



Decisio

Studienarbeit

Abteilung Informatik

Hochschule für Technik Rapperswil

Herbstsemester 2014

Autoren: Raphael Horber, Sascha Kaufmann
Betreuer: Prof. Dr. Eduard Glatz
Projektpartner: Forventis GmbH Zürich, Panter AG Zürich



Aufgabenstellung zur Studienarbeit HS 2014

„Decisio - App für die optimale Entscheidungsfindung“

Gruppe: Horber, Raphael / Kaufmann, Sascha

Kurzbeschreibung

Entscheidungen sind in der heutigen Multi-Options Gesellschaft schwerer zu treffen als in vergangenen Zeiten. Unter anderem müssen immer mehr Entscheidungen durch Einzelpersonen getroffen werden, da niemand anderes dies übernimmt. Die zu entwickelnde App soll Entscheidungswilligen, jedoch überforderten Usern helfen, richtige Entscheidungen zu treffen, indem sie Handlungsempfehlungen abgibt. Richtig, in unserem Sinne, ist eine Entscheidung dann, wenn sie mit den eigenen Motiven und Motivationen in Einklang steht. Die App vergleicht deshalb eine konkrete Entscheidungssituation mit der persönlichen Motivlage des Users.

Hintergrund

Die App basiert auf der Vermutung, dass eine Entscheidung umso besser ist, je näher sie am Motivprofil eines Entscheiders liegt. Das Motivprofil basiert auf dem *Zürcher Modell der sozialen Motivation von Norbert Bischof [1]* zu dem ein validierter Fragebogen vorliegt. Dieser wird von der App zur Erstellung des Motivprofils verwendet. Der Vergleich des Motivprofils mit einer Entscheidungssituation erfolgt über eine Rating-Liste. Die Handlungsempfehlungen basieren auf dem *Forventis durchBlick ®-Prinzip [2]*, welches eine allgemeine Entscheidungsregel verkörpert. Diese Regel wird durch Werte aus der Rating-Liste modifiziert und so individuell auf den User angepasst.

Mögliche Arbeitspakete

Motivprofil

Die App soll einen Fragebogen enthalten, mit dessen Hilfe sich die User ein persönliches Motivprofil erstellen können.

Entscheidungs-Presets

Die App soll eine umfassende Auswahl an möglichen Entscheidungssituationen enthalten. Dies sind z.B. Entscheidungen zu Fragen, wie „Soll ich meinen Partner heiraten?“, „Soll ich ein Studium beginnen oder eine Stelle annehmen?“ oder „Soll ich einen Rennbolide oder eine Familienkarosse kaufen?“. Zu den Entscheidungssituationen sind relevante Einflussfaktoren bereits aufgeführt, was die Vollständigkeit der Entscheidungsgrundlagen verbessern soll.

Entscheidungs-Personalisierung

Die Entscheidungs-Presets sollen durch den User beliebig ergänzt werden können. Dafür soll er zum Preset aus einer nach Domänen gruppierten Auswahl an möglichen Entscheidungsfaktoren (z.B. Finanzen, Gefühle, Werte, Mobilität, etc.) auswählen und passende Faktoren dem Preset hinzufügen können. Es soll zudem die Möglichkeit bestehen, eigene Entscheidungssituationen und Entscheidungsfaktoren hinzuzufügen. Eigene Entscheidungssituationen und Entscheidungsfaktoren sollen an das System weitergereicht werden, damit sich die Sammlung ausbauen lässt.

Einschätzung

Die Entscheidungsfaktoren sollen jetzt auf ihren Erfüllungsgrad eingeschätzt werden. Aufgrund dieser Einschätzung nimmt die App nun den Vergleich mit dem Motiv-Profil des Users vor.

Output

Darauf basierend kommuniziert die App eine Situationseinschätzung und gibt eine klare Handlungsempfehlung ab, z.B. „Die Situation ist eindeutig. Heiraten Sie Ihren Partner – jetzt oder nie“... oder so ähnlich.

Feedback

Der User soll nach der Empfehlung beurteilen, ob die Empfehlung für ihn stimmig war. Diese Information soll ins System zurückfliessen, wissenschaftlich ausgewertet werden können und der Verbesserung des Algorithmus dienen.

Der umzusetzende Funktionsumfang kann - falls gewünscht - bei Projektstart abschliessend in Diskussion mit dem Praxispartner definiert werden. Es gibt zurzeit keine Vorgaben betreffend die einzusetzenden Technologien. Das Frontend kann beispielsweise auch für iOS, Android oder Windows Phone umgesetzt werden.

Allgemeine Vorgaben

Die Arbeit ist gemäss den allgemeinen Vorgaben [3] durchzuführen. Dies beinhaltet auch Vorgaben zur Berichtsgestaltung.

Termine

15. Sept. 2014	Arbeitsbeginn
19. Dez. 2014	Abgabe des Berichts an den Betreuer (bis 17:00)

Weitere Termine siehe Terminangaben auf dem HSR-Web (intern).

Betreuung

Betreuer: Prof. Dr. Eduard Glatz

Während der Durchführung der Arbeit findet nach Möglichkeit regelmässig jede Woche eine Besprechung mit dem Betreuer statt. Dazu werden entsprechende Termine bei Arbeitsbeginn festgelegt. Die Studierenden verfassen vor jeder Besprechung eine Traktandenliste, die spätestens

am Vortag dem Betreuer per Email zu senden ist. Jede Besprechung wird von den Studierenden in einem Protokoll dokumentiert.

Praxispartner

Forventis GmbH, Kontaktperson: Dr. med. Samuel Huber

Panter AG, Kontaktperson: Hr. Flavio Trolese

Referenzen

- [1] http://de.wikipedia.org/wiki/Z%C3%BCrcher_Modell_der_sozialen_Motivation
- [2] <http://www.forventis.ch/druechblick.html>
- [3] Glatz, E., „Vorgaben zur Arbeitsdurchführung“, Ausgabe des 11. September 2014.

Rapperswil, den 11. Sept. 2014

A handwritten signature in black ink, appearing to be 'E. Glatz', written in a cursive style.

Eduard Glatz

Abstract

Heutzutage sind Entscheidungen schwerer zu treffen als früher. Decisio soll unentschlossenen Einzelpersonen helfen, zu ihnen passende Entscheidungen zu fällen. Der persönliche Aspekt für die Handlungsempfehlung basiert auf dem Zürcher Modell der sozialen Motivation von Norbert Bischof [1]. Dafür konnte ein Fragebogen zum Erstellen des Motivprofils des Benutzers verwendet werden, welchen uns die Praxispartner zur Verfügung stellten.

Im Vorfeld unserer Arbeit haben die Praxispartner bereits ein Design mittels Vorstudie an der ZHAW evaluiert [2]. Aufgrund dieses Designs galt es eine Android App zu entwickeln.

Zudem sollten möglichst viele Daten (Entscheidungssituationen, Faktoren, Fragebogen, usw.) dynamisch gehalten werden, d.h. von einem SFTP-Server im XML-Format heruntergeladen, anschliessend geparkt und die geparkten Daten in einer lokalen Datenbank abgelegt werden. Dadurch können diese Daten aktualisiert werden, ohne Updates der App zu veröffentlichen.

Den Algorithmus zur Berechnung der Handlungsempfehlung konnten wir mit den Praxispartnern besprechen und anschliessend in Code umsetzen. Dafür mussten wiederum vom SFTP-Server geladene Daten verwendet werden.

Unser Systemtest hat gezeigt, dass die App noch einen Fehler in der Synchronisierung der Daten vom SFTP-Server aufweist. Zudem haben einige Personen, welchen die App von den Praxispartnern gezeigt wurden, das UI der App bemängelt. Daher ist Sie für eine Markteinführung noch nicht ausgereift genug, stellt aber eine gute Basis für eine Folgearbeit dar.

Management Summary

Ausgangslage

Heutzutage sind Entscheidungen schwerer zu treffen als früher. Decisio soll unentschlossenen Einzelpersonen helfen, die zu ihnen passenden Entscheidungen zu fällen.

Eine Entscheidung soll mit den Motivationen des Benutzers übereinstimmen, deshalb kann eine Handlungsempfehlung nicht generell abgegeben werden, sondern soll persönlich berechnet werden. Um dies zu ermöglichen, muss der Benutzer zu Beginn einen Fragebogen ausfüllen, welchen uns die Praxispartner zur Verfügung stellten. Jede Frage darin zielt auf einen bestimmten Zweck ab und muss zwischen 1 (nicht zutreffend) bis 5 (voll zutreffend) eingestuft werden. Daraus soll die App das Motivationsprofil des Benutzers, basierend auf dem Zürcher Modell der sozialen Motivation von Norbert Bischof [1], berechnen.

Im Vorfeld unserer Arbeit haben die Praxispartner bereits ein Design für die App mittels Vorstudie an der ZHAW [2] evaluiert. Dieses Design sollte in der Android App verwendet werden.

Zudem hatten die Praxispartner bereits ein Vorgehen zur Berechnung der Handlungsempfehlung, mit Berücksichtigung des Motivationsprofils, bestimmt. Dieses Vorgehen sollte in der App umgesetzt werden.

Vorgehen

Das Design der Vorstudie wurde im Material Design [3] gestaltet. Dieses Design konnte aber, bei der Planung des Vorgehens, noch nicht für die Entwicklung von Android Apps verwendet werden. Zudem gab es noch keine Android Geräte mit der entsprechenden Android Version, die dieses Design unterstützten. Deshalb versuchten wir mit dem älteren Holo Design [4], das Design der Vorstudie so ähnlich wie möglich nachzubilden.

Im Motivationsprofil Fragebogen sind mehrere Fragen für jedes der 5 Motive (Sicherheit, Unternehmungslust, Macht, Geltung und Leistung) enthalten. Um daraus das Motivationsprofil des Benutzers zu erstellen, wird der Durchschnitt der Antworten zum jeweiligen Motiv berechnet. Die erhaltenen Werte können mit dem Durchschnittswert des Motivs verglichen werden, um dem Benutzer eine Angabe zur verhältnismässigen Einstufung anzugeben.

Eine Entscheidungssituation besteht aus einer Frage, zu welcher der Benutzer eine Antwort wünscht, und diversen Faktoren, die sich auf diese Entscheidung auswirken. Die Faktoren bestimmen, wie stark die einzelnen Motivationen des Benutzers ins Gewicht der Entscheidung fallen.

Führt der Benutzer eine Entscheidungssituation durch, wird ihm ein Faktor nach dem anderen zur Einstufung angezeigt. Dabei muss er angeben, wie gut er mit dem Faktor umgehen kann. Hat er alle Faktoren bewertet, gibt ihm die App eine auf ihn zugeschnittene Handlungsempfehlung an. Diese kann positiv, positiv mit Warnung oder negativ sein, wie bei einem Ampelsystem.

Die verwendeten Daten in der App, wie z.B. die Entscheidungssituationen, sollten dynamisch gehalten werden, damit die Praxispartner diese mit minimalem Aufwand ändern können. Um dies zu gewährleisten, werden die Daten beim Start der App synchronisiert, d.h. es werden bei Bedarf mehrere XML-Dateien von einem SFTP-Server heruntergeladen und die darin enthaltenen Daten in einer lokalen Datenbank abgelegt.

Ergebnisse

Die entwickelte App weist noch einen Fehler sowie in der Synchronisierung der Daten vom Server als auch einen kleineren, ebenfalls im Systemtest gefundenen, Usability Fehler auf. Ausserdem konnten nicht alle Anforderungen mit Priorität 2 umgesetzt werden. Dies betrifft das Erstellen von neuen Entscheidungssituationen und Faktoren von Benutzern, d.h. sie können nur vorgegebene Entscheidungssituationen vom Server ausführen.

