

DreamSwipe

Tinder für Filme

Leon Gieringer, Robin Meckler, Vincent Schreck

Studienarbeit

14. Mai 2021

Inhaltsverzeichnis

1 Einleitung	1
1.1 Motivation	1
1.2 Methode	1
2 Theoretische Grundlagen	2
2.1 Framework	2
2.2 Language	2
2.3 IDE	2
2.4 Database	2
2.5 Firebase	2
2.6 Recommendationssystem	2
3 Konzept	2
4 Funktionen/Komponenten	2
4.1 Swipe/Aussuchen/Voting	2
4.2 Matches/Chat	2
4.3 Film-/Serienvorschläge	2
4.4 Gruppenorgien	2
4.5 Gespeicherte Filme/Filmliste	2
4.6 Barrierefreiheit	2
4.6.1 Barrierefreiheit in mobilen Anwendungen	2
4.6.2 Barrierefreiheit in Filmen und Serien	2
4.6.3 Barrierefreiheit bei StreamSwipe	3
5 Benutzeroberflächen	5
5.1 Aspekte von Benutzeroberflächen	5
5.2 Oberflächen von StreamSwipe	6
5.2.1 Login-Screen	7
5.2.2 Home-Screen	9
5.2.3 Swipe-Screen	10
5.2.4 Chat	12
5.2.5 Benutzerprofil	12
5.3 Filmliste	12
6 CodeBeispiele	12
7 Probleme	12
8 Fazit	12
9 Literaturverzeichnis	12

1 Einleitung

Vincent Schreck

Die Zahl der Scheidungen in Deutschland hat sich während den Corona-Einschränkungen 2020 verfünffacht [1]. Neben dem erhöhten Ansturm auf Rechtskanzleien haben sich auch unverheiratete Paare zu Zeiten des Lockdowns getrennt und die Paartherapien sind flächendeckend ausgebucht. Die Menschen suchen sich Partner aus, die sie zwar attraktiv finden, mit denen sie jedoch kaum gemeinsame Interessen und Ansichten teilen. Sind diese Leute gezwungen Zeit miteinander zu verbringen realisieren sie, dass ihre Beziehung nicht passt.

Wir wollen diesen gravierenden Fehler in seinem Keim erstickten und revolutionieren das Dating-Game mit einem Verfahren, bei dem persönliche Vorlieben im Vordergrund stehen und das Aussehen zweitrangig ist.

1.1 Motivation

Bereits vor tausenden von Jahren haben sich die Menschen Partner gesucht und mit der ersten Monogamie kam auch die erste Beziehung und wahrscheinlich auch die ersten Beziehungsprobleme.

Eine der wichtigsten Grundlagen einer Beziehung sind gleiche Ansichten, Interesse und Vorlieben, anstatt Aussehen und Geld, denn Schönheit vergeht und Charakter besteht. Jedoch ist das Problem dabei, dass man erst weiß wie gut man zueinander passt, nachdem man sich kennengelernt hat. Viele möglicherweise sehr glückliche Beziehungen finden gar nicht statt, da die Person durch ein eigentlich weniger wichtiges Kriterium herausgefiltert wurde. Sucht man die Ursache dieses Problems, ist man schnell bei der Art des Kennenlernens. Der erste Eindruck ist gewöhnlicherweise die optische Natur. Dementsprechend ist Aussehen in der Realität das erste Filterverfahren, was jedoch durch StreamSwipe an eine spätere Position tritt.

Wir bieten die Lösung zu einem jahrtausendealten Problem der Menschheit.

1.2 Methode

Gerade in den letzten Jahren genießt das Medium Film und Serie einen immer höheren Stellenwert in der Gesellschaft. Durch Video-on-Demand Plattformen wie Netflix, Disney+ und Amazon Prime Video sind Filme und Serien omnipräsent geworden und der Nachschub scheint endlos zu sein. Der Zugriff auf komplette Serien wurde dadurch stark vereinfacht und der Nutzer kann einer Serie oder Filmreihe treu bleiben, da er keine Folge mehr verpassen kann. So kann dieses Medium bereits bei vielen Menschen eine Charaktereigenschaft werden und Charaktereigenschaften vieler Zuschauer passen sich an Filmcharaktere an.

Bereits 2017 haben die 18- bis 39-Jährigen an durchschnittlich 4 Tagen pro Woche eine Serie angeschaut [1]. Aus diesen Vorlieben lässt sich sehr viel auslesen. Bei StreamSwipe wird dies ausgenutzt und über eine Film- und Serienauswahl des Nutzers ein Geschmack berechnet, der über einen Algorithmus mit anderen ähnlichen Geschmäcken gematcht wird. Sobald ein Match entstanden ist, öffnet sich ein privater Chat und die beiden Personen können sich austauschen und verabreden.

2 Theoretische Grundlagen

Autor-Name

2.1 Framework

2.2 Language

2.3 IDE

2.4 Database

2.5 Firebase

2.6 Recommendationssystem

3 Konzept?

Autor-Name

4 Funktionen/Komponenten

Autor-Name

4.1 Swipe/Aussuchen/Voting

4.2 Matches/Chat

4.3 Film-/Serienvorschläge

4.4 Gruppenorgien

4.5 Gespeicherte Filme/Filmliste

4.6 Barrierefreiheit

Barrierefreiheit im Allgemeinen bedeutet, dass ein Gegenstand, eine Einrichtung oder Informationsquelle für Menschen mit Behinderung ohne Unzulänglichkeiten nutzbar, zugänglich oder auffindbar ist ([2], §4). In der Softwareentwicklung versteht man darunter Applikationen für Menschen mit Einschränkungen zugänglich und bedienbar zu machen. Bezogen auf die Entwicklung von mobilen Apps gilt es dabei den akustischen, optischen oder motorischen Einschränkungen der Benutzer entgegenzuwirken.

