichten (b) Karte mit Sehenswürdigkeiten (c) Veranstaltungskalender. Quelle: Eigene Screenshots

## 2.2 Tourismus-Apps

Stellvertretend für Tourismus-Apps wird hier die App „Urban Challenger“ vorgestellt. [2] Diese App bietet für die Stadt Berlin eine Stadtrallye mit Aufgaben aus verschiedenen Kategorien. Es gibt auch Aufgaben, bei denen fremde Menschen – insbesondere Einheimische – angesprochen oder in die Aufgabe mit eingebunden werden müssen. Diese Aufgaben sind in einzelne (digitale) Spielkarten aufgeteilt und können nach Belieben ausgewählt werden. Zu jeder Aufgabe gibt es eine Beschreibung, Hinweise und eine Auflistung der einzelnen Schritte. Über ein Menü können die Aufgaben nach Kategorien gefiltert werden. So können Nutzende die Stadt nach eigenen Interessen in einem spielerischen Kontext erkunden und mit Einheimischen in Kontakt kommen. Angebote aus dem Tourismus sind meist nur für die Nutzung von einem oder wenigen Tagen gedacht. Die Aufgaben der App „Urban Challenger“ helfen dabei, die wichtigsten Sehenswürdigkeiten und ebenfalls einige lokale Besonderheiten zu entdecken. Allerdings führt das Konzept nicht zu einer langfristigen Nutzung: Es gibt keine Anreize für die Nutzenden, die App täglich, wiederholt oder langfristig zu nutzen. Auch bleiben die sozialen Interaktionen begrenzt, da es keine Möglichkeit in der App gibt, mit den Menschen in Kontakt zu bleiben. Ebenso wie bei den Stadt-Apps wäre hier eine vorgegebene Struktur, die Nutzenden den Einstieg erleichtert und die Aufgaben in sinnvoller Reihenfolge organisiert, hilfreicher. Da es keine kostenlose Version gibt, entsteht eine weitere Hürde vor der Nutzung dieser App.

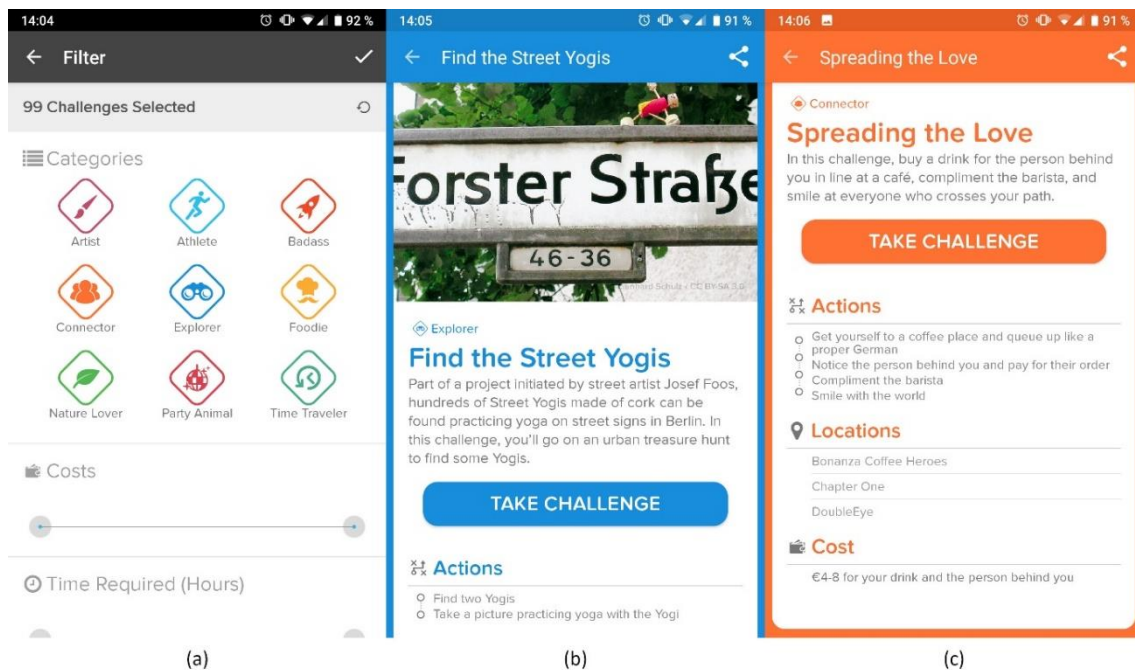


Abbildung 3: Die App Urban Challenger. (a) Kategorienauswahl (b) Eine Aufgabe zur Erkundung von Straßenschildern (c) Aufgabenbeschreibung und Informationen für eine Connector-Challenge. Quelle: Eigene Screenshots

## 2.3 Apps für Geflüchtete

Mit der stark gestiegenen Zuwanderung und der zahlreichen Aufnahme von Geflüchteten in den Jahren 2015 bis 2017 entstanden Apps und weitere digitale Angebote für Geflüchtete. Diese Apps haben als Ziel, den Geflüchteten wichtige Informationen für ihre Ankunft und praktische Tipps für das Leben in Deutschland leichter zugänglich und in ihrer Sprache zur Verfügung zu stellen. Die „Ankommen-App“ des Bundesamtes für Migration und Flüchtlinge ist solch eine App. [6] Sie wurde zusammen mit dem Goethe-Institut entwickelt. Die Inhalte der „Ankommen-App“ sind in drei Bereiche unterteilt. Unter „Leben in Deutschland“ gibt es Tipps zur Mobilität in Deutschland, zum Schulsystem, den Religionen, zur Politik und vielem mehr. Weiterhin gibt es einen Bereich für Asyl, Beruf und Ausbildung. Im dritten Bereich können Geflüchtete über mehrere Kapitel Deutsch lernen. Die gesamte App und alle Inhalte sind in Deutsch, Französisch, Englisch, Arabisch und Persisch (Farsi) verfügbar.

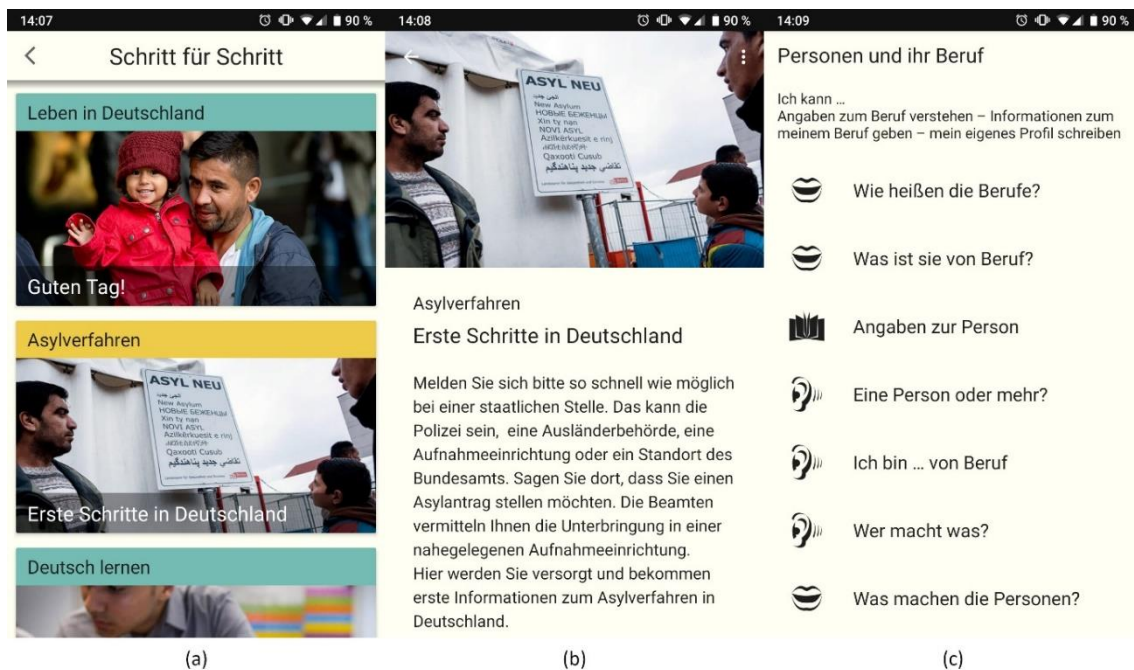


Abbildung 4: Die App Ankommen des BAMF. (a) Der Schritt für Schritt Wegbegleiter schlägt eine Reihenfolge aller Kapitel vor. (b) Informationen zum Asylverfahren (c) Ein Kapitel aus dem Deutschkurs. Quelle: Eigene Screenshots

Eine ähnliche App für Geflüchtete in Schleswig-Holstein hatte eine Firma aus Kiel entwickelt. [32] Die App „Moin Refugee“ stellte die wichtigsten Informationen auf sechs verschiedenen Sprachen zur Verfügung und half Zugewanderten und Geflüchteten dabei öffentliche Behörden und Einrichtungen zu finden.

Die Apps für Geflüchtete vereinen viele klassische Angebote in digitaler Form. Stadtführer, wichtige Informationen, Ratgeber und Wörterbuch wurden hier zusammengefasst und digital aufbereitet. Bei der „Ankommen App“ gibt es im Gegensatz zu den anderen vorgestellten Apps auch eine geführte Version, bei der Schritt für Schritt an die Inhalte durchgearbeitet werden können. Für die Nutzenden wurde eine sinnvolle Reihenfolge der verfügbaren Inhalte gewählt und optional zur Verfügung gestellt. Diese geführte Version hilft beim Einstieg. Allerdings fehlen hier Möglichkeiten, mit der lokalen Bevölkerung Bekanntschaften zu machen. Des Weiteren fordern diese Apps eine hohe Disziplin, da sich die Inhalte hauptsächlich auf Informationen konzentrieren und es keine motivierenden oder spielerischen Elemente gibt.

# 3 Herausforderungen und Lösungen

In der Einleitung wurden schon einige Herausforderungen und Probleme genannt, mit denen Menschen nach einem Umzug konfrontiert werden. In diesem Kapitel werden nun aus diesen Herausforderungen und Problemen konkrete Anforderungen für die zu entwickelnde App abgeleitet.

Unmittelbar nach einem Umzug fallen zunächst administrative Aufgaben, wie zum Beispiel das Ummelden beim Bürgeramt oder das Einschreiben an der Hochschule an. Für diesen Bereich gibt es in der Regel genügend Informationen und Hilfestellung von der Stadt selbst. Dieser Bereich soll nicht Teil der zu entwickelnden App sein.

## 3.1 Unzufriedenheit und gemindertes Wohlbefinden

Um herauszufinden wieso Zugezogene weniger zufrieden und glücklich mit ihrer Lebenssituation sind, haben Hendriks, Ludwigs und Veenhoven (2016) das tägliche Leben und die Freizeitgestaltung von einheimischen und zugezogenen Studierenden in Düsseldorf untersucht. [24] Dabei zeigten sich einige Unterschiede. Die einheimischen Studierenden, die insgesamt ein subjektiv höheres Wohlbefinden haben und durchschnittlich glücklicher sind, verbrachten viermal mehr Zeit mit „social drinks“ und auf Partys (soziale Veranstaltungen) verglichen mit den zugezogenen Studierenden. [24] Außerdem haben die einheimischen Studierenden im Schnitt täglich doppelt so lange Sport getrieben und auch fast doppelt so viel Zeit ihren Hobbies gewidmet. Die dadurch entstehende Differenz verbrachten die zugezogenen Studierenden hauptsächlich an ihrem Computer. Weiterhin konnte beobachtet werden, dass die Einheimischen sich täglich eine Stunde länger außer Haus aufhielten (abgesehen vom Arbeitsplatz und der Zeit im Verkehr). Aus ihrer Studie schließen Hendriks et al., dass Einheimische unter anderem deshalb glücklicher sind, weil sie mehr Zeit außer Haus, auf sozialen Veranstaltungen, beim Sport oder mit ihren Hobbies verbringen. [24] Hendriks et al. sind daher der Meinung, dass Menschen nach einem Umzug von Hilfe bei der Freizeitgestaltung und Unterstützung beim Aufbau neuer Sozialkontakte profitieren. [24]

Eine weitere Studie aus Deutschland beschäftigte sich allgemeiner mit der Lebenszufriedenheit einer Stadtbevölkerung. Bei dem Projekt „Glücklich in Wuppertal“<sup>1</sup> machten 1800 Bürgerinnen und Bürger aus Wuppertal Angaben zu ihrem Wohlbefinden. Die Ergebnisse zeigen, dass die Bürgerinnen und Bürger Wuppertals glücklicher sind, wenn sie sich der Stadt verbunden fühlen, sie Zeit draußen verbringen (und nicht Zuhause oder bei der Arbeit), sich mit befreundeten Personen treffen, sie sich als Teil der Gemeinschaft fühlen und sich engagieren. [22, 23] Gerade soziale Kontakte sind für das Wohlbefinden der Teilnehmenden entscheidend. [22]

## 3.2 Einsamkeit

In der Einleitung wurde Einsamkeit als eine gesellschaftliche Herausforderung vorgestellt und der Umzug in eine fremde Stadt als ein Hauptgrund identifiziert. In diesem Kapitel geht es um mögliche Lösungen zur Minderung und Prävention von Einsamkeit.

Einsamkeit wird meist als Mangel an sozialen Kontakten und Unzufriedenheit über die Beziehungen mit anderen Menschen verstanden. [36] Soziale Beziehungen zu pflegen oder diese neu zu knüpfen und dadurch langfristige, enge Verbindungen mit anderen Menschen aufzubauen ist daher essenziell, um Einsamkeit zu meiden. Gerade nach einem Umzug muss hier oft neu begonnen werden. Deshalb gelten Maßnahmen, die Möglichkeiten für soziale Verbindungen und Interaktionen mit anderen Menschen fördern als besonders effektiv gegen Einsamkeit. [36] Ähnlich wie bei der Zufriedenheit und dem Wohlbefinden hängt die (häufigere) Teilnahme an Freizeitaktivitäten auch mit geringerer Einsamkeit zusammen. [15, 36] Eyerund und Orth (2019) leiten aus den Ergebnissen ihrer Studie über die Einsamkeit in Deutschland ab, dass sich Maßnahmen, welche die Verfügbarkeit und Erreichbarkeit von Freizeitangeboten verbessern, positiv auf geringere Einsamkeit auswirken. [15] Weiterhin besteht eine Wechselwirkung zwischen persönlichem Wohlbefinden und Einsamkeit. Eine Studie aus Großbritannien ergab, dass Kinder und Jugendliche (bis 24 Jahre) mit niedrigerer Lebenszufriedenheit, Glücksempfinden und Selbstwertgefühl häufiger angaben, einsam zu sein. [36] Soziale, emotionale und mentale Probleme können die Teilnahme am Sozialleben und an sozialen

---

<sup>1</sup> <https://www.gluecklich-in-wuppertal.de>



Aktivitäten erschweren und so wiederum Einsamkeit fördern. [36] Deshalb sollte eine entsprechende Lösung die Teilnahme an sozialen Aktivitäten erleichtern.

### **3.3 Stress und Aufwand**

Ganz andere Herausforderungen bilden der Stress und der zusätzliche Aufwand, die mit einem Umzug und größeren Veränderungen im Leben einhergehen. Hendriks et al. folgern, dass Stress zusammen mit dem Zeit- und Energieaufwand, der ein Umzug verlangt, dafür sorgen können, dass Zugezogene keine mentale und physische Energie mehr dafür aufbringen können, sich nach Freizeitangeboten umzuschauen oder an sozialen Aktivitäten teilzunehmen. [24] Außerdem haben Zugezogene oft keinen Überblick über alle Angebote und verfügen über zu wenig Ortskenntnisse. Gerade zu Beginn fehlen Zugezogenen oft sozialen Kontakte, die sie motivieren, zu privaten Veranstaltungen mitzunehmen, und über die sie an Aktivitäten außerhalb ihres Kenntnisstands und Erfahrungsbereiches teilnehmen können.

Entsprechende Lösungen dürfen daher nur wenig Aufwand, insbesondere Zeit, Energie und Geld erfordern und es muss für Nutzende einfach sein (d.h. ohne mentalen Stress zu erzeugen) neue Aktivitäten auszuprobieren und neue Menschen kennenzulernen. Bestenfalls bereitet allein die Nutzung einer solchen Anwendung den Nutzenden schon Freude und sorgt so für eine zusätzliche Motivation.

# 4 Design der App

Mithilfe der Erkenntnisse aus den vorangegangenen Kapiteln und der Liste der Anforderungen aus dem nächsten Unterkapitel wurden ein Konzept erarbeitet und anschließend erste Entwürfe der App skizziert. Um ein besseres Problemverständnis und eine genauere Vorstellung der späteren Verwendung der App zu bekommen, wurde die Persona-Methode angewandt. Die erstellten Personas wurden genutzt, um auf Basis des Konzepts einen Design-Prototyp zu entwickeln. Dieses Kapitel beschreibt den Designprozess und die Überlegungen, die in das Konzept und das Design der App eingeflossen sind.

In den vorangegangenen Kapiteln wurden die Anforderungen an eine entsprechende App ausführlich erläutert. Während des Designprozesses wurden diese Anforderungen wie folgt zusammengefasst definiert: Die App hilft den Nutzenden dabei (1) mehr über die (neue) Heimat zu lernen, (2) sich mehr außer Haus zu bewegen und (3) mit anderen Menschen in Kontakt zu kommen. Dabei muss (4) der Einstieg in die App einfach gestaltet sein und (5) Nutzende sollten fortlaufend motiviert werden.

## 4.1 Das Konzept

In der Grundidee besteht die App aus einem Spiel mit täglichen Aufgaben. Diese Aufgaben heißen im Spiel „Challenges“. Challenges sollen Nutzende spielerisch dazu motivieren die Stadt zu erkunden und kennenzulernen. Eine Freunde-Funktion hilft in Kontakt mit anderen Menschen zu kommen. Ein Spielsystem mit Punkten, Statistiken und Ranglisten soll dafür sorgen, dass Nutzende langfristig Spaß beim Lernen, Entdecken, und Kontakteknüpfen hat. Das Spiel beginnt auf dem Smartphone und stellt somit anfangs keine große Hürde dar. Je länger die Nutzenden spielen, desto mehr neue Orte, Geschichten und Menschen lernen sie über das Spiel kennen.

### **4.1.1 Spielverlauf**

Der Spielverlauf ist recht simpel. Neue Nutzer und Nutzerinnen landen direkt im Spiel. Jeden Tag gibt es neue Challenges. Diese Challenges gelten nur für den aktuellen Tag. Für jede gemeisterte Challenge erhalten die Nutzenden jeweils Punkte. Eine Rangliste zeigt die Punkte aller Nutzenden im Vergleich. Neben dem Spielen von Challenges können Nutzende untereinander Freundschaften schließen. Dafür gibt es ebenfalls Punkte. Nutzende können sich Statistiken und eine persönliche Historie in einem eigenen Profil ansehen.

### **4.1.2 Challenges**

Es gibt drei Arten von Challenges: Quiz-Challenges, Geo-Challenges und Content-Challenges.

Die Quiz-Challenge besteht aus einem einfachen Quiz aus 5 Multiple-Choice-Fragen zu Freiburg. Bei jeder Frage gibt es 15 Sekunden Zeit zum Antworten. Bei Abgabe einer Antwort oder Ablauf der Antwortzeit wird die Frage aufgelöst. Die Quiz-Challenge soll den Nutzenden helfen, mehr über die Geschichte und Kultur der Stadt zu lernen.

Bei der Geo-Challenge müssen die Nutzenden einen Ort in der echten Welt aufsuchen. Die Aufgabe gilt als erledigt, sobald der Ort geographisch erreicht wurde. Hierfür soll die GPS-Funktion von Smartphones genutzt werden. Diese positionsbezogene Challenge ist dafür gedacht, dass sich die Nutzenden draußen bewegen und die Sehenswürdigkeiten, sowie weitere interessante Orte in der Stadt entdecken können. Standardmäßig gibt es eine klare Beschreibung des Ortes, zusätzlich zum Namen und einem Bild. Außerdem wird den Nutzenden ihre Entfernung zum Ort angezeigt. Mit weniger Angaben bzw. Hilfestellungen zum Ort bekommt die Geo-Challenge einen Rätsel-Charakter. Beispielsweise wird nur ein Bild von dem Ort angezeigt und die Nutzenden müssen selbst herausfinden, wo dieses Bild aufgenommen wurde. So werden die Nutzenden dazu motiviert, sich frei in der Stadt zu bewegen und ihre Umgebung bewusst zu erkunden.

Bei der Content-Challenge bekommen die Nutzenden einen Text angezeigt und müssen diesen Inhalt als gelesen markieren (akzeptieren), um die Challenge zu erledigen. Die Content-Challenge kann genutzt werden, um die Nutzenden über etwas zu informieren oder ihnen Spielelemente zu erklären.

Bei allen Challenges gibt es eine Einladungs-Funktion. Mit dieser Einladungs-Funktion kann ein Link zu der entsprechenden Challenge versendet werden. Nutzende können so andere Personen zum Beispiel über Whatsapp oder per E-Mail zur Challenge einladen.

### **4.1.3 Freundschaften**

Im Spiel gibt es die Möglichkeit Freundschaftsanfragen an andere Nutzende zu senden bzw. Freundschaftsanfragen zu erhalten. Nutzende bekommen für das Versenden und Annehmen von Freundschaftsanfragen Punkte. Befreundete Nutzende können sich in der speziellen Freundes-Rangliste miteinander messen.

## **4.2 Personas**

Im vorherigen Kapitel wurde das Konzept der App vorgestellt. Mithilfe von Personas wird dieses Konzept nun auf mögliche Anwendungsszenarien getestet. Personas sind Profile von fiktiven Nutzenden, die eine Zielgruppe idealtypisch repräsentieren sollen. Personas bestehen aus einer Personenbeschreibung, den Zielen der Person und einem oder mehreren Szenarien, in denen die Lösung (hier: die App) zum Erreichen dieser Ziele beiträgt. Das Erstellen solcher Personas soll dabei helfen, die Bedürfnisse, Erwartungen und Ziele der Nutzenden besser zu verstehen. Dadurch können Anforderungen an eine Lösung besser erkannt und genauer definiert werden. [33] Obwohl ähnliche Methoden schon zuvor bestanden, wird heutzutage in vielen Bereichen hauptsächlich die Persona-Methode eingesetzt, welche 1999 erstmals von Alan Cooper in seinem Buch „The inmates are running the asylum“ beschrieben wurde. [11, 33]

Es wird empfohlen eine Persona als primäre Persona zu definieren, welche den hauptsächlichsten Anwendungsfall beschreibt. Darüber hinaus können eine oder mehrere sekundäre Personas erstellt werden. Die sekundären Personas sind dazu da weitere Aspekte oder ähnliche Anwendungsfälle abzudecken. Die Anforderungen, die aus der primären Persona hervorgehen, sind denen der sekundären Personas übergeordnet.

Bei der Persona-Methode muss beachtet werden, dass alle Annahmen, die beim Erstellen einer Persona gemacht wurden, evaluiert oder durch Daten und Analysen von echten Nutzenden begründet werden sollten.

Für die zu entwickelnde Anwendung wurden im Rahmen des Designprozesses eine primäre und eine sekundäre Persona erstellt. Beide Personas stellen Szenarien dar, die aus der Problemstellung hervorgehen. Es wurden im Vorfeld keine Interviews oder ähnliche Datenerhebungen durchgeführt. Ebenfalls wird auf eine vollständige Beschreibung der Personas verzichtet. Die erstellten Personas sollen vielmehr das Anwendungsszenario verdeutlichen und wurden für die Entwicklung des Design-Prototyp zur Hilfe gezogen. Im Folgenden werden sowohl die primäre als auch die sekundäre Persona vorgestellt.

#### **4.2.1 Primäre Persona**

Anna Fuchs ist 19 Jahre alt und wuchs mit ihren Eltern und ihren beiden Brüdern in Saarbrücken auf. Sie wandert gerne, reist viel und hält sich in ihrer Freizeit am liebsten draußen auf. Nach Abschluss ihres Abiturs, ist Anna nach Freiburg gezogen, um dort Geografie zu studieren. Anna plant sowohl den Bachelor als auch den Master in Freiburg zu absolvieren. Deshalb wird Freiburg mindestens für die nächsten fünf Jahre ihre neue Heimat.

Während der Einführungswoche ihres Studiengangs hat Anna schon ein paar ihrer Mitstudierenden kennengelernt. Da sie ein offener Mensch ist, konnte sie schnell neue Freunde finden. Allerdings fühlt sich Anna in Freiburg noch nicht heimisch. Das macht sie unglücklich. Das Studium ist deutlich anspruchsvoller als erwartet. Anna verbringt die meiste Zeit in der Uni, draußen ist sie selten unterwegs. Sie kennt sich in Freiburg noch überhaupt nicht aus und fühlt sich den Einheimischen und der Kultur noch sehr fremd. Anna möchte gerne mehr von der Stadt und der Umgebung sehen und auch mehr über Freiburg und die Kultur lernen. Sie würde sich gerne mehr mit ihrer neuen Heimat identifizieren.

Anna beschließt, sich in ihrer Freizeit die Sehenswürdigkeiten und besonderen Orte Freiburgs anzuschauen. Außerdem hat sie begonnen, sich auf Wikipedia über die Geschichte und Kultur von Freiburg zu informieren. Nach ein paar Tagen ist ihr das Lesen der Wikipedia-Artikel zu trocken und zu anstrengend.