Die Benutzeroberfläche wurde von einigen Personen, welchen die App von den Praxispartnern gezeigt wurde, bemängelt, da diese noch sehr schlicht gehalten wurde.

Ausblick

Die App ist für eine Markteinführung noch nicht ausgereift und ansprechend genug, stellt aber eine gute Basis für eine Folgearbeit dar.

Inhaltsverzeichnis

Aufgabenstellung	2
Abstract.....	5
Management Summary	6
Inhaltsverzeichnis	8
I. Technischer Bericht	10
<hr/>	
1. Einleitung.....	10
1.1 Übersicht	10
2. Anforderungs-Analyse	11
2.1 Allgemeine Beschreibung	11
2.2 Use Cases.....	12
2.3 Hintergrund-Aufgaben der Android App.....	17
2.4 Entscheidungsfaktor-Einschätzungen	18
2.5 Entscheidungsregel	19
2.6 Weitere Anforderungen	20
2.7 Change Requests	20
3. Domain-Analyse	22
3.1 Domain Modell	22
3.2 Systemsequenzdiagramme	25
4. Design und Umsetzung.....	28
4.1 Systemübersicht	28
4.2 Architektonische Ziele & Einschränkungen	28
4.3 Logische Architektur.....	30
4.4 Prozesse und Threads.....	46
4.5 Datenspeicherung	47
4.6 Datenaustausch-Format	48
4.7 Wireframes.....	52
4.8 Systemtest	56
4.9 Unit Tests.....	75
5. Schlussfolgerungen.....	76
5.1 Ergebnisse.....	76
5.2 Schlussfolgerungen.....	76
5.3 Ausblick.....	76
II. Anhang.....	77
<hr/>	
6. Build Anleitung	77

7.	Werkzeuge/Tools.....	78
8.	Glossar.....	79
9.	Abbildungsverzeichnis.....	80
10.	Tabellenverzeichnis.....	81
11.	Literaturverzeichnis.....	82
11.1	Weitere Literatur.....	83
11.2	Lizenzen.....	83

I. Technischer Bericht

1. Einleitung

Wir treffen täglich etliche Entscheidungen. Dies ist nicht immer leicht, da wir auch die Folgen der Entscheidung tragen müssen. Daher sollte die Entscheidung gut zu einem selbst passen.

Um die Beweggründe des Benutzers zu bestimmen, muss dieser zu Beginn einen Motivprofil-Fragebogen ausfüllen. Daraus kann der Durchschnitt für die 5 Motive (Sicherheit, Unternehmungslust, Macht, Geltung und Leistung [1]) des Benutzers berechnet werden.

Anschliessend kann der Benutzer Entscheidungssituationen durchführen und dabei die zur Entscheidung beitragenden Faktoren, anhand eines Ampelsystems, einstufen. Die entwickelte App berechnet danach anhand seiner Faktor-Einstufungen und seinem persönlichen Motivprofil eine Handlungsempfehlung. Diese kann positiv (mach es!), positiv mit Warnung (sei Vorsichtig) oder negativ (mach es nicht!) sein. Als Zusammenfassung werden die als Orange- und Rot-gewerteten Faktoren, zusammen mit einer Beschreibung der Auswirkung, aufgelistet.

Für die Praxispartner war nicht nur eine funktionierende App, sondern auch der Back-End-Bereich von Bedeutung. Damit wollten sie z.B. den Fragebogen zur Erstellung des Motivprofils validieren können und allenfalls neue Fragen darin testen. Dies sollte mittels Upload der angegebenen Antworten zum Fragebogen, sowie einigen anonymisierten Daten zum Benutzer erfolgen.

1.1 Übersicht

Anforderungs-Analyse	In diesem Kapitel sind die Anforderungen spezifiziert (grob sind dies die Use Cases und die NFRs).
Domain-Analyse	In diesem Kapitel wurde die Problemstellung analysiert.
Design und Umsetzung	Dieses Kapitel dokumentiert die Architektur der App und getroffene Entscheide, sowie Designentwürfe (z.B. Wireframes).
Schlussfolgerung	In diesem Kapitel sind die Ergebnisse unserer Arbeit, sowie daraus gezogene Schlussfolgerungen beschrieben.

2. Anforderungs-Analyse

2.1 Allgemeine Beschreibung

2.1.1 Produkt Perspektive

Die App soll auf allen Geräten ab Android 4.0.3 (API-Level 15) und höher funktionieren. Das Design der App soll dem der erhaltenen Vorarbeit der ZHAW [2] entsprechen.

Falls dies mit dem API-Level 15 nicht möglich ist, soll dieser erhöht werden, um das Design zu erreichen.

2.1.2 Produkt Funktion

Die folgende Liste enthält lediglich eine Übersicht der Funktionen des Produkts. Die genauen Funktionalitäten können aus den Use Cases entnommen werden.

- Android App
 - Der Benutzer kann mit einem Fragebogen ein persönliches Motivprofil erstellen.
 - Der Benutzer kann eine vorgegebene Frage auswählen und die zugehörigen Faktoren bewerten. Danach erhält er die Handlungsempfehlung als Antwort.
 - Zu den bestehenden Entscheidungssituationen sollen neue Entscheidungssituationen mit deren Faktoren erfasst werden können.
- Back-End: Die Daten werden in XML-Dateien abgelegt. Ein User-Interface dazu ist optional. Folgendes muss möglich sein (mit XML-Files):
 - Auflisten der Entscheidungssituationen, die von Benutzern erfasst wurden.
 - Ändern und hinzufügen der vordefinierten Entscheidungssituationen und Entscheidungsfaktoren. Dies beinhaltet auch die möglichen Erläuterungen und Gewichtungen der Entscheidungsfaktoren.
 - Ändern des Fragebogens, der das Motivprofil des Benutzers bestimmt.
 - Ändern der Erläuterungen zu der Einstufung des Benutzers beim Motivprofil.

2.1.3 Benutzer Charakteristik

- Die App ist für Personen ab 18 Jahren konzipiert. Somit können auch Entscheidungssituationen vorhanden sein, die Mündigkeit erfordern.
- Die Benutzer benötigen Grundkenntnisse, um ein Gerät mit Android zu bedienen.

2.1.4 Einschränkungen

Die Applikation wird keine Anleitung enthalten. Die Bedienung sollte soweit Android spezifisch sein, dass jeder Android-Benutzer die Applikation bedienen kann.

2.2 Use Cases

2.2.1 Use Case Diagramm

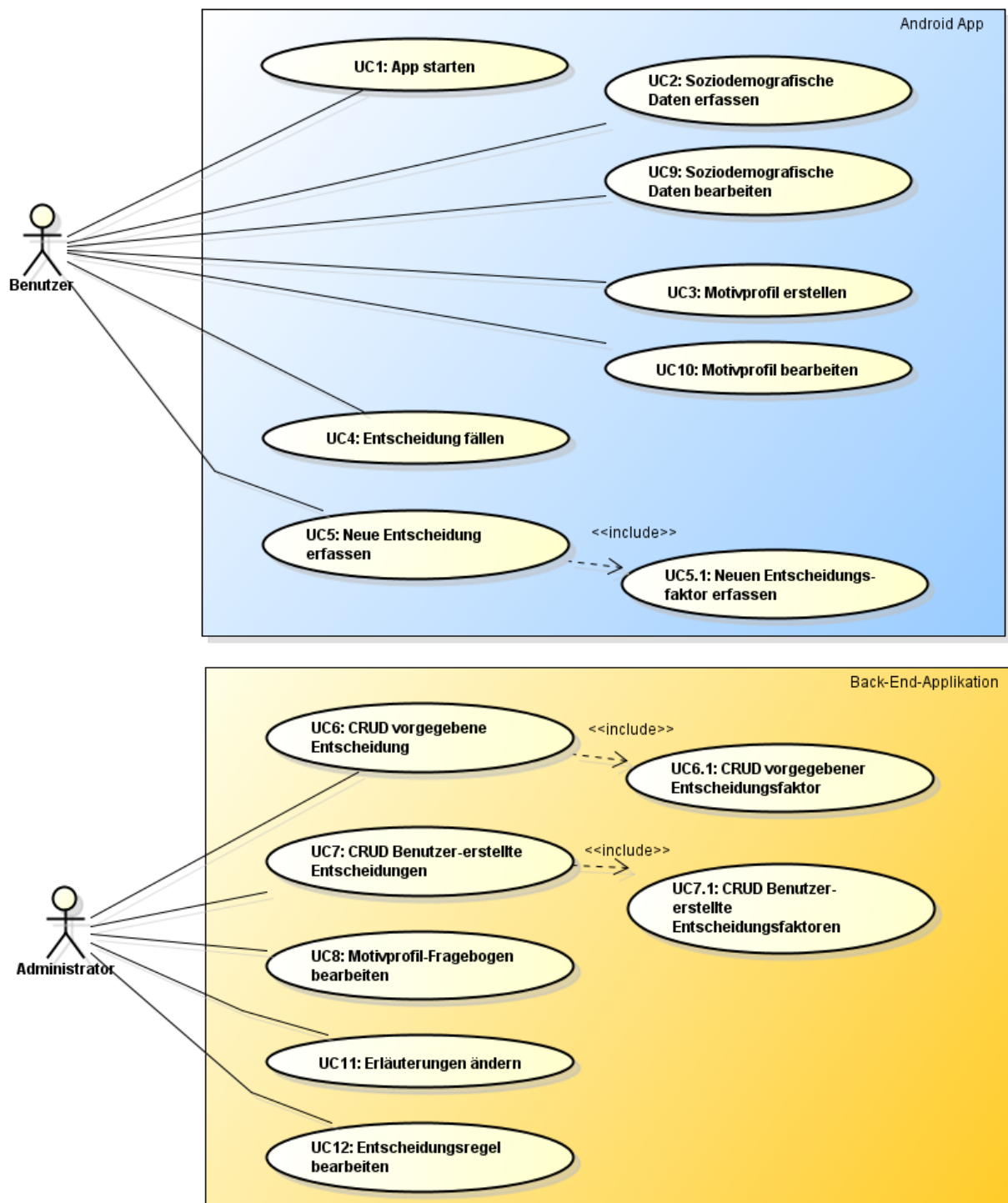


Abbildung 1: Use Case Diagramm

2.2.2 Aktoren & Stakeholder

- Benutzer: Benutzer der Android App
- Administrator: Benutzer der Back-End-Applikation

2.2.3 Beschreibungen (Brief)

Die Hintergrund-Aufgaben des Systems, welche durch einzelne Use Cases ausgelöst werden, sind im Kapitel "2.3 Hintergrund-Aufgaben der Android App" beschrieben.

2.2.3.1 UC1: App starten

Sobald der Benutzer die App startet, prüft das System, ob der Benutzer den Disclaimer Text bereits akzeptiert hat. Falls nicht, zeigt das System diesen Text an, welchen der Benutzer zur Verwendung der App akzeptieren muss.

Danach prüft das System, ob der Benutzer "UC2: Soziodemografische Daten erfassen" und "UC3: Motivprofil erstellen" schon ausgeführt hat. Wenn nicht, dann fordert das System den Benutzer dazu auf.

2.2.3.2 UC2: Soziodemografische Daten erfassen

Der Benutzer gibt seine soziodemografischen Daten ein. Dies beinhaltet Geschlecht, Jahrgang, Einkommensklasse und höchster Abschluss.

2.2.3.3 UC3: Motivprofil erstellen

Der Benutzer beantwortet die Fragen des Motivprofil-Fragebogens. Das System erstellt daraus sein Motivprofil und zeigt dies dem Benutzer an. Der Benutzer hat die Möglichkeit, den Fragebogen noch einmal auszuführen, falls er mit dem erstellten Motivprofil nicht zufrieden ist.