4.6.1 Barrierefreiheit in mobilen Anwendungen

Mit der Verbreitung von Smartphones ist die Benutzung mobiler Apps stark angestiegen und mittlerweile in nahezu jedem Haushalt aufzufinden. Obwohl etwa 9,5% aller in Deutschland lebenden Menschen einen Schwerbehindertenausweis besitzen (Stand 24.06.2020)[1] was etwa 7,9 Millionen Menschen entspricht, ist die Implementierung von barrierefreier Bedienung nicht selbstverständlich. Gerade Programmierern/innen aus dem privaten Sektor sind diese Funktionen oft nicht bekannt, es besteht kein Interesse oder sie werden schlichtweg vergessen. Software, die für öffentliche Einrichtungen entwickelt wird, ist durch das Behindertengleichstellungsgesetz von 2002 dazu verpflichtet ihr Softwareangebot bis spätestens dem 23. Juni 2021 barrierefrei zu gestalten ([2], §12a Abs.1). Hierzu zählen sämtliche Webseiten sowie mobile Anwendungen.

4.6.2 Barrierefreiheit in Filmen und Serien

Auch die Zugänglichkeit von Filmen und Serien für Menschen mit eingeschränkter Wahrnehmung wurde in den letzten Jahren stark verbessert. Hierbei lässt sich zwischen optischer und akustischer Einschränkung differenzieren. Für hörgeschädigte Personen werden bereits seit mehreren Jahrzehnten Untertitel eingesetzt. Was früher für einzelne Filme durch eine Funktion

des Teletextes erreicht wurde, wird heutzutage durch eine integrierte Funktion des Videoplayers verwirklicht. Immer mehr Videos werden mit Untertiteln veröffentlicht. Manche Anbieter wie beispielsweise die Internetplattform YouTube bieten durch Spracherkennung automatisch generierte Untertitel an, was eine flächendeckende Untertitelung ermöglicht.

Auch für Menschen mit eingeschränktem Sehvermögen werden Filme und Serien mithilfe von Audiodeskriptionen vermehrt zugänglich gemacht. Hierbei wird die bereits vorhandene Tonspur mit Bildbeschreibungen und Kommentaren versehen. Was bis vor wenigen Jahren noch etwas Besonderes war und nur für ausgewählte Filme bestimmt war, ist heutzutage Standard. Größere Video-On-Demand-Plattformen wie Netflix oder Amazon Prime bieten diese Möglichkeit bei nahezu allen Eigenproduktionen an. Zusätzlich werden bestehende Filme neu mit Audiodeskriptionen versehen.

Hieraus lässt sich leicht erkennen, dass Filme und Serien heutzutage auch von Menschen mit Einschränkungen genutzt werden. Was auf den ersten Blick vielleicht nicht bedacht wird oder als unwichtig abgestempelt wird, kann einen nicht unerheblichen Vergrößerungsfaktor für den Kundenstamm bewirken. Für die Entwicklung einer mobilen App, bei der Filme und Serien bewertet werden, spielt also die Barrierefreiheit eine wichtige Rolle und darf auf keinen Fall vernachlässigt werden.

4.6.3 Barrierefreiheit bei StreamSwipe

Bei der Entwicklung von StreamSwipe werden mehrere mögliche Einschränkungen der User betrachtet und entsprechend reagiert. Ziel ist es, dass sowohl der Kunde sowie der Anbieter maximal davon profitieren. Hierfür soll die App für ein möglichst großes Publikum zugänglich gemacht werden, jedoch auch sogenanntes Over-Engineering vermieden werden, da zu viele Funktionen eine App unübersichtlich, teuer und langsamer werden lassen.

Allgemein wird Leserlichkeit durch große Schriftgrößen, hohe Farbkontraste, große Schaltflächen oder universelles Design erreicht. Alleine in Deutschland tragen 44,5 Millionen Menschen regelmäßig eine Brille oder Kontaktlinsen und benötigen somit Sehhilfen [3]. Unterstützung auf Seiten der App kann hierfür durch vergrößerbaren Text geschehen. Da aber davon ausgegangen werden kann, dass Personen, die sich auf Sehhilfen verlassen, bereits eine Brille oder Kontaktlinsen besitzen, wird die Textgröße vorerst nicht variabel gehalten. Außerdem gibt es bei Android- und Apple-Smartphones bereits eingebaute Vergrößerungsfeatures, die Bildausschnitte vergrößert darstellen können. Aus diesem Grund wird in diesem Projekt kein Fokus auf dieses Feature gelegt.

Farbblindheit kann jedoch in vielen Formen auftreten. Um der bekannten Farbfehlsicht entgegenzuwirken, werden Farben aus Problembereichen wie Rot und Grün nicht nebeneinander benutzt. Allgemein wird ein schlichtes Design gewählt und Farben nur zu Akzentuierung und als Stilmittel benutzt, statt als Informationsträger wie beispielsweise in den Abbildungen 1a erkennbar ist. Geringe Sehschärfe durch Achromatopsie kann wie weiter oben beschrieben umgangen werden.

Ist die Sehkraft noch weiter eingeschränkt oder gar nicht mehr vorhanden, werden Semantiken eingesetzt. Hierbei erhält jedes Element auf dem Bildschirm eine Beschreibung, die vorgelesen werden kann. Bei Zahlen und Texten werden diese vorgelesen, sofern keine weitere Information hinterlegt ist. Besonders hilfreich ist dies jedoch bei Abbildungen. Ausgeführt wird das Auslesen von einem Screenreader. Mobile Geräte haben diese Funktion bereits standardmäßig eingebaut (VoiceOver bei Apple und TalkBack bei Android) und wandeln die Semantiken mittels Sprachsynthese in akustische Signale um. Bei Desktopanwendungen wie z.B. JAWS für Windows können diese Informationen zusätzlich auch durch eine Braillezeile wiedergegeben werden.