#### **Annas Ziele:**

- Mit mehr Freude und Spaß über die Geschichte und Kultur von Freiburg lernen
- Ohne groß zu planen mehr Orte in Freiburg und der Umgebung entdecken

## **Szenario 1**

1. Anna bekommt von einer Kommilitonin einen Link geschickt.
2. Sie öffnet den Link auf ihrem Smartphone und landet in der App.
3. Sie kann sofort ein Quiz über Freiburg starten.
4. Anna spielt das Quiz durch. Das Quiz macht ihr Spaß und sie lernt etwas Neues über Freiburg.
5. Nach dem Quiz erhält Anna Punkte für ihre richtigen Antworten.
6. Anna hat die Möglichkeit ein weiteres Quiz zu spielen. Weil ihr das erste Quiz Spaß gemacht hat, spielt sie auch das zweite Quiz.
7. Nach dem zweiten Quiz erhält Anna ebenfalls Punkte. Die Punkte des zweiten Quiz addieren sich zu ihrem Punktestand.
8. Am nächsten Tag öffnet Anna wieder den Link. Es stehen neue Quizze zu Verfügung.
9. Anna spielt nun täglich ein Quiz, bei dem sie jedes Mal etwas Neues über Freiburg und die Kultur erfährt.
10. Annas Mitstudierende spielen auch täglich. In einer Rangliste können alle sehen, wer die meisten Punkte sammelt.

## **Szenario 2**

1. Anna möchte mehr Zeit draußen verbringen und die Sehenswürdigkeiten ihrer neuen Heimat erkunden.
2. Nachdem sie ein paar Tage lang Quizze gespielt hat, fällt Anna auf, dass die App ihr eine Geo-Challenge präsentiert.
3. Die Challenge hat den Titel „Besuche das Colombischlössle“.
4. Anna hat noch nie vom Colombischlössle gehört. In der Beschreibung der Challenge liest sie die nötigen Informationen dazu und sieht ein Bild vom Colombischlössle.
5. Anna beschließt am nächsten Tag mit ihren Mitstudierenden das Colombischlössle aufzusuchen. Die Beschreibung und die angezeigte Entfernung zum Ziel in der App helfen ihnen dabei.
6. Als Anna und ihre Mitstudierende am Colombischlössle angekommen sind, zeigt ihr die App an „Du bist da!“. Anna freut sich und sie erhält Punkte.

7. Weil sie schon dort sind, erkunden Anna und ihre Mitstudierende das Colombischlössle, besuchen das archäologische Museum im Innern und liegen noch eine Weile auf der Wiese im Park davor.
8. In den darauffolgenden Tagen erscheinen weitere Geo-Challenges, die Anna und ihre Mitstudierende an verschiedene Orte in Freiburg führen.

#### **4.2.2 Sekundäre Persona**

Martin Neumann ist 35 Jahre alt und von Beruf Softwareentwickler. Vor sechs Monaten ist er für eine neue Arbeitsstelle nach Freiburg gezogen. Zuvor hat er in Berlin gelebt. Die Arbeit macht ihm Spaß, jedoch kommt er mit seinen neuen Kolleginnen und Kollegen nicht so gut klar. Er findet keinen Anschluss und fühlt sich sozial ausgegrenzt.

Weil er in Freiburg noch keine Freunde und Hobbies gefunden hat, konzentriert Martin sich zurzeit voll auf seinen Job. An den Wochenenden liest er Bücher aus seinem Fachbereich oder kocht sich selbst asiatische Gerichte. Manchmal läuft Martin auch durch die Stadt und sucht Orte, die er bisher noch nicht entdeckt hat. Dafür hat er extra einen Stadtführer in der Bücherei neben seiner Wohnung gekauft. Martin liest auch gerne die Informationen und Hintergründe zu den Sehenswürdigkeiten, die er sich anschaut. Das gibt ihm das Gefühl mehr in Freiburg zuhause zu sein und sich nicht mehr so sehr als Außenseiter zu fühlen.

Martin ist nicht der Typ Mensch, der andere einfach so anspricht. Für ihn gehört zu einem Gespräch immer auch ein Anlass oder ein Thema, über das man sich unterhält. Deshalb ist Martin auch auf der Suche nach einem passenden Hobby, bei dem er andere Leute kennenlernt.

#### **Martins Ziele:**

- In der Freizeit über einen lockeren Anlass mit anderen Menschen in Kontakt kommen

#### **Szenario 1**

1. Martin wird auf die App aufmerksam und öffnet sie.

2. Nachdem er ein paar Quizze über Freiburg gespielt hat, möchte er sich am Wochenende wieder auf die Suche von Orten machen, die er noch nicht kennt. Diesmal nutzt er die App dafür.
3. Er läuft zu Fuß durch die Stadt zu dem Ort, den er für die Challenge im Spiel aufsuchen muss. Ein paar Hintergrundinformationen dazu hat er aus den Quiz-Challenges schon erfahren.
4. Am Ort angekommen bemerkt er andere, die auch mit ihrem Smartphone an diesen Ort laufen.
5. Ein Mann in Martins alter spricht ihn an. Er heißt Christian und fragt, ob Martin das gleiche Spiel spielt und ob Martin sich mit ihm befreunden möchte.
6. Martin freut sich darüber. Die beiden unterhalten sich noch ein paar Minuten über die App und den Ort, an dem sie sich befinden. Danach verknüpfen sich beide in der App über die Freunde-Funktion. Nun können sie sich in der Freundesliste in der App sehen und in der Rangliste gegenseitig ihren Fortschritt verfolgen.
7. Am nächsten Tag sieht Martin, dass Christian ihn zu einer weiteren Geo-Challenge eingeladen hat.
8. Martin klickt auf den Link und landet direkt in der Geo-Challenge.
9. Martin und Christian verabreden sich, die Challenge zu einer bestimmten Uhrzeit gemeinsam zu erledigen.
10. Nach einigen Tagen begegnen sich Martin und Christian wieder zufällig an einem anderen Ort. Sie unterhalten sich länger und gehen danach zusammen einen Kaffee trinken.



## 4.3 Der Design-Prototyp

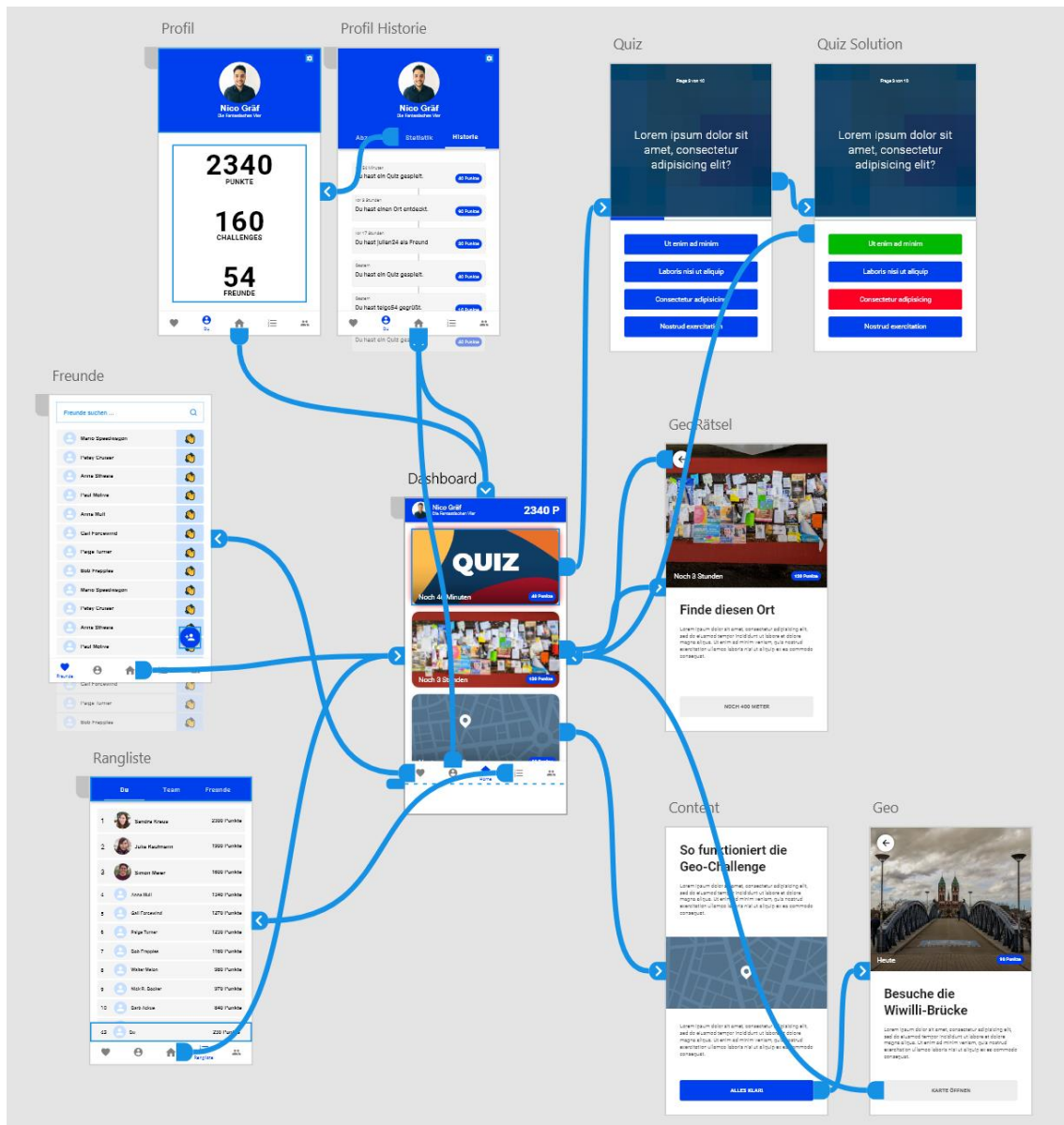


Abbildung 5: Screenshot des Design-Prototyps in Adobe XD. Die blauen Verbindungen zeigen die Verlinkungen der Seiten im Prototyp-Modus an.

Mithilfe des Designprogramms Adobe XD<sup>2</sup> wurde vor dem Beginn der Entwicklung ein Design-Prototyp erstellt. Adobe XD bietet neben dem Designen von Nutzeroberflächen (User Interfaces) auch die Möglichkeit aus den Designs einen interaktiven Prototypen zu bauen. Das in Kapitel 4.1 vorgestellte Konzept wurde in diesem Schritt in eine prototypische App überführt. Dabei wurde für jeden Bereich der App das User Interface

<sup>2</sup> <https://www.adobe.com/de/products/xd.html>

entworfen und die einzelnen Komponenten zu einem Prototyp miteinander verbunden. Das von Google entwickelte, weitverbreitete Designsystem Material Design<sup>3</sup> diente dabei als Grundlage. Der Design-Prototyp diente insgesamt eher zur Orientierung, statt als feststehende Entwicklungs-Vorlage. Deshalb gibt es teilweise Unterschiede zur entwickelten App.

Für den Design-Prototyp wurden auch Funktionen und Screens entworfen, die für eine eventuelle Weiterentwicklung der App angedacht sind. Gleichzeitig sind während der Entwicklung Funktionen hinzugekommen, die im Vorfeld nicht designet wurden. Im Folgenden werden die wichtigsten Bereiche des Design-Prototyps vorgestellt.

---

<sup>3</sup> <https://material.io/>

### 4.3.1 Das Dashboard

Das Dashboard ist der Haupt- und Startbildschirm der App. Nutzende landen in diesem Bereich, wenn sie die App starten. Im Dashboard werden alle spielbaren (aktiven) Challenges angezeigt. Jede Challenge wird durch eine Kachel dargestellt, auf welcher auch die noch verbleibende Zeit und die möglichen Punkte angezeigt werden. Ebenfalls sehen Nutzende ihre aktuellen Punkte und ihren Nutzernamen (in der App „Username“ genannt) am oberen Bildschirmrand in einer für Material Design typischen „App Bar“. Außerdem befindet sich am unteren Bildschirmrand eine Navigationsleiste (eine bei Material Design sogenannte „Bottom Navigation Bar“), um zwischen den einzelnen Bereichen zu navigieren.

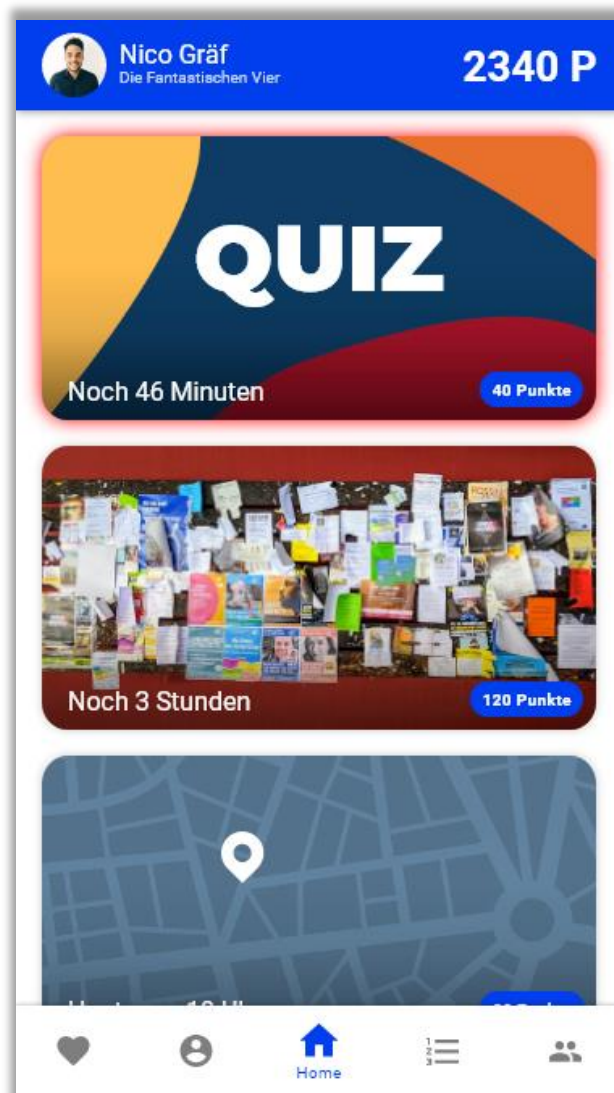


Abbildung 6: Entwurf des Dashboards.

### 4.3.2 Die Challenge-Seite

Die Challenge-Seite wird angezeigt, wenn ein Nutzer beziehungsweise eine Nutzerin auf eine Challenge im Dashboard klickt. Die Challenge-Seite bietet alle Informationen zur jeweiligen Challenge und bietet die Möglichkeit durch eine entsprechende Aktion diese Challenge zu erledigen. Jeder Challenge-Typ (Quiz-Challenge, Geo-Challenge, Content-Challenge) hat dabei sein individuelles Design.

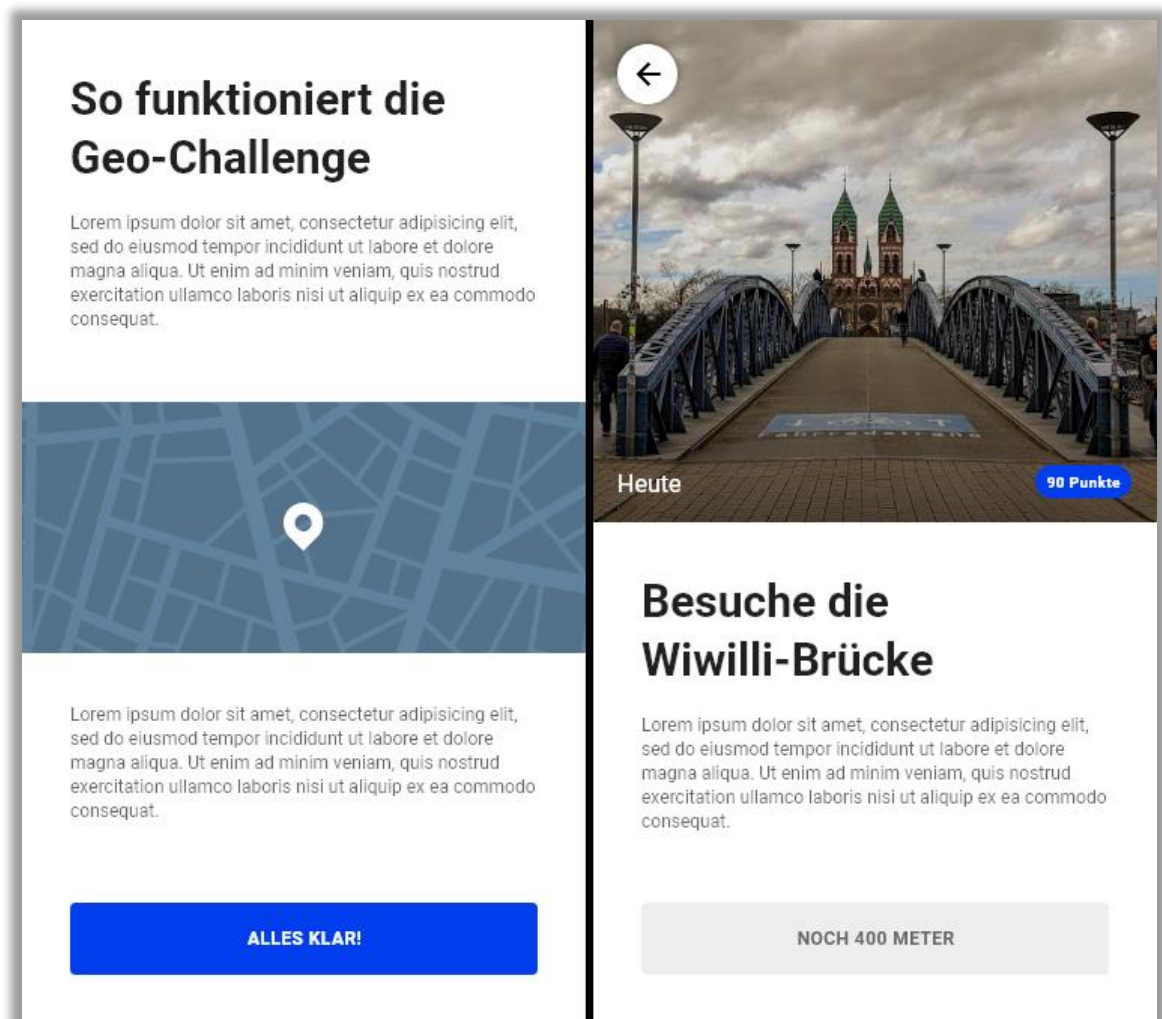


Abbildung 7: Entwurf der Challenge-Seite. Links: Content-Challenge. Rechts: Geo-Challenge.

### 4.3.3 Das Quiz

Das Quiz ist ein spezieller Bereich in der Web-App. Ein Quiz wird gestartet, wenn Nutzende bei einer Quiz-Challenge auf der Challenge-Seite auf den „Quiz-Starten“-Button klicken. Nach abschließen des Quiz landen die Nutzenden wieder im Dashboard. Die Fragen des Quiz werden nacheinander angezeigt. Es gibt eine Anzeige zur Anzahl der Fragen (z.B. „Frage 3 von 5“) und eine Countdown-Leiste, welche die noch verbleibende Zeit für die aktuelle Frage anzeigt.

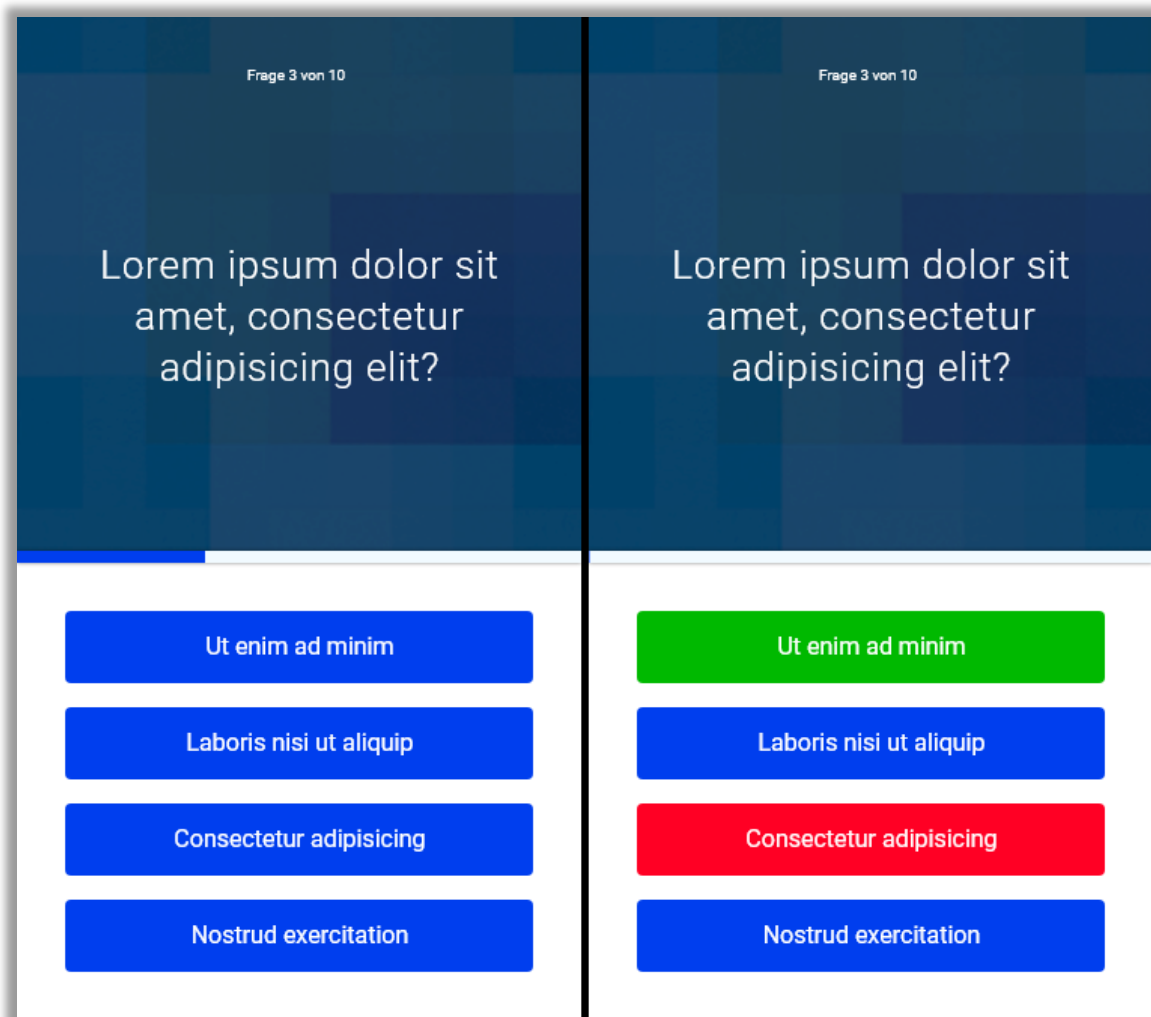


Abbildung 8: Entwurf der Quiz-Seite. Links: Eine Quiz-Frage. Rechts: Die Auflösung der Quiz-Frage.

#### 4.3.4 Die Profil-Seite

Jeder Nutzer (und jede Nutzerin) hat ein eigenes Profil. Auf der Profilseite sehen Nutzende ihre Punkte und Statistiken zu ihrem Fortschritt im Spiel. Es gibt eine Zeitleiste, in der die Nutzenden nachsehen können, für welche Aktion sie wann und wie viele Punkte erhalten haben.

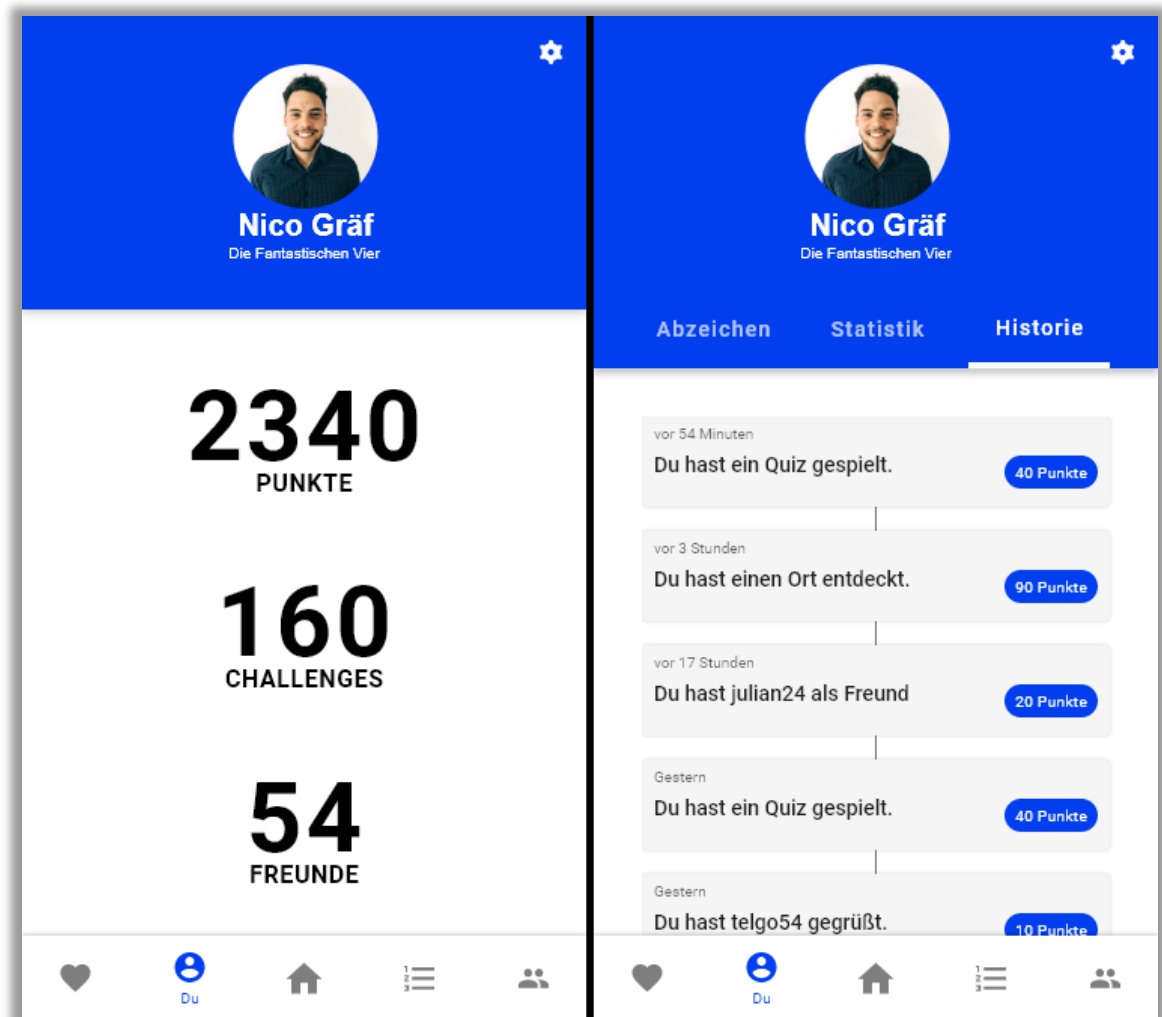


Abbildung 9: Entwurf der Profilseite. Statistiken (links) und Zeitleiste zur Aktivität des Nutzers (der Nutzerin) (rechts).