2.2.3.4 UC4: Entscheidung fällen

Der Benutzer hat die Möglichkeit, Entscheidungen aus vorhandenen (vorgegebene- und benutzer-erstellte-) Entscheidungssituationen, auszuwählen. Sobald der Benutzer alle zur Entscheidungssituation gehörenden Faktoren bewertet hat, gibt ihm das System eine klare Handlungsempfehlung.

2.2.3.5 UC5: Neue Entscheidung erfassen

Der Benutzer hat die Möglichkeit, neue Entscheidungssituationen zu erfassen. Dazu muss er der Entscheidungssituation einen Namen geben und die zur Entscheidungssituation massgebenden Entscheidungsfaktoren auswählen.

2.2.3.6 UC5.1: Neuen Entscheidungsfaktor erfassen

Der Benutzer hat die Möglichkeit, neue Entscheidungsfaktoren zu erfassen. Diese werden mit den Standardwerten (grün 0, orange 1, rot 3) für die Entscheidungsregel gewichtet (ohne Korrekturbetrag).

Die zwei möglichen Detail-Erläuterungen werden mit Standard-Werten gefüllt.

2.2.3.7 UC6: CRUD vorgegebene Entscheidung

Der Administrator hat die Möglichkeit neue zu erstellen, bestehende zu bearbeiten oder zu löschen. Dazu ändert er die Konfigurations-Dateien auf dem Back-End-Server gemäss der Dokumentation.

2.2.3.8 UC6.1: CRUD vorgegebener Entscheidungsfaktor

Der Administrator ändert die Konfigurations-Dateien anhand der Dokumentation, um die vorgegebenen Entscheidungsfaktoren zu bearbeiten. Ebenfalls kann das Mapping der Faktoren in den 5 Motiven zur Bestimmung des Korrekturbetrages (via Ratingtabelle) bearbeitet werden.

Der Administrator hat die Möglichkeit neue zu erstellen, bestehende zu bearbeiten oder zu löschen.

Pro Entscheidungsfaktor können die zwei möglichen Detail-Erklärungen (für Orange- und Rot-Wertungen) in der Datei angegeben werden, ansonsten werden Standard-Erläuterungen verwendet.

2.2.3.9 UC7: CRUD Benutzer-erstellte Entscheidungen

Der Administrator hat die Möglichkeit, alle von Benutzern erstellten Entscheidungssituationen, anhand der Konfigurations-Dateien, einzusehen. Es wird pro Benutzer eine Datei angelegt.

2.2.3.10 UC7.1: CRUD Benutzer-erstellte Entscheidungsfaktoren

Der Administrator hat die Möglichkeit, alle von Benutzern erstellten Entscheidungsfaktoren, anhand der Konfigurations-Dateien, einzusehen. Es wird pro Benutzer eine Datei angelegt.

2.2.3.11 UC8: Motivprofil-Fragebogen bearbeiten

Der Administrator ändert die Konfigurations-Dateien anhand der Dokumentation, um die Fragen des Motivprofil-Fragebogens zu bearbeiten.

Der Administrator hat die Möglichkeit neue zu erstellen, bestehende zu bearbeiten oder zu löschen.

2.2.3.12 UC9: Soziodemografische Daten bearbeiten

Der Benutzer öffnet die Ansicht seiner soziodemografischen Daten. Er kann diese ändern und speichern.

2.2.3.13 UC10: Motivprofil bearbeiten

Der Benutzer öffnet die Ansicht seines Motivprofils. Er will dieses bearbeiten. Dazu kann er einzelne Antworten auf die Fragen im Motivprofil-Fragebogen ändern und die Änderungen speichern.

2.2.3.14 UC11: Erläuterungen ändern

Der Administrator ändert die Konfigurations-Dateien anhand der Dokumentation, um die Erläuterungen zu der Einstufung des Motivprofils zu bearbeiten. Die Erläuterungen werden zu den Bereichen "unter dem Durchschnitt", "Durchschnittlich" und "über dem Durchschnitt" definiert, welche mit Werten für die Bereichsgrenzen getrennt werden.

2.2.3.15 UC12: Entscheidungsregel bearbeiten

Der Administrator hat die Möglichkeit, die Ratingtabelle für die Bestimmung der Korrekturbeträge zu ändern. Dazu ändert er die entsprechende Konfigurations-Datei auf dem Back-End-Server anhand der Dokumentation. Zudem hat er die Möglichkeit, die Schwellwerte der Entscheidungsregel über Konfigurationen in der Applikation zu verändern. Dabei muss aber eine neue Version der App publiziert werden. Siehe auch Gewichtung der Entscheidungsfaktoren für die "Dreier-Regel".

2.2.4 Beschreibungen (Fully Dressed)

2.2.4.1 UC1: App starten

Use Case Name: UC1: App starten

Primary Actor: Benutzer

Main Success Scenario:

1. Benutzer startet die App.
2. App stellt fest, dass Benutzer den Disclaimer Text bereits einmal akzeptiert hat.
3. App stellt fest, dass Benutzer die soziodemografischen Daten bereits erfasst hat.
4. App stellt fest, dass der Benutzer das Motivprofil bereits erfasst hat.
5. Benutzer kann mit "UC4: Entscheidung fällen" oder "UC5: Neue Entscheidung erfassen" fortfahren.

Extensions (or Alternative Flows):

- 2a. Benutzer hat den Disclaimer Text noch nie akzeptiert:
 1. System zeigt den Disclaimer Text in einem Dialog an, den der Benutzer akzeptieren muss.
 - 2a. Benutzer akzeptiert den Disclaimer Text → weiter bei 3. im Main Success Scenario.
 - 2b. Benutzer akzeptiert den Disclaimer Text nicht → System beendet die App.
- 3a. Benutzer hat die soziodemografischen Daten noch nicht erfasst:
 1. System prüft, ob eine Internet-Verbindung besteht.
 - 2a. Internet-Verbindung besteht: System fordert Benutzer zu "UC2: Soziodemografische Daten erfassen" auf. Sobald die Daten eingegeben sind → weiter bei 4. im Main Success Scenario.
 - 2b. Keine Internet-Verbindung: System zeigt Informationsmeldung an und schliesst die App.
- 4a. Benutzer hat noch kein Motivprofil erfasst:
 1. System fordert Benutzer zu "UC3: Motivprofil erstellen" auf. Sobald erstellt → weiter bei 5. im Main Success Scenario.

2.2.4.2 UC3: Motivprofil erstellen

Use Case Name: UC3: Motivprofil erstellen

Primary Actor: Benutzer

Main Success Scenario:

1. System zeigt den Fragebogen zum Erstellen des Motivprofils an.
2. Benutzer gewichtet die gestellten Fragen jeweils von 1 bis 5.
3. Benutzer hat alle gestellten Fragen gewichtet und will Motivprofil erstellen.
4. System speichert das Motivprofil.
5. System erstellt Motivprofil und zeigt Übersicht mit verhältnismässiger Einstufung an.

2.2.4.3 UC4: Entscheidung fällen

Use Case Name: UC4: Entscheidung fällen

Primary Actor: Benutzer

Main Success Scenario:

1. System zeigt den Katalog der vorhandenen (vorgegebene- und benutzer-erstelte-) Entscheidungssituationen an.
2. Benutzer wählt eine Entscheidungssituation aus.
3. System führt den Benutzer durch die zur Entscheidungssituation gehörenden Entscheidungsfaktoren. Die Reihenfolge entspricht der Reihenfolge in der Konfigurationsdatei.
4. Benutzer gibt seine Einschätzung zu den Entscheidungsfaktoren, gemäss Kapitel "2.4 Entscheidungsfaktor-Einschätzungen" an.
5. Hat der Benutzer alle Faktoren bewertet, rechnet das System den Korrekturfaktor für die "Dreier-Regel" aus (siehe Kapitel "2.5 Entscheidungsregel").
6. System zeigt anhand der "Dreier-Regel" eine konkrete Handlungsempfehlung und seine Bewertungen (der Faktoren zu dieser Entscheidungssituation) zusammengefasst an.

2.2.4.4 UC5: Neue Entscheidung erfassen

Use Case Name: UC5: Neue Entscheidung erfassen

Primary Actor: Benutzer

Main Success Scenario:

1. Benutzer gibt den Namen der Entscheidungssituation ein.
2. System zeigt die zur Auswahl stehenden Faktoren mit Hilfe von Auto-Vervollständigung an.
3. Benutzer wählt die für die Entscheidungssituation ausschlaggebenden Faktoren aus.
4. Benutzer will Entscheidungssituation erfassen.
5. System speichert die neu erfasste Entscheidungssituation.

Extensions (or Alternative Flows):

- 3a. Optional kann der Benutzer auch noch via "UC5.1: Neuen Entscheidungsfaktor erfassen" neue Faktoren hinzufügen.

2.2.4.5 UC5.1: Neuen Entscheidungsfaktor erfassen

Use Case Name: UC5.1: Neuen Entscheidungsfaktor erfassen

Primary Actor: Benutzer

Main Success Scenario:

1. Benutzer gibt den Namen des Faktors ein.
2. Das System trägt zu den 5 Motiven die Standard-Korrekturbeträge von 0 ein.
3. Benutzer will Entscheidungsfaktor erfassen.
4. System speichert den neu erfassten Entscheidungsfaktor, inkl. den drei möglichen Detail-Erläuterungen.

2.2.5 Priorisierung der Use Cases

2.2.5.1 Priorität 1

- UC1: App starten
- UC2: Soziodemografische Daten erfassen
- UC3: Motivprofil erstellen
- UC4: Entscheidung fällen
- UC6: CRUD vorgegebene Entscheidung
- UC6.1: CRUD vorgegebener Entscheidungsfaktor
- UC8: Motivprofil-Fragebogen bearbeiten
- UC11: Erläuterungen ändern
- UC12: Entscheidungsregel bearbeiten

2.2.5.2 Priorität 2 (optional)

- UC5: Neue Entscheidung erfassen
- UC5.1: Neuen Entscheidungsfaktor erfassen
- UC7: CRUD Benutzer-erstellte Entscheidungen
- UC7.1: CRUD Benutzer-erstellte Entscheidungsfaktoren
- UC9: Soziodemografische Daten bearbeiten
- UC10: Motivprofil bearbeiten

2.3 Hintergrund-Aufgaben der Android App

Vor den unten aufgeführten Aufgaben, prüft das System jeweils, ob eine Internet-Verbindung besteht. Besteht keine Verbindung, werden die Aufgaben nicht ausgeführt.

Die Aufgaben werden jeweils durch den entsprechenden Use Case ausgelöst.

2.3.1 UC1: App starten

Beim Start lädt das System die neuesten Einstellungen vom Server herunter. Dies beinhaltet die vorgegebenen Entscheidungssituationen und deren Entscheidungsfaktoren inkl. deren Gewichtungen, sowie der Ratingtabelle zur Bestimmung der Korrekturbeträge und dem aktuellsten Fragebogen zum Erstellen des Motivprofils.

2.3.2 UC3: Motivprofil erstellen

Sobald der Benutzer den Motivprofil-Fragebogen ausgefüllt und das Motivprofil gespeichert hat, sendet das System die Antworten des Benutzers auf die gestellten Fragen zum Back-End-Server.

2.3.3 UC5: Neue Entscheidung erfassen

Ist die Entscheidungssituation vollständig erfasst, sendet das System die erfasste Entscheidungssituation und die dazugehörigen Entscheidungsfaktoren an den Server.

2.3.4 UC5.1: Neuen Entscheidungsfaktor erfassen

Ist der Entscheidungsfaktor vollständig erfasst, sendet das System den erfassten Entscheidungsfaktor an den Server.

2.3.5 UC9: Soziodemografische Daten bearbeiten

Speichert der Benutzer seine soziodemografischen Daten, sendet das System die neuen Daten an den Server.