Bei Flutter ist das Hinzufügen von Semantiken bereits eingebaut. Hierfür kann ein String dem

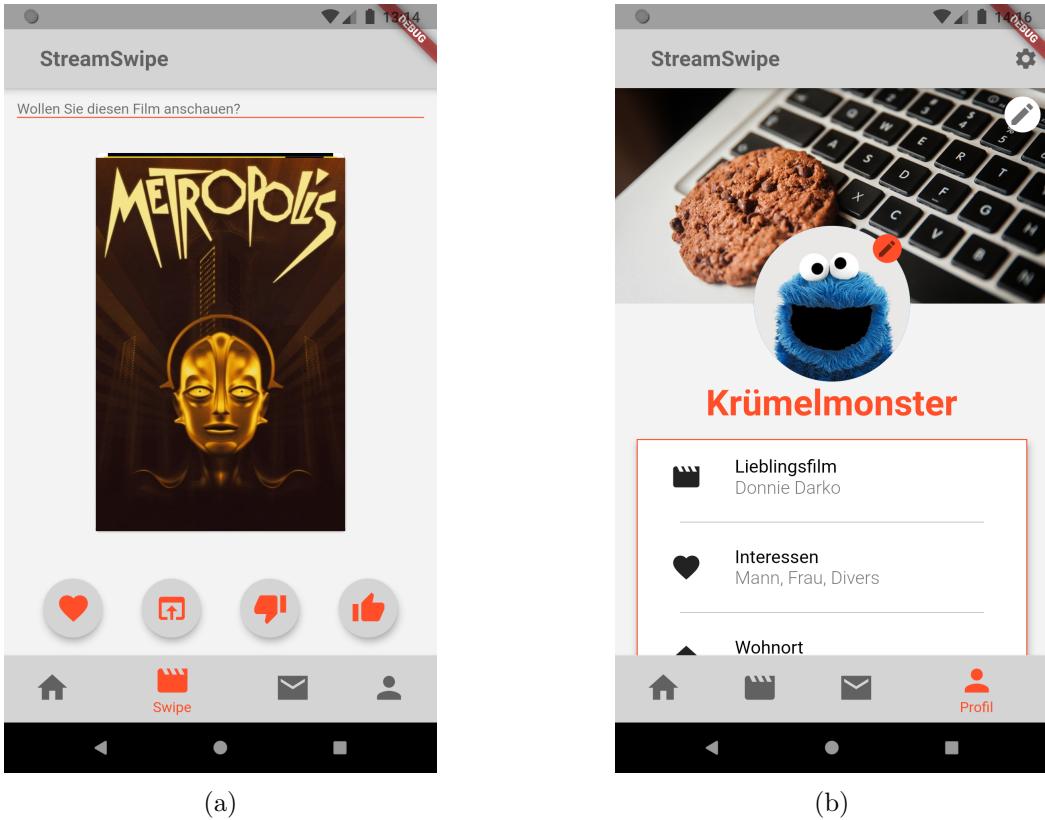


Abbildung 1: Screenshots aus der App StreamSwipe als Beispiele zu (a) schlichem Design, bei dem farbige Akzente nicht der Informationenübertragung dienen um die Zugänglichkeit für farbblinde Menschen zu verbessern und für einen Icon in (b), welcher sonst durch sehgeschädigte Menschen nicht wahrnehmbar ist, wird exemplarisch eine Semantik implementiert.

jeweiligen Bereich zugeordnet werden. In Beispiel 1 ist hierfür der Code des Buttons, der zu den Einstellungen führt. In Abbildung 1b ist dieser Button ganz rechts oben im Eck zu sehen. Der GestureDetector erkennt Interaktionen mit dem Touchscreen, wobei hier nur auf Antippen reagieren soll, deshalb die Funktion `onTap:(){}, die auf den Einstellungsbildschirm leitet. Diese Implementierung ist hier aber nicht von Relevanz und wird übersprungen. In dem GestureDetector ist ein Icon eingebettet, von der Form Settings, was einem Zahnrad entspricht. Dieses Icon erhält eine Farbe und anschließend eine Semantik aus allem was in den Anführungszeichen steht. Ein Screenreader kann AE erkennen und ihn als den Umlaut Ä aussprechen. So wird im kompletten Programm für jedes relevante Element vorgegangen. Teilweise müssen den Semantiken Variablen übergeben werden, da sich die vorzulesende Information ändert wie beispielsweise bei den Filmtiteln.`

Listing 1: Codeausschnitt in Dart von einem Button mit Semantiken.

```
GestureDetector(
  onTap: () {
    ...
  },
  child: Icon(
    Icons.settings,
    color: Provider.of(context).colors.textSmall,
    semanticLabel: "Einstellungen. Zum Auswählen doppeltippen.",
  )
)
```

) ,

Bei einer sauberen Implementierung wird auf diese Weise vorgegangen und eine bereits vorhandene Funktion verwendet. Dies vereinfacht nicht nur die Leserlichkeit des Codes, sondern bietet auch die höchste Modularität, da hierbei normalerweise standardisierte Schnittstellen für Betriebssysteme oder andere Anwendungen verwendet werden. In diesem Fall müssen die Screen-reader von Android und Apple damit arbeiten können.

Um für Personen mit eingeschränktem Hörvermögen oder vollständiger Gehörlosigkeit die App zugänglich zu machen, wird auf akustisches Feedback als notwendige Informationsübertragung verzichtet. Innerhalb der App werden keine Geräusche erzeugt, außer der oben beschriebenen Funktion der Semantiken. Beim Erhalten einer neuen Nachricht oder eines neuen Matches kann weiterhin optional eine akustische Benachrichtigung erhalten werden. Hierbei wird die betriebssystemeigene Funktion übernommen, sodass in der App keine neuen Einstellungen vorgenommen werden müssen.

Auch feinmotorische Einschränkungen werden versucht zu umgehen. Die Navigation und die Filmbewertung in StreamSwipe können durch großflächige Wischbewegungen ausgeführt werden. Wo diese Lösung nicht möglich ist, werden verhältnismäßig große Buttons eingesetzt. Lediglich beim Registrieren und Einloggen werden feine Bewegungen erforderlich. Hierbei öffnet sich allerdings die als Standard eingestellte digitale Tastatur, die in vielen Fällen eine Spracheingabe besitzt, sodass die sehr kleinen Tasten nicht benutzt werden müssen.

Sollte sich in Zukunft jedoch Kritik in Form von negativen Nutzerbewertungen herauskristallisieren, kann eines der noch nicht implementierten Features über ein Update nachgerüstet werden.