### 4.3.5 Die Freunde-Seite

Die Freunde-Seite zeigt alle befreundeten Nutzende und Freundschaftsanfragen an. Außerdem können von hier aus neue Freundschaftsanfragen versendet werden. Ein Klick auf einen blauen Button – ein bei Material Design sogenannter „Floating Action Button“ – am unteren rechten Bildschirmrand löst die Funktion der Freundschaftsanfrage aus.

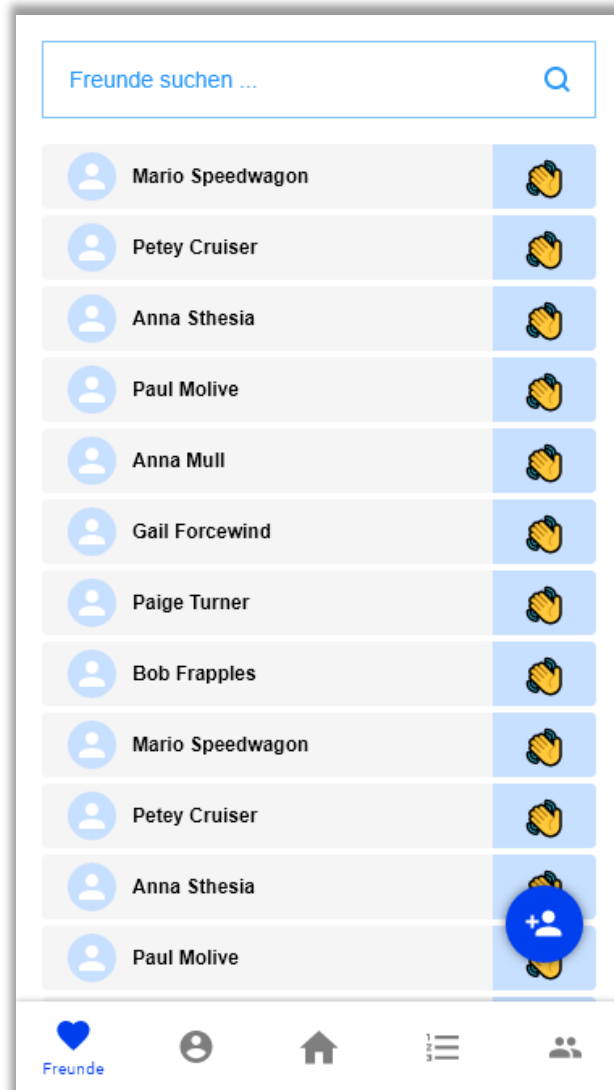


Abbildung 10: Entwurf der Freunde-Seite

### 4.3.6 Die Ranglisten-Seite

Wie schon im Konzept beschrieben, gibt es eine allgemeine Rangliste und eine Rangliste nur für die Freunde jedes Nutzers und jeder Nutzerin. Auf der Ranglisten-Seite sind diese beiden Ranglisten zu finden. Im Entwurf wird die Position der Nutzenden allseits präsent im unteren Bildschirmbereich angezeigt. Eine Tab-Navigation steuert die Anzeige der jeweiligen Rangliste.

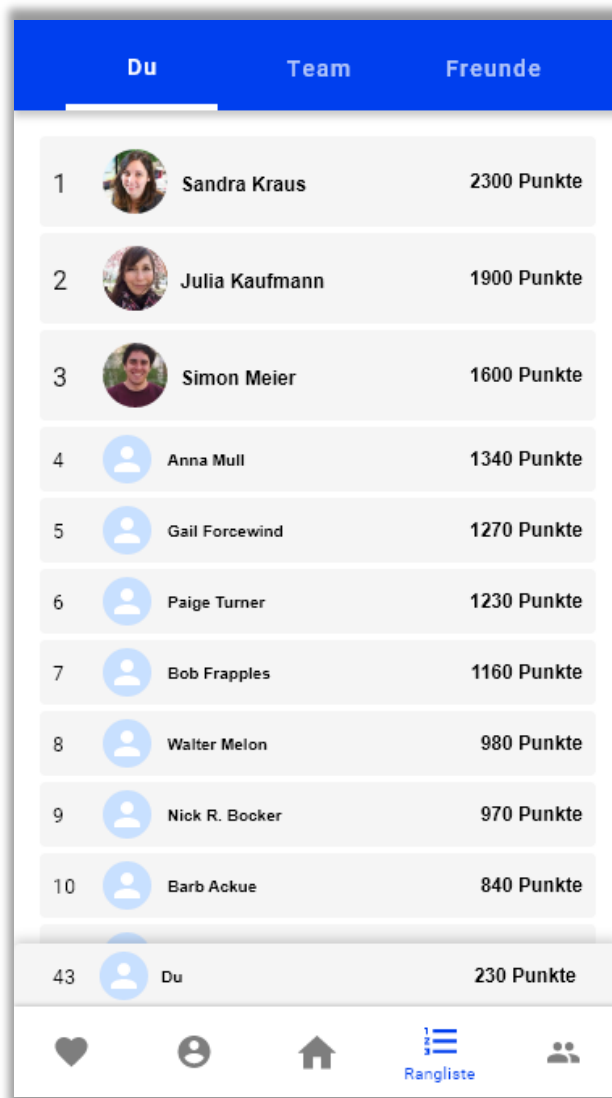


Abbildung 11: Entwurf der Ranglisten-Seite



# 5 Entwicklung der App

Die im vorangegangenen Kapitel entworfene App wurde als Web-App implementiert. Web-Apps haben den Vorteil, dass sie betriebssystemunabhängig auf allen Smartphones und Tablets und ohne vorherige Installation genutzt werden können. Eine Web-App muss aus Entwicklersicht nur einmal implementiert werden, um auf allen Geräten mit einem Browser ausgeführt werden zu können – abgesehen von wenigen browserspezifischen Details. Web-Apps lassen sich von jedem Smartphone aus leicht über den Browser aufrufen und ausführen. Im Gegensatz zu nativen Smartphone-Anwendungen lässt sich so die Hürde für neue Nutzer und Nutzerinnen senken. Außerdem wurden bei der Entwicklung der Web-App die Anforderungen von sogenannten „Progressive Web-Apps“ berücksichtigt. Progressive Web-Apps können von Nutzenden auch installiert und danach – mit gewissen Einschränkungen – gleichwertig zu einer nativen App genutzt werden.

Zusätzlich zur eigentlichen App wurde eine Verwaltungsoberfläche, das sogenannte „Admin-Panel“ entwickelt. In dieser Verwaltungsoberfläche können berechtigte Personen die Inhalte der App verwalten. Die Verwaltungsoberfläche wurde ebenfalls als Web-App entwickelt.

App und Admin-Panel basieren auf der JavaScript-Bibliothek für Benutzeroberflächen React. Beide sind von ihrer Architektur her Single Page Web-Apps. Ein modernes und Smartphone-orientiertes Design konnte durch den Einsatz des UI-Frameworks Material-UI umgesetzt werden. Auf eine iOS-spezifische Benutzeroberfläche wurde vorerst verzichtet. Serverseitig wird die App-Entwicklungs-Plattform „Firebase“ der Firma „Google“ eingesetzt. Sowohl im Frontend (Web-Apps) als auch im Backend (Firebase) wurde während der Entwicklung fast ausschließlich die Programmiersprache TypeScript verwendet.

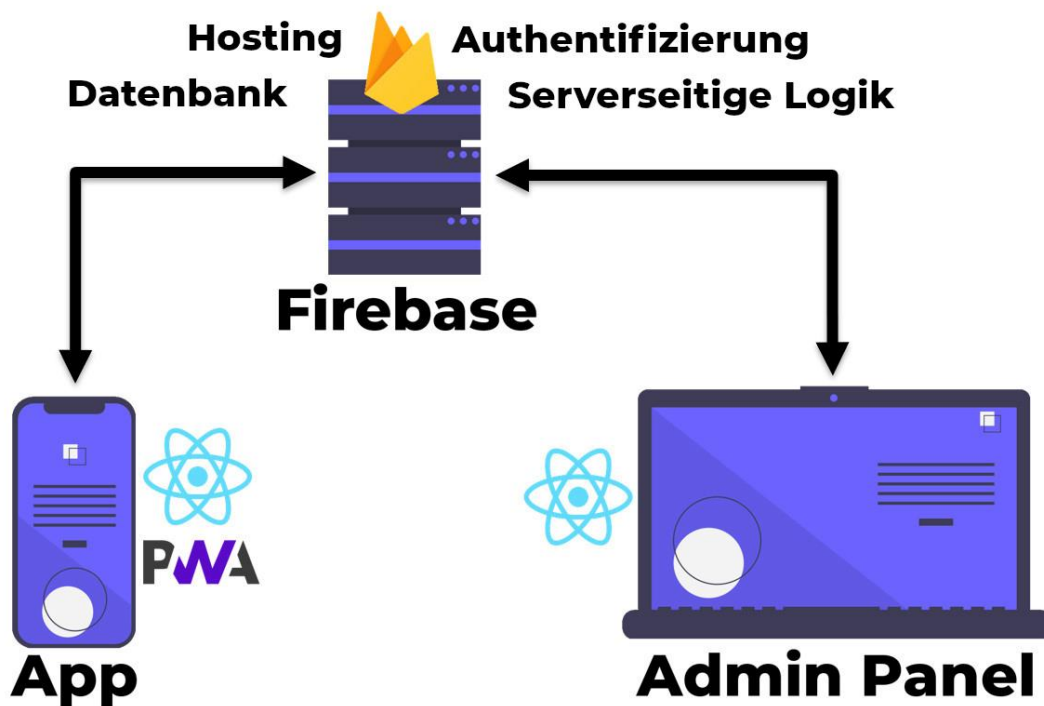


Abbildung 12: Genereller Aufbau des entwickelten Systems. Die schwarzen Pfeile zeigen die Kommunikation zwischen Firebase und der App bzw. dem Admin-Panel an.

## 5.1 Grundlagen

### 5.1.1 Web-Apps

Moderne Webseiten sind nicht mehr nur statische Seiten, sondern werden durch den Einsatz von JavaScript und Frontend-Frameworks dynamisch und interaktiv. Web-Apps sind interaktive Webseiten, die einer nativen Desktop- oder Smartphone-Anwendung ähnlich sind, jedoch vollständig im Browser laufen. Im Gegensatz zu statischen Webseiten werden bei Web-Apps Teile der Anwendungslogik clientseitig, d.h. im Browser der Nutzenden ausgeführt. [45] Daten können im Hintergrund an den Server geschickt und vom Server angefragt werden, während clientseitig mit der Web-App interagiert wird. Dabei können auch Teile der Webseite (bzw. Web-App) aktualisiert werden, ohne dass die Seite neu geladen werden muss. Web-Apps können auch über Browserschnittstellen auf die Kamera des Smartphones zugreifen, den GPS-Standort abrufen oder Berechnungen auf der Grafikkarte ausführen. [44] Im besten Fall nehmen Nutzende keinen Unterschied zu einer nativen App wahr.

### 5.1.2 Single Page Web-Apps

Eine klassische Web-App verteilt sich auf mehrere, untereinander verlinkte HTML-Dokumente (Seiten). Single Page Web-App bezeichnet eine Architektur von Web-Apps, bei der die Web-App aus nur einem HTML-Dokument besteht. Sämtliche Inhalte werden bei Single Page Web-Apps dynamisch nachgeladen oder erzeugt. Seiten- und Ansichtswechsel können durch clientseitiges Routing innerhalb der Web-App gesteuert werden, ohne dass eine Anfrage an den Webserver gesendet (und abgewartet) werden muss. [16] Dabei wird die Präsentationsschicht der Anwendung vollständig beim Client umgesetzt. Ebenfalls werden Teile der Logikschicht vom Webserver in die Web-App verschoben. Dadurch kann die Web-App schneller und flüssiger auf Nutzereingaben reagieren. Weil keine Seitenwechsel stattfinden, kann auch per WebSocket eine bidirektionale Verbindung mit einem Webservice aufgebaut und dauerhaft gehalten werden. Durch gezieltes Caching im Browser kann außerdem eine Offline-Nutzung der Web-App möglich gemacht werden. [16] Der Webserver, welcher traditionell in jeden Prozess der Anwendung involviert war, spielt bei Single Page Web-Apps eine untergeordnete Rolle. Bei einer zusätzlichen Abspaltung der Datenschicht und der Logikschicht (Geschäftslogikschicht) in Webservices dient der Webserver lediglich zur einmaligen Auslieferung der Web-App an den Client. [43]

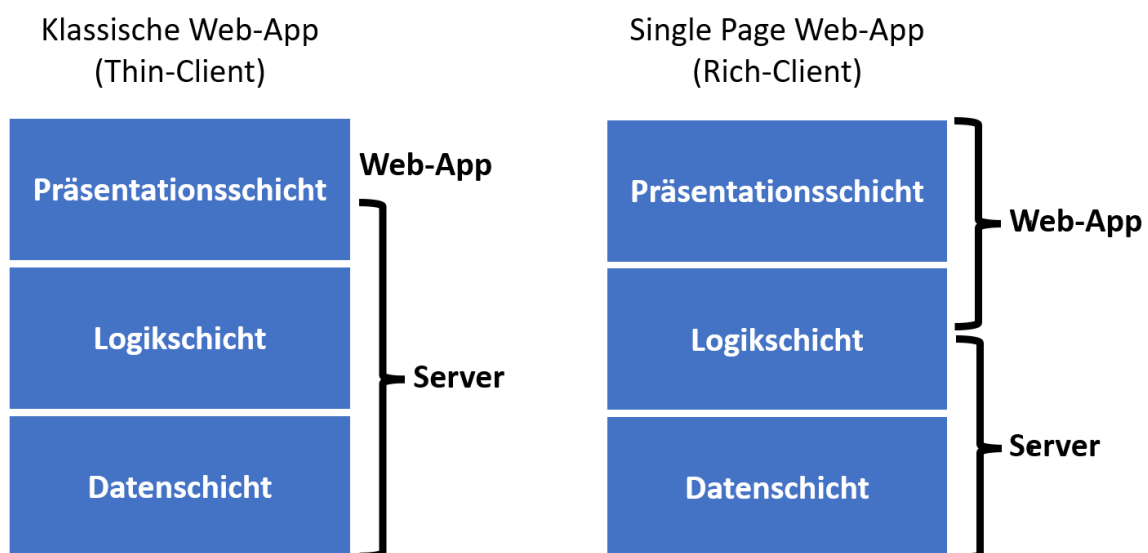


Abbildung 13: Vergleich der Drei-Schichten-Architektur von klassischen Web-Apps und Single Page Web-Apps.

Der Ablauf einer Single Page Web-App kann beispielsweise wie folgt aussehen:

1. Ein Nutzer (eine Nutzerin) ruft im Browser die Webadresse der Web-App auf.
2. Der Webserver liefert die Web-App als HTML-Dokument mit CSS und JavaScript aus.
3. Die Web-App erscheint im Browser der benutzenden Person und fordert Daten von einem Webservice an. Zum Beispiel über eine HTTP-Anfrage an eine entsprechende REST-API.
4. Der Webservice kommuniziert mit einer Datenbank und sendet die angeforderten Daten zurück an die Web-App.
5. Die Web-App präsentiert die Daten.
6. Der Nutzer (die Nutzerin) interagiert mit der Web-App.
7. Die Web-App verarbeitet die Aktionen der benutzenden Person lokal oder
8. kommuniziert gegebenenfalls mit einem (weiteren) Webservice.

In diesem Beispiel liegt die Präsentationsschicht vollständig in der Web-App. Die Datenbank bildet die Datenschicht. Und die Logikschicht ist verteilt auf die Web-Services und die Web-App. Abbildung 14 soll diesen Ablauf visualisieren.

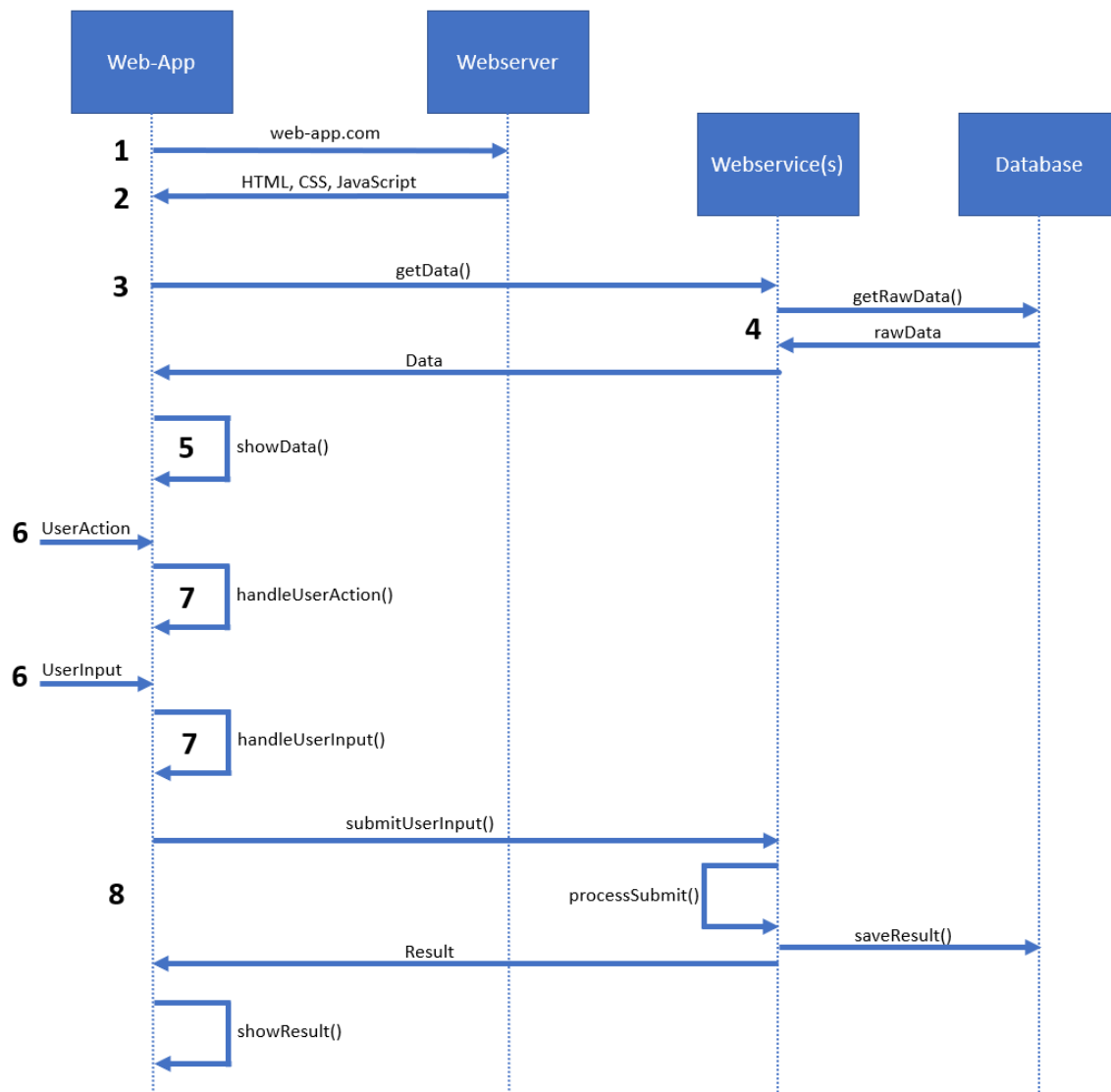


Abbildung 14: Beispielhafter Ablauf einer Single Page Web-App. Die Nummerierung entspricht der Beschreibung oben.

### 5.1.3 Progressive Web-Apps

Progressive Web-App ist eine Bezeichnung für Web-Apps, die bestimmte Anforderungen erfüllen, damit sie sich für die Nutzenden wie native Anwendungen verwenden lassen. Insbesondere auf dem Smartphone und nach der Installation kommen Progressive Web-Apps den nativen Smartphone-Anwendungen (native Apps) nahezu gleich. Sie bieten Offline-Funktionalität, können Benachrichtigungen auf dem Smartphone anzeigen oder im Hintergrund Berechnungen durchführen, selbst wenn der Browser beziehungsweise die installierte Web-App geschlossen ist. [35] Im Vergleich mit nativen Apps bieten Progressive Web-Apps den Entwickelnden und den Nutzenden allerdings weniger Leistung, Geschwindigkeit und im Allgemeinen weniger Funktionen: Dieser Nachteil

entsteht dadurch, dass Progressive Web-Apps nur über die Browserschnittstellen auf Systemressourcen und Smartphone-Funktionen zugreifen können. Diese Schnittstellen sind jedoch begrenzt und für einige Funktionen wie beispielsweise Fingersensoren gegenwärtig noch nicht entwickelt. Deshalb sind Progressive Web-Apps vor allem für einfache Anwendungen geeignet. Bei komplexen, rechen- oder ressourcenintensiven Anwendungen wie 3D-Spiele zum Beispiel sind native Apps daher besser geeignet.

Der Browserhersteller „Google“ beschreibt die Hauptmerkmale von Progressive Web Apps als „reliable“, „fast“ und „engaging“. [35] Die Web-App soll also zuverlässig (englisch „reliable“) laden und in ihrer Grundfunktionalität selbst bei schlechter oder nicht bestehender Internetverbindung genutzt werden können. Dabei soll die Web-App schnell (englisch „fast“) laden und unmittelbar auf Nutzerinteraktionen reagieren. Außerdem sollen laut Google Nutzende gerne, häufig und immer wieder mit der Anwendung interagieren. Progressive Web-Apps müssen sicher sein. Das bedeutet in erster Linie, dass sie nur über das verschlüsselte Protokoll „HTTPS“ (kurz für „Hypertext Transfer Protocol Secure“) ausgeliefert werden dürfen.

#### **5.1.4 Web-App Manifest**

Um einer nativen App vom Aussehen und Verhalten gleich zu kommen, benötigt eine Web-App ein Web-App Manifest. Das Web-App Manifest ist eine Datei, in der Angaben zum Namen der App, dem App-Logo, dem Farbschema und weiteren Parametern und Einstellungen zur Web-App gemacht werden können. [29] Einige dieser Angaben werden erfordert, andere sind wiederum optional. Dieses Manifest wird als einfache JSON-Datei (JSON steht für „JavaScript Object Notation“) in den Metadaten der HTML-Seite verlinkt. Der Browser und das Betriebssystem (Android, iOS, Windows) verwenden das Web-App Manifest hauptsächlich, um das Aussehen des Browsers an die Web-App anzupassen und bei einer Installation den Namen und das Logo der App auf dem Startbildschirm des Smartphones anzuzeigen.

```

{
  "short_name": "Freiburg Challenge",
  "name": "Freiburg Challenge",
  "description": "Orte entdecken und Freiburg kennenlernen.",
  "icons": [
    {
      "src": "favicon.ico",
      "sizes": "48x48",
      "type": "image/x-icon"
    },
    {
      "src": "logo192.png",
      "type": "image/png",
      "sizes": "192x192"
    },
    {
      "src": "logo512.png",
      "type": "image/png",
      "sizes": "512x512"
    }
  ],
  "start_url": ".",
  "display": "standalone",
  "orientation": "portrait",
  "theme_color": "#3f51b5",
  "background_color": "#ffffff",
  "permissions": ["geolocation"]
}

```

Abbildung 15: Web-App Manifest der entwickelten App

### 5.1.5 TypeScript

TypeScript ist eine auf JavaScript basierende Programmiersprache. Sie wurde von der Firma Microsoft entwickelt und erweitert JavaScript um einige nützliche Eigenschaften, die teilweise von anderen Programmiersprachen beeinflusst sind.<sup>4</sup> Da TypeScript eine Obermenge zu JavaScript bildet, ist jeder JavaScript-Code auch ein gültiger TypeScript-Code. Der TypeScript-Compiler übersetzt TypeScript zu JavaScript. Das Hauptmerkmal von TypeScript ist die (optionale) statische Typisierung, von der auch der Name stammt. Die statische Typisierung erleichtert die Entwicklung (größerer) Web-Apps vor allem, indem viele Programmierfehler (Bugs) schon während der Kompilierung erkannt werden, die bei JavaScript (dynamischen Typisierung) erst bei der Ausführung auftreten würden. Entwicklungsumgebungen wie „Visual Studio Code“ der Firma Microsoft bieten Tools, die Fehler schon während des Tippens erkennen und den Entwickelnden aufzeigen.<sup>5</sup>

<sup>4</sup> <https://www.typescriptlang.org/>

<sup>5</sup> <https://code.visualstudio.com/>

```
const myArray: string[] = ['hello world']  
  
myArray.push(42)
```

Argument of type '42' is not assignable to parameter of type 'string'. ts(2345)  
Peek Problem No quick fixes available

Abbildung 16: TypeScript Error in Visual Studio Code

## 5.1.6 React

React (auch ReactJS genannt) ist eine von Facebook maßgeblich entwickelte Open-Source JavaScript-Bibliothek, welche die Entwicklung von interaktiven, modularen Benutzeroberflächen (User Interfaces) erleichtert. Diese Benutzeroberflächen bestehen aus hierarchisch aufgebauten Komponenten. React baut dann zur Laufzeit die Web-App (oder Webseite) aus den einzelnen Komponenten zusammen. Wenn sich die Daten einer Komponente ändern, sorgt React für eine effiziente Aktualisierung der Benutzeroberfläche. Die große Entwicklercommunity von React bietet viele zusätzliche Bibliotheken und Frameworks, die React für die Entwicklung von Web-Apps erweitern. Die mit React erzeugten Komponenten können aber auch einzeln, miteinander kombiniert oder hierarchisch verschachtelt in bestehende Webseiten oder Web-Apps integriert werden. React erlaubt auch die Programmierung in TypeScript. Zur besseren Lesbarkeit wird im Folgenden bei Erklärungen nur JavaScript erwähnt. Gemeint sind jedoch beide Programmiersprachen. Im Folgenden bezeichnet React-App eine mit React entwickelte Web-App.