2.4 Entscheidungsfaktor-Einschätzungen

Die Entscheidungsfaktoren können vom Benutzer wie folgt bewertet werden:

- Kein Problem (**Grün**): Der Benutzer kann mit dem Entscheidungsfaktor ohne Probleme umgehen.
- Nicht Ideal (**Orange**): Der Benutzer kann mit dem Entscheidungsfaktor bedingt umgehen.
- Problem (**Rot**): Der Benutzer kann mit dem Entscheidungsfaktor nicht umgehen.
- Nicht Relevant: Der Benutzer findet, dass dieser Entscheidungsfaktor sich nicht auf die Entscheidung auswirkt und daher keinen Einfluss auf den Entscheidungsvorschlag haben soll.

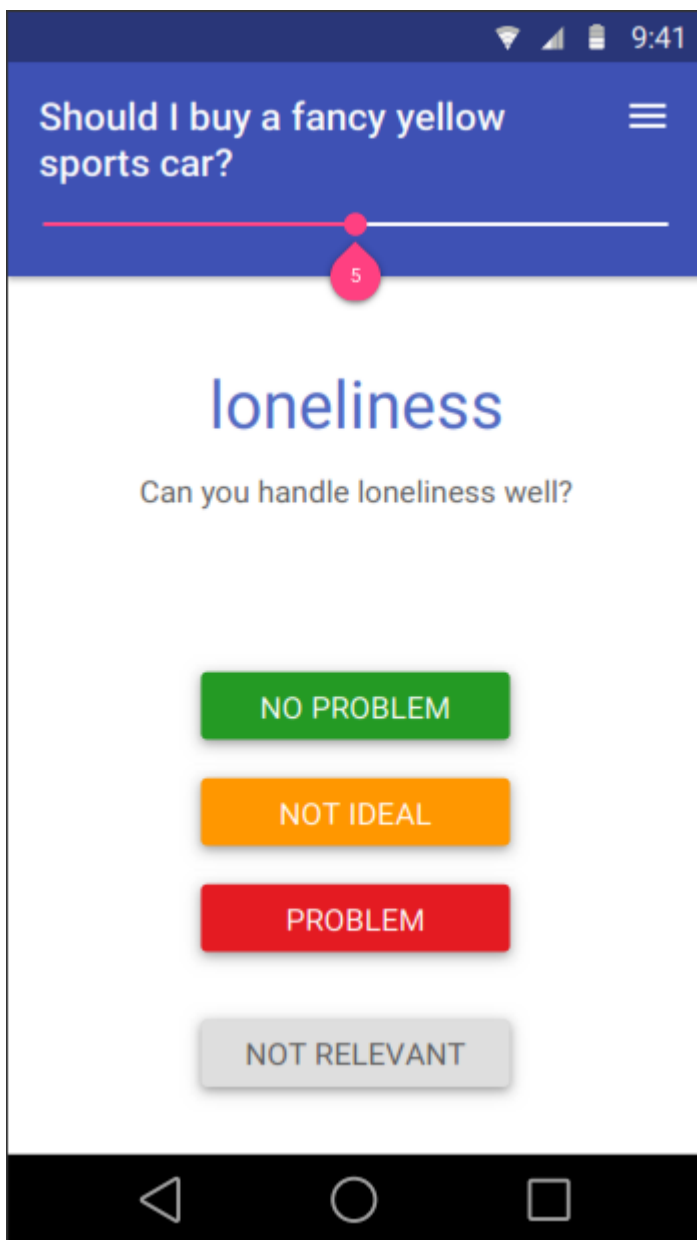


Abbildung 2: Beispiel Abbildung aus der Vorarbeit, der ZHAW Studenten [2]

2.5 Entscheidungsregel

2.5.1 "Dreier-Regel"

Die "Dreier-Regel" ist eine Art, wie die Benutzereinschätzungen der Entscheidungsfaktoren verarbeitet werden.

Vereinfacht ausgedrückt wird wie folgt vorgegangen:

1. Vom Benutzer als "Nicht Relevant" eingeschätzte Entscheidungsfaktoren werden nicht berücksichtigt.
2. Alle relevanten Einschätzungen der Entscheidungsfaktoren des Benutzers sind Grün → Handlungsempfehlung ist **positiv**.
3. Ein Entscheidungsfaktor wurde vom Benutzer als Orange eingestuft → Handlungsempfehlung ist trotzdem **positiv**.
4. Zwei Entscheidungsfaktoren wurden vom Benutzer als Orange eingestuft → Handlungsempfehlung ist **positiv**, jedoch **mit einer Warnung**.
5. Drei oder mehr Entscheidungsfaktoren wurden vom Benutzer als Orange eingestuft → Entscheidung ist negativ.
6. Ein oder mehr Entscheidungsfaktoren wurden vom Benutzer als Rot eingestuft → Handlungsempfehlung ist negativ.

Das System zeigt dem Benutzer die berechnete Handlungsempfehlung (positiv, positiv mit Warnung oder negativ) an.

2.5.2 Gewichtung der Entscheidungsfaktoren für die "Dreier-Regel"

Im obigen Kapitel 2.5.1 "Dreier-Regel" wurde die Entscheidungsregel vereinfacht beschrieben. Diese Entscheidungsregel arbeitet jedoch eigentlich mit gewichteten Werten der Entscheidungsfaktoren.

Es wird wie folgt vorgegangen:

1. Für jeden Entscheidungsfaktor, der als Orange eingestuft wurde, wird ein Korrekturbetrag pro Motiv aus der Ratingtabelle ausgelesen. Diese werden summiert und zu 1 addiert und stellen die Gewichtung des Entscheidungsfaktors dar.
2. Diese effektiven Gewichtungen pro Entscheidungsfaktor werden zusammengezählt.
3. Ist der Wert kleiner als 1.5 ist die Handlungsempfehlung positiv. Liegt der Wert zwischen 1.5 und 2.4 ist die Handlungsempfehlung positiv mit Warnung. Ist der Wert 2.5 oder höher ist die Handlungsempfehlung negativ.

Die oben genannten Schwellwerte können in „UC12: Entscheidungsregel bearbeiten“ geändert werden.

Faktor-Mapping	Profil-Wert	Korrektur-Betrag
0	1	0
0	2	0
0	3	0
0	4	0
0	5	0
1	1	-0.4
1	2	-0.3
1	3	0
1	4	0.1
1	5	0.3
2	1	-0.3

Abbildung 3: Auszug der Ratingtabelle

2.6 Weitere Anforderungen

2.6.1 Qualitätsmerkmale

2.6.1.1 Benutzbarkeit

- Die Benutzeroberfläche der Applikation soll in Englisch geschrieben werden. Die App soll in Zukunft auch in Deutsch und Französisch verfügbar sein.
- Das Design der ZHAW-Vorarbeit [2] soll möglichst ähnlich umgesetzt werden.

2.6.1.2 Zuverlässigkeit

- Die Software sollte soweit ausgereift sein, dass sie bei einem Verbindungsunterbruch zum Server nicht abstürzt.

2.6.1.3 Übertragbarkeit

- Die Android App soll auf jedem Gerät mit Android-Version 4.0.3 oder höher lauffähig sein.
- Die Verbreitung der Android App im Google Play Store wird den Praxispartnern überlassen.

2.6.1.4 Änderbarkeit

- Das Projekt enthält eine Schichtenarchitektur, sodass es leicht ist Mängel zu beheben.

2.6.2 Schnittstellen

- Die Android App verwendet die Geräte-Netzwerkschnittstelle, um mit dem Server zu kommunizieren.
- Der Benutzer kann die Applikation via Touchscreen bedienen. Die Verwendung von zusätzlichen Eingabegeräten wird nicht unterstützt.

2.6.3 Randbedingungen

- Decisio wird in Java implementiert.
- Als Entwicklungsumgebung wird Android Studio verwendet.
- Der Back-End-Server bietet via SFTP Zugriff auf die XML-Dateien. Der Server wird von den Praxispartnern zur Verfügung gestellt.

2.7 Change Requests

Tabelle 1 soll die von der ursprünglichen Anforderungsspezifikation abweichenden Änderungswünsche aller Beteiligten dokumentieren.

Die Spalte Aufwand stellt die **Schätzung des zusätzlichen Aufwands** zum Eingangsdatum dar. Der Status "**REQ**" heisst, dass der Change Request in die Anforderungen übernommen wurde. "**NIMP**" (not implemented) wurde nicht umgesetzt.

ID	Beschreibung	Antragssteller	Eingang	Status	Aufwand
CR01	Anstelle der Suche eines Faktors über die Domäne, soll eine Liste mit Auto-Vervollständigung verwendet werden.	Samuel Huber	16.10.2014	REQ	8h

ID	Beschreibung	Antragssteller	Eingang	Status	Aufwand
CR02	Anstelle von spezifischen Antworten zu den Entscheidungssituationen, sollen spezifische Detail-Erläuterungen zu den Orange/Rot gewerteten Faktoren erfasst werden.	Samuel Huber	16.10.2014	REQ	2h
CR03	Die App soll in Zukunft nicht nur in Englisch, sondern auch in Deutsch und Französisch verfügbar sein.	Samuel Huber	17.10.2014	REQ	4h
CR04	Anstelle von Durchschnittswerten sollen die Bereichsgrenzen für den Durchschnitt vom Server heruntergeladen werden.	Samuel Huber	24.10.2014	REQ	2h
CR05	Wertung "Nicht Relevant": Soll nicht als "Grün" gewertet werden, sondern als eigene Option (mit eigener Gewichtung) in die Entscheidung eingehen.	Samuel Huber	30.10.2014	REQ	2h
CR06	Qualitätssicherung mit Frage an den Benutzer: Damit keine Mülldaten eingegeben und übermittelt werden.	Samuel Huber	30.10.2014	NIMP	4h
CR07	Motivprofil-Fragebogen: Antworten des Benutzers sollen an Back-End-Server gesendet werden.	Samuel Huber	30.10.2014	REQ	16h
CR08	Disclaimer: Text der beim Applikationsstart angezeigt und akzeptiert werden soll.	Samuel Huber	30.10.2014	REQ	6h
CR09	Anstelle von FTP sollen die dynamischen Daten via Dropbox synchronisiert werden.	Flavio Trolese	30.10.2014	--	--
CR10	CR09 widerrufen: Die dynamischen Daten sollen via SFTP synchronisiert werden.	Flavio Trolese	04.11.2014	REQ	12h
CR11	Motivprofil-Fragebogen: Soll beim Editieren die früher ausgefüllten Werte anzeigen (nice to have).	Samuel Huber	20.11.2014	REQ	2h
CR12	Entscheidung durchführen: Die Reihenfolge der Faktoren soll der Reihenfolge in der Konfigurationsdatei entsprechen (nice to have).	Samuel Huber	20.11.2014	REQ	2h

Tabelle 1: Change Requests aller Beteiligten

3. Domain-Analyse

3.1 Domain Modell

Der Android Lifecycle bringt einige Schwierigkeiten in der Datenhaltung in der Domain mit sich. Die selben Domain Objekte können nur über Umwege in mehreren Activities verwendet werden. Eine Variante wäre, die Objekte als Serializable oder Parcelable zu definieren. Dadurch können sie in einem Intent zum Starten einer Activity übergeben werden.

Eine andere Option wäre der Einsatz eines gebundenen Service. Dieser kann die Objekte auch nur am Leben halten, solange mindestens eine Activity noch nicht beendet wurde.

Zudem macht es innerhalb einer Activity keinen Sinn, die Liste aller Entscheidungen und deren Faktoren aus der Datenbank in die Domain zu laden. Diese müssten erneut aus der Datenbank gelesen werden, wenn z.B. das Gerät gedreht und dadurch das Layout der Activity neu erstellt werden muss.

Daher soll die Android App nur die momentan wichtigen Informationen, zum erfolgreichen Ausführen der jeweiligen Use Cases in der Domain halten. Bei Bedarf lädt sie die fehlenden Daten aus der Datenbank.