5 Benutzeroberflächen

Vincent Schreck

Die Benutzeroberfläche einer Software muss im Grunde genommen nur einen Informationsfluss in zwei Richtungen erzeugen. Die eine Richtung liefert Informationen an den User und über die andere kann der User Informationen an das System weitergeben. Um auf dem heutigen Markt Fuß fassen zu können, sollte eine Oberfläche jedoch wesentlich mehr Aspekte erfüllen.

5.1 Aspekte von Benutzeroberflächen

Die Vielschichtigkeit einer Benutzeroberfläche kann ausschlaggebend für den Erfolg einer Applikation sein, abhängig davon welche Erfahrungen der User mit der Oberfläche macht und welche Eindrücke sie hinterlässt. Hieraus resultiert wie lange ein User auf der App bleibt und wie oft er zurück kommt. Neben der Nutzungszeit erhöht eine positive User Experience die Weiterempfehlungsrate.

Bei erfolgreicher Software besteht ein großer Teil der Entwicklung in der Planung der Oberfläche, da die User Experience nicht zu umgehen ist. Auf die eine oder andere Art erlebt der User immer eine Erfahrung. Neben den offensichtlichen Aus- und Eingabefunktionen werden beispielsweise folgende Kriterien betrachtet:

Simpel: Ausgegebene Information kann zum Beispiel durch Icons, Farben oder Symbole vereinfacht werden. Eine Oberfläche sollte weder überladen sein, noch sollten alle Ein- und Ausgaben auf verschiedenen Screens verteilt sein. Bei der Entwicklung wird eine gesunde Mischung aus maximaler Funktionalität und einfacher, übersichtlicher Darstellung angestrebt.

Einheitlich: Die Bedienung und das Lesen von Applikationen kann erheblich vereinfacht werden wenn einheitliche Bedien- oder Ausgabeelemente verwendet werden. Nicht nur innerhalb einer App ist es sinnvoll konsistente Elemente in der Oberfläche zu verwenden, auch Funktionen von anderen Apps können die Bedienung vereinfachen. Bekannte Funktionen bei Smartphone-Applikationen sind zum Beispiel die Vergrößerung mit zwei Fingern oder das „Daumen nach oben“-Symbol als positive Rückmeldung. Durch das Einbauen solcher Features wird eine App intuitiv und ohne Einführung bedienbar.

Benutzergesteuert: Alle ausgeführten Aktionen sollten vom Benutzer ausgehen. Ein gutes Interface unterstützt den User lediglich bei seiner Bedienung, schränkt ihn aber nicht ein. Mit der heutigen Technologie ist die Verführung groß viele Funktionen automatisch ablaufen zu lassen. Was eigentlich der Sinn einer Applikation ist, kann jedoch auch negative Folgen haben. Zu viel Automatisierung verursacht das Gefühl von Kontrollverlust und Unsicherheit, was sich negativ auf das Vertrauen und somit auf die Benutzungszeit von dem User auswirkt.

Klarheit: Eine mobile App muss ohne Anleitung bedienbar sein. Sobald Unklarheiten beim User entstehen und Funktionen oder Ausgaben nicht erkannt werden können, verliert die Anwendung auf dem freien Markt.

Der User sollte zu jeder Zeit wissen welche Optionen ihm zur Verfügung stehen und welche Folgen seine Aktionen haben. Besonders wichtig ist das Feedback infolge einer Aktion. Auch wenn diese Aspekte offensichtlich erscheinen, können sie bei der Entwicklung einer App leicht übersehen werden. Verwendet werden einfache und für den User bekannte Funktionen, wie die Beschriftung aller Buttons oder das haptische, akustische oder optische Feedback beim drücken einem dieser Buttons.

Benutzerfreundlich/Barrierefreiheit: Die Bedienung der App sollte für Menschen mit Einschränkungen im vollen Umfang möglich sein. In Abschnitt 4.6 wird auf dieses Thema tiefer eingegangen. Aber auch Benutzer ohne Einschränkungen erwarten eine einfache und übersichtliche Bedienung, die auch beispielsweise Eingabefehler mit mehreren Versuchen verzeiht.

Ästhetik: Das Design spielt bei dieser Betrachtung gleich mehrere wichtige Rollen. Es sollte eine angenehme Arbeitsumgebung für den User erstellen, Ein- und Ausgaben verdeutlichen und gleichzeitig mithilfe eines eigenen Stils ein einzigartiges Image für die App schaffen (sogenanntes Branding) um deren Individualität und Wiedererkennungswert zu steigern. Das Design erschafft ein Erlebnis während der Benutzung und weckt Gefühle im User.

Gerade weil viele dieser Aspekte unterbewusst wirken, ist eine ausgiebige Betrachtung unumgänglich.

Eine Schwierigkeit, die sich bei der Entwicklung ergibt sind die zwei unterschiedlichen Ziele. Einerseits sollten bestehende Design- und Bedienelemente übernommen werden um die Bedienung intuitiv und übersichtlich zu gestalten, andererseits aber auch neue Ideen und Innovationen eingebracht werden, um sich von anderen Apps abzuheben und bleibenden Wiedererkennungswert aufzubauen.

5.2 Oberflächen von StreamSwipe

Die Smartphone-App lässt sich in mehrere Bereiche aufteilen, die sich in ihren Funktionen unterscheiden. Auf Basis der oben beschriebenen Grundlagen wurden diese Bereiche entworfen und werden in diesem Kapitel analysiert. Auch wenn manches davon als gewöhnlich oder naheliegend

erscheint, so ist jedes Element mit Bedacht gewählt, erstellt und angepasst worden.

5.2.1 Login-Screen

Bei erstmaliger Benutzung der App öffnet sich der Login-Screen. An diesem Punkt wird der erste Eindruck für den Benutzer gesetzt, wobei bei StreamSwipe ein schlichtes Design gewählt wurde. Man sieht helle Grautöne mit einem Akzentfarbton, welche sich durch alle Bildschirme der App ziehen werden. Abhängig davon, ob der User in den Systemeinstellungen den dunklen Modus gewählt hat, werden anstatt den hellen Grautönen, dunkle bis schwarze Farben dargestellt, siehe auch Abbildungen 3c und 4e.