### 5.1.6.1 React Komponenten

React erlaubt es wiederverwendbare User-Interface-Komponenten zu bauen. Diese Komponenten sind standardmäßig unabhängig voneinander und in sich geschlossen. Komponenten werden als JavaScript-Klasse oder JavaScript-Funktion definiert, können Argumente (props genannt, kurz für englisch „properties“) entgegennehmen und geben selbst HTML-Elemente oder andere React Komponenten zurück. Komponenten können auch Zustände speichern (englisch „stateful components“) und je nach Argumenten und Zustand ihr Aussehen (d.h. ihre HTML-Ausgabe) und ihre Funktionalität bestimmen (englisch „conditional rendering“). React Komponenten können beliebig ineinander



verschachtelt und mehrfach wiederverwendet werden. Das Entwickeln von React Komponenten kann durch die JavaScript-Syntaxerweiterung „JSX“ (kurz für „JavaScript Syntax Extension“) deutlich vereinfacht werden. Durch den Einsatz von JSX können Komponenten wie selbst definierte HTML-Tags verwendet werden (siehe Abbildung 18). Diesen selbst definierte HTML-Tags können die Argumente (props) dann in Form von HTML-Attributen übergeben werden. JSX wird im Folgenden Kapitel näher erklärt. In der Web-App dieser Arbeit wurden alle React Komponenten als TypeScript-Funktion mit JSX implementiert.

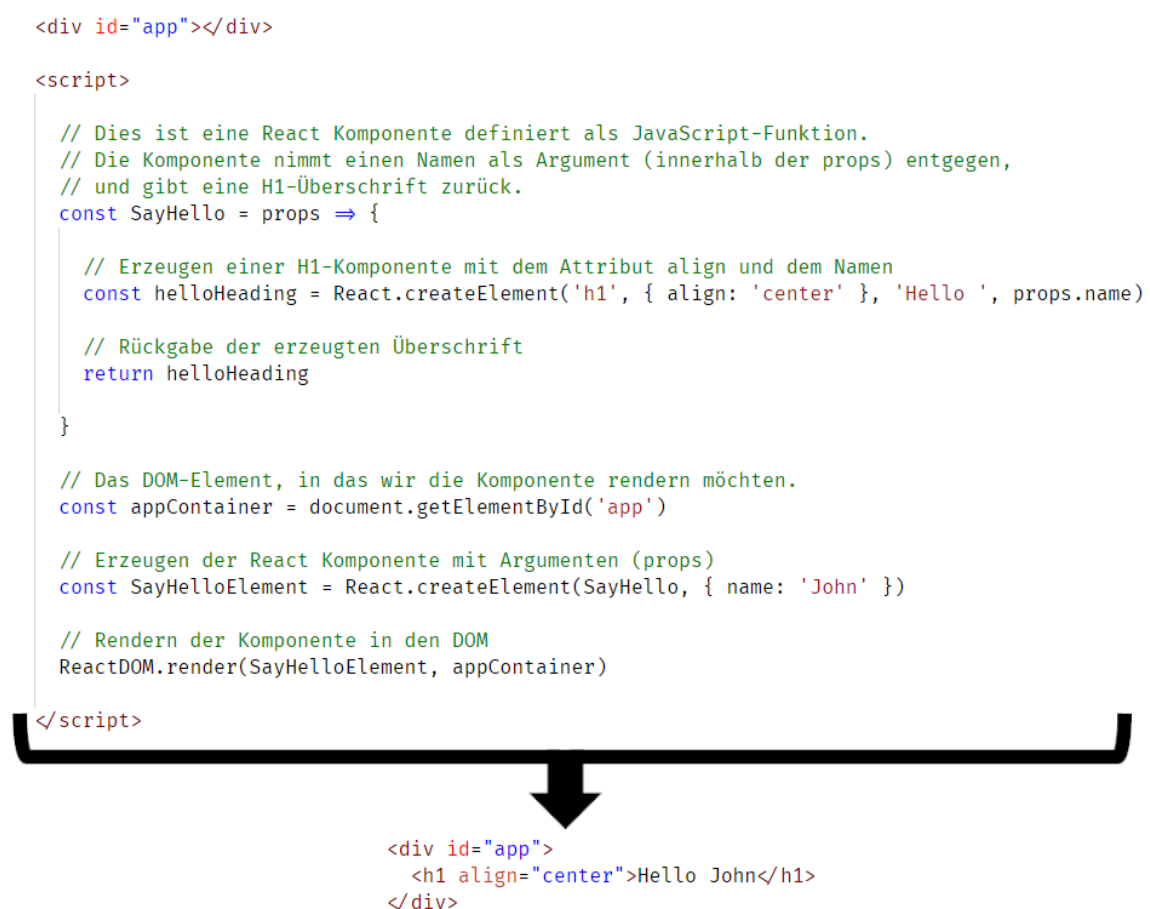


Abbildung 17: Beispiel einer React Komponente ohne JSX.

### 5.1.6.2 JSX

JSX steht für JavaScript XML und ist eine and XML angelehnte Syntax-Erweiterung für JavaScript. Entwickelnde können JSX in React Komponenten als eine Art Template-Sprache einsetzen, die sich wie HTML schreiben lässt und annähernd die volle JavaScript-Funktionalität bietet. JavaScript-Ausdrücke können in JSX innerhalb von

geschweiften Klammern verwendet werden. Der JavaScript-Übersetzer „Babel“<sup>6</sup> erzeugt aus dem JSX-Code React-Elemente die vom React Virtual DOM dann zu „echten“ DOM-Elementen gerendert werden (siehe nächstes Kapitel). Beispielsweise wird folgender JSX-Code `<h1>1 + 1 = {1 + 1}</h1>` zu `<h1>1 + 1 = 2</h1>` gerendert. Das Beispiel aus Abbildung 17 kann mithilfe von JSX deutlich vereinfacht werden. Die gleiche Komponente mit JSX wird in Abbildung 18 dargestellt.

```
<script>

  const appContainer = document.getElementById('app')

  // React Komponente mit JSX
  const SayHello = props => <h1 align="center">Hello {props.name}</h1>

  // Rendern der Komponente als selbst definierter HTML-Tag
  ReactDOM.render(<SayHello name="John" />, appContainer)

</script>
```

Abbildung 18: Beispiel einer React Komponente mit JSX.

### 5.1.6.3 Virtual DOM

Das „Document Object Model“ (kurz DOM) ist eine technische Schnittstelle zu Webseiten. Wenn eine Webseite geladen wird, erzeugen Browser das DOM aus dem HTML-Dokument als eine Baumstruktur von Objekten, die HTML-Elemente sowie deren Attribute und Inhalte repräsentieren. Als Programmierschnittstelle bietet das DOM beispielsweise JavaScript-Funktionen, die es ermöglichen HTML-Elemente zu verändern, hinzuzufügen oder zu löschen. Mit React werden Webseiten und Web-Apps entwickelt, bei denen – für gewöhnlich – viele Veränderungen am DOM vorgenommen werden. Die Entwickler/-innen von React haben deshalb hier eine Zwischenebene eingefügt: der Virtual DOM. Der Virtual DOM sorgt für eine effiziente und schnelle Aktualisierung des „echten“ DOM und damit der Benutzeroberfläche einer Web-App. In React werden Komponenten neu erzeugt bzw. neu gerendert, wenn sich die Daten, ihr Zustand oder eine Unter-Komponente ändern. Statt jedes Mal den gesamten DOM neu zu erzeugen, vergleicht der Virtual DOM die neue Version der Komponente mit der aktuellen DOM-Komponente und aktualisiert im „echten“ DOM nur die notwendigen Stellen. Aktualisierungen des DOMs sind dadurch in React schnell und effizient.

---

<sup>6</sup> <https://babeljs.io/>

#### **5.1.6.4 Hooks**

In früheren Versionen von React gab es Features, die nur bei Komponenten möglich waren, welche als JavaScript-Klasse implementiert wurden. Die in Version 16.8 eingeführten Hooks sind JavaScript-Funktionen, die diese und weitere Features für funktionale Komponenten ermöglichen. Die meistgenutzten Hooks sind der useState-Hook und der useEffect-Hook. Wenn sich die Daten einer Komponente ändern, erzeugt React eine neue Instanz dieser Komponente. Sämtliche Variablen und vorherigen Daten werden dabei „gelöscht“. Mit dem useState-Hook kann ein persistenter Zustand in einer Komponente gespeichert werden. Der useEffect-Hook ermöglicht es eine bestimmte Logik (Code) jedes Mal auszuführen, wenn sich eine bestimmte Variable (oder mehrere) ändert. Entwickler/-innen können auch eigene Hooks (Custom Hooks) schreiben. Custom Hooks können andere Hooks beinhalten. Oft werden der useState-Hook und der useEffect-Hook zusammen in einem Custom Hook eingesetzt. Dadurch kann Logik in mehreren Komponenten wiederverwendet und – zum Beispiel – vom Template einer Komponente abstrahiert werden. Custom Hooks machen den Programmiercode oft verständlicher und übersichtlicher.

#### **5.1.6.5 React Router**

Bei clientseitigem Routing werden Seitenanfragen von der Web-App selbst verarbeitet, statt vom Webserver. React Router ist eine JavaScript-Bibliothek, welche React um Komponenten erweitert, die clientseitiges Routing vereinfachen. Die wichtigsten drei Komponenten heißen „BrowserRouter“, „Route“ und „Link“. BrowserRouter sorgt dafür, dass reguläre URL-Pfade (z.B. beispiel.de/user/123) für das clientseitige Routing genutzt werden können. Route definiert einen solchen URL-Pfad (z.B. /user). Innerhalb einer Route können dann die gewünschten React Komponenten dieses Pfades gerendert werden. Die Komponente Link stellt einen einfach HTML-Link zu einem URL-Pfad dar. Links können zur Navigation genutzt werden.

#### **5.1.6.6 Create React App**

Für die Entwicklung einer modernen und umfangreichen Web-App ist es nötig geworden mehrere Tools zu installieren und zu konfigurieren. Beispielsweise müssen TypeScript und JSX übersetzt, Tests automatisiert, benötigte Module zusammengebracht und der

erzeugte Code optimiert werden. Facebook entwickelt dafür ein auf Node.js<sup>7</sup> basierendes Tool namens „Create React App“. Mit Create React App ein neues und dem Stand der Technik entsprechendes React-Projekt aufgesetzt werden. [12] Das Programm erstellt, installiert und konfiguriert alle Tools und Dateien, die für die Entwicklung einer modernen React Web-App von Facebook empfohlen sind. Zusätzlich erfüllen alle mit Create React App erstellten Web-Apps die Kriterien für Progressive Web Apps. Standardmäßig werden das Web-App Manifest und ein sogenannter „Service Worker“ erzeugt. Durch einen Service Worker können alle statischen Ressourcen lokal gespeichert und die App damit auch ohne Internetverbindung – zumindest teilweise – genutzt werden. Die App dieser Bachelorarbeit, sowie das Admin-Panel, wurde ebenfalls mit Create React App erzeugt.

### **5.1.7 Firebase**

Firebase ist eine cloudbasierte App-Entwicklungs-Plattform für Web-Apps und native iOS-/Android-Apps. Die Plattform bietet Dienste wie Datenbanken, Hosting, Nutzerverwaltung, serverseitige Logik und mehr in der Cloud. Dazu bietet Firebase ein JavaScript SDK für die einfache Integration der Firebase-Dienste in Web-Apps. Um die Firebase Dienste nutzen zu können, muss ein Projekt angelegt werden. Jedes Projekt ist unabhängig und alle Firebase Dienste können innerhalb des Projektes einzeln ein- und ausgeschaltet werden. Beim Erstellen eines neuen Projektes müssen keine Server aufgesetzt oder konfiguriert werden (englisch „serverless computing“). Firebase passt die notwendigen Ressourcen automatisch an.

Im Folgenden werden die Firebase-Dienste vorgestellt, die in der entwickelten Web-App genutzt werden.

#### **5.1.7.1 Cloud Firestore**

Cloud Firestore ist eine hierarchisch organisierte, dokumentenbasierte NoSQL-Datenbank. Dokumente ähneln stark der JavaScript Object Notation (JSON). Diese Dokumente gehören jeweils einer sogenannten „Collection“ an. Dokumente können Daten beliebiger und beliebig verschachtelte Struktur beinhalten. Weiterhin kann jedes Dokument auch untergeordnete Collections haben (sogenannte „Subcollections“). Daten können so hierarchisch strukturiert werden. [17, 18]

---

<sup>7</sup> <https://nodejs.org/>

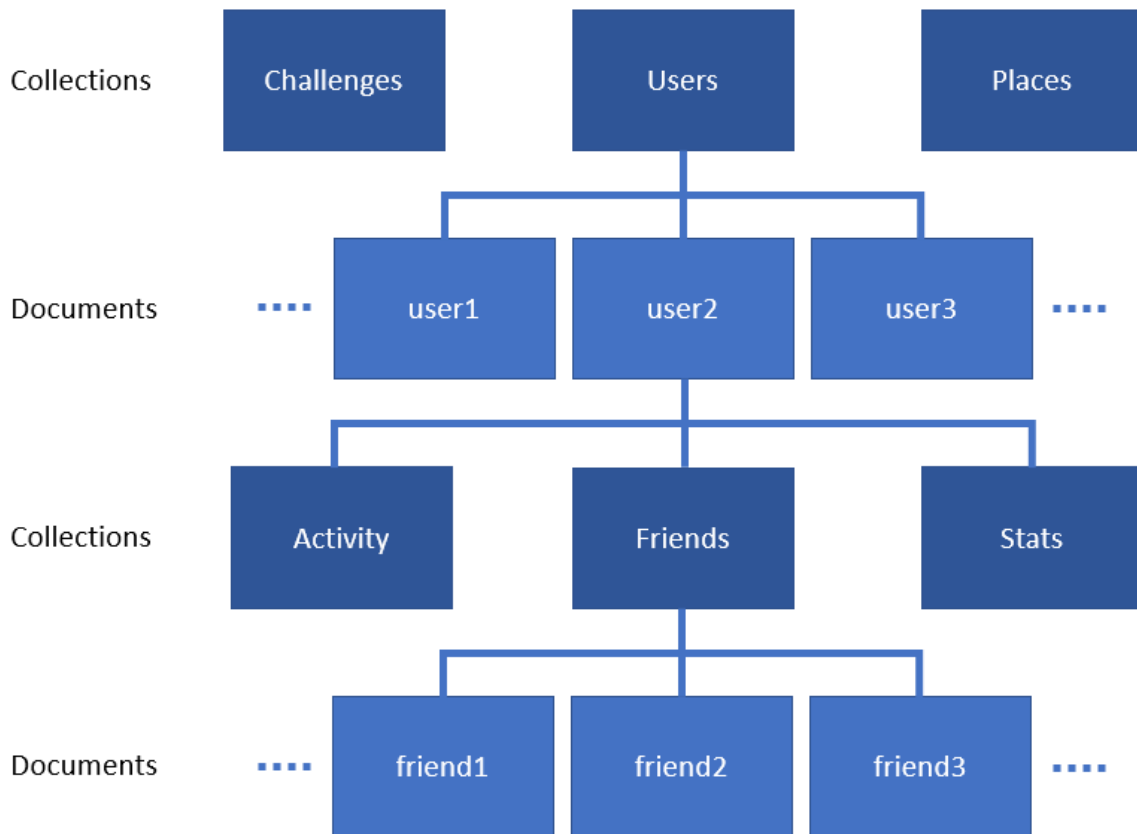


Abbildung 19: Beispiel einer hierarchischen Datenstruktur in Cloud Firestore.

Cloud Firestore erstellt für jedes Feld in einem Dokument automatisch einen Index. Felder in Dokumenten sind vergleichbar mit Key-Value-Paaren in JSON. Dokumente einer Collection können bei einer Datenbankabfrage (Query) nach jedem Feld gefiltert und sortiert werden. Das folgende Beispiel zeigt eine Datenbankabfrage mit dem Firebase JavaScript SDK. Gefragt sind die ersten 10 Nutzer/-innen, welche am 10. Mai 2020 beigetreten sind, absteigend sortiert nach ihren Punkten.

```

firestore
  .collection('users')
  .where('joinedAt', '=', '2020-03-10')
  .orderBy('points', 'desc')
  .limit(10)
  .get()
  .then(({ documents }) => showUsers(documents))

```

Abbildung 20: Beispiel einer Datenbank-Abfrage mit dem Firebase JavaScript SDK.

Die Sicherheit der Daten wird durch die sogenannten „Cloud Firestore Security Rules“ gewährleistet. Die Cloud Firestore Security Rules sind Regeln, welche festlegen unter welchen Bedingungen Daten aus einem bestimmten Dokument gelesen bzw. in ein Dokument geschrieben werden dürfen. Diese Regeln können von den Entwickelnden individuell angepasst werden.

```

rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {
    match /{document=**} {
      allow read, write: if request.auth.uid ≠ null;
    }
  }
}

```

Abbildung 21: Beispiel für Cloud Firestore Security Rules. Authentifizierte Nutzer können alle Dokumente in der Datenbank lesen.

### 5.1.7.2 Authentication

Firebase Authentication ist ein Authentifizierungs-System für die Nutzerverwaltung und die Zugangskontrolle in Firebase Projekten. Neue Nutzer/-innen können sich je nach Einstellung per E-Mail und Passwort, anonym (Anonymous Authentication) oder über Drittanbieter wie Google und Facebook über den OAuth-Standard registrieren und anmelden. Nutzende bekommen bei der Registrierung eine einzigartige ID zugewiesen. Mit dieser ID können Nutzer/-innen in der Datenbank und bei API-Anfragen identifiziert werden. [19]

### **5.1.7.3 Cloud Functions**

Bei Firebase wird die gesamte serverseitige Logik, d.h. der serverseitige Code, in einzelne Funktionen unterteilt und bei Bedarf voneinander getrennt ausgeführt. Dieses Modell heißt „Functions as a Service“ und steht der klassischen, monolithischen Serveranwendung gegenüber. „Cloud Functions“ heißt der Firebase-Dienst, der dafür zuständig ist. Funktionen – auch „microservices“ genannt – können auf verschiedene Ereignisse reagieren und daraufhin ausgeführt werden. Beispielsweise auf Ereignisse in der Datenbank (wenn ein neues Dokument hinzugefügt wurde), bei der Nutzerauthentifizierung (wenn ein neuer Nutzer registriert wurde) oder wiederholt zu einem bestimmten Zeitpunkt. Häufige Anwendungsfälle sind die Verarbeitung von HTTP-Anfragen, um eigene API-Endpunkte zu entwickeln, die Verarbeitung von neuen Daten in der Datenbank oder auch Datenbankbereinigungen. Funktionen können in JavaScript oder TypeScript geschrieben werden. [20]

### **5.1.7.4 Hosting**

„Firebase Hosting“ ermöglicht das Hosten und Bereitstellen von Web-Apps und weiteren statischen Inhalten wie Bilder über ein Content Delivery Network (CDN). Eine verschlüsselte Verbindung über HTTPS, das HTTP/2-Protokoll und das Kompressionsverfahren „gzip“ sind standardmäßig integriert. Jedes Firebase Projekt erhält eine kostenlose .web.app-Domain. Eigene Domains können mit Firebase Hosting verknüpft werden. Es können auch mehrere gehostete Seiten pro Projekt existieren. [21]

### **5.1.7.5 JavaScript SDK**

Das Firebase JavaScript SDK erleichtert die Integration der Firebase Dienste in Web-Apps. Das SDK besteht aus JavaScript-Bibliotheken der einzelnen Firebase Dienste. Diese Bibliotheken bieten Funktionen zum Lesen und Schreiben von Daten von der und in die Datenbank (Cloud Firestore), zum Authentifizieren von Nutzenden, zum Kommunizieren mit serverseitiger Logik (Cloud Functions) und mehr. Das SDK bietet auch die Möglichkeit nach dem Beobachter-Muster bestimmte Änderungen in der Datenbank zu beobachten (abonnieren) und bei einer Aktualisierung der Daten entsprechende Aktionen auszuführen. Ebenfalls können Features wie das lokale Caching der Daten aktiviert werden. Durch das Caching können sich Nutzende auch offline authentifizieren und auf die lokal gespeicherten Daten zugreifen. Die Firebase JavaScript SDK abstrahiert die gesamte Kommunikation mit den Firebase Diensten und Servern.

Entwickler/-innen müssen sich keine Gedanken um Auth-Tokens, Cookies, Caching oder HTTP-Anfragen machen. Notwendige HTTP-Header und Auth-Tokens werden bei allen Anfragen automatisch mitgesendet.

## **5.2 Aufbau und Funktionsweise der App**

Die App besteht aus den sechs Seiten, die aus dem Design-Prototypen bekannt sind: Dashboard, Profil, Freunde, Rangliste, Challenge und Quiz. Seiten sind Komponenten, welche in keine übergeordnete Komponente integriert sind. Jede Seite hat ihre eigene Route, d.h. ihren eigenen URL-Pfad, eigene Logik, eigene Daten und entsprechende Sub-Komponenten, aus denen die Seite zusammengestellt wird. Eine Navigationsleiste am unteren Bildschirmrand erlaubt es den Nutzenden zwischen den Seiten zu navigieren. Eine Top-Level-Komponente integriert alle Seiten in den clientseitigen Router und implementiert dazu die Logik zur Nutzerauthentifizierung. Diese Top-Level-Komponente wird von React in die HTML-Seite eingebaut. Daten und Logik werden bis auf wenige Ausnahmen in den Seiten gehalten. Sub-Komponenten werden diese Daten nur zur Darstellung übergeben.



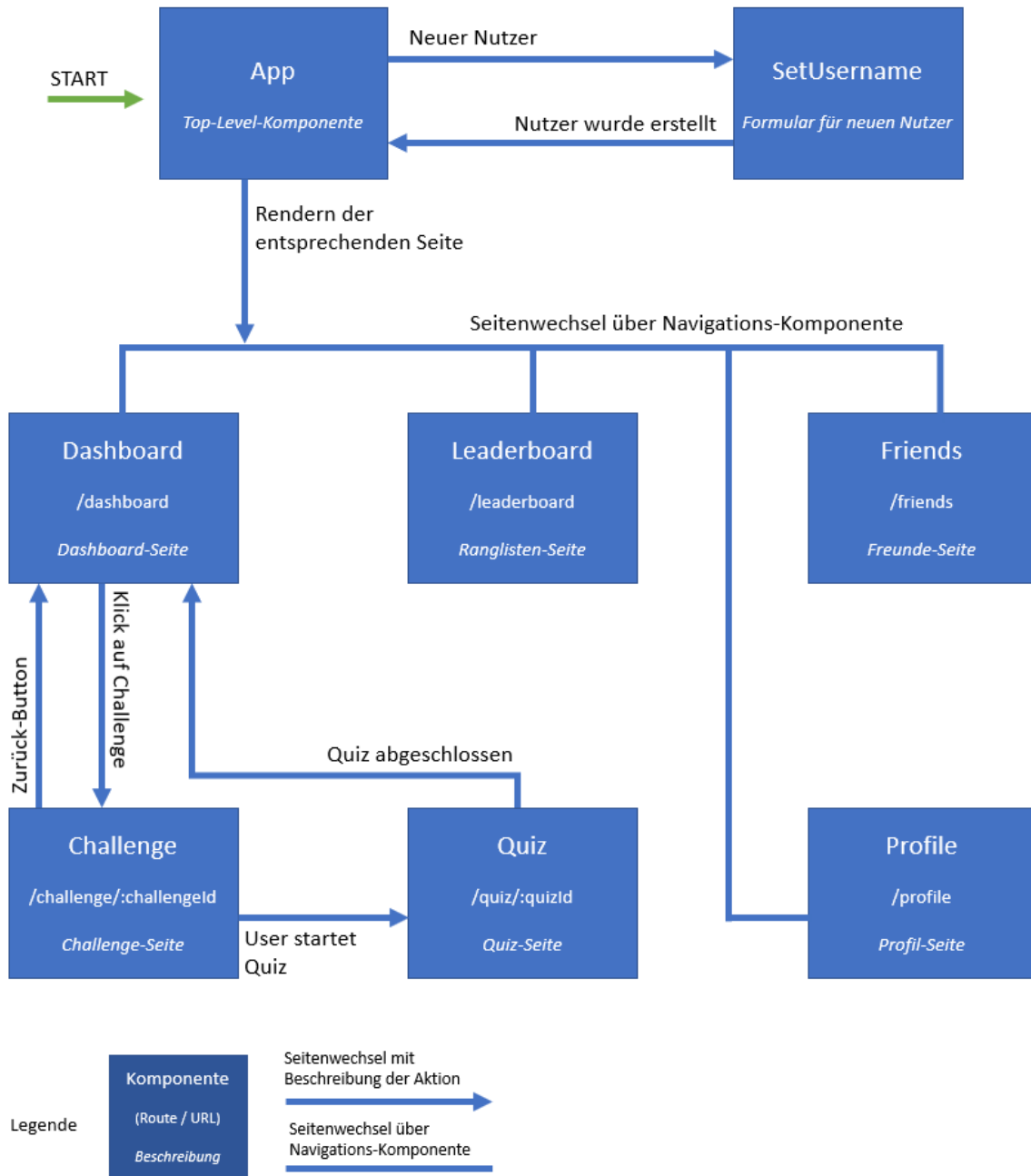


Abbildung 22: Schematischer Aufbau der App.