Für die Übersichtsliste aller vorhandenen Entscheidungssituationen genügen einige Headerinformationen pro Eintrag. Will der Benutzer eine Entscheidung fällen, dann kann die komplette Entscheidungssituation mit all ihren Faktoren aus der Datenbank geladen werden.

3.1.1 Strukturdiagramm

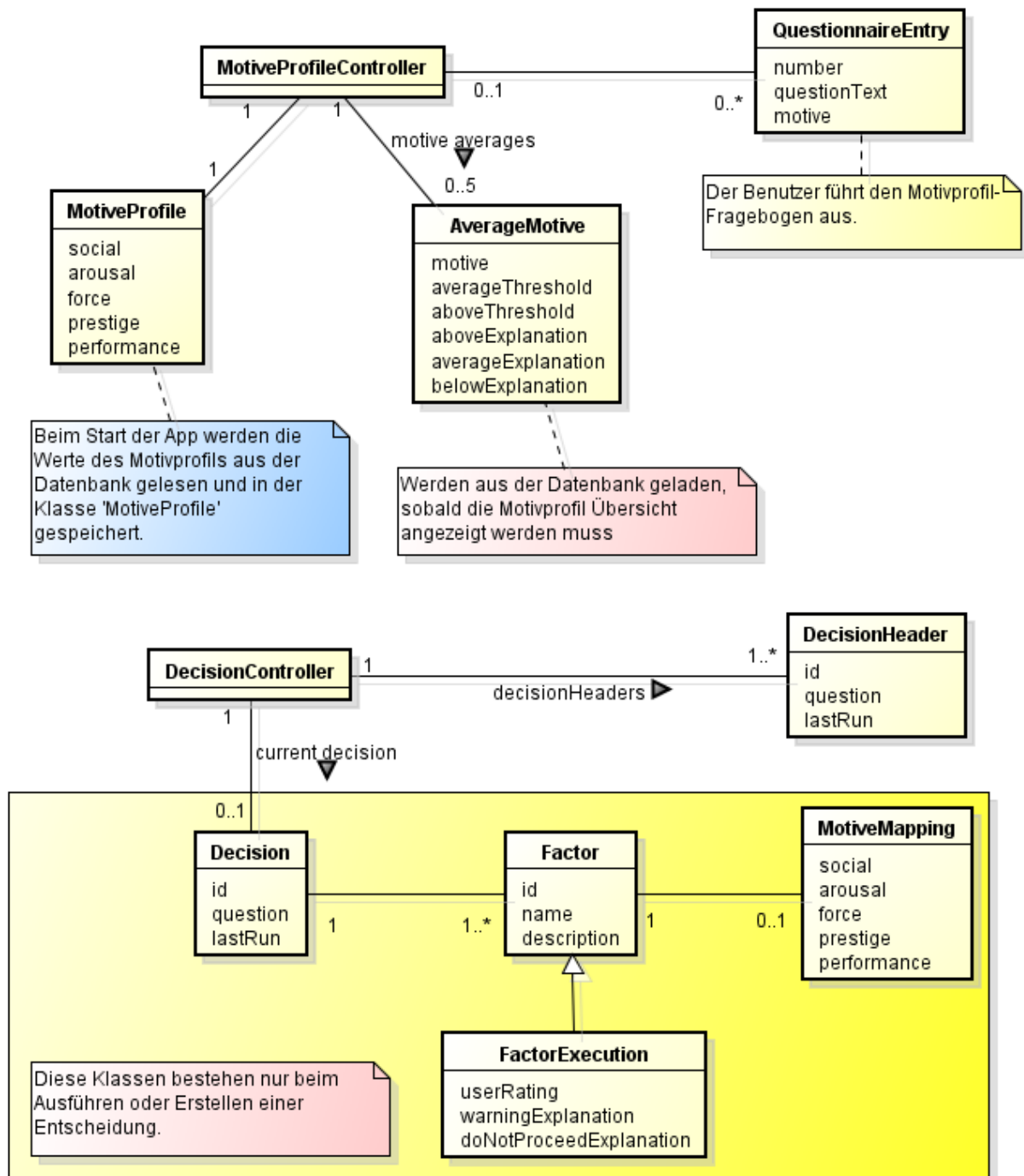


Abbildung 4: Domain Modell

3.1.1.1 Erklärungen

Die Domainobjekte haben wir, anhand der Use Cases, thematisch in zwei Gruppen aufgeteilt:

- **Entscheidungen:** Alle Domainobjekte, die zum Ausführen einer Entscheidung oder zum Erstellen einer neuen Entscheidungssituation nötig sind, werden über die Klasse *DecisionController* angesprochen.
- **Motivprofil:** Alle Domainobjekte, die zum Erstellen des Motivprofils und zur Visualisierung des Motivprofils nötig sind, werden über die Klasse *MotiveProfileController* angesprochen.

3.1.2 Wichtige Konzepte

Factor	Stellt einen vorgegebenen oder vom Benutzer erstellten Faktor dar.
FactorExecution	Ein Faktor während der Ausführung einer Entscheidungssituation. Zusätzlich werden hier die Einstufung des Benutzers und die anzuzeigenden Detail-Erklärungen gespeichert.
MotiveMapping	Definiert zu jedem Motiv den Wert "Faktor-Mapping" für die Ratingtabelle.
DecisionHeader	Die Headerinformationen der Entscheidungssituationen. Dies wird bei der Auflistung der ausführbaren Entscheidungssituationen verwendet.
MotiveProfile	Enthält die Motivprofil-Werte des Benutzers, welche aufgrund seiner Antworten zum Motivprofil-Fragebogen berechnet wurden.
AverageMotive	Stellt die Durchschnitts-Motivprofile dar. Pro Motiv gibt es ein Objekt mit den Schwellwerten des Durchschnittsbereichs und den Erklärungen zu den Rangierungen.
QuestionnaireEntry	Repräsentiert eine Frage des Motivprofil-Fragebogens.

3.2 Systemsequenzdiagramme

3.2.1 UC1: App starten

Wie in Abbildung 5 zu sehen ist, werden beim ersten Starten der App alle für die App benötigten Daten vom Back-End-Server heruntergeladen. Erst wenn alle Daten herunter geladen wurden, kann der Benutzer die App benutzen. Danach wird die Internet-Verbindung nicht mehr zwingend benötigt. Falls jedoch bei einem weiteren Starten der App eine Internet-Verbindung besteht, werden die lokalen Daten durch die neuesten verfügbaren Daten ersetzt (siehe Abbildung 6).