Auf dem Login-Screen (siehe Abbildung 2a) sind neben einer Überschrift mehrere beschriftete Textfelder und Buttons zu sehen, welche allesamt mit Semantiken versehen wurden, um durch einen Screenreader erkannt und identifiziert werden zu können. Die gewählte Anordnung wird universell bei Apps, Programmen und Webseiten benutzt, sodass die Felder auch ohne die einge tragenen Hinweistexte korrekt ausgefüllt werden könnten. Beim Antippen der Textfelder, öffnet sich die Standardtastatur des Betriebssystems. Sind alle Felder korrekt ausgefüllt, wird der User in die eigentliche App weitergeleitet, ansonsten wird durch individualisierte Fehlermeldung auf eventuelle Falscheingaben hingewiesen. Nach Erstellen eines neuen Accounts, durchläuft der User einen ähnlich aufgebauten Bildschirm (siehe Abbildung 2b) und wird danach aufgefordert weitere Informationen zur Profilvollständigung einzugeben (siehe Abbildung 2c und 2d). Auch hierbei werden bekannte Bedienelemente wie Textfelder, Dropdownmenüs und Checkboxen verwendet, wie in den Abbildungen 2e und 2f beispielhaft dargestellt sind.

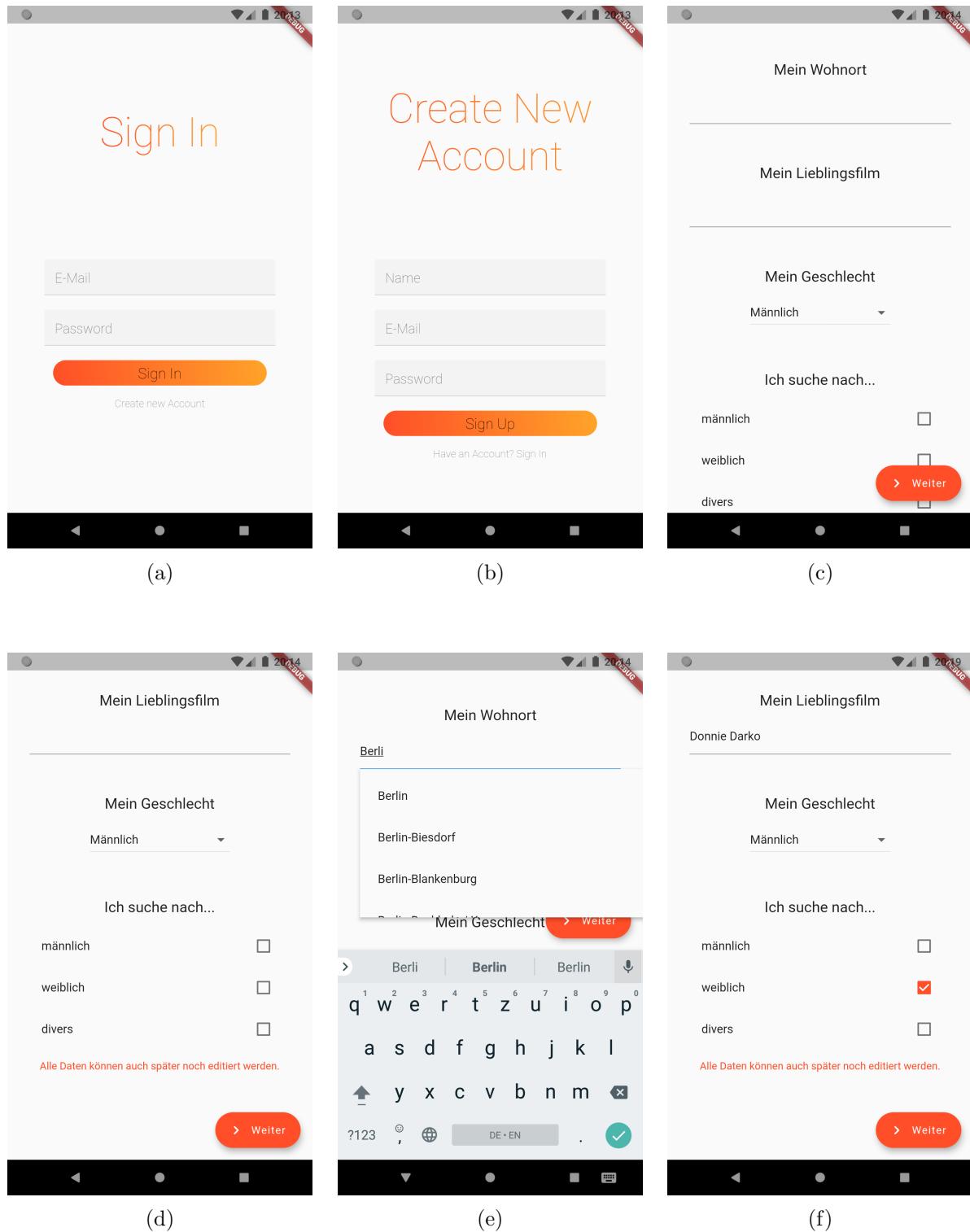


Abbildung 2: Der Login-Screen von StreamSwipe und alle damit zusammenhängenden Seiten. Man sieht (a) das Einloggen bei bestehendem Account, (b) das Erstellen eines Accounts, (c) und (d) das Formular für die weiter benötigten Profildaten, (e) Ein Texteingabefeld mit Auto-vervollständigung als Dropdownmenü und (f) eine ausgefüllte Formular.