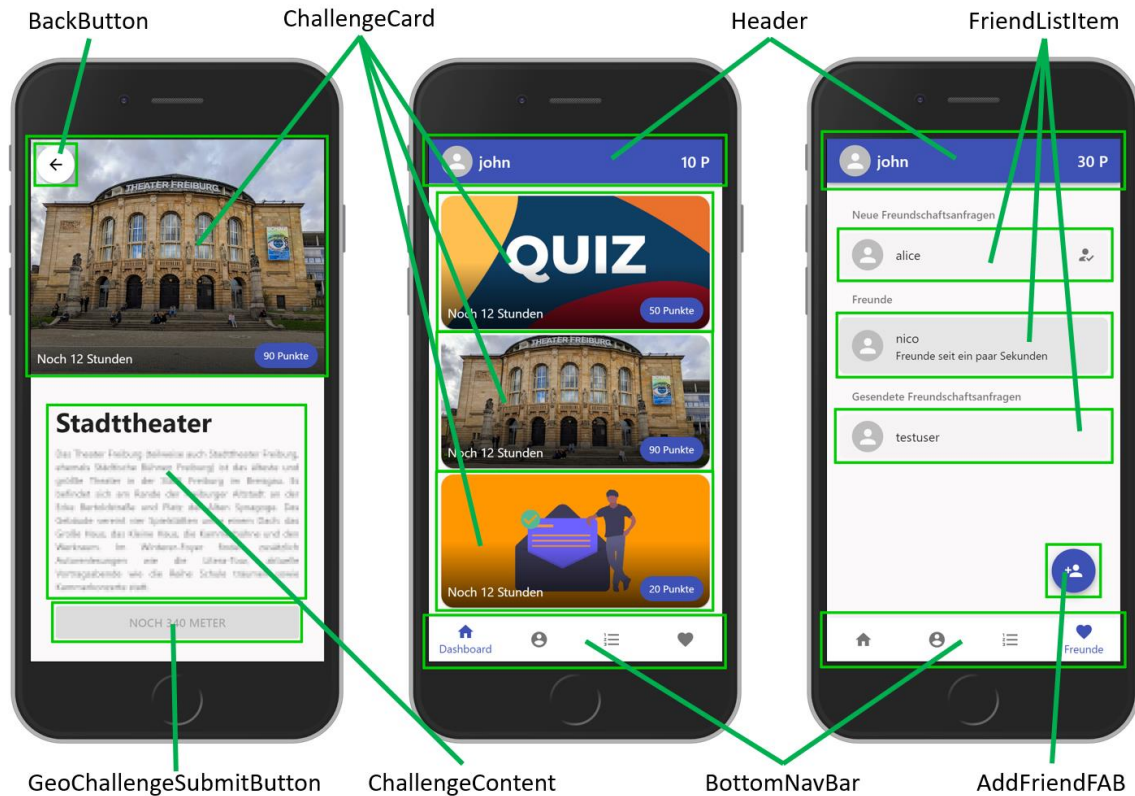


Abbildung 23: Aufteilung der Challenge-Seite (links), der Dashboard-Seite (mitte) und der Freunde-Seite (rechts) in ihre Komponenten.

Die Funktionsweise der App sieht in etwa wie folgt aus:

1. Ein Nutzer (eine Nutzerin) ruft im Browser die Webadresse der Web-App auf oder klickt auf das entsprechende Icon auf dem Startbildschirm seines Smartphones. Firebase Hosting liefert die App aus.
2. Sobald die App fertig geladen wurde, wird der Nutzer (die Nutzerin) per „Anonymous Authentication“ im Hintergrund authentifiziert. Nutzende müssen weder E-Mail noch Passwort eingeben, um sich zu authentifizieren. Wenn Nutzende zum ersten Mal die App öffnen, erzeugt Firebase einen neuen anonymen Account. Firebase Authentication liefert danach die Nutzeridentifikation (englisch „user id“) zurück.
3. Bei neuen Nutzern und Nutzerinnen erscheint ein Formular in der App, welches die Person dazu auffordert, einen Nutzernamen („Username“) zu wählen. Der gewählte Nutzername wird beim Absenden des Formulars an eine Cloud Function (createUserWithUsername) gesendet. Die Cloud Function kontrolliert, ob dieser Nutzername schon vergeben ist. Falls nicht, erzeugt die Cloud Function ein neues Dokument für den Nutzer (die Nutzerin) in der Users-Collection. Falls der

- gewählte Nutzernamen bereits existiert, wird eine Fehlermeldung an die App zurückgesendet.
4. Die App holt sich mithilfe der Nutzeridentifikation das User-Dokument des Nutzers (der Nutzerin) aus der Datenbank (Firestore) und zeigt den Nutzernamen und die Punkte in der App an.
  5. Die App lädt die aktuellen Challenges aus der Datenbank und zeigt sie im Dashboard an. Das Firebase SDK kümmert sich automatisch um die Authentifizierung bei Anfragen an Firestore.
  6. Der Nutzer (die Nutzerin) spielt eine Challenge. Nach Beenden der Challenge wird das Ergebnis in die Datenbank geschrieben. Eine Cloud Function (`handleChallengeResultSubmissions`) startet beim Schreiben eines solchen Challenge-Ergebnisses durch das entsprechende Firestore Ereignis. Die Cloud Function verarbeitet das Ergebnis, berechnet die Punkte und aktualisiert das User-Dokument. Die App wartet auf diese Änderungen und aktualisiert die Anzeige der Punkte in der App.
  7. Der Nutzer (die Nutzerin) möchte eine Freundschaftsanfrage versenden. Dafür gibt er (sie) auf der Freunde-Seite den entsprechenden Nutzernamen ein und klickt auf „versenden“. Der eingegebene Nutzernamen wird an eine Cloud Function gesendet. Diese Cloud Function (`submitFriendRequest`) erzeugt die entsprechenden Dokumente für die Freundschaftsanfrage in Firestore. Falls es keinen Nutzer und keine Nutzerin mit diesem Nutzernamen gibt oder ein anderer Fehler auftritt, wird eine entsprechende Fehlermeldung an die App zurückgesendet. Die App wartet auf die neu erstellten Dokumente in Firestore und aktualisiert daraufhin das User Interface. Dem Nutzer (der Nutzerin) wird nun die versendete Freundschaftsanfrage in der App angezeigt.

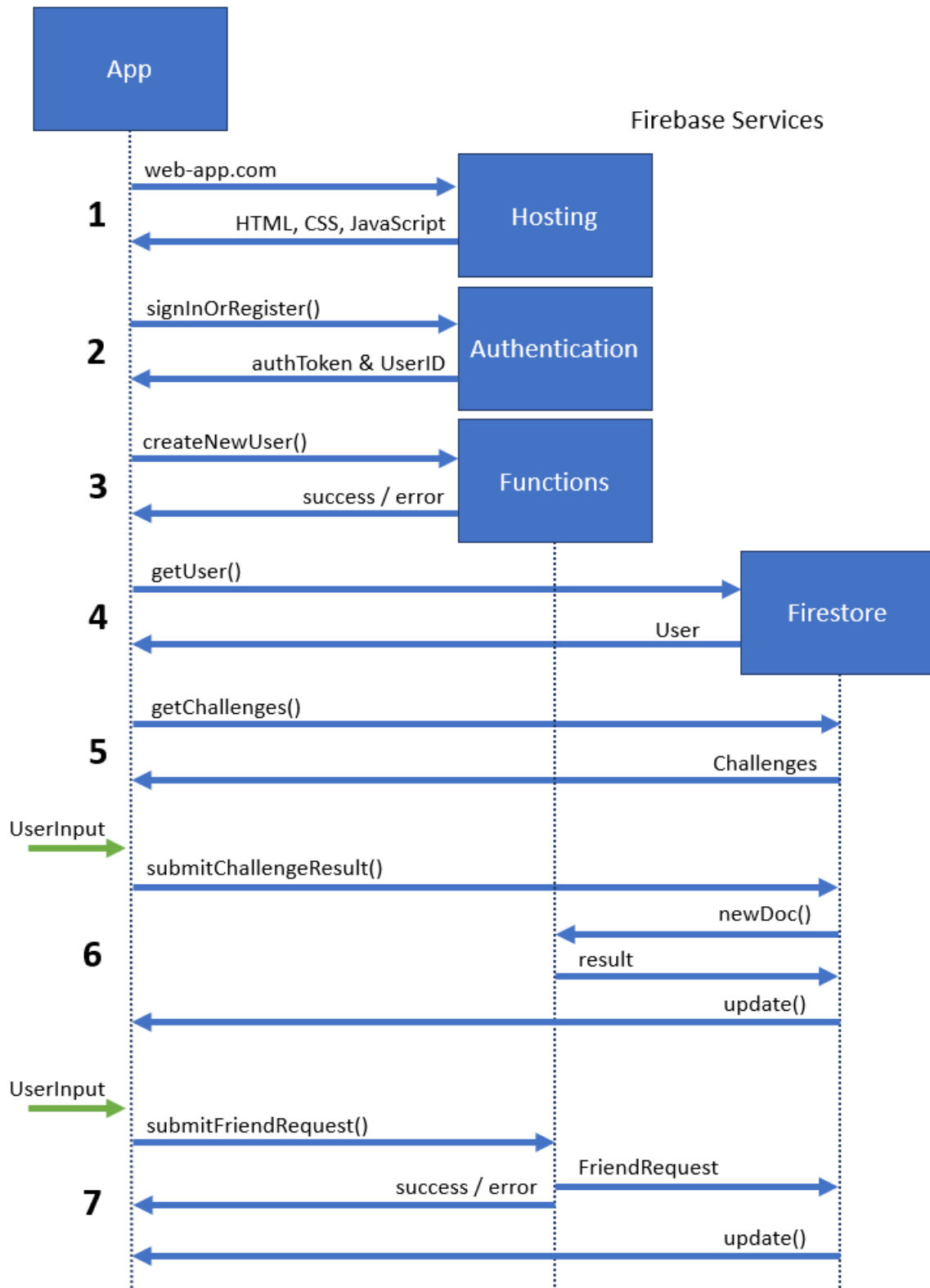


Abbildung 24: Funktionsweise der App. Die Nummerierung entspricht der Beschreibung oben.

Jegliche Kommunikation mit den Firebase Diensten läuft über das Firebase JavaScript-SDK. Eine selbst geschriebene Firebase API bildet die benötigten Funktionen der SDK

als TypeScript-Funktionen ab. Eine eigene Firebase API macht den Code verständlicher und übersichtlicher, indem die API darunterliegende Schichten für die spezifische Anwendung abstrahiert. Ebenfalls wurden Custom Hooks geschrieben, die sich um die Abfrage, Speicherung und Aktualisierung der Daten in den Seiten bzw. Komponenten kümmern. Die Custom Hooks nutzen die selbst geschriebene Firebase API.

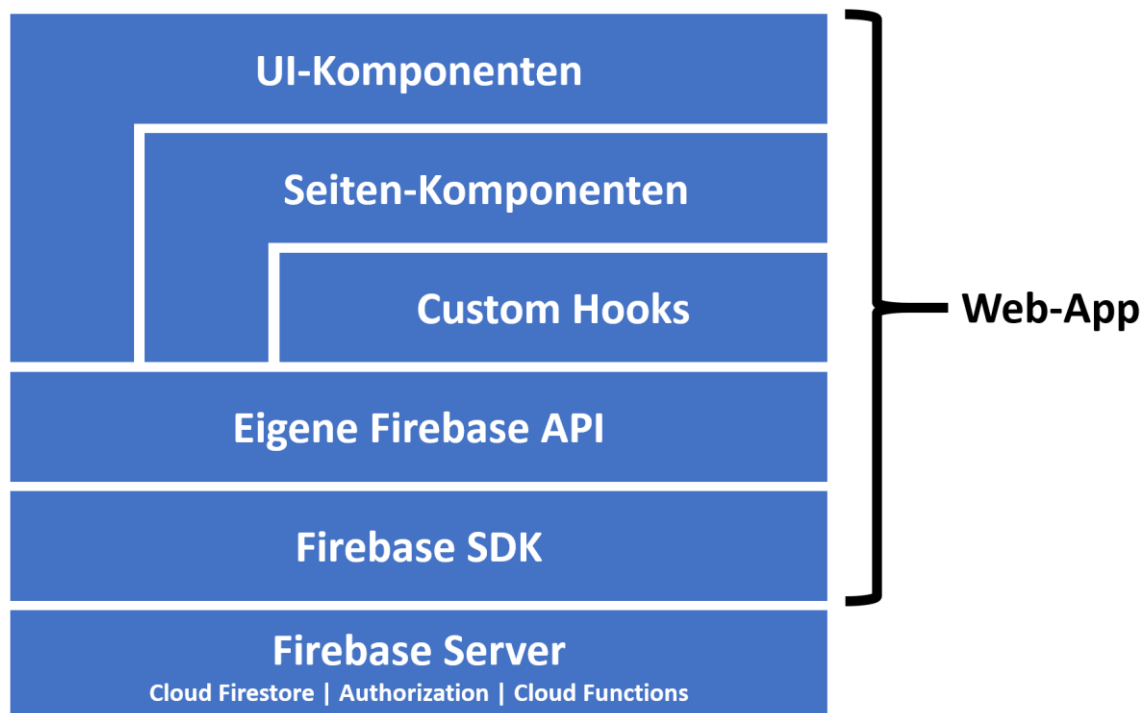


Abbildung 25: Die verschiedenen Kommunikations- und Datenebenen der App. Benachbarte Ebenen kommunizieren miteinander oder tauschen Daten miteinander aus.

Beim Aufrufen einer Seite bzw. beim Öffnen der Web-App werden Nutzende zunächst anonym angemeldet (oder – beim erstmaligen Aufrufen – registriert und danach angemeldet). Nachdem Nutzende angemeldet (d.h. authentifiziert) wurden, lädt die aufgerufene Seite alle benötigten Daten und rendert ihre Komponenten.

Folgendes Beispiel soll diesen Prozess verdeutlichen:

1. Der Nutzer (die Nutzerin) ruft das *Dashboard* auf.
2. Der Router rendert die *Dashboard*-Komponente
3. In der *Dashboard*-Komponente startet der *useChallenges*-Hook.
4. Der *useChallenges*-Hook ruft die Funktion *getActiveChallenges* der Firebase API auf.

5. Die Firebase API nutzt die *firestore*-Bibliothek des Firebase SDK, um die aktuellen Challenges aus der Datenbank zu laden.
6. Das Firebase SDK kommuniziert mit dem Firebase Server und gibt die Daten weiter.
7. Der *useChallenges*-Hook erhält die aktuellen Challenges und speichert sie mittels des *useState*-Hooks in der *Dashboard*-Komponente.
8. Die *Dashboard*-Komponente rendert alle Challenge-Objekte in *ChallengeCard*-Komponenten.
9. Der Nutzer (die Nutzerin) sieht das Dashboard mit den aktuellen Challenges.

### 5.2.1 Datenstruktur

Die Cloud Firestore Datenbank besteht aus folgenden Collections:

**Users:** Beinhaltet alle Daten zu den Nutzenden.

**Challenges:** Beinhaltet alle Challenges (aktive, abgelaufene und geplante).

**Questions:** Beinhaltet die einzelnen Fragen, aus denen ein Quiz besteht.

**Contents:** Beinhaltet die Inhalte, aus denen eine Content-Challenge besteht.

**Places:** Beinhaltet Informationen zu den Orten, aus denen eine Geo-Challenge besteht.

Dabei werden jeder Ort, jeder Inhalt und alle Nutzende, sowie jede Challenge und jede Frage in einem eigenen Dokument festgehalten. Ein Challenge-Dokument einer Geo-Challenge beinhaltet beispielsweise diese Felder:

**type:** Definiert den Challenge-Typ („geo“, „quiz“ oder „content“).

**startsAt:** Zeitstempel für den Start der Challenge.

**endsAt:** Zeitstempel für das Ende der Challenge.

**points:** Punkte für das Erledigen der Challenge.

**placeId:** ID des entsprechenden Ort-Dokuments in der Places-Collection.

**title:** Titel der Challenge / des Ortes.

**body:** Beschreibung der Challenge / des Ortes.

**image:** URL zum Bild der Challenge / des Ortes.

**coordinates:**

**latitude:** Breitengrad des Ortes.

**longitude:** Längengrad des Ortes.

Und sieht in der Cloud Firestore Datenbank beispielsweise folgendermaßen aus:

```
body: "Hinter das Bild geben Sie ein Foto ein, das Sie für den  
Hintergrund wählen"

coordinates
  latitude: 47.994118
  longitude: 7.847457

endsAt: 1589500800000

image: "https://firestore.googleapis.com/v1/projects/your-project-id/databases/(default)/documents/your-document-id?alt=media"

placeId: "ChIJ... "

points: 90

startsAt: 1589414400000

title: "Hintergrund des Monats"

type: "geo"
```

Abbildung 26: Screenshot eines Challenge-Dokuments in der Firebase Web-App.

Ein User-Dokument hat folgenden Inhalt:

**userId:** Die einzigartige Firebase Authentication User-ID des Nutzers beziehungsweise der Nutzerin. Gleichbedeutend mit der Dokument-ID.

**username:** Der gewählte Nutzername.

**joinedAt:** Beitrittsdatum des Nutzers (der Nutzerin) im Format YYYY-MM-DD.

**points:** Die bisher erspielte Punktzahl des Nutzers (der Nutzerin).

Außerdem besitzt jedes User-Dokument drei Sub-Collections:

**activity:** Beinhaltet Daten zu den einzelnen Aktionen der Nutzenden. Zum Beispiel: „Freundschaftsanfrage gesendet“ oder „Geo-Challenge erledigt“.

**friends:** Beinhaltet Informationen zu befreundeten Nutzenden.

**stats:** Beinhaltet ein einziges Dokument mit Statistiken des Nutzers (der Nutzerin).

Collections	Documents	Document
challenges	3vwHYYP5r0eb09o79tiJbbUpa103	<b>Sub-Collections</b> activity friends stats  <b>Document-Data</b> joinedAt: "2020-03-10" points: 70 userId: "d2nMEACKrSgo2KJJfXJ8F3Uarom1" username: "testuser"
contents	BHGMS2e7bpfWp6kpIW1uwWS7nHy2	
places	Erv22E7Su7RV8LffdCWIAQ4bB0J3	
questions	Jxd5XyMhW4Vu3nDuYXu1g1LRXHv1	
users >	Xkf5uiDwidfbU014yVbRnmSZggZ2	
	ZPSfJpD3s4gIBeDCaPupsKdUiF52	
	d2nMEACKrSgo2KJJfXJ8F3Uarom1 >	
	kZCFV4KEcxbr3hYaQSHZTTC16uB3	
	pyMFJPnJP5W47otHbdXs0EspZ2L2	
	tFuK4xrVxPUVLJUte5ZBkF1ruR83	
	tQEbIDLyy1htZMfNrE4maxqtkup2	

Abbildung 27: Screenshot der Cloud Firestore Datenbank in der Firebase Web-App mit angefügten Beschriftungen. Der Screenshot zeigt ein User-Dokument und dessen Sub-Collections.

## 5.3 Die App aus Sicht der Nutzenden

Wenn Nutzende die App zum ersten Mal starten – durch Klicken auf einen Link oder das Icon auf dem Startbildschirm des Smartphones – werden sie zunächst aufgefordert, einen Nutzernamen zu wählen. Für den Nutzernamen sind nur Groß- und Kleinbuchstaben, sowie Unterstriche erlaubt. Ohne diesen Schritt kann die App nicht benutzt werden. Es gibt keine Möglichkeit diesen Schritt zu umgehen. Nachdem erfolgreich ein Nutzernamen gewählt wurde, landen die Nutzenden im Dashboard.



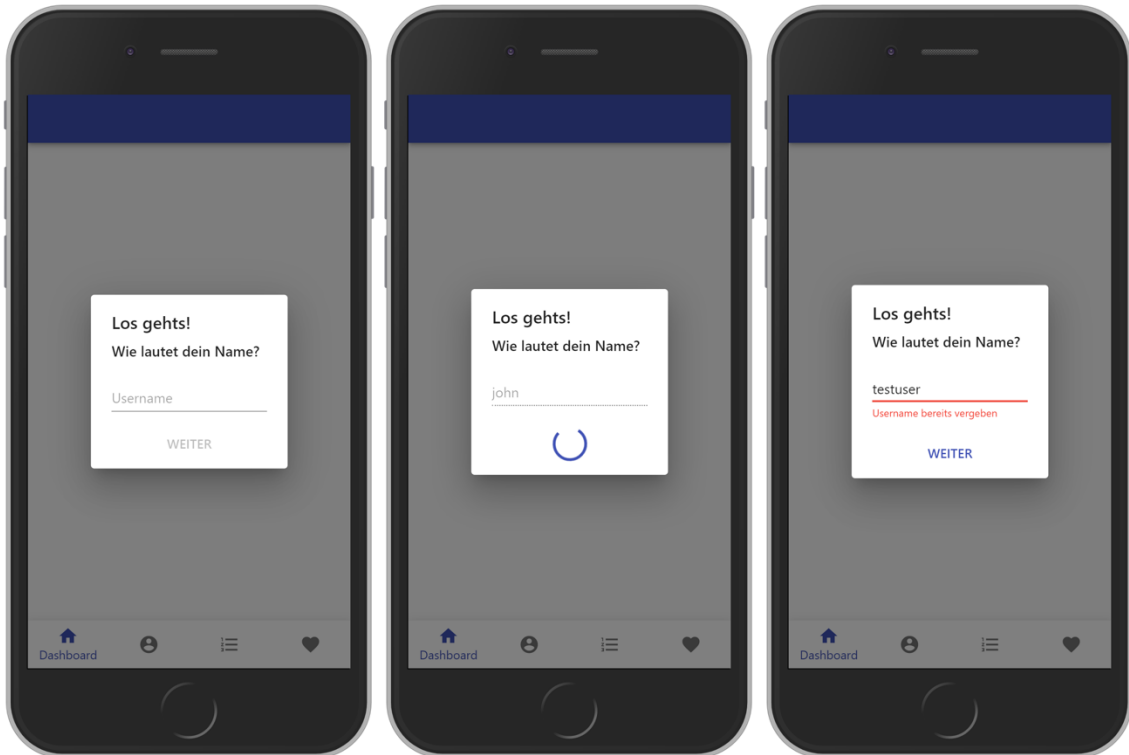


Abbildung 28: Screenshots der App beim erstmaligen öffnen. Links: Nutzende müssen einen Nutzernamen wählen. Mitte: Warten auf die Antwort der Cloud Function. Rechts: Information, falls der gewählte Nutzername schon vergeben ist.

### 5.3.1 Dashboard-Seite

Im Dashboard werden die aktuellen Challenges, der Nutzername und die Punkte des Nutzers (der Nutzerin) zusammen mit einer Navigationsleiste angezeigt. Die Challenges werden als Karten dargestellt. Jede Karte hat als Hintergrund ein Bild, welche den Typ der Challenge andeutet. Außerdem werden den Nutzenden die noch verbleibende Zeit und die möglichen Punkte der Challenges angezeigt. Nutzende können nun über die Navigationsleiste auf andere Seiten wechseln oder auf eine Challenge klicken.



Abbildung 29: Screenshots der Dashboard-Seite. Links: Neu beigetreten. Rechts: Nach Absolvieren von zwei Challenges.

### 5.3.2 Challenge-Seite

Klicken Nutzende auf eine der angezeigten Challenges im Dashboard wechselt die App auf die Challenge-Seite und zeigt die angeklickte Challenge an. Je nach Challenge-Typ (Quiz-Challenge, Geo-Challenge oder Content-Challenge) werden die passenden Komponenten in der Challenge-Seite angezeigt. Am oberen Bildschirmrand wird die Challenge wie im Dashboard als Karte dargestellt. Darunter erscheinen Name und Beschreibung des Ortes bei einer Geo-Challenge beziehungsweise Titel und Inhalt der Content-Challenge. Bei einer Quiz-Challenge wird nur eine Überschrift mit dem Text „Quiz-Challenge“ angezeigt. Jeweils darunter folgt ein Button, mit dem Nutzende das Quiz starten beziehungsweise die Challenge abschließen kann. Bei einer Quiz-Challenge wird das Quiz durch einen Klick auf diesen Button gestartet. Bei einer Content-Challenge können Nutzende die Challenge mit diesem Button bestätigen („Alles klar!“-Button). Bei einer Geo-Challenge ist der Button zunächst gesperrt und zeigt die geographische Entfernung zum Ort der Geo-Challenge an. Befinden sich die Nutzenden am richtigen Ort, wird der Button aktiviert und die Challenge kann mit einem Klick abgeschlossen werden. Wird eine Geo-Challenge zum ersten Mal geöffnet, muss der Standortermittlung

(Zugriff auf das GPS-Signal des Smartphones) durch die App zugestimmt werden. Lehnt ein/e Nutzer/-in die Standortermittlung ab, wird er/sie über eine Anzeige darüber informiert, dass die Geo-Challenge nicht ohne die Standortermittlung spielbar ist. Wurde die Challenge erledigt, wird der Button ausgegraut und gesperrt. Außerdem kann zu jedem Zeitpunkt durch einen Button am linken oberen Rand des Bildschirms zurück zum Dashboard navigiert werden.