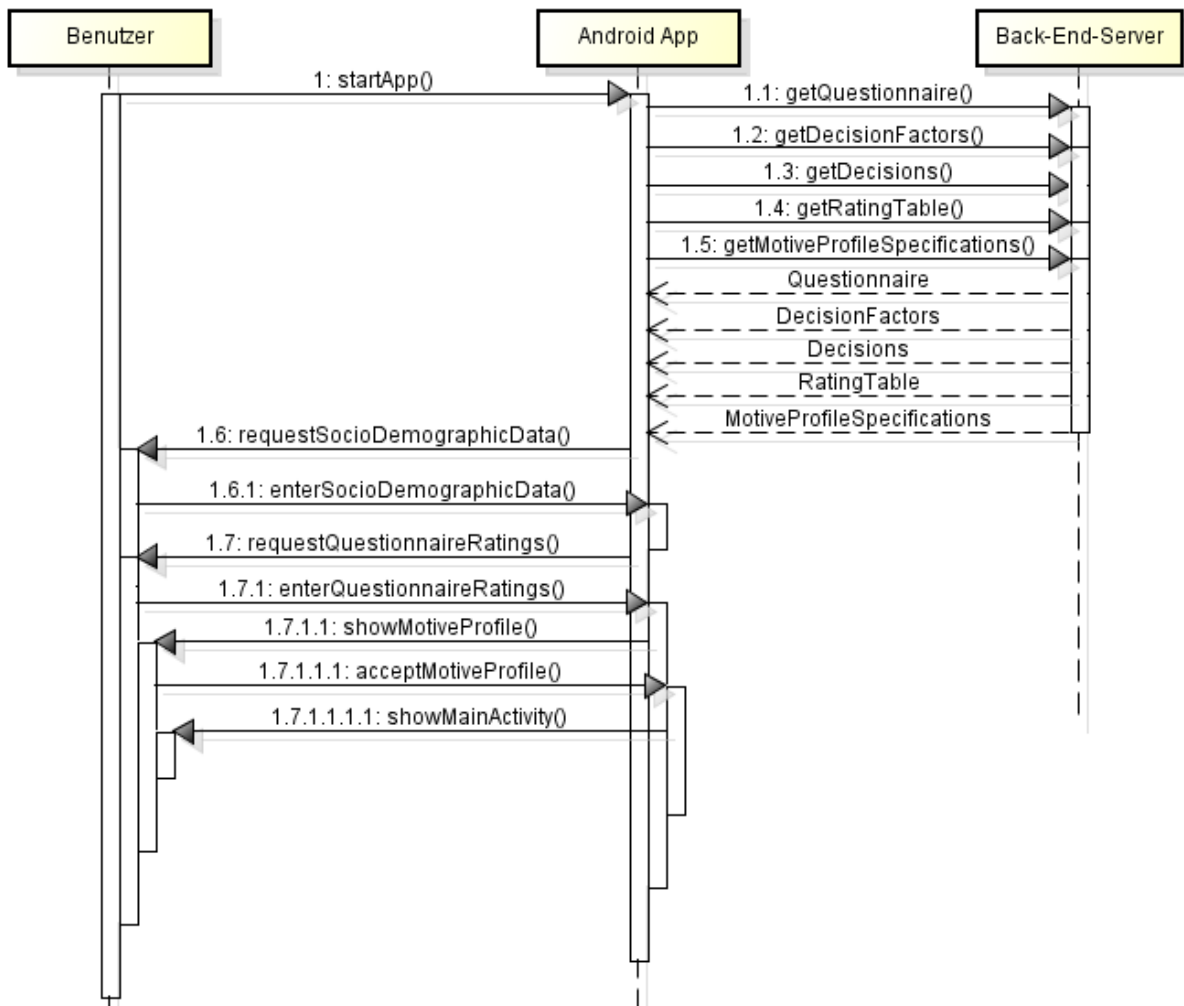


Abbildung 5: SSD: Erstes Starten der App

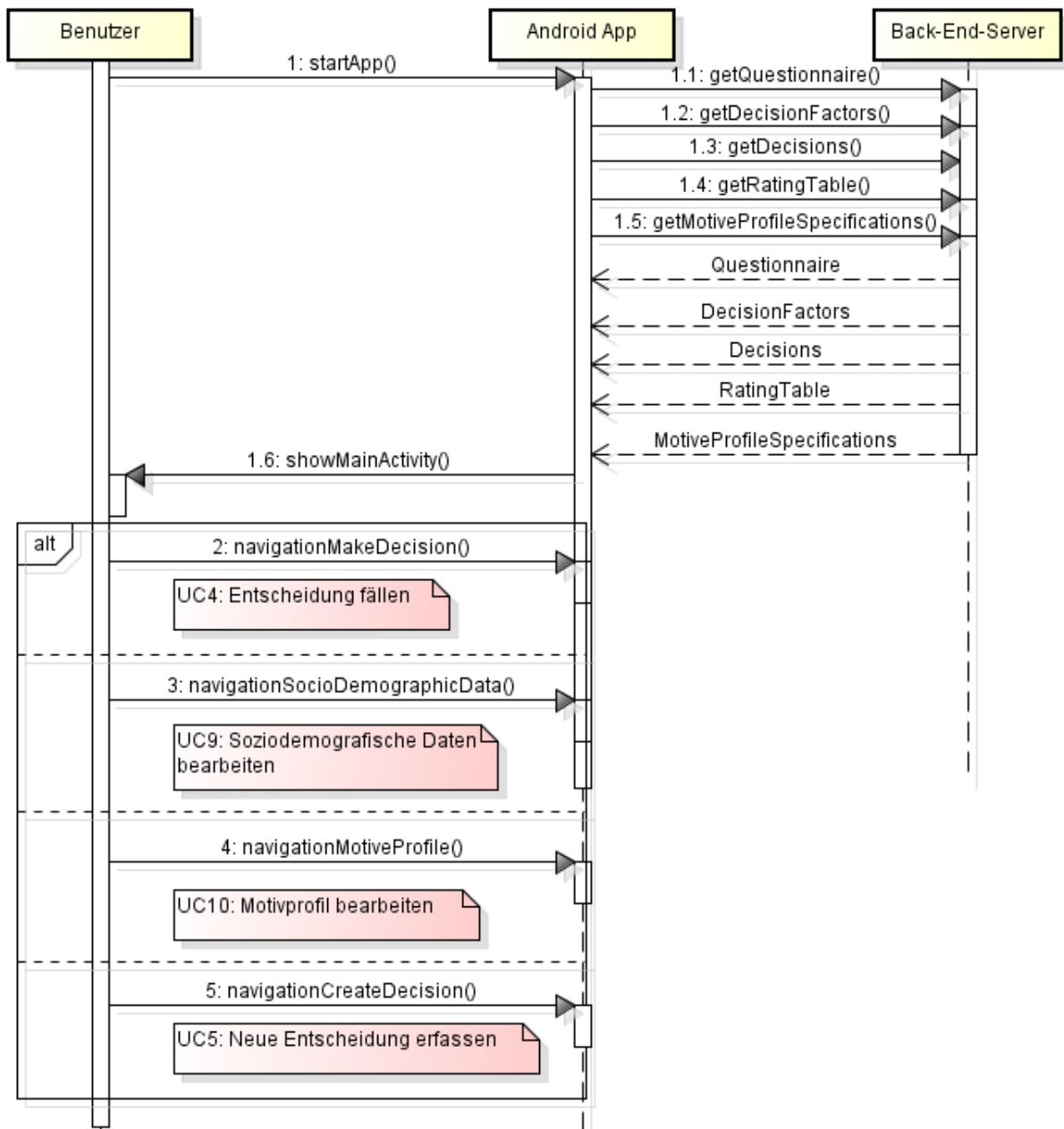


Abbildung 6: SSD: Weiteres Starten der App

3.2.2 UC4: Entscheidung fällen

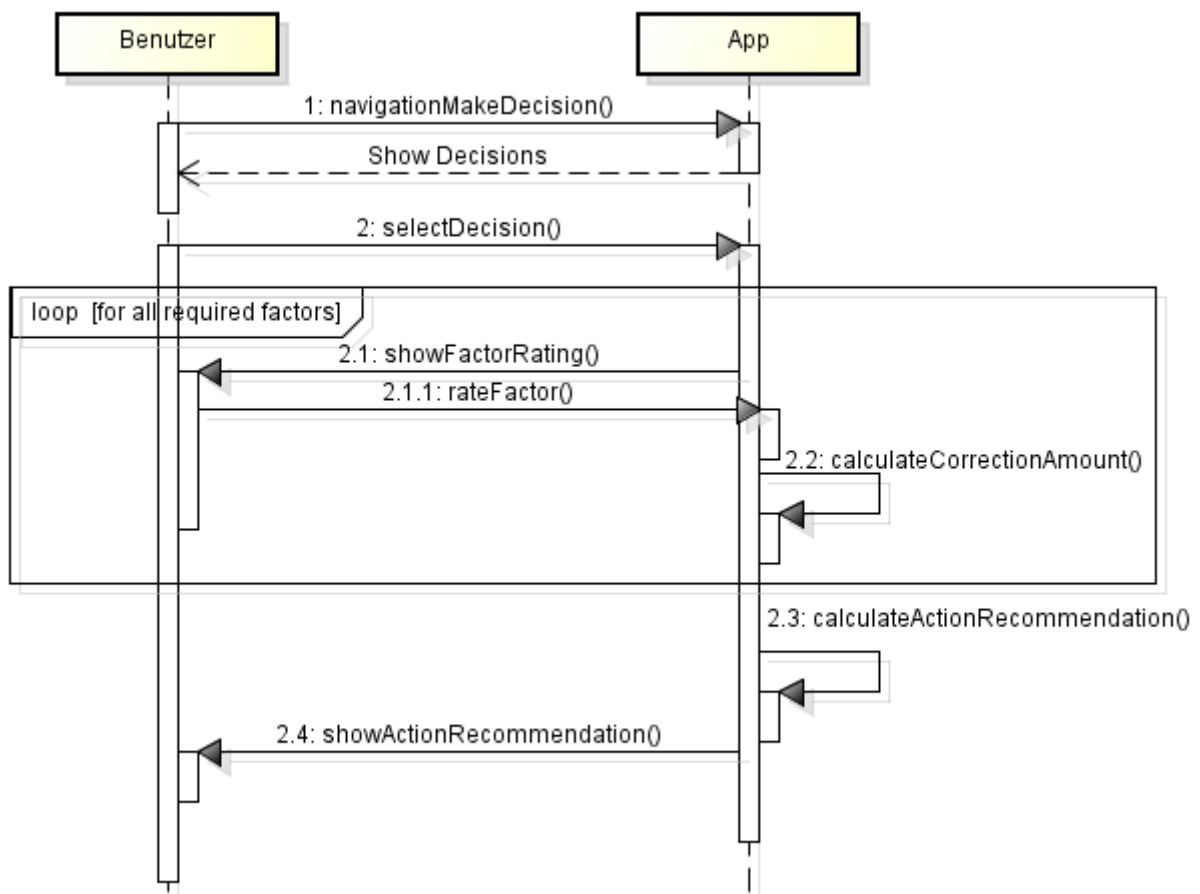


Abbildung 7: Ablauf "UC4: Entscheidung fällen"

4. Design und Umsetzung

4.1 Systemübersicht

Der Benutzer arbeitet mit der App auf seinem Smartphone. Beim Öffnen der App synchronisiert diese mit dem SFTP-Server die neuesten Daten. Der Admin kann die Daten aktualisieren indem er die XML-Dateien auf dem SFTP-Server bearbeitet.

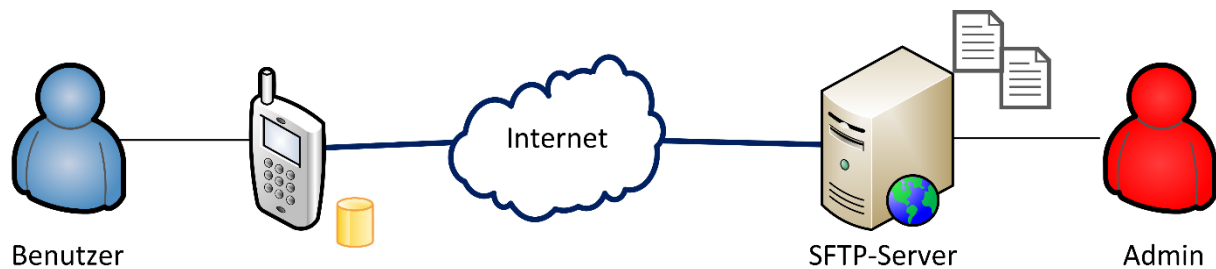


Abbildung 8: Systemübersicht: Überblick der beteiligten Systeme und Benutzer

4.2 Architektonische Ziele & Einschränkungen

4.2.1 Ziele

Es soll möglich sein die Entscheidungssituationen und deren Entscheidungsfaktoren ohne Update der App zu aktualisieren. Zusätzlich sollen die Schwellwerte der Entscheidungsregel änderbar sein.

4.2.2 Einschränkungen

Ist der Server nicht verfügbar oder hat der Benutzer keine Internet-Verbindung aktiviert, werden die Daten nicht synchronisiert. Dies beinhaltet das Herunterladen von den vorgegebenen Entscheidungssituationen und deren Entscheidungsfaktoren, sowie dem Hochladen der Entscheidungssituationen und –faktoren, die der Benutzer erstellt hat.

4.2.2.1 Dynamische Konfigurationen

Die Android App soll viele Daten vom Back-End-Server herunterladen. So muss z.B. bei einer Änderung der vorgegebenen Entscheidungssituationen keine neue Version der Applikation veröffentlicht werden. Sondern man kann lediglich die entsprechende XML-Datei auf dem Back-End-Server anpassen.

In Tabelle 2 wird definiert, welche Daten via XML-Dateien vom Back-End-Server heruntergeladen werden und welche Daten über eine neue Version der Applikation geändert werden können. Die genauen Formate der XML-Dateien sind im Kapitel "4.6 Datenaustausch-Format" definiert.

Konfiguration	Variante
Vorgegebene Entscheidungssituationen	XML-Datei
Vorgegebene Entscheidungsfaktoren	XML-Datei
Motivprofil-Fragebogen	XML-Datei
Motivprofil Beschreibungen und Angabe der Durchschnittswerte	XML-Datei
Ratingtabelle (zum Berechnen der Korrekturbeträge)	XML-Datei
Kurzantworten für Handlungsempfehlung (Positiv, Warnung und Negativ). Beispiel: You can proceed with slight risk	App Konfiguration
Standard-Detailerläuterungen der Entscheidungsfaktoren (Warnung und Negativ)	App Konfiguration
Schwellwerte für Positiv mit Warnung und Negativ	App Konfiguration
Gewichtungen der Entscheidungsfaktor Bewertungen (Grün, Orange, Rot, Nicht relevant)	App Konfiguration

Tabelle 2: Konfigurationen

4.3 Logische Architektur

Die Applikation ist in drei Schichten aufgeteilt: view – application – data.

In Abbildung 9 sind die Schichten (inkl. Sub-Packages), und deren Abhängigkeiten abgebildet. Die jeweiligen Packages werden in den folgenden Unterpunkten beschrieben.

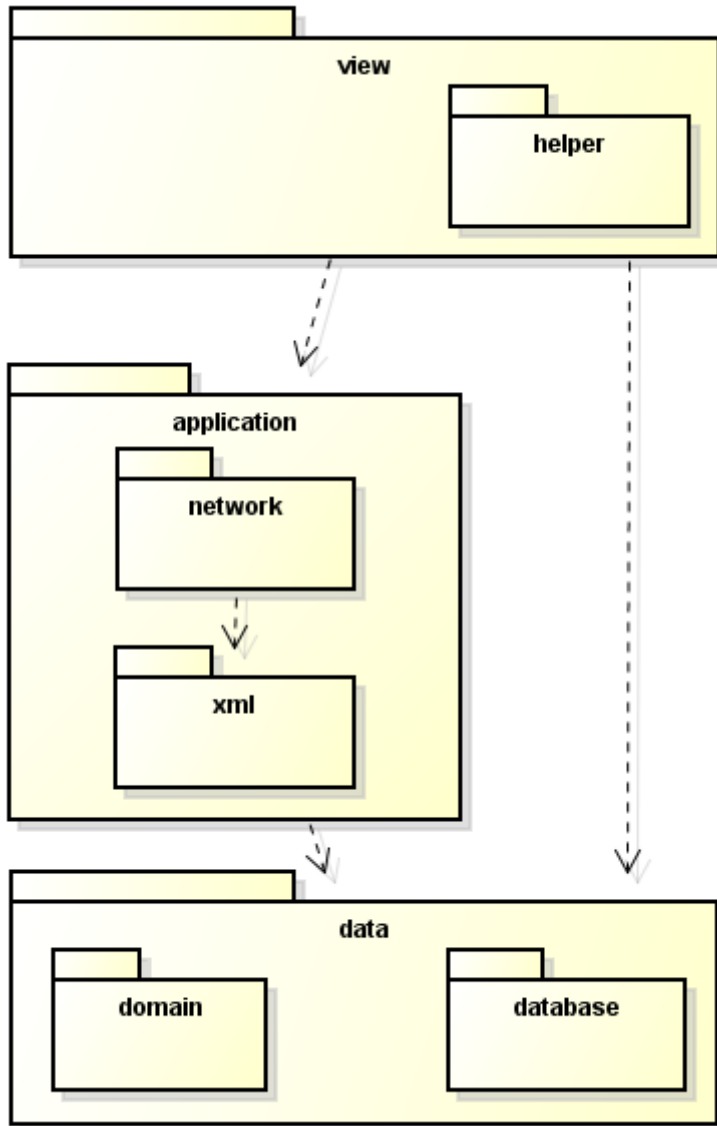


Abbildung 9: Logische Architekturschichten

4.3.1 data

Dieses Package enthält alle Klassen zur Speicherung und Bereitstellung der Daten zur Laufzeit. Nebst den Sub-Packages enthält es die Klasse *Preferences*.