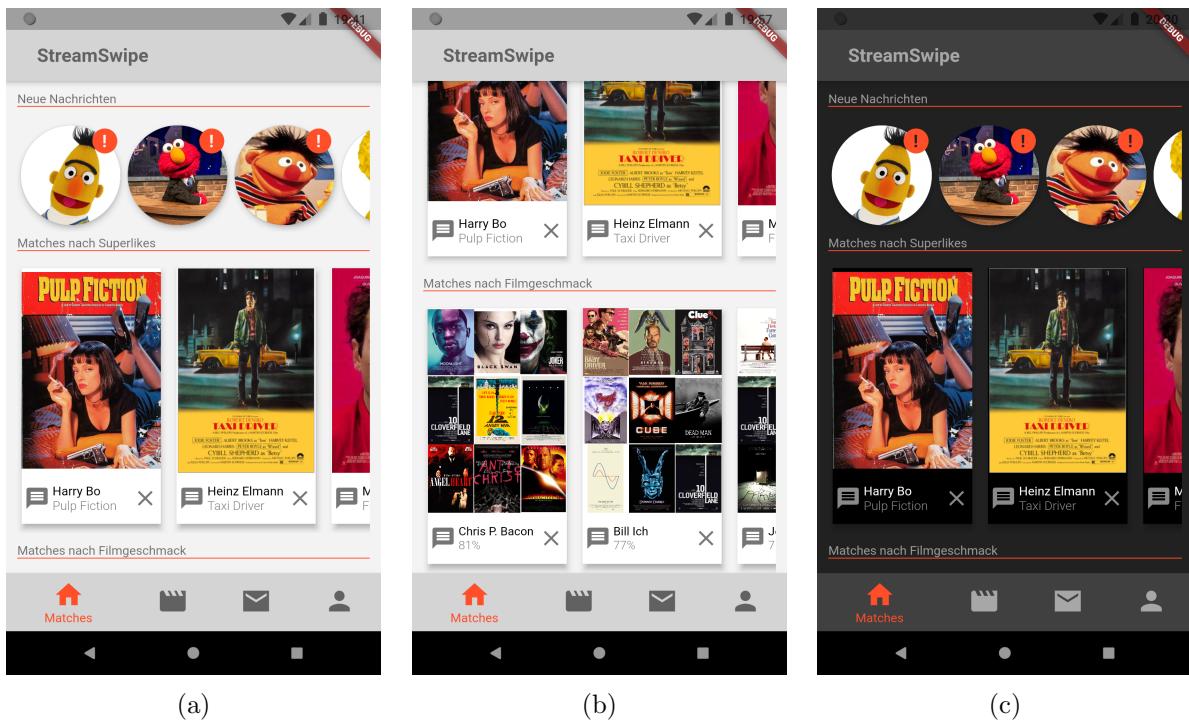


Abbildung 3: Der Home-Screen, der beim Öffnen der App zuerst gezeigt wird und Neuigkeiten wie neue Nachrichten und Matches zusammenfasst. Um den gesamten Inhalt dieser Seite sehen zu können, wird in (a) der obere Abschnitt und in (b) der untere Abschnitt gezeigt. Hat der User in den Systemeinstellungen den dunklen Modus aktiviert, so wird (c) der Home-Screen wie alle anderen Screens angepasst.

5.2.2 Home-Screen

Da davon ausgegangen wird, dass der User sich nicht nach jeder Nutzung ab- und wieder anmeldet, erscheint im alltäglichen Gebrauch der in Abbildung 3 dargestellte Bildschirm zuerst. Demnach bietet es sich an Ereignisse wie neue Matches und neue Nachrichten hier anzusehen. Diese werden wie Abbildungen 3a und 3b zeigen klar strukturiert in Abschnitte eingegliedert, welche mit Überschriften kenntlich gemacht sind. Die einzelnen Matches befinden sich mit allen dazugehörigen Funktionen und Informationen jeweils auf einer Karte. Durch diese Karten kann mit einer von anderen Apps bekannten horizontalen Swipemechanik navigiert werden. Weiterführende Funktionen wie das Starten eines Chats oder das Löschen des Matches werden durch Antippen von allgemein verständlichen Icons ausgeführt.

Auch auf diesem Screen findet sich einerseits das bereits eingeführte Farbschema wieder und es werden andererseits ebenfalls Semantiken verwendet. Bei den neuen Nachrichten wird jeweils der Benutzername vorgelesen und bei den neuen Matches je nach ausgewähltem Bereich der Filmtitel, die Icons oder der Text dazwischen.

Am unteren Bildschirmrand ist eine sogenannte Bottom-Navigation-Bar zu sehen. Sie ermöglicht eine kompakte und anschauliche Navigation durch die relevanten Bildschirme. Außerdem zeigt sie an welcher Bildschirm aktuell ausgewählt ist, wobei diese Information wie in Abschnitt 4.6.3 erarbeitet nicht ausschließlich auf einer Farbänderung basieren sollte und deshalb das ausgewählte Icon durch Hinzufügen von Text hervorgehoben wird. Liest der Screenreader die Semantik hiervon, gibt er die Bezeichnung des aktuellen Bildschirms sowie die Anzahl der weiteren Möglichkeiten an.

5.2.3 Swipe-Screen

Auf dem Swipe-Screen (Abbildungen 4) findet die Bewertung der Filme statt. Durch das hier verwendete Matchingsystem mithilfe des Filmgeschmacks unterscheidet sich StreamSwipe von anderen Apps und erhält so einen innovativen, individuellen Charakter, womit diese Seite das Herzstück der App bildet.

Das zuvor eingeführte Farbschema bleibt auch hier erhalten, wie Abbildung 4a zeigt. Eine Überschrift im selben Stil wie bereits aus Abschnitt 5.2.2 bekannt, verdeutlicht durch eine Frage nach welcher Motivation die Filmauswahl getroffen werden soll. Zentral im Bild ist eine Liste von Postern der zu beurteilenden Filme. Wie bereits durch die Datingapp Tinder verbreitet, werden die vier Antwortmöglichkeiten durch eine Swipe-Bewegung in eine der vier Richtungen ausgewählt. Abhängig von der Position des Fingers auf dem Touchscreen bewegt sich das FilmPoster innerhalb des Bildschirms, was den Effekt einer frei beweglichen Karte hervorruft. Um klarzustellen welche Swipe-Richtung für welche Entscheidung steht, verfärbt sich der jeweilige Indikator in der unteren Reihe bei Verschiebung des FilmPoster. Beide diese Animationen sind in Abbildung 4d zu sehen. Die Indikatoren sind mit Icons versehen, zeigen aber durch Drücken welche Entscheidung sie repräsentieren und in welche Richtung der User dafür swipen muss, wie Abbildung 4c am Beispiel des rechten Indikators zeigt.

Durch Antippen des FilmPosters werden weitere Informationen zu dem jeweiligen Film dargestellt, wie in Abbildung 4b zu sehen. Gleichfalls wird durch ein einfaches Antippen wieder zurück zu den Postern gewechselt. Eine Rotations-Animation verdeutlicht die Illusion der Karten.