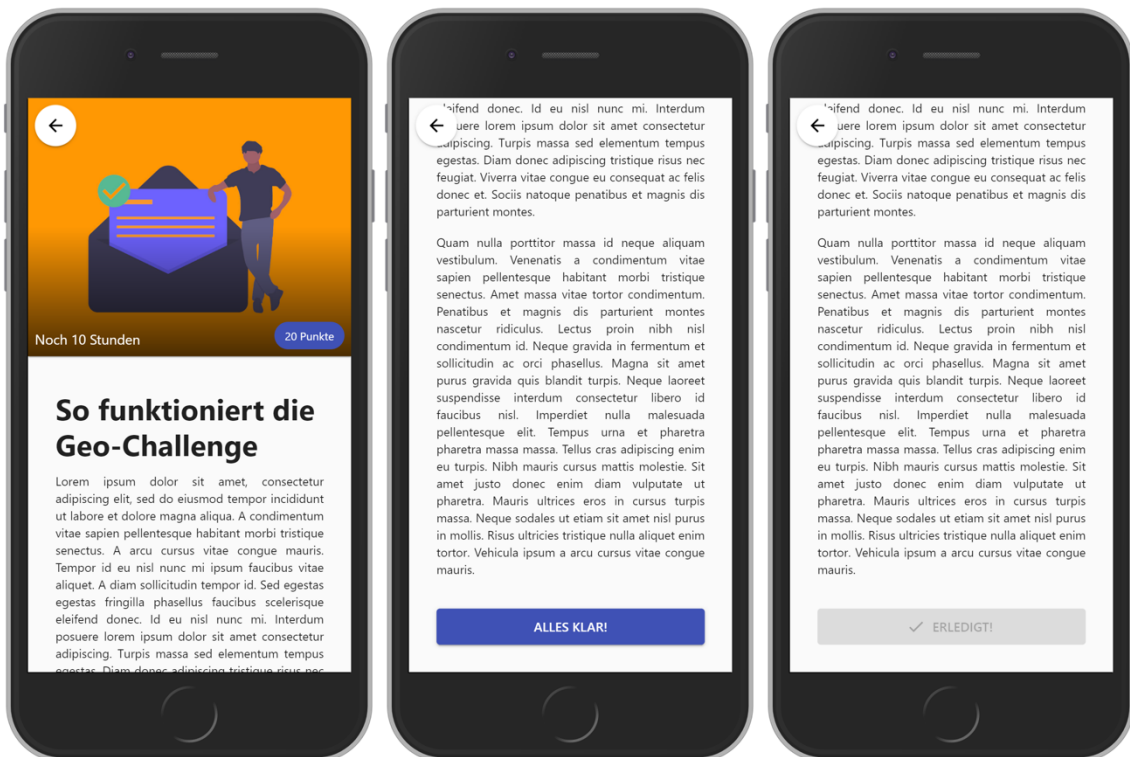


Abbildung 30: Screenshot einer Content-Challenge. Links: Ansicht nach dem Öffnen der Seite. Mitte: Teil des Inhaltes mit dem „Alles klar!“-Button zum Bestätigen der Challenge. Rechts: Der Button zeigt an, dass diese Challenge bereits erledigt wurde.

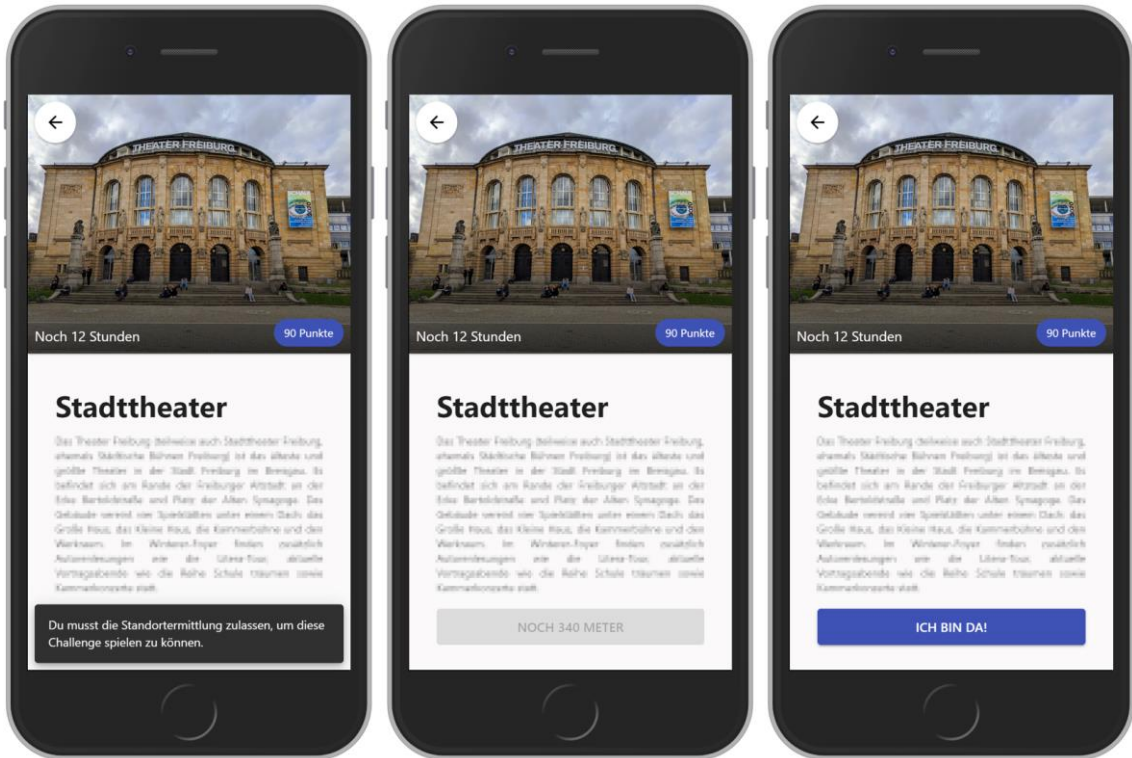


Abbildung 31: Screenshots einer Geo-Challenge. Links: Nutzer/-in wird darüber informiert, dass die Standortermittlung benötigt wird. Mitte: Nutzer/-in befindet sich noch zu weit vom Ort entfernt. Rechts: Nutzer/-in befindet sich am richtigen Ort.



Abbildung 32: Screenshot einer Quiz-Challenge.

### 5.3.3 Quiz-Seite

Klicken Nutzende bei einer Quiz-Challenge auf den Button mit der Aufschrift „Quiz starten“, wechselt die App zur Quiz-Seite und startet das Quiz automatisch, indem die erste Frage angezeigt wird. Ein Quiz besteht aus fünf Fragen, welche nacheinander beantwortet werden müssen. Zu jeder Frage gibt es vier Antwortmöglichkeiten. Für jede Frage haben die Nutzenden 15 Sekunden Zeit eine Antwort auszuwählen. Eine Zeitleiste am oberen Bildschirmrand repräsentiert die noch verbleibende Antwortzeit (Countdown) für die aktuelle Frage. Läuft die Zeit ab, bevor eine Antwort abgegeben wurde, wird diese Frage als falsch beantwortet gewertet. Klicken Nutzende auf eine der Antworten oder läuft die Zeit ab, wird die Frage aufgelöst. Die richtige Antwort erscheint in Grün und die drei falschen Antworten in Rot. Nach circa zwei Sekunden folgt die nächste Frage. Nach Beantworten der letzten Frage landen die Nutzenden im Dashboard. Die gespielte Quiz-Challenge wird nun als erledigt angezeigt und die Punkte für richtigen Antworten werden auf das Punktekonto gutgeschrieben.

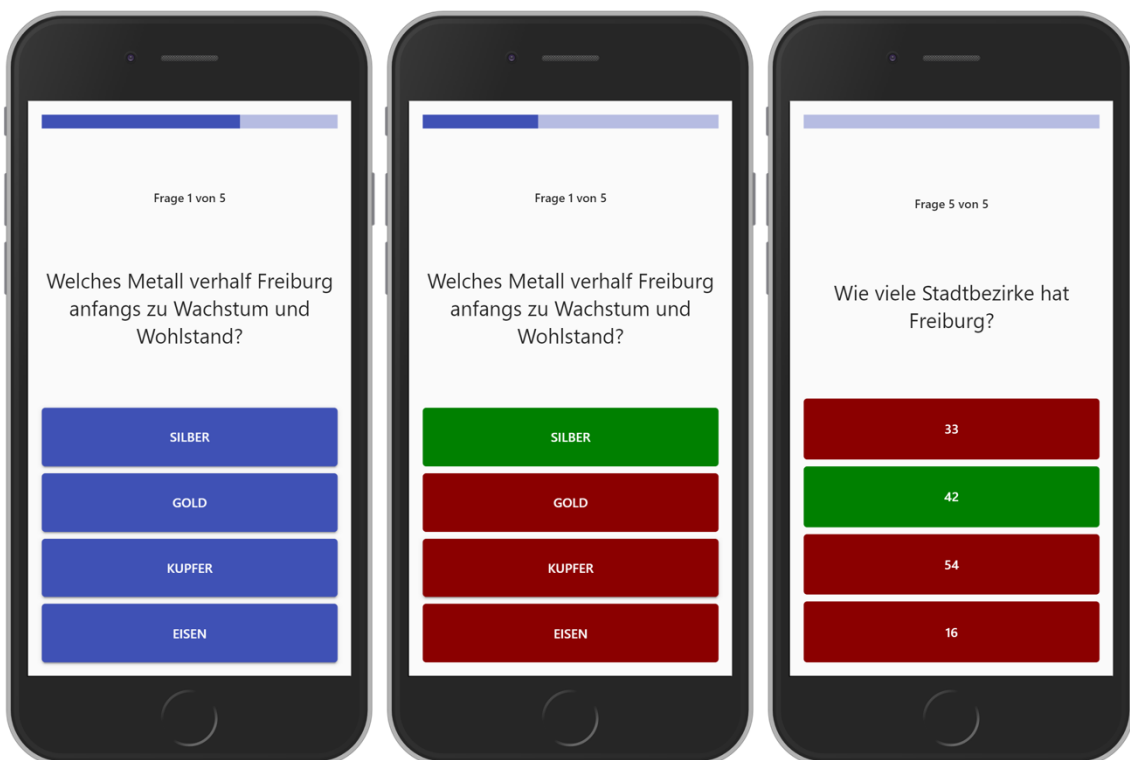


Abbildung 33: Screenshots der Quiz-Seite. Links: Die erste Frage eines Quiz. Mitte: Auflösung nach Antwortabgabe. Rechts: Auflösung nach Zeitablauf.

### 5.3.4 Freunde-Seite

Über die Navigation können Nutzende auf die Freunde-Seite wechseln. Ein Text auf der Freunde-Seite informiert darüber, dass über einen Button am unteren rechten Bildschirmrand die erste Freundschaftsanfrage versendet werden kann. Mit einem Klick auf diesen Button öffnet sich ein Dialog zum Versenden einer Freundschaftsanfrage. Hier muss der Nutzername eines anderen Nutzers oder einer anderen Nutzerin eingegeben werden. Mit dem Klick auf „Anfrage senden“ kann eine Freundschaftsanfrage versendet werden. Gesendete und empfangene Freundschaftsanfragen werden den Nutzenden über beziehungsweise unter der Liste ihrer Freunde angezeigt. Empfangene Freundschaftsanfragen werden dabei ganz oben angezeigt und Nutzende haben die Möglichkeit jede Freundschaftsanfrage einzeln anzunehmen.

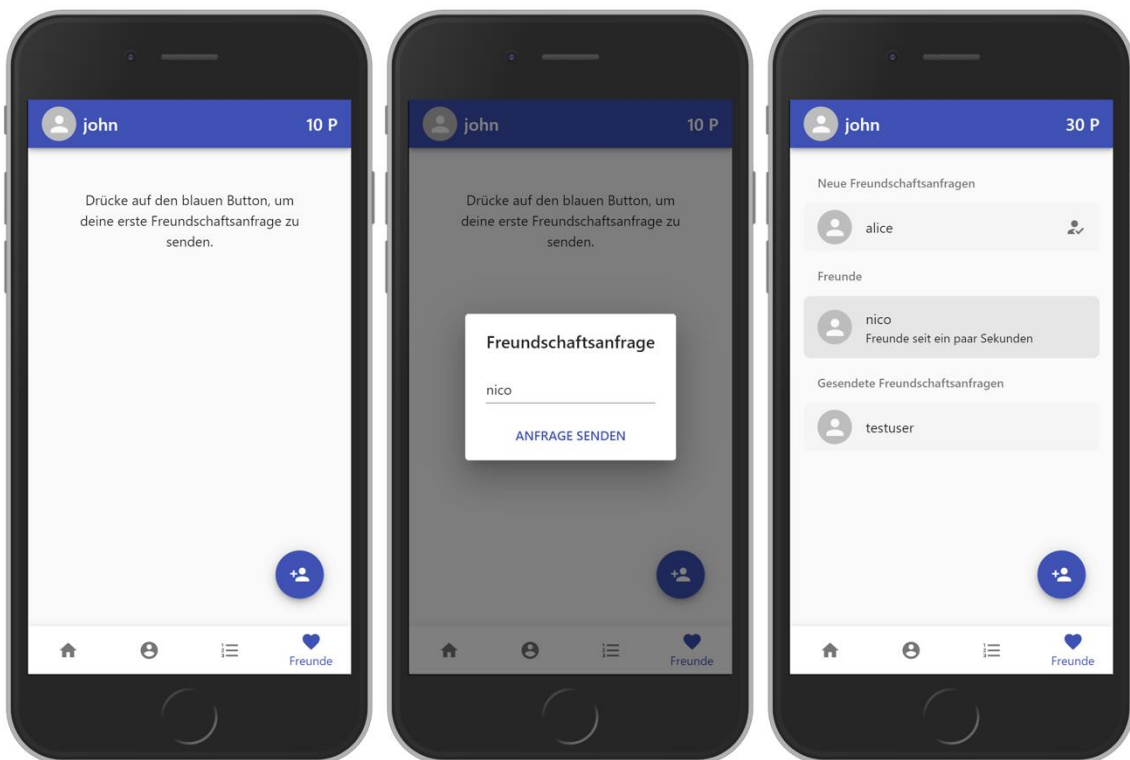


Abbildung 34: Screenshots der Freunde-Seite. Links: Nutzer/-in hat noch keine Freundschaftsanfrage versendet. Mitte: Versenden einer Freundschaftsanfrage. Rechts: Freundesliste, sowie gesendete und empfangene Freundschaftsanfragen.

### 5.3.5 Profil-Seite

Über die Navigation gelangen Nutzende auch auf die Profil-Seite. Auf der Profil-Seite können Statistiken und Punkte zu den eigenen Aktionen in der App eingesehen werden. Es gibt eine Statistik zu den erledigten Challenges, den besuchten Orten, den gespielten

Quizze und der Anzahl der Freunde der Nutzenden. Die aus dem Design-Prototyp bekannte Zeitleiste wurde ebenfalls umgesetzt. Unter der Überschrift „Deine Aktivität“ können Nutzende chronologisch geordnet ihre Aktionen und Punkte in der App nachvollziehen. Die neuesten Ereignisse erscheinen dabei ganz oben.

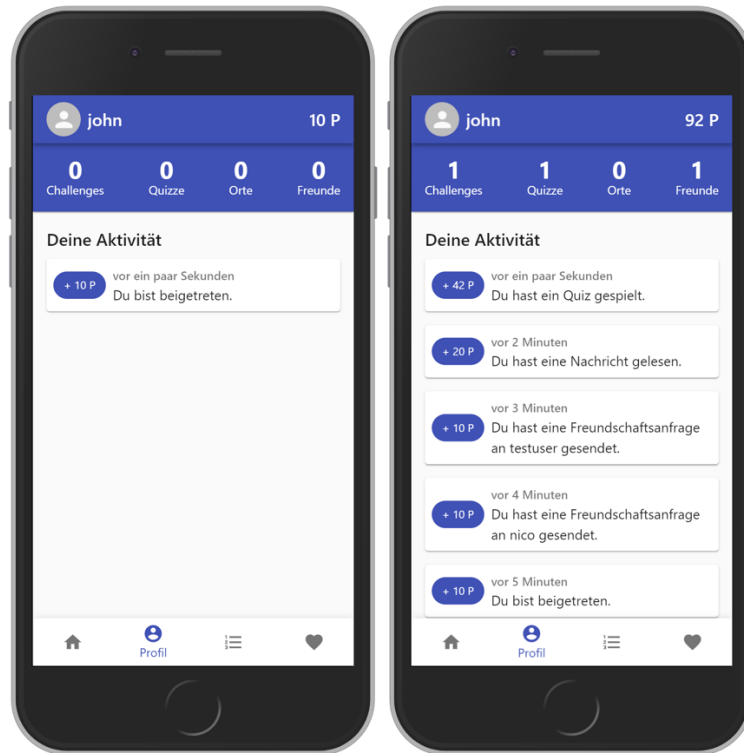


Abbildung 35: Screenshots der Profil-Seite. Links: Neu beigetreten. Rechts: Statistik und Aktivität nach mehreren Aktionen.

### 5.3.6 Ranglisten-Seite

Die Ranglisten-Seite ist ebenfalls über die Navigationsleiste erreichbar. Die Seite ist unterteilt in zwei Bereiche. Unter dem Reiter „Alle“ wird die Rangliste aller Nutzenden angezeigt. Unter dem Reiter „Freunde“ wird die Rangliste der Freunde angezeigt. Die eigene Position der Nutzenden wird in der Rangliste jeweils blau hervorgehoben. Mit einer Wischgeste kann zwischen den Tabs gewechselt werden.

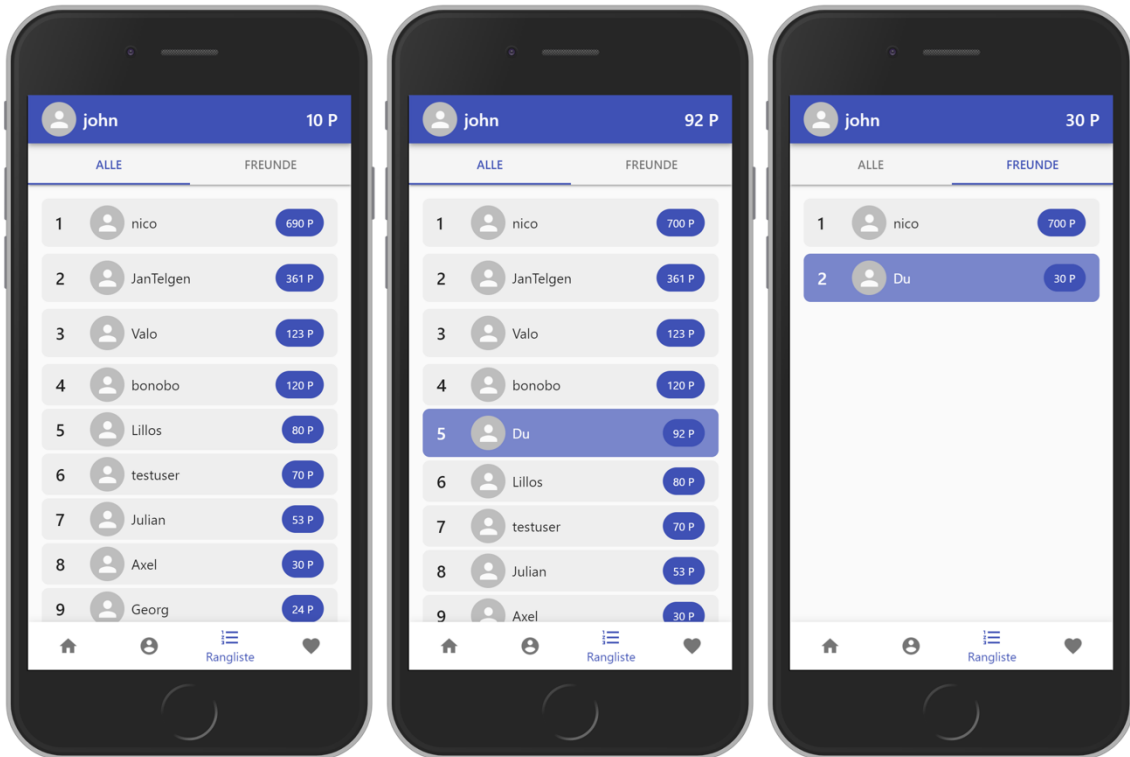


Abbildung 36: Screenshots der Ranglisten-Seite. Links: Nutzer/-in hat nur wenige Punkte. Mitte: Nutzer/-in hat sich Punkte erspielt. Rechts: Freundes-Rangliste mit Freunden des Nutzers (der Nutzerin).

### 5.3.7 Challenge-Einladungen

Die Funktion der Challenge-Einladungen wurde im User-Interface durch einen Button auf der Challenge-Seite umgesetzt. Dieser blaue Button befindet sich rechts oben. Klicken Nutzende auf diesen Button, öffnet sich mittels „Web Share API“ der native Share-Dialog des Smartphones. Dabei wird eine Nachricht generiert, welche die Punktzahl der Challenge und einen Link zur Challenge-Seite dieser Challenge beinhaltet (siehe Abbildung 37). Nutzende können diese Nachricht über den gewünschten Kanal teilen.



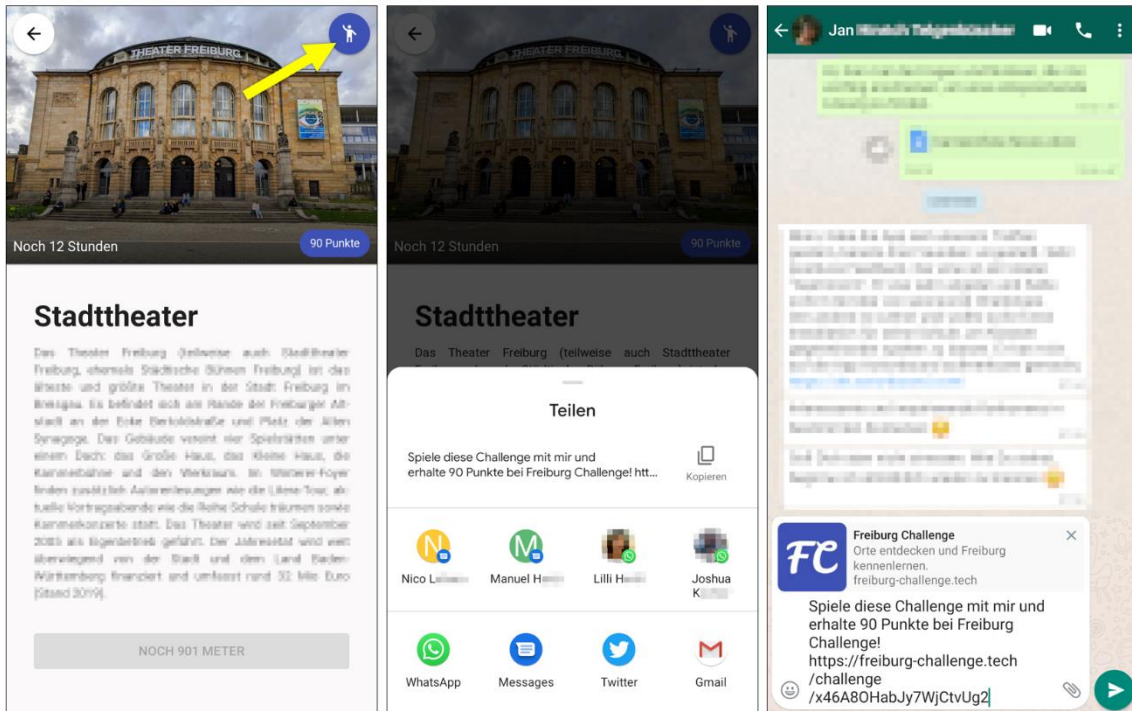


Abbildung 37: Links: Der Einladungs-Button auf der Challenge-Seite (durch den gelben Pfeil gekennzeichnet). Mitte: Nativer Share-Dialog bei Android 10. Rechts: Teilen der Nachricht bei Whatsapp.

## 5.4 Admin-Panel

Das Admin-Panel wurde aus der Notwendigkeit heraus entwickelt, Challenges für die App zu erstellen, zu bearbeiten und zu löschen. Das Admin-Panel ist als ein Werkzeug für die Entwicklung der App gedacht und wurde nicht für Endnutzer konzipiert. Während der Entwicklung der App wurde die Funktionalität des Admin-Panels nach Bedarf erweitert.

Das Admin-Panel ist unterteilt in fünf Ansichten. Jede Ansicht repräsentiert eine Collection in der Datenbank. Es gibt eine Ansicht für die Quiz-Fragen (Questions-Collection), die Orte (Places-Collection) und die Inhalte der Content-Challenges (Contents-Collection). Die Challenges-Collection ist unterteilt in eine Ansicht über die geplanten Challenges und eine Ansicht über die abgelaufenen Challenges. Über ein Menü auf der linken Seite kann zwischen diesen Ansichten gewechselt werden. Alle Ansichten sind einheitlich aufgebaut und funktionieren nach dem gleichen Prinzip. Die Daten (Dokumente) der aktuellen Ansicht (Collection) werden in einer Liste oder Tabelle präsentiert. Beispielsweise werden in der Fragen-Ansicht alle Quiz-Fragen aus der

Questions-Collection geladen und aufgelistet. Mit einem Klick auf ein Element (eine Quiz-Frage) dieser Liste oder Tabelle, kann das jeweilige Dokument bearbeitet oder gelöscht werden. Über einen Button am unteren rechten Bildschirmrand können neue Daten (Dokumente), zum Beispiel eine neue Quiz-Frage, in jeder Ansicht hinzugefügt werden. Berechtigte Personen können sich mittels E-Mail-Adresse und Passwort einloggen.