4.3.1.1 Klassenstruktur

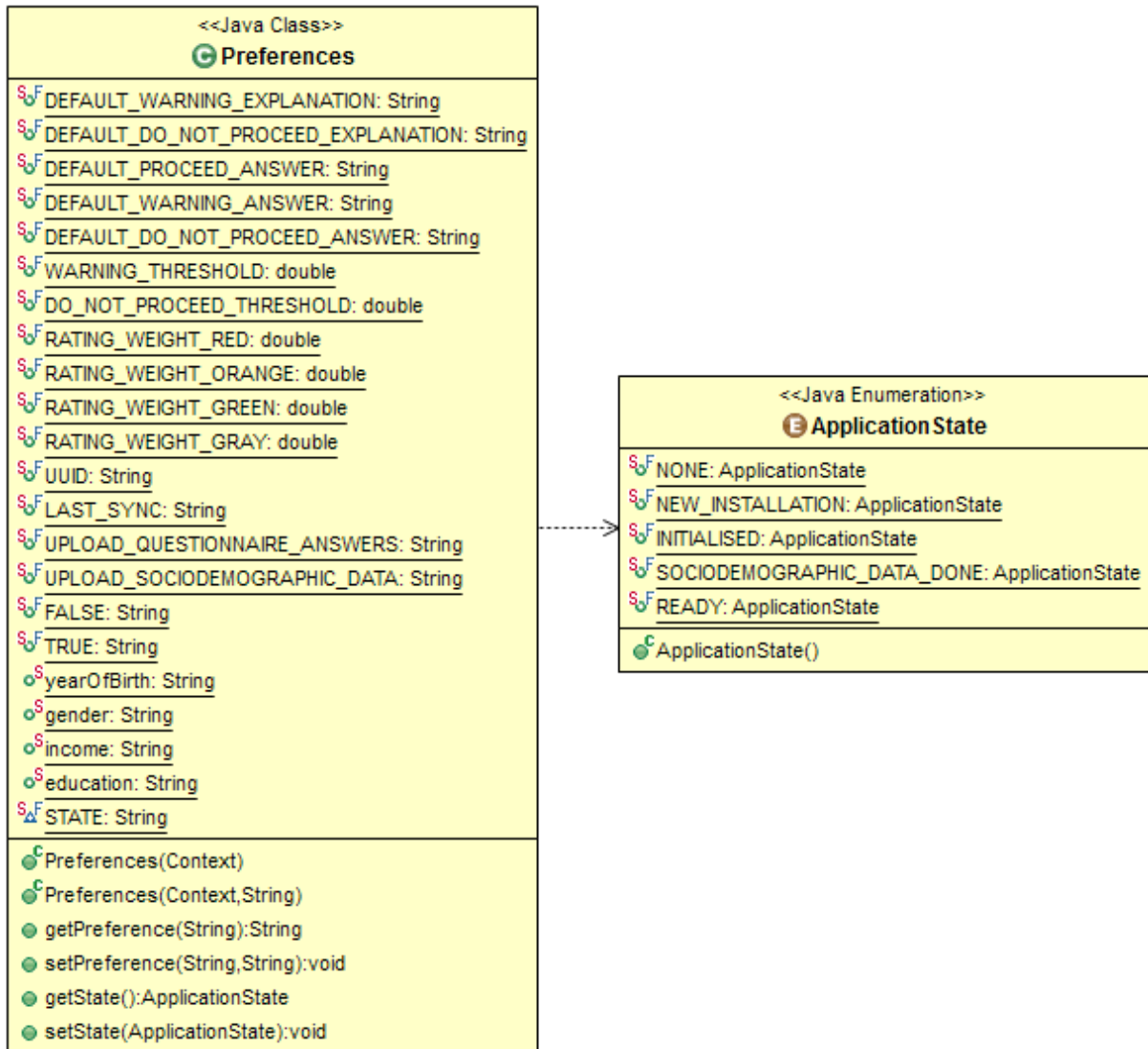


Abbildung 10: Klassendiagramm Package *data*

Preferences Dies ist eine Wrapper-Klasse für *SharedPreferences*. Sie definiert die Namen alle benötigten SharedPreferences und bietet Funktionen zum Lesen und Speichern dazu an. Dies sind einzelne zu speichernde Datenwerte, die nicht als Datenbanktabellen realisiert sind. Z.B. die Soziodemografischen Daten, welche vom Benutzer eingegeben wurden.

4.3.2 data.database

Das Package *database* definiert die Datenbankstruktur und ermöglicht Zugriffe auf die Datenbank. In der Klasse *DBSchema* sind alle Datenbanktabellen definiert. Die verschiedenen Datenbank-Controller greifen alle via *DBSchema* auf diese Datenbankdeklaration zu. Diese Abhängigkeiten und die Klassen der Datenbank-Tabellen wurden zur besseren Übersicht in Abbildung 11 weggelassen.

4.3.2.1 Klassenstruktur

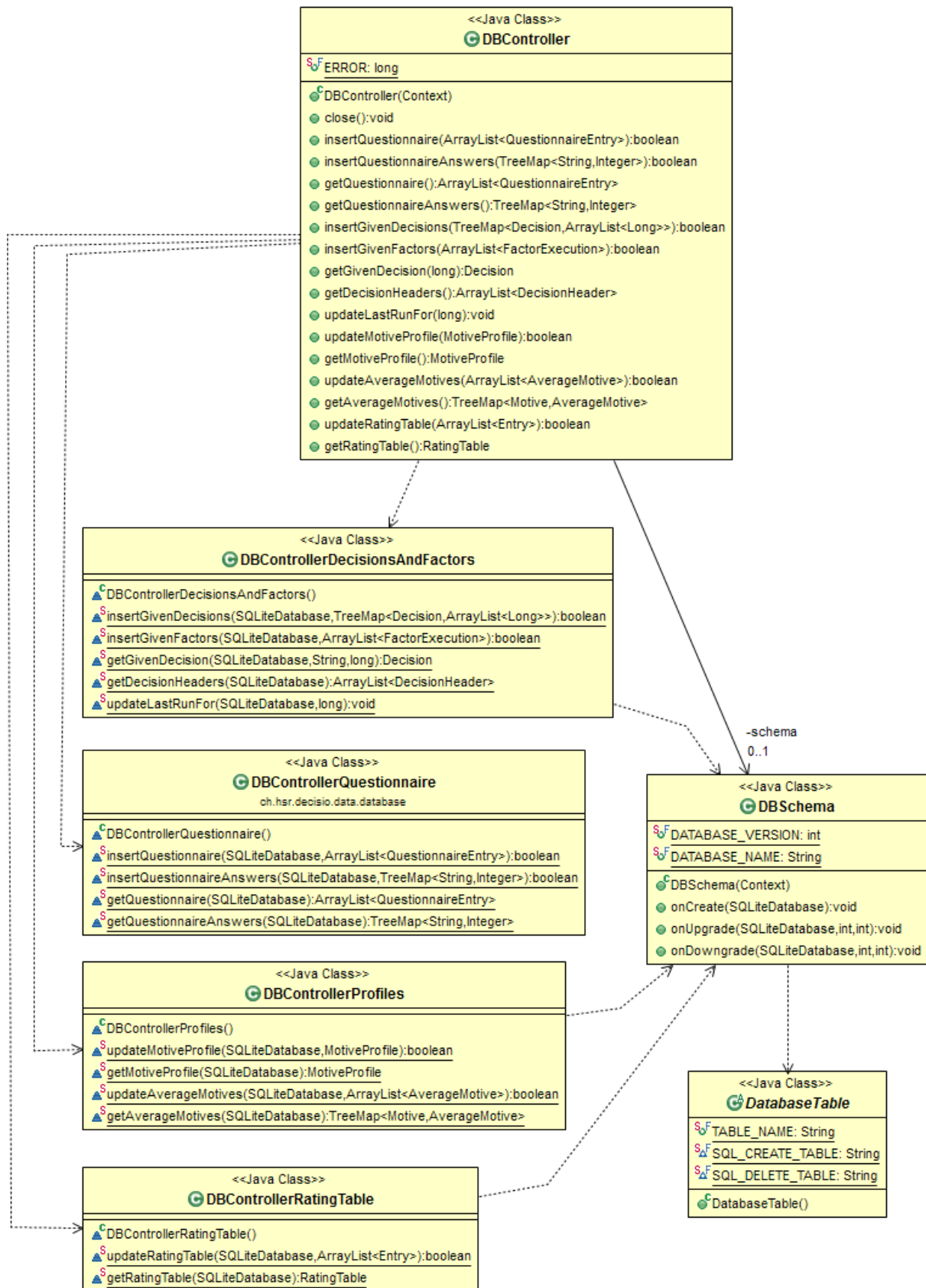


Abbildung 11: Klassendiagramm Package `data.database`

DBSchema	<p>Sie legt das Datenbank-Schema fest und erstellt die Datenbank beim ersten Starten oder nach einem Update der App.</p> <p>Sie erstellt die Tabellen und deren Struktur und stellt die Datenbank dem <i>DBController</i> zur Verfügung.</p>
DBController	<p>Der <i>DBController</i> agiert als Facade-Objekt auf die Datenbanktabellen. Somit kann zentralisiert auf alle Datenbank-Operationen zugegriffen werden. Sie delegiert die Methodenaufrufe an entsprechende Sub-Controller weiter.</p> <p>Dieser Ansatz wurde gewählt, um den Code in der Klasse <i>DBController</i> möglichst gering zu halten. Als weiterer Vorteil sind in den höheren Schichten nicht mehrere Instanzen von unterschiedlichen DBControllern nötig.</p>
DBController- DecisionsAndFactors	<p>Dieser Sub-Controller enthält alle Methoden zum Eintragen und Abfragen der gegebenen Entscheidungssituationen und gegebenen Faktoren, sowie das Aktualisieren des Datums der letzten Durchführung einer Entscheidungssituation.</p>
DBController- Questionnaire	<p>Dieser Sub-Controller definiert alle nötigen Operationen zum Eintragen und Abfragen der Fragen und Antworten des Benutzers des Motivprofil-Fragebogens.</p>
DBController- Profiles	<p>Dieser Sub-Controller enthält die Funktionalität zum Aktualisieren des Motivprofils des Benutzers und der Motivprofil Beschreibungen.</p>
DBController- RatingTable	<p>Dieser Sub-Controller ist verantwortlich für das Aktualisieren und Abfragen der Einträge der Ratingtabelle.</p>

4.3.2.2 Wichtige interne Abläufe

Die Klasse *DBSchema* erweitert den *SQLiteOpenHelper* des Android-Frameworks. Der *DBController* instanziiert ein *DBSchema* welches eine *SQLiteDatabase* zur Verfügung stellt.