Alle diese für die Bedienung der App grundlegenden Steuerungen verlangen keine feinmotorischen Eingaben und können problemlos von Personen mit motorischen Einschränkungen genutzt werden. Auch dieser Bildschirm ist vollkommen mit Semantiken ausgestattet. Anstelle des FilmPosters wird der Name des Films ausgelesen und für die vier Indikatoren am unteren Rand werden jeweils deren Funktion und durch welche Swipe-Richtung sie erreicht werden vorgelesen. Sämtliche Textfelder können ebenfalls problemlos von einem Screenreader gelesen werden.

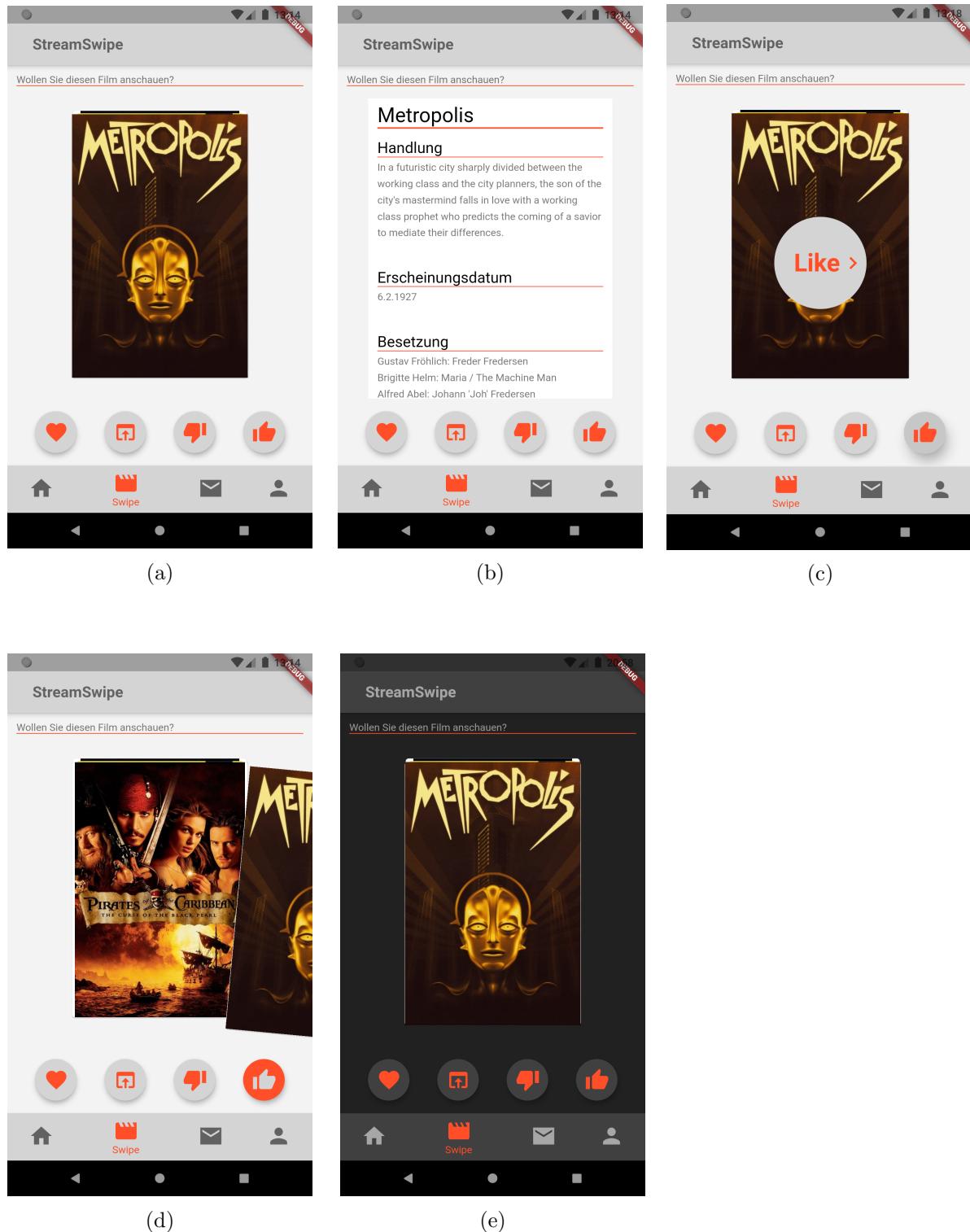


Abbildung 4: Darstellungen und Funktionen des Swipe-Screens mit (a) der Standarddarstellung, (b) weiteren Filminformationen, (c) einer Animation beim Drücken einer der Indikatoren und (d) der Swipe-Animation. Hat der User in den Systemeinstellungen den dunklen Modus aktiviert, so wird (e) der Swipe-Screen wie alle anderen Screens angepasst.

5.2.4 Chat

Chat kommt noch. Wollte glaub Leon noch was fragen

5.2.5 Benutzerprofil

Auf der Profilseite werden ein Profilbild, ein Hintergrundbild und für das Matching relevante persönliche Informationen dargestellt. Es gibt eine Version, die nur von anderen Nutzern sichtbar ist, mit denen ein Match stattgefunden hat, und eine Version, die über die Bottom-Navigation-Bar erreichbar werden kann. Die Letztere wird in Abbildung 5 dargestellt und unterscheidet sich von der Version für andere Nutzer darin, dass Profil- und Hintergrundbild bearbeitet werden können.