Freiburg Challenge Admin Panel		LOGOUT		
Geplante Challenges				
	Titel	Startet	Typ	Punkte
	Quiz-Challenge <span>live</span>	vor 11 Stunden	quiz	50
	Freiburger Münster <span>live</span>	vor 11 Stunden	geo	90
	Schließe Freundschaften! <span>live</span>	vor 11 Stunden	content	20
	Augustinerplatz <span>live</span>	vor 11 Stunden	geo	60
	Quiz-Challenge <span>live</span>	vor 11 Stunden	quiz	50
	Quiz-Challenge	in 13 Stunden	quiz	50
	Colombischlössle	in 13 Stunden	geo	60
	Quiz-Challenge	in 13 Stunden	quiz	50

[+ NEUE CHALLENGE PLANEN](#)

Abbildung 38: Übersicht über geplante Challenges im Admin-Panel.

Nach dem Anmelden werden alle geplanten Challenges in einer Tabelle angezeigt. Geplante Challenges sind alle Challenges, deren Endzeitpunkt in der Zukunft liegt. Die Challenges sind nach ihrem Startzeitpunkt sortiert. Die nächsten Challenges stehen oben. Die Tabelle zeigt zudem Titel und Bild, Challenge-Typ und Punkte der Challenges.

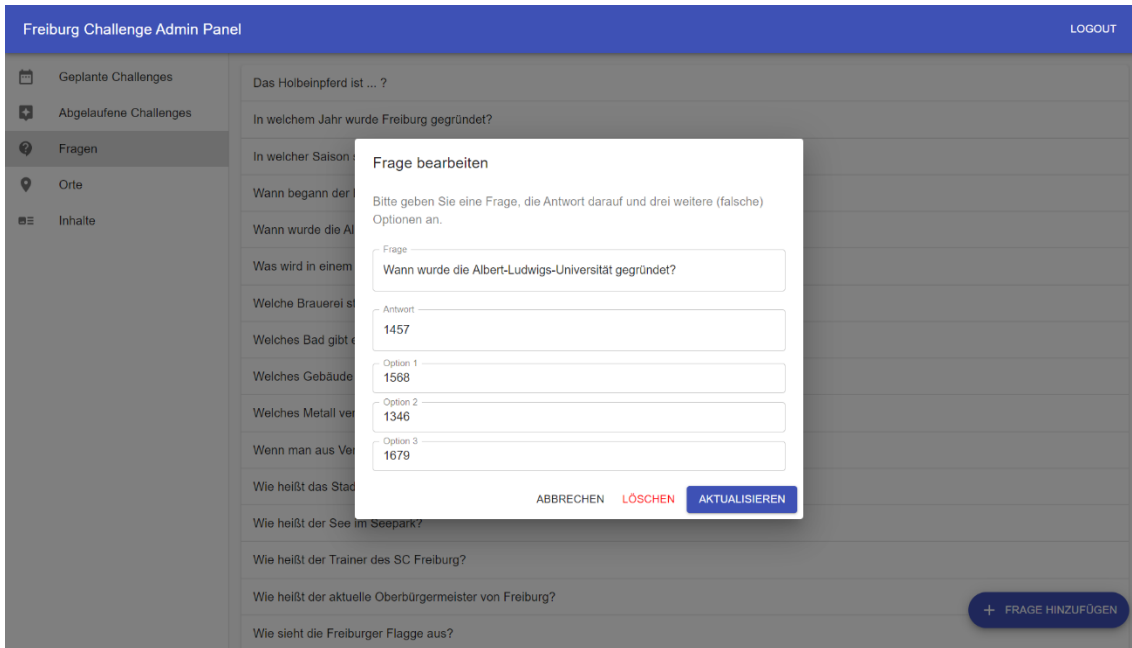


Abbildung 39: Bearbeiten einer Quiz-Frage im Admin-Panel. Die Liste der Quiz-Fragen und der Button zum Hinzufügen einer Quiz-Frage sind im Hintergrund zu sehen.

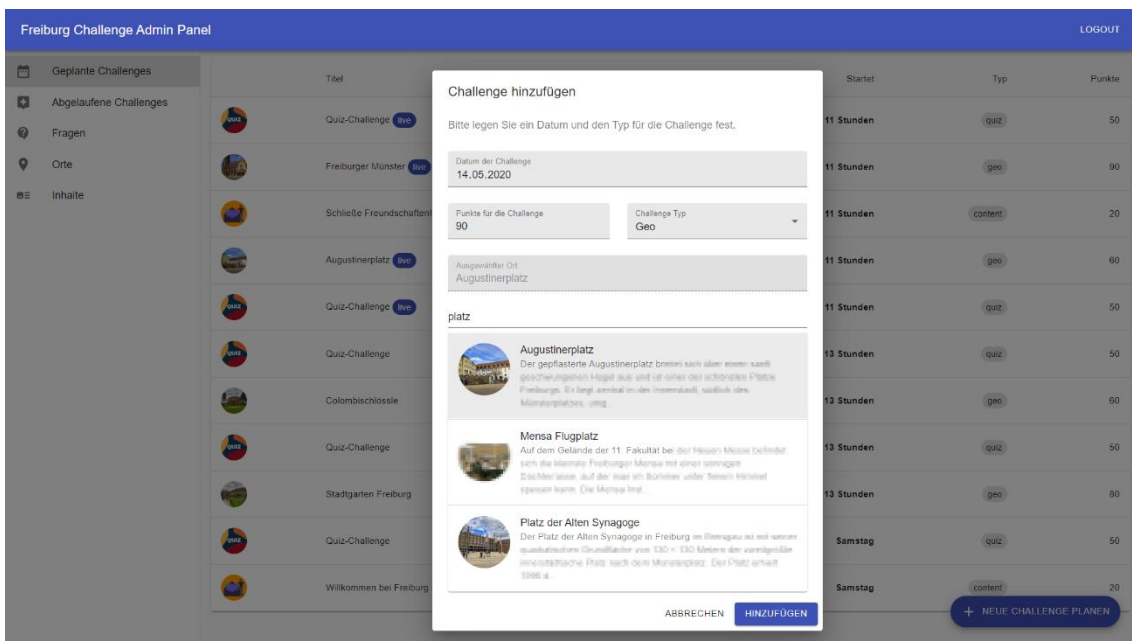


Abbildung 40: Erstellen einer Geo-Challenge im Admin-Panel.

## 5.5 Tests und Programmierstil

Während der Entwicklung wurden für entscheidende Funktionen Modultests geschrieben. Diese Modultests sollen sicherstellen, dass die kritischen Bereiche des Quellcodes der App fehlerfrei funktionieren. Zur Durchführung dieser Tests wurde das quelloffene

JavaScript-Test-Framework „Jest“<sup>8</sup> eingesetzt. Jest, ebenfalls von Facebook mitentwickelt, ist zum Testen von React-Apps geeignet und wird von Create-React-App automatisch mitinstalliert und eingerichtet.

Zusätzlich zu den Tests wird mithilfe des JavaScript-Linters „ESLint“<sup>9</sup> und dem Quelltextformatierer „Prettier“<sup>10</sup> ein konsistenter Programmierstil sichergestellt.

---

<sup>8</sup> <https://jestjs.io/>

<sup>9</sup> <https://eslint.org/>

<sup>10</sup> <https://prettier.io/>

# 6 Evaluation

Zur Evaluation der App wurden ein Usability-Test, sowie ein mehrtägiger Praxistest durchgeführt. Ebenfalls wurde ein Usability-Test für das Admin-Panel durchgeführt. Dabei sollten diese Fragen geklärt werden:

- Funktioniert die App auf unterschiedlichen Geräten ohne technische Fehler?
- Ist das User Interface verständlich, ansprechend und intuitiv bedienbar?
- Wo gibt es Verbesserungsbedarf?
- Inwieweit wurden die Anforderungen umgesetzt?

Ob die Nutzung der App langfristig die Identifikation mit der (neuen) Heimat und das Wohlbefinden der Nutzenden fördert, kann im Rahmen dieser Arbeit aus zeitlichen Gründen nicht getestet werden. Jedoch wurde in der Befragung nach dem Praxistest auf die Anforderungen eingegangen.

Mit einem Usability-Test wird in erster Linie die Nutzerfreundlichkeit oder auch Gebrauchstauglichkeit (englisch „usability“) einer Anwendung überprüft. Dabei sollen auch Schwächen der Benutzeroberfläche (User-Interface) aufgedeckt und Möglichkeiten zur Verbesserung der Anwendung erarbeitet werden. [27] Für die Usability-Tests wurden Einzeltreffen mit Testpersonen abgehalten. Vor dem Test bekamen die Testpersonen eine kurze Erläuterung des Testverlaufs und anschließend einen Link zur App auf das Smartphone zugeschickt (bzw. einen Zugang zum Admin-Panel). Zunächst sollten die Testpersonen die App (bzw. das Admin-Panel) frei bedienen und ohne Anweisungen ausprobieren. Danach bekamen die Testpersonen Aufgaben gestellt. Eine Aufgabe lautet beispielsweise „Füge einen anderen Spieler als Freund hinzu“. Die Testpersonen mussten dabei laut denken, d.h. sie mussten ihre Gedanken und Aktionen kommentieren. Jeder Test wurde mit Erlaubnis der jeweiligen Testperson mit einem Audiorekorder aufgezeichnet. Zusätzlich wurde der Test stichwortartig protokolliert. Anschließend wurden der Test und die App mit den Testpersonen besprochen, insbesondere wo und

weshalb es zu Problemen oder Unklarheiten kam und welche Änderungen die Testpersonen an der App vornehmen würden.

Die Testpersonen nutzen während den Tests ihre persönlichen Smartphones.

Im Vorfeld wurden Quizfragen zu Freiburg und einige Freiburger Sehenswürdigkeiten als Orte in die Datenbank eingepflegt. Außerdem wurden drei kurze Texte geschrieben, welche einzelne Funktionen der App erklären. Mit diesen Inhalten wurden dann Quiz-Challenges, Geo-Challenge und Content-Challenges über das Admin-Panel erstellt und über mehrere Tage verteilt.

## **6.1 Usability-Test der App**

### **6.1.1 Testperson 1**

Testperson 1 ist männlich, 25 Jahre alt und besitzt ein Android-Smartphone.

Nach dem Öffnen der App gab die Testperson gleich ihren Namen in das angezeigte Formular für den Nutzernamen ein und landete daraufhin im Dashboard. Die Testperson bemerkte sofort, dass sie zehn Punkte hat und empfand es als motivierend, nicht bei null Punkten zu starten. Nach kurzer Zeit erkannte die Testperson, dass sie sich im Dashboard befindet und wechselte intuitiv auf die Profil-Seite, indem sie auf das entsprechende Symbol in der Navigationsleiste klickte. Auf der Profilseite konnte die Testperson sofort erkennen, wofür sie die ersten zehn Punkte erhalten hatte. Danach wechselte die Testperson auf die Ranglisten-Seite und sah sich die gesamte Rangliste, sowie die Freundes-Rangliste an. Die Navigation und den Aufbau der App fand die Testperson intuitiv. Über den gesamten Testverlauf traten keine technischen Fehler auf.

Die Testperson hatte keine Schwierigkeiten ein Quiz zu starten und erkannte sofort, dass es fünf Fragen gibt und die Antwortzeit begrenzt ist. Die Beantwortungszeit wurde von der Testperson als angemessen befunden. Nach Beendigung des Quiz und nachdem die Punkte aktualisiert wurden, schaute die Testperson auf der Profil-Seite nach, ob es einen neuen Eintrag unter den Aktivitäten gab und anschließend, welche Platzierung sie nun in

der Rangliste belegte. Im Dashboard las die Testperson die noch verbleibende Zeit der Content-Challenge und vermutete, dass dies eine Art Tagesaufgabe sei. Die Testperson öffnete die Content-Challenge, las den Text durch und klickte auf den „Alles klar“-Button, um die Content-Challenge zu erledigen. Dabei wunderte sich die Testperson darüber, dass die von ihr als Tagesaufgabe wahrgenommene Challenge nur durch das Klicken des Buttons schon erledigt wurde. Das Konzept der Content-Challenge war der Testperson nicht verständlich. Auf der Challenge-Seite bemerkte die Testperson den Einladen-Button und klickt aus Neugier darauf. Dabei öffnete sich der Share-Dialog des Android-Gerätes. Die Testperson vermutete, dass man eine Challenge mit jemandem gemeinsam spielen oder zumindest teilen könne. Auch merkte die Testperson an, dass man dadurch wahrscheinlich auch Freunde an eine Challenge erinnern könnte. Beim Öffnen der Geo-Challenge erschien der Systemdialog, welcher die Berechtigung der Standortermittlung erfragt. Die Testperson willigte ein. Dabei merkte die Testperson an, dass dies alle anderen Apps „auch schon tun“. Die Testperson überprüfte im Anschluss, ob das GPS des Smartphones aktiviert war. Das Prinzip der Geo-Challenge verstand die Testperson auf Anhieb. Die Statistik und die Übersicht über den Verlauf der Aktivitäten und Punkte auf der Profil-Seite wurde als sehr hilfreich empfunden. Beim Versuch eine Freundschaftsanfrage zu versenden gab die Testperson zunächst einen Nutzernamen mit falscher Groß- und Kleinschreibung ein und erhielt daraufhin eine Fehlermeldung. Nach Überprüfung des falsch eingegebenen Nutzernamens und weiteren Nutzernamen aus der Rangliste konnte die Testperson problemlos Freundschaftsanfragen versenden. Dabei bemängelte die Testperson eine fehlende Suchfunktion für Nutzernamen vor dem Absenden einer Freundschaftsanfrage. Das Annehmen einer Freundschaftsanfrage bereitete der Testperson keine Schwierigkeiten.

Die Testperson sah fast nach jeder Aktion auf der Ranglisten-Seite nach, ob sie nun eine bessere Platzierung belegte. Generell empfand die Testperson die Rangliste als motivierend. Die Freundes-Rangliste wurde ohne Schwierigkeiten gefunden.

### **Weitere Probleme**

Die Testperson hatte mehrfach versucht über das Klicken auf den eigenen Nutzernamen in der App-Bar auf die Profil-Seite zu gelangen und ebenfalls über das Klicken auf den jeweiligen Nutzernamen in der Rangliste die Profile anderer Spieler aufzurufen.

## **Änderungswünsche und Verbesserungsvorschläge**

- Nutzernamen-Suche bei Freundschaftsanfragen
- Möglichkeit zum Ablehnen von Freundschaftsanfragen
- Freundes-Rangliste und Freunde-Seite zusammenführen
- Benachrichtigungen, z.B. bei neuen Punkten
- Bilder bei Quiz-Fragen
- Profil-Seite nach Klick auf eigenen Nutzernamen in Titelleiste
- Öffentliche Profile und Aktivitäten von befreundeten Spielern
- Großansicht nach Klick auf Bild in Geo-Challenge

### **6.1.2 Testperson 2**

Testperson 2 ist weiblich, 20 Jahre alt und besitzt ein Android-Smartphone.

Die App wurde generell als übersichtlich und verständlich empfunden. Die einfache Anmeldung per Nutzernamen wurde als Vorteil gesehen. Über den gesamten Testverlauf traten keine technischen Fehler auf.

Beim Festlegen des Nutzernamens nach dem Öffnen der App hat die Testperson versucht Zahlen und einen Punkt in den Nutzernamen zu integrieren. Nachdem dies mehrfach nicht funktionierte, akzeptierte die Testperson diesen Umstand und gab einen anderen Nutzernamen ein. Im Dashboard bemerkte die Testperson, dass sie von Beginn an zehn Punkte hat. Es wurde versucht auf die Punkte in der App-Bar zu klicken, dabei passierte nichts. Im Dashboard klickte die Testperson zuallererst auf eine Quiz-Challenge und spielte das Quiz daraufhin durch. Das User-Interface wurde von der Testperson intuitiv bedient. Nach dem Quiz freute sich die Testperson über die erhaltenen Punkte. Nach dem zweiten Quiz hätte die Testperson auch gerne noch weitere Quizze gespielt. Die Fragen der Quizze kamen nach Aussagen der Testperson gut an. Beim Öffnen der ersten Geo-Challenge ging die Testperson sehr vertraut mit dem Systemdialog zur Standortermittlung um und willigte ein. Die Testperson las den Text der Geo-Challenge und verstand das Konzept der Entfernungsangabe. Nachdem die Testperson den Inhalt der Content-Challenge überflogen und auf den „Alles klar!“-Button geklickt hatte, gab es eine Verwunderung. Die Testperson verstand nicht, weshalb auf dem Button nun „Erledigt“ stand. Das Konzept der Content-Challenge war nicht verständlich. Beim weiteren



ausprobieren der App entdeckte die Testperson den Button zur Challenge-Einladung und klickte aus Neugier darauf. Der sich dadurch öffnende Share-Dialog war der Testperson bekannt. Die Testperson erwartete allerdings nun die Punkte, welche in der Einladungsnachricht stehen, durch das Einladen selbst zu erhalten. Nachdem dies nicht geschah, wiederholte die Testperson die Einladung und wunderte sich danach, weshalb sie keine Punkte dafür erhalten hatte. Auf der Profil-Seite las die Testperson zunächst die Statistiken und die Punkte. Der oberste Eintrag „Du hast eine Nachricht gelesen“ in der Aktivitäten-Liste verhalf der Testperson dazu, die Unklarheit über die Content-Challenge zu beseitigen. Die Profil-Seite wurde als übersichtlich wahrgenommen. Während des Tests überprüfte die Testperson häufig ihre Aktionen und erhaltenen Punkte auf der Profilseite. Besonders die Übersicht über die Aktionen (Aktivitäten-Liste) gefiel der Testperson. Die Rangliste war für die Testperson selbsterklärend. Die Testperson suchte und fand sich schnell selbst in der Rangliste. Die Freundes-Rangliste wurde beim ersten Besuch der Ranglisten-Seite nicht beachtet. Nachdem eine Freundschaftsanfrage versandt wurde, fragte sich die Testperson, weshalb in der Freundes-Rangliste nur sie selbst auftauchte. Auf der Freunde-Seite wurde der Button zum Versenden von Freundschaftsanfragen gut wahrgenommen und verstanden. Die Testperson klickte auf diesen Button und dachte daraufhin, dass sie nach einem Nutzernamen suchen soll. Zunächst wurde ein Nutzernamen mit falscher Groß- und Kleinschreibung eingegeben und es erschien eine Fehlermeldung. Die Testperson versuchte es daraufhin mit einem anderen Nutzernamen, an den sie sich noch aus der Rangliste erinnern konnte. Nach Klicken auf den „Anfrage senden“-Button wunderte sich die Testperson darüber, dass die Freundschaftsanfrage ohne weiteren Zwischenschritt versendet wurde. Die Testperson freute sich über die dadurch erhaltenen Punkte. Das Annehmen von Freundschaftsanfragen gelang ohne Probleme.

### **Weitere Probleme**

Es wurde versucht, über einen Klick auf den eigenen Nutzernamen in der App-Bar auf die Profilseite zu gelangen. Durch eine Challenge-Einladung wechselt das System in die App, in welcher die Einladung geteilt wurde (in diesem Fall „Whatsapp“). Die App dieser Arbeit ist so für Nutzende auf Bildschirm ihres Smartphones nicht mehr sichtbar.

## **Änderungswünsche und Verbesserungsvorschläge**

- Möglichkeit zum Ändern des Profilbildes
- Freundes-Rangliste auf die Freunde-Seite verschieben und dabei Freunde als Freundes-Rangliste von Ansicht der Freundschaftsanfragen trennen.
- Mehrere Challenges, insbesondere Quiz- und Geo-Challenges pro Tag.
- Benachrichtigung über neuen Freundschaftsanfragen
- Profil-Seite nach Klick auf eigenen Nutzernamen in App-Bar

### **6.1.3 Testperson 3**

Testperson 3 ist männlich, 25 Jahre alt und besitzt ein iPhone (iOS 13).

Der Testverlauf ist der gleiche wie bei Testpersonen 1 und 2. Deshalb wird an dieser Stelle auf eine ausführliche Verlaufsbeschreibung verzichtet.

Die App lief flüssig und wurde generell als übersichtlich und verständlich empfunden. Das Öffnen der App und der Einstieg in das Spiel verliefen ohne Probleme. Die Übersicht über den Verlauf der Aktivitäten und Punkte auf der Profil-Seite wurde als sehr hilfreich empfunden. Das schnelle Erspielen von Punkten zu Beginn und die Rangliste motivierten die Testperson.

#### **Probleme**

Im Safari-Browser wurde bei der Geo-Challenge kein Systemdialog zur Aktivierung der Standortermittlung angezeigt. In den Systemeinstellungen des iPhones musste zuerst die Standortnutzung für Safari aktiviert werden (Einstellungen > Datenschutz > Ortungsdienste). Danach konnte die Standortermittlung auch in der Web-App aktiviert werden. Nach der Installation der App musste die Testperson einen neuen Nutzernamen festlegen und neu beginnen. Im Browser konnte mit dem ursprünglichen Account weitergespielt werden. Die Testperson versuchte über einen Klick auf den eigenen Nutzernamen in der App-Bar auf die Profilseite und über einen Klick auf die eigenen Punkte auf die Ranglisten-Seite zu gelangen. Außerdem wurde versucht, über das Klicken auf den jeweiligen Nutzernamen in der Rangliste oder auf der Freunde-Seite die Profile anderer Spieler aufzurufen. Beim Versuch eine Freundschaftsanfrage zu versenden gab die Testperson einen Nutzernamen mit falscher Groß- und Kleinschreibung ein. Nach

erledigen der ersten Geo-Challenge war die Testperson darüber verwundert, dass keine weiteren Aufgaben folgten. Auf der Challenge-Seite wurden der Zurück-Button und der Einladen-Button oval statt rund dargestellt.

### **Änderungswünsche und Verbesserungsvorschläge**

- Quiz-Spiele gegen Freunde
- Challenge-Einladung innerhalb der App
- Gemeinsames Spielen von Challenges mit Freunden
- Mehr Funktionen und Interaktionen mit Freunden
- Sammlung über besuchte Orte im Profil
- Mehr Challenges pro Tag zur Auswahl

#### **6.1.4 Testperson 4**

Testperson 4 ist weiblich, 24 Jahre alt und besitzt ein iPhone (iOS 12).

Der Testverlauf ist der gleiche wie bei Testpersonen 1 und 2. Deshalb wird an dieser Stelle auf eine ausführliche Verlaufsbeschreibung verzichtet.

Die App lief auch hier flüssig und wurde generell als übersichtlich und verständlich empfunden. Das Öffnen der App und der Einstieg in das Spiel verliefen ebenfalls ohne Probleme. Die Übersicht über den Verlauf der Aktivitäten und Punkte auf der Profil-Seite wurde als hilfreich und interessant empfunden. Die Testperson hat sich über Punkte für das Versenden und Annehmen von Freundschaftsanfragen gefreut. Der Hinweistext auf der Freunde-Seite, der die Freundschaftsanfrage erklärt, war hilfreich für die Testperson. Die Rangliste, insbesondere die Freundes-Rangliste, motivierte die Testperson. Die Testperson betonte am Ende des Tests, dass ihr gefallen hat durch das Spielen etwas über Freiburg gelernt zu haben.

#### **Probleme**

Die Testperson musste für die Geo-Challenge ebenfalls die Standortermittlung in den Systemeinstellungen des iPhones für Safari aktivieren. Ein Systemdialog wie bei Android-Geräten wurde erst nach dieser Einstellungsänderung angezeigt. Auch bei diesem Test gab die Testperson beim Versuch eine Freundschaftsanfrage zu versenden

einen Nutzernamen mit falscher Groß- und Kleinschreibung ein. Die Testperson versuchte außerdem auch über einen Klick auf den eigenen Nutzernamen in der App-Bar auf die Profilseite zu gelangen und über einen Klick auf den jeweiligen Nutzernamen in der Rangliste oder auf der Freunde-Seite die Profile anderer Nutzer und Nutzerinnen aufzurufen. Die Content-Challenge sorgte auch in diesem Test für Verwirrung. Bei der Aufgabe, jemanden zu einer Challenge einzuladen, dachte die Testperson zuerst, dass die Einladung innerhalb der App über die Freunde-Funktion abläuft.

### **Änderungswünsche und Verbesserungsvorschläge**

- Öffentliche Profile und Aktivitäten von befreundeten Nutzenden
- Sammlung über besuchte Orte im Profil
- Mehr Informationen zur App zu Beginn und Erklärungen zu den Funktionen
- Trennung von Content-Challenges und den anderen Challenges
- Ansicht darüber, welche Nutzende auch schon an einem bestimmten Ort waren
- Benachrichtigung über neuen Freundschaftsanfragen
- Weitere Informationen oder Aufgaben anschließend an eine Geo-Challenge

### **6.1.5 Zusammenfassung**

Auch wenn einige Schwachstellen deutlich wurden, kann insgesamt ein positives Fazit gezogen werden. Der Aufbau und die Funktionsweise der App wurden von allen Testpersonen intuitiv angenommen. Die App lief auf allen Geräten flüssig und funktionierte ohne technische Probleme. Alle Testpersonen freuten sich über Punkte für Challenges und Freundschaftsanfragen und wurden durch das Fortschreiten in der Rangliste und den Vergleich in der Freundes-Rangliste motiviert. Als besonders hilfreich und interessant wurde die Übersicht über den Verlauf der Aktivitäten und Punkte auf der Profil-Seite wahrgenommen.