Die Sub-Controller enthalten nur statische Methoden, damit der *DBController* nicht für jeden Sub-Controller eine Instanz halten muss.

Bei der Aktualisierung von Daten, wie z.B. Motivprofil-Fragebogen, werden die Tabellen geleert. Ob die entsprechende Tabelle geleert wird oder nicht, liegt in der Verantwortlichkeit der Sub-Controller.

4.3.3 data.domain

Das Package *domain* bildet die Problem-Domain ab. Das Domainmodell wurde um einige Klassen erweitert.

4.3.3.1 Klassenstruktur

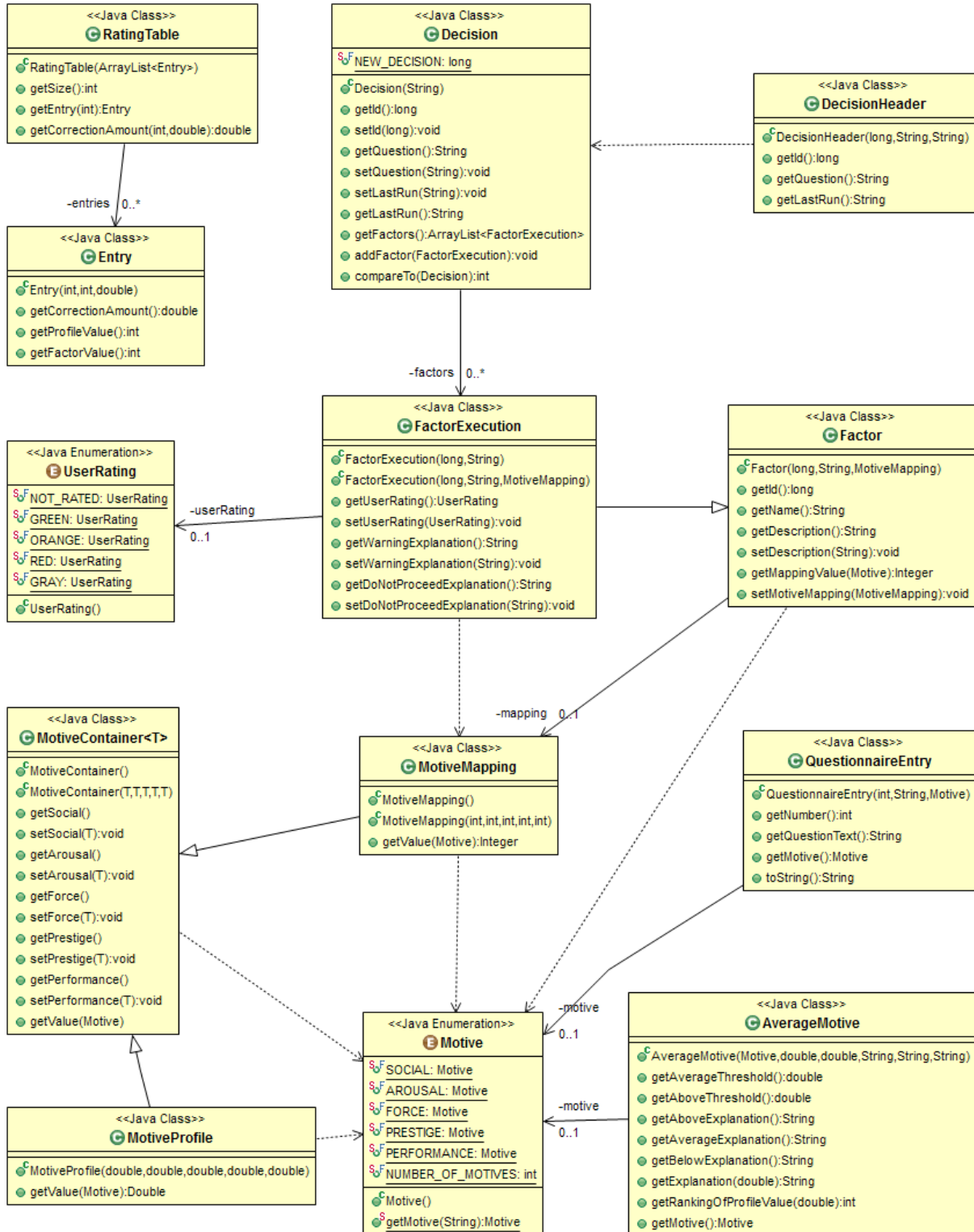


Abbildung 12: Klassendiagramm Package *data.domain*

Decision	Enthält alle Informationen einer Entscheidungssituation während der Ausführung.
MotiveContainer	Generische Klasse, zur Wiederverwendung von Code für die Zugriffe auf die Werte der einzelnen Motive in: a) Klasse <i>MotiveProfile</i> b) Klasse <i>MotiveMapping</i> (Mapping des Faktors auf die verschiedenen Motive)

4.3.4 application

Das *application* Package beinhaltet die gesamte Koordination der einzelnen Use Cases, sowie der Netzwerkzugriffe und dem Parsen der XML-Dateien in den entsprechenden Sub-Packages.

4.3.4.1 Klassenstruktur

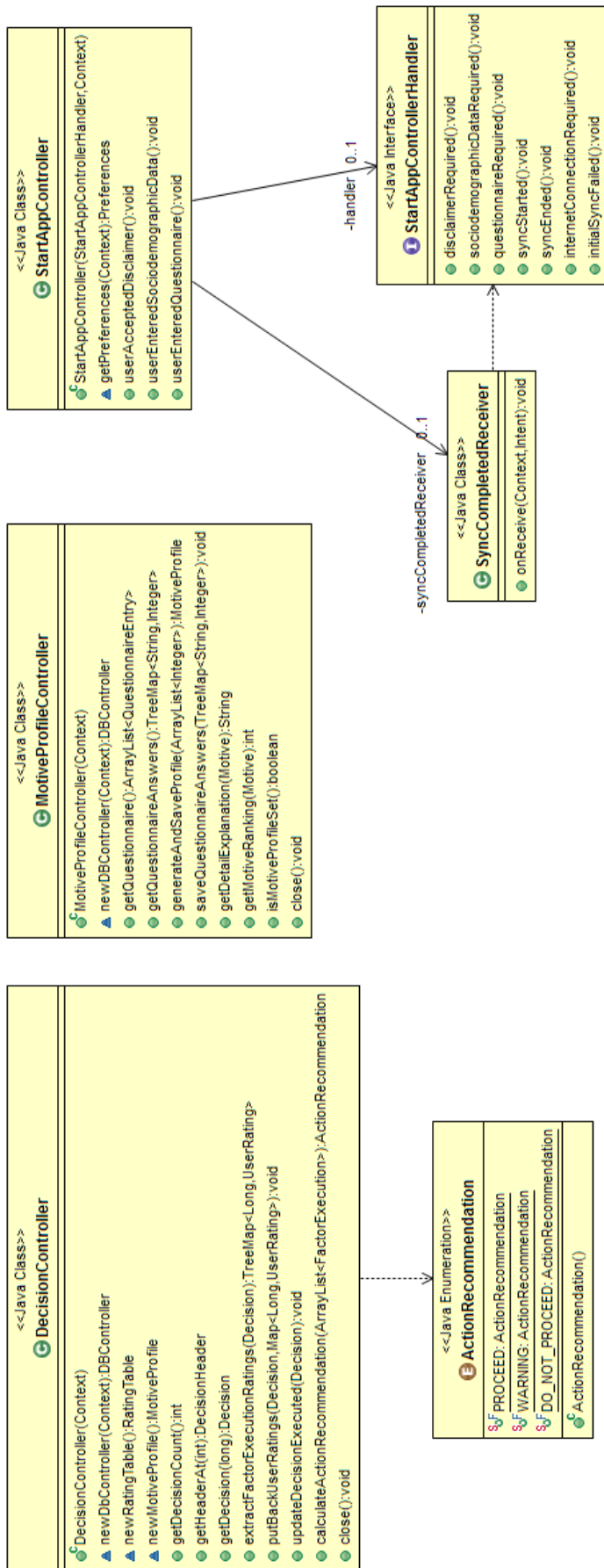


Abbildung 13: Klassendiagramm Package application

- DecisionController** Enthält die Logik, um eine Entscheidung durchzuführen. Dies beinhaltet das Auflisten aller vorhandenen Entscheidungssituationen und das Berechnen der Handlungsempfehlung.
- MotiveProfile-Controller** Enthält die Logik, um ein Motivprofil zu erstellen und zu visualisieren. Dies beinhaltet auch das Laden des Motivprofil-Fragebogens und der Durchführung der Vergleiche des erstellten Motivprofils mit den heruntergeladenen Durchschnittswerten.
- StartAppController** Enthält die Logik, um den Use Case "UC1: App starten" zu koordinieren. Dies beinhaltet das jeweilige Auslösen der Synchronisierungs-Vorgänge, sowie der Koordination der Aufforderungen zur Dateneingabe an den Benutzer.

4.3.4.2 Schnittstellen

StartAppController-Handler Wird vom *StartAppController* verwendet, um der implementierenden Activity anhand Callbacks mitzuteilen, welchen Inhalt sie dem Benutzer anzeigen soll.

4.3.4.3 Wichtige interne Abläufe

Der *StartAppController* arbeitet als Zustandsmaschine. Der aktuelle Status wird in einer *SharedPreferences* persistent gespeichert. Die Abbildung 14 soll die Funktionsweise des *StartAppControllers* visualisieren.

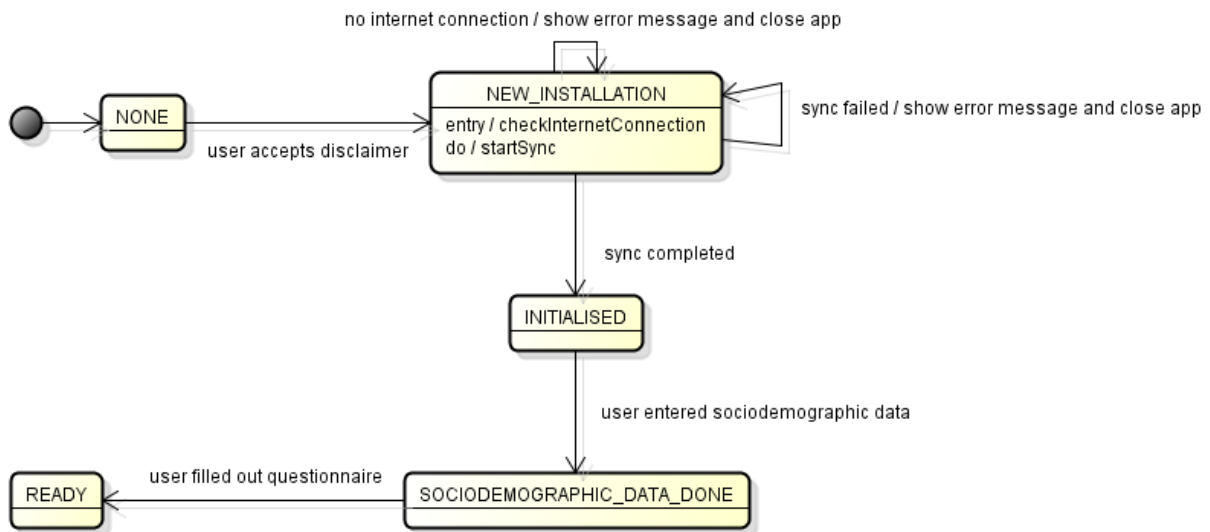


Abbildung 14: Zustandsdiagramm *StartAppController*

4.3.5 application.network

Das Package *network* ist für die gesamte Kommunikation mit den Back-End-Server zuständig.

4.3.5.1 Klassenstruktur