Das Farbschema und das Design wurden an die bisherigen Seiten angepasst. Um die Oberfläche simpel und selbsterklärend zu halten, wird jede dargestellte Information mit einem passenden Icon und einem Hinweis versehen (siehe Abbildung 5a). Die Icons zum Bearbeiten der Bilder sind wie auch in vielen anderen Apps platziert und designt. Sie öffnen die systemeigene Bildergalerie des Smartphones um den User aus einem bekannten Umfeld Bilder auswählen lassen zu können. Beim initialen Öffnen einer Profilseite sollen Namen, Profilbild und ein Hintergrundbild ins Auge springen. Sie stellen die ersten Informationen dar, die dem Betrachter wichtig sind, weshalb sie wie in Abbildung 5a deutlich sichtbar ist beim Öffnen mehr als die Hälfte des Bildschirms einnehmen. Anschließend wird der Fokus auf detailliertere Informationen gerichtet. Auf der Profilseite von StreamSwipe wird hierfür heruntergescrollt um den Block mit den Profildaten sehen zu können. Bei dieser Aktion blendet eine Animation das Profilbild aus und verschmälert das Hintergrundbild. Der Benutzername wird ebenfalls aus dem Fokus gezogen, bleibt aber wie in Abbildung 5b zu sehen mit dem verbleibenden Hintergrundbildausschnitt erhalten. Dies hilft dem Betrachter unterbewusst bei dem Fokuswechsel und schafft ein modernes, responsives Feedback bei der User Experience.

Um das durchgängig schlichte Design der App zu erhalten ist der Zugang zu den Einstellungen ausschließlich auf der Profilseite zu finden. Hierfür ist im rechten oberen Bildschirmbereich das repräsentative Icon. Der hierdurch erreichbare Bildschirm (Abbildung 5c) ist gleich aufgebaut wie die Informationeneingabe nachdem ein neuer Account erstellt wurde (Abbildungen 2c und 2d). Die dort angegebenen Informationen können hier wieder angepasst werden.

5.3 Filmliste

6 CodeBeispiele

Autor-Name

7 Probleme

Autor-Name

8 Fazit

Autor-Name

9 Literaturverzeichnis

- [1] Meinungsforschungsinstituts Civey (2020): <https://www.presseportal.de/pm/145489/4627304>, letzter Zugriff: 13. Mai 2021
- [1] Splendid Research (2017): <https://www.springerprofessional.de/konsumforschung/marketingstrategie/konsumenten-auf-der-serien-welle/15146374>, letzter Zugriff: 14. Mai 2021
- [1] Statistisches Bundesamt (2020): <https://www.destatis.de/DE/Themen/Gesellschaft-Umwelt/Gesundheit/Behinderte-Menschen/Tabellen/schwerbehinderte-alter-geschlecht-quote.>

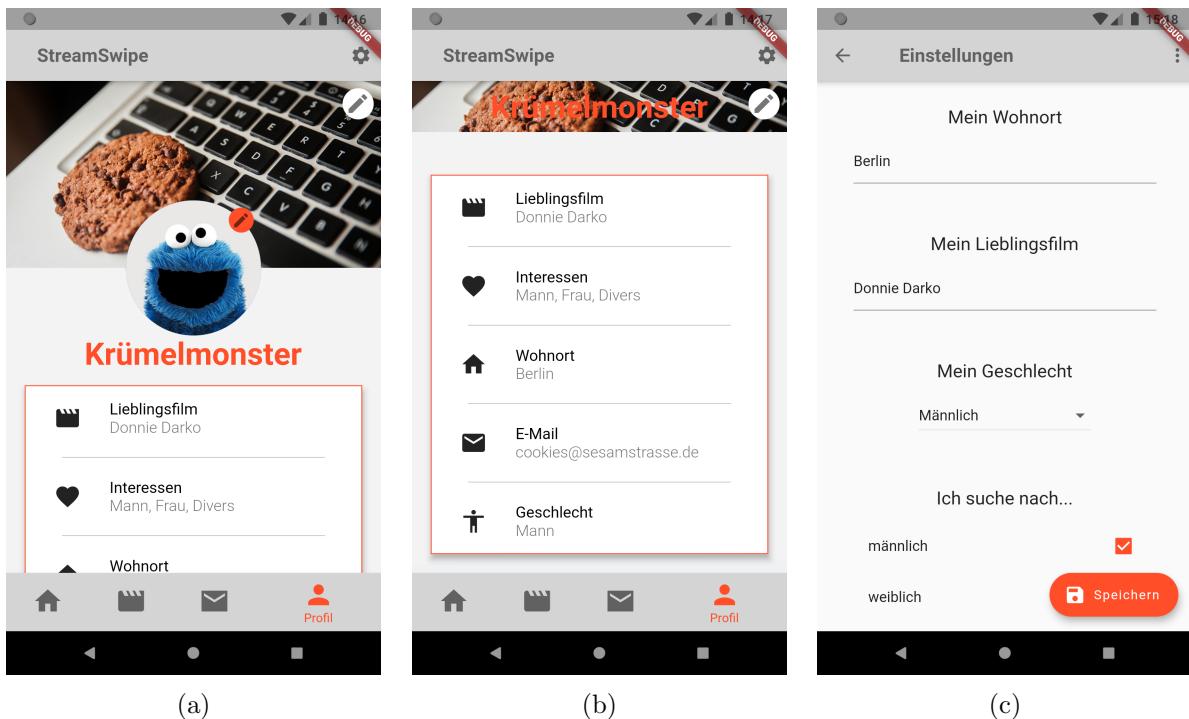


Abbildung 5: Profilseite wie sie für den Nutzer selbst angezeigt wird (a) im normalen Zustand und (b) nach vollständigem Einklappen des Profilkopfes durch eine Animation während dem Herunterscrollen. Mit den von hier aus erreichbaren Einstellungen (c) können die anfänglich gegebenen Profilangaben abgepasst werden.

html;jsessionid=885260788D4FFC7F670576B72E5089F4.live741, letzter Zugriff: 17. April 2021

- [2] Behindertengleichstellungsgesetz (2002): <https://www.gesetze-im-internet.de/bgg/BGG.pdf>, letzter Zugriff: 19. April 2021
- [3] Institut für Demoskopie Allensbach (2019): *Untersuchung zum Sehbewusstsein der Deutschen*, file:///C:/Users/Vincent/AppData/Local/Temp/ZVA_Brillenstudie_2019-1.pdf, letzter Zugriff: 11. Mai 2021
- [99] Mustermann, Max (2020): Methode und Nutzung der Literatur-Zitierweise, 2. Aufl., Boston: Harvard's Eleven Publications.
- [99] Autor (2008): Name des Buches, 22. Aufl., Berlin: Foxtrott.