Tabelle 1: Häufig genannte positive Eigenschaften, Probleme und Verbesserungsvorschläge während des Usability-Tests der App.

Positives	Probleme und Schwachstellen	Verbesserungsvorschläge
Design und Struktur	Standortermittlung bei iOS-Geräten	Suchfunktion bei Freundschaftsanfragen
Übersicht über Aktivitäten	Content-Challenge	Öffentliche Profile anderer Nutzer/-innen
Rangliste	Freundschaftsanfragen	Ausbau der Statistiken im Profil
Quiz-Fragen	Navigation über App-Bar	Benachrichtigung über neuen Freundschaftsanfragen und Punkte
Informationen zu Freiburg		Mehr Interaktionsmöglichkeiten mit Freunden Einführung / Erklärung beim ersten Öffnen

Die häufigsten Probleme gab es bei den Content-Challenges und beim Versenden von Freundschaftsanfragen. Eine Funktion zum Suchen nach Nutzernamen vor dem Versenden einer Freundschaftsanfrage könnte letztere Schwachstelle beheben. Bei den Content-Challenges sollte hingegen das Konzept neu durchdacht und überarbeitet werden. Empfehlenswert wäre außerdem den Nutzernamen in der App-Bar mit der Profil-Seite und die Punkte in der App-Bar mit der Ranglisten-Seite zu verlinken. Das einzige Problem, das die Testpersonen nicht selbstständig lösen konnten, trat bei iOS-Geräten bezüglich der Aktivierung der Standortermittlung auf. Da die Geo-Challenges eine Hauptfunktion der App darstellen, sollte in der weiteren Entwicklung mit hoher Priorität darauf eingegangen werden. Außerdem sollten weitere Tests mit iOS-Geräten durchgeführt werden. Häufig genannte positive Eigenschaften, Probleme und Verbesserungsvorschläge sind in Tabelle 1 zusammengefasst.

## 6.2 Usability-Test des Admin-Panels

Der Usability-Test des Admin-Panels wurde mit einer Testperson durchgeführt. Die Testperson ist männlich, 42 Jahre alt und war mit dem Konzept schon vertraut. Der Test wurde auf einem Windows-Laptop mit dem „Google Chrome“-Browser durchgeführt.

Die einheitliche und einfache Gestaltung des User Interface über alle Bereiche des Admin-Panels hinweg wurde als sinnvoll und hilfreich empfunden. Die tabellarische Übersicht der Challenges war verständlich. Die Testperson hatte keine Probleme damit Challenges zu erstellen, zu bearbeiten, zu löschen und zu reaktivieren. Ebenfalls waren das Erstellen, Bearbeiten und Löschen von Fragen und Inhalten ohne Schwierigkeiten möglich. Der Logout-Button wurde leicht gefunden und auch an dieser Stelle erwartet.

### **Schwachstellen und Probleme**

Die Testperson hat versucht mit dem Klick auf eine Tabellenüberschrift die Challenges zu sortieren. Die vordefinierte Sortierung war der Testperson nicht sofort klar. Auch war für die Testperson unverständlich, wie man einem Ort ein Bild hinzufügen kann. Die Testperson hat nach einer Funktion gesucht Quizze zu erstellen. Es wurde erwartet mit den Fragen Quizze erstellen zu können, aus denen man anschließend Quiz-Challenges erstellen kann. Der Unterschied zur Erstellung von Geo- und Content-Challenges sorgte an dieser Stelle für Verwirrung. Beim Erstellen einer neuen Geo-Challenge wurde versucht hierfür gleichzeitig einen Ort anzulegen. Außerdem dachte die Testperson, dass man mit einem Klick auf das X, welches neben dem Datum-Input angezeigt wurde, die Challenge löschen kann. Bei diesem Datum-Input wurde ebenfalls die Beschriftung nicht korrekt dargestellt.

### **Änderungswünsche und Verbesserungsvorschläge**

- Suchfunktion für Quiz-Fragen
- Sortierfunktion für Challenges
- Funktion zum Hochladen von Bildern und Mediathek für hochgeladene Bilder
- Quiz-Bereich und Funktion zum Erstellen eines Quiz

## **6.3 Umgesetzte Anforderungen**

Die Recherche zu den Herausforderungen, denen Menschen nach einem Umzug in eine andere Stadt begegnen, hat zu einer Liste von Anforderungen an eine entsprechende App als Lösung geführt. In diesem Kapitel wird zunächst festgehalten, durch welche Designaspekte und Funktionen der App versucht wurde diesen Anforderungen gerecht zu werden und wodurch die App dahingehend verbessert werden könnte. Tabelle 2 fasst

diesen Teil der Evaluation zusammen. Anschließend wird die Erfüllung der Anforderungen mit den Ergebnissen eines Praxistests evaluiert.

Quiz-Challenges sollen in erster Linie Wissen über die (neue) Heimat vermitteln. Zusätzlich bieten die Beschreibungen der Orte bei den Geo-Challenges Informationen zu den Orten und deren Geschichte. Geo-Challenges fördern außerdem nicht nur die Bewegung außer Haus, sondern auch die Ortskenntnisse der Nutzenden, weil diese durch das Spielen der Geo-Challenges auch an Orte gelangen, die sie vorher noch nicht kannten und bei der Suche möglicherweise neue Wege entdecken. Freundschaften in der App sollen den Nutzenden eine Möglichkeit geben, mit anderen Nutzern und Nutzerinnen in Kontakt zu kommen oder in Kontakt zu bleiben. Die Freundes-Rangliste soll den Nutzenden dazu einen Anreiz geben. Geo-Challenges bieten einen Anlass dafür, dass sich die Nutzenden im öffentlichen Raum bewegen und begünstigen dadurch die Wahrscheinlichkeit sozialer, zwischen-menschlicher Interaktionen. Das Gesamtkonzept der App besteht aus einem Spiel. Nutzende werden durch Challenges herausgefordert und durch Punkte belohnt. Die Rangliste soll die Nutzenden dazu motivieren, mehr Punkte zu sammeln. Auf der Profil-Seite können Nutzende ihren Fortschritt verfolgen. Eine niedrige Einstiegshürde für neue Nutzer und neue Nutzerinnen wird durch mehrere Designaspekte erzielt. Die Umsetzung als Web-App sorgt zunächst dafür, dass keine App heruntergeladen und installiert werden muss. Außerdem kann die App mit jedem Smartphone genutzt werden. Da auf einen Registrierungsprozess neuer Nutzer/-innen verzichtet wurde und neue Nutzer/-innen direkt im Spiel landen, fällt eine weitere Hürde weg. Ein weiterer, wichtiger Aspekt sind positionsunabhängige Challenges wie die Quiz-Challenges. Damit können neue Nutzer/-innen auch unabhängig von ihrem Standort beginnen.

*Tabelle 2: Anforderungen an die App und die Funktionen und Designaspekte, durch welche diese jeweils umgesetzt wurden.*

<b>Anforderung</b>	<b>Umgesetzt durch</b>
Wissen über (neue) Heimat vermitteln	Quiz-Challenges, Geo-Challenges
Bewegung außer Haus fördern	Geo-Challenges

Soziale Aktivitäten fördern	Freundschaften, Freundes-Rangliste, Geo-Challenges
Motivierend / Spielerisch	Challenges, Punkte, Rangliste, Profil mit Statistik und Historie
Niedrige Einstiegshürde	Web-App, keine Registrierung, positionsunabhängige Challenges

Um die Anforderungen besser zu erfüllen sind einige Weiterentwicklungen denkbar. Content-Challenges mit Texten, Bildern und Videos könnten zusätzlich zu den Quiz-Challenges dazu eingesetzt werden Wissen über Freiburg und die Geschichte und Kultur der Stadt zu vermitteln. Die Bewegung außer Haus könnte durch weitere positionsbezogene Funktionen noch mehr gefördert werden. Zwischenmenschliche Interaktionen – und damit soziale Aktivität – könnten durch Veranstaltungen in Form von Event-Challenges noch weiter gefördert werden. Bei Event-Challenges müssen Nutzende einen Ort zu einem bestimmten Zeitpunkt besuchen. Dadurch würden die Nutzenden vor Ort aufeinandertreffen. Zusätzlich zu Freundschaften könnte es auch Teams in der App geben, bei denen mehrere Nutzende in irgendeiner Form miteinander spielen. Außerdem sind Quiz-Spiele denkbar, bei denen zwei (befreundete) Nutzer/-innen gegeneinander antreten oder sich herausfordern können. Der Usability-Test hat auch gezeigt, dass sich manche Testpersonen gerne die Profile anderer Nutzer/-innen angesehen hätten. Eine solche Funktion könnte also auch die sozialen Aktivitäten (zumindest innerhalb der App) fördern. Abzeichen könnten neben den Punkten eine weitere Motivationsebene darstellen. Vorstellbar sind beispielsweise Abzeichen für zehn neu entdeckte Orte oder das Schließen von zehn Freundschaften. Diese Spielprinzipien zielen zunächst auf die intrinsische Motivation der Nutzenden ab. Die Motivation der Nutzenden könnte aber auch durch externe Anreize gesteigert werden. Beispielsweise könnte es Gewinne für die Nutzenden mit den meisten Punkten am Ende jedes Monats geben. Denkbar sind auch öffentliche Profile. Diese würden es Nutzenden ermöglichen sich die Erfolge und den Fortschritt anderer Nutzer/-innen anzusehen und sie gleichzeitig dazu motivieren ihr eigenes Profil aufzuwerten. Der Einstieg für neue Nutzer/-innen könnte durch ein geführtes „Onboarding“ mit Erklärungen zu den einzelnen Funktionen und Challenges in der App weiter erleichtert werden.



Tabelle 3: Funktionen und Designaspekte, die die App im Sinne der Anforderungen ergänzen könnten.

Anforderung	Verbesserung durch
Wissen über (neue) Heimat vermitteln	Content-Challenges
Bewegung außer Haus fördern	Weitere positionsbezogene Funktionen, Event-Challenges
Soziale Aktivitäten fördern	Quiz-Spiele gegen andere Nutzer/-innen, Event-Challenges, Teams, öffentliche Profile
Motivierend / spielerisch	Gewinne, Abzeichen, öffentliche Profile
Niedrige Einstiegshürde	Einleitung / Tutorial, Erklärungen der Funktionen

## 6.4 Praxistest

Um herauszufinden, inwieweit die Anforderungen an die App umgesetzt wurden, wurde zusätzlich ein Praxistest durchgeführt. Im Praxistest wurde die App von sechs Testpersonen über mehrere Tage selbstständig genutzt. Im Anschluss an die Testphase wurden die Testpersonen zur Erfüllung der Anforderungen befragt. Die Gruppe der Testpersonen besteht aus Personen, die neu nach Freiburg gezogen sind, Personen, die vor 1-3 Jahren nach Freiburg gezogen sind und Personen, die in Freiburg aufgewachsen sind.

Generell funktionierte die App während des Praxistests ohne Schwierigkeiten. Eine Testperson mit einem iPhone wurde in Safari allerdings zwei Tage hintereinander dazu aufgefordert, sich neu anzumelden. Nachdem die Testperson die App als Progressive Web App installiert hatte, konnte sie die App während der restlichen Zeit ohne Probleme nutzen. Alle Testpersonen gaben bei der Befragung an durch die App mehr über Freiburg gelernt zu haben. Die Quizze wurden als spaßbringend und lehrreich wahrgenommen und gerne genutzt. Ebenfalls fanden einige der Testpersonen die Beschreibungen der Orte in den Geo-Challenges interessant und wissenswert. Zwei Testpersonen haben täglich Umwege auf dem Weg zur Arbeit, zur Uni oder auf dem Nachhausweg für die Geo-

Challenges in Kauf genommen. Die anderen Testpersonen erledigten nur Geo-Challenges in unmittelbarer Nähe. Die Bewegung außer Haus wurde also bei einem Teil der Testperson gefördert. Soziale Interaktionen fanden während des Praxistests nicht statt. Lediglich eine Testperson hatte mit einer fremden Person über die App gesprochen und eine weitere Person zu einer Challenge eingeladen. Alle Testpersonen gaben bei der Befragung an, dass ihnen die Nutzung der App Spaß bereitet habe. Insbesondere die Quiz-Challenges wurden von allen Testpersonen gerne gespielt. Auch die spezielle Form der Geo-Challenge, bei der der Ort weder benannt noch beschrieben wurde, sondern von den Nutzenden selbst gesucht und gefunden werden musste, bekam besonders positives Feedback. Die Rangliste wurde von allen Testpersonen als enorm motivierend empfunden. Einige Testpersonen gaben an, dass der Wettbewerb sie anspornte, noch mehr Punkte zu sammeln. Dieses Verhalten konnte während des Usability-Tests ebenfalls beobachtet werden. Die Testpersonen hatten außerdem Spaß daran, schon zu Beginn schnell Punkte erspielen zu können und auch für jeden Aktion, wie beispielsweise das Versenden einer Freundschaftsanfrage, Punkte zu erhalten. Wichtig war für einige Testpersonen auch, dass Quiz-Challenges von Zuhause gespielt werden konnten. Für die weitere Entwicklung wünschten sich die Testpersonen mehr Interaktionsmöglichkeiten mit anderen Nutzern und Nutzerinnen und System-Mitteilungen („Push-Benachrichtigungen“) bei neuen Challenges.

# 7 Ergebnis

In dieser Arbeit wurden typische Herausforderung analysiert, denen Menschen nach einem Umzug begegnen, um zu untersuchen, inwiefern eine App dabei helfen kann. Eine Erarbeitung der Anforderungen an solch eine App mündete anschließend in ein Konzept und ein Design für eine App, die Menschen in Freiburg dabei helfen soll, die Stadt besser kennenzulernen und mit anderen Menschen in Kontakt zu kommen. Diese App wurde daraufhin als Web-App implementiert.

Das Ergebnis ist eine Spiele-App mit Informationen, Quizen und positionsbezogenen Aufgaben zu Freiburg, in dem die Nutzenden Punkte und Freunde sammeln können und sich in einer Rangliste gegenseitig motivieren.

Die Evaluation hat gezeigt, dass die App die meisten Anforderungen abdeckt. Den Testpersonen hat die Nutzung der App nach eigenen Aussagen Spaß bereitet und motivierende Faktoren konnten erfasst werden. Der Praxistest bestätigte, dass die App Wissen über Freiburg vermitteln, sowie die Nutzenden dazu bringen kann, sich mehr in der Stadt zu bewegen. Damit auch die sozialen Aktivitäten und Kontakte der Nutzenden durch die App gefördert werden, bedarf es jedoch einer Weiterentwicklung. Die Usability-Tests haben nützliche Informationen zur Verbesserung der Benutzeroberfläche der App und des Admin-Panels hervorgebracht. Dabei zeigte sich auch eine Notwendigkeit für weitere Tests mit iOS-Geräten. Aus den Ergebnissen der Tests konnte eine Anforderungsliste für die weitere Entwicklung abgeleitet werden. Unterschieden wurde dabei einerseits Verbesserungen der Nutzeroberfläche und der technischen Umsetzung und andererseits mögliche Funktionen und Spielelemente, um die sich die bestehende App in Bezug auf die Anforderungen sinnvoll erweitern ließe.

Die Entwicklung mit React im Frontend und Firebase als serverseitige Lösung hat sich als geeignete Lösung erwiesen.

## 8 Literatur

1. (2019) Pokemon GO officially hits 1 billion downloads worldwide. <https://nintendosoup.com/pokemon-go-officially-hits-1-billion-downloads-worldwide/>. Zugegriffen: 26. März 2020
2. /gebrüderheit Urban Challenger App. <http://urbanchallenger.com/app/>. Zugegriffen: 26. März 2020
3. Aksel Ş, Gün Z, Irmak TY, Cengelci B (2007) Migration and psychological status of adolescents in Turkey. *Adolescence* 42(167)
4. Ametsreiter H, Bitkom-Präsidium (2019) Smartphone-Markt: Konjunktur und Trends. Bitkom. [https://www.bitkom.org/sites/default/files/2019-02/Bitkom-Pressekonferenz%20Smartphone-Markt%2020%2002%202019%20Pr%C3%A4sentation\\_final.pdf](https://www.bitkom.org/sites/default/files/2019-02/Bitkom-Pressekonferenz%20Smartphone-Markt%2020%2002%202019%20Pr%C3%A4sentation_final.pdf). Zugegriffen: 26. März 2020
5. Angelika Wörthmüller Einsamkeit – woher sie kommt und wie sie wieder geht. [https://www.planet-wissen.de/gesellschaft/psychologie/emotionen\\_wegweiser\\_durchs\\_leben/einsamkeit-114.html](https://www.planet-wissen.de/gesellschaft/psychologie/emotionen_wegweiser_durchs_leben/einsamkeit-114.html). Zugegriffen: 26. März 2020
6. Ankommen App. <http://ankommenapp.de>. Zugegriffen: 26. März 2020
7. Bitkom (2018) Stadt, Land, Frust: Zwei von drei Bürgern finden ihren Wohnort nicht digital. Pressemitteilung. <https://www.bitkom.org/Presse/Presseinformation/Stadt-Land-Frust-Zwei-von-drei-Buergern-finden-ihren-Wohnort-nicht-digital.html>. Zugegriffen: 26. März 2020
8. Bundesamt für Migration und Flüchtlinge Willkommen in Deutschland. Internetseite. <http://www.bamf.de/DE/Willkommen/willkommen-node.html>. Zugegriffen: 26. März 2020
9. Bundesamt für Migration und Flüchtlinge (2014) Willkommen in Deutschland. Informationen für Zuwanderer, 5. Aufl. Bonifatius, Paderborn
10. cm city media Stadt sind Wir. <https://stadtsindwir.de/data/stadtapp.php>. Zugegriffen: 26. März 2020
11. Cooper A (2004) The inmates are running the asylum. Why high-tech products drive us crazy and how to restore the sanity, 2. Aufl. Sams Publishing, Indianapolis
12. Create React App Webseite. <https://create-react-app.dev/>. Zugegriffen: 26. März 2020
13. Dietzke A (2019) Echte Liebe auch fernab vom Fußball. Pokémon GO Fest in Dortmund. <https://www.tagesspiegel.de/wirtschaft/pokemon-go-fest-in-dortmund-echte-liebe-auch-fernab-vom-fussball/24534834.html>. Zugegriffen: 26. März 2020
14. EQLS2017 (2012) Ich fühle mich eng mit den Menschen in meiner Wohngegend verbunden (Soziale Ausgrenzung und Unterstützung). Europäische Erhebung zur Lebensqualität 2016. [https://www.eurofound.europa.eu/de/data/european-quality-of-life-survey?locale=DE&dataSource=EQLS2017NC&media=png&width=740&question=Y16\\_Q36e&plot=euBars&countryGroup=linear&onlyEU=onlyEU&subset=Y16\\_Activity\\_status&subsetValue=Al](https://www.eurofound.europa.eu/de/data/european-quality-of-life-survey?locale=DE&dataSource=EQLS2017NC&media=png&width=740&question=Y16_Q36e&plot=euBars&countryGroup=linear&onlyEU=onlyEU&subset=Y16_Activity_status&subsetValue=Al). Zugegriffen: 26. März 2020
15. Eyerund T, Orth AK (2019) Einsamkeit in Deutschland: Aktuelle Entwicklung und soziodemographische Zusammenhänge. IW-Report, No. 22/2019. Institut der deutschen Wirtschaft (IW), Köln
16. Gerndt K, Bremus T (2015) Single Page Applications. Webapplikationen auf Steroiden. entwickler.press, Frankfurt am Main
17. Google Cloud Firestore Data Model. <https://firebase.google.com/docs/firestore/data-model>. Zugegriffen: 26. März 2020
18. Google Cloud Firestore Dokumentation. <https://firebase.google.com/docs/firestore>. Zugegriffen: 26. März 2020
19. Google Firebase Authentication Dokumentation. <https://firebase.google.com/docs/auth>. Zugegriffen: 26. März 2020

20. Google Firebase Cloud Functions Dokumentation. <https://firebase.google.com/docs/functions>. Zugegriffen: 26. März 2020
21. Google Firebase Hosting Dokumentation. <https://firebase.google.com/docs/hosting>. Zugegriffen: 26. März 2020
22. Haake H, Ludwigs K, Schneidewind U, Lohmann A (2019) Glücklich in Wuppertal. Ein urbanes Wohlbefindens-Panel. In: Herrmann H, Üblacker J (Hrsg) FGW-Studie Integrierende Stadtentwicklung 07. Forschungsinstitut für gesellschaftliche Weiterentwicklung (e.V), Düsseldorf
23. Happiness Research Organisation (2017) Glücklich in Wuppertal - Erste Ergebnispräsentation
24. Hendriks M, Ludwigs K, Veenhoven R (2016) Why are Locals Happier than Internal Migrants? The Role of Daily Life. *Social Indicators Research* 125(2):481–508. doi:10.1007/s11205-014-0856-7
25. HM Government (2018) A connected society: a strategy for tackling loneliness. Department for Digital, Culture, Media and Sport, London
26. Holt-Lunstad J, Smith TB, Layton JB (2010) Social relationships and mortality risk: a meta-analytic review. *PLoS Med* 7(7):e1000316. doi:10.1371/journal.pmed.1000316
27. Jacobsen J (2018) In-House Usability Tests: der komplette Leitfaden. <https://testingtime.com/blog/usability-test/>. Zugegriffen: 26. März 2020
28. Knight J, Gunatilaka R (2010) Great Expectations? The Subjective Well-being of Rural–Urban Migrants in China. *World Development* 38(1):113–124. doi:10.1016/j.worlddev.2009.03.002
29. LePage P, Beaufort F Add a web app manifest. <https://web.dev/add-manifest/>. Zugegriffen: 26. März 2020
30. Liebig S, Schupp J, Goebel J, Richter D, Schröder C, Bartels C, Fedorets A, Franken A, Giesselmann M, Grabka M, Jacobsen J, Kara S, Krause P, Kröger H, Kroh M, Metzling M, Nebelin J, Schacht D, Schmelzer P, Schmitt C, Schnitzlein D, Siegers R, Wenzig K, Zimmermann S, Deutsches Institut für Wirtschaftsforschung Sozio-oekonomisches Panel (SOEP), Daten der Jahre 1984–2017. SOEP Socio-Economic Panel Study
31. Liu Y, Zhang F, Wu F, Liu Y, Li Z (2017) The subjective wellbeing of migrants in Guangzhou, China: The impacts of the social and physical environment. *Cities* 60:333–342. doi:10.1016/j.cities.2016.10.008
32. Markenwerk GmbH Moin Refugee App. <https://markenwerk.net/projekte/moin-refugee/>. Zugegriffen: 26. März 2020
33. Nielsen L (2013) Personsas. In: The Interaction Design Foundation (Hrsg) *The Encyclopedia of Human-Computer Interaction*, 2. Aufl, Aarhus, Dänemark
34. OECD (2019) OECD Better Life Index. nach den Faktoren Gemeinsinn, Zivilengagement, Gesundheit und Lebenszufriedenheit. <http://www.oecdbetterlifeindex.org/de/#/00050055500>. Zugegriffen: 26. März 2020
35. Richard S, LePage P What are Progressive Web Apps? <https://web.dev/what-are-pwas/>. Zugegriffen: 26. März 2020
36. Snape D, Manclossi S (2018) Children’s and young people’s experiences of loneliness: 2018, London. Office for National Statistics. <https://www.ons.gov.uk/peoplepopulationandcommunity/wellbeing/articles/childrensandyoungpeoplesexperiencesofloneliness/2018#what-makes-children-and-young-people-feel-lonely>. Zugegriffen: 26. März 2020
37. Stadtwerke Jena Mein Jena App. <https://www.meinjena.de/index>. Zugegriffen: 26. März 2020
38. Statistisches Bundesamt (2018) Bevölkerung und Erwerbstätigkeit. Bevölkerung mit Migrationshintergrund (Ergebnisse des Mikrozensus 2017). Fachserie 1, Reihe 2.2. Statistisches Bundesamt, Wiesbaden
39. Statistisches Bundesamt (2019) Bevölkerung und Erwerbstätigkeit. Wanderungen 2017. Fachserie 1, Reihe 1.2. Statistisches Bundesamt, Wiesbaden

40. Statistisches Jahrbuch 2019. Beiträge zur Statistik der Stadt Freiburg im Breisgau. Stadt Freiburg im Breisgau, Amt für Bürgerservice und Informationsverarbeitung, Freiburg
41. Stimson RJ, Minnery J (1998) Why people move to the'sun-belt': A case study of long-distance migration to the Gold Coast, Australia. *Urban Studies* 35(2):193–214
42. Tassi P (2018) 'Pokémon GO' Is More Popular Than It's Been At Any Point Since Launch In 2016. <https://www.forbes.com/sites/insertcoin/2018/06/27/pokemon-go-is-more-popular-than-its-been-at-any-point-since-launch-in-2016>. Zugegriffen: 26. März 2020
43. W3L AG (2015) Single-page Webanwendungen. [https://www.w3l.de/de/fileadmin/user\\_upload/Single-page\\_Webanwendungen\\_2015.pdf](https://www.w3l.de/de/fileadmin/user_upload/Single-page_Webanwendungen_2015.pdf). Zugegriffen: 26. März 2020
44. Wikipedia Web App Wikipedia Artikel Englisch. [https://en.wikipedia.org/wiki/Web\\_application](https://en.wikipedia.org/wiki/Web_application). Zugegriffen: 26. März 2020
45. Wikipedia Web-Apps Wikipedia. <https://de.wikipedia.org/wiki/Webanwendung>. Zugegriffen: 26. März 2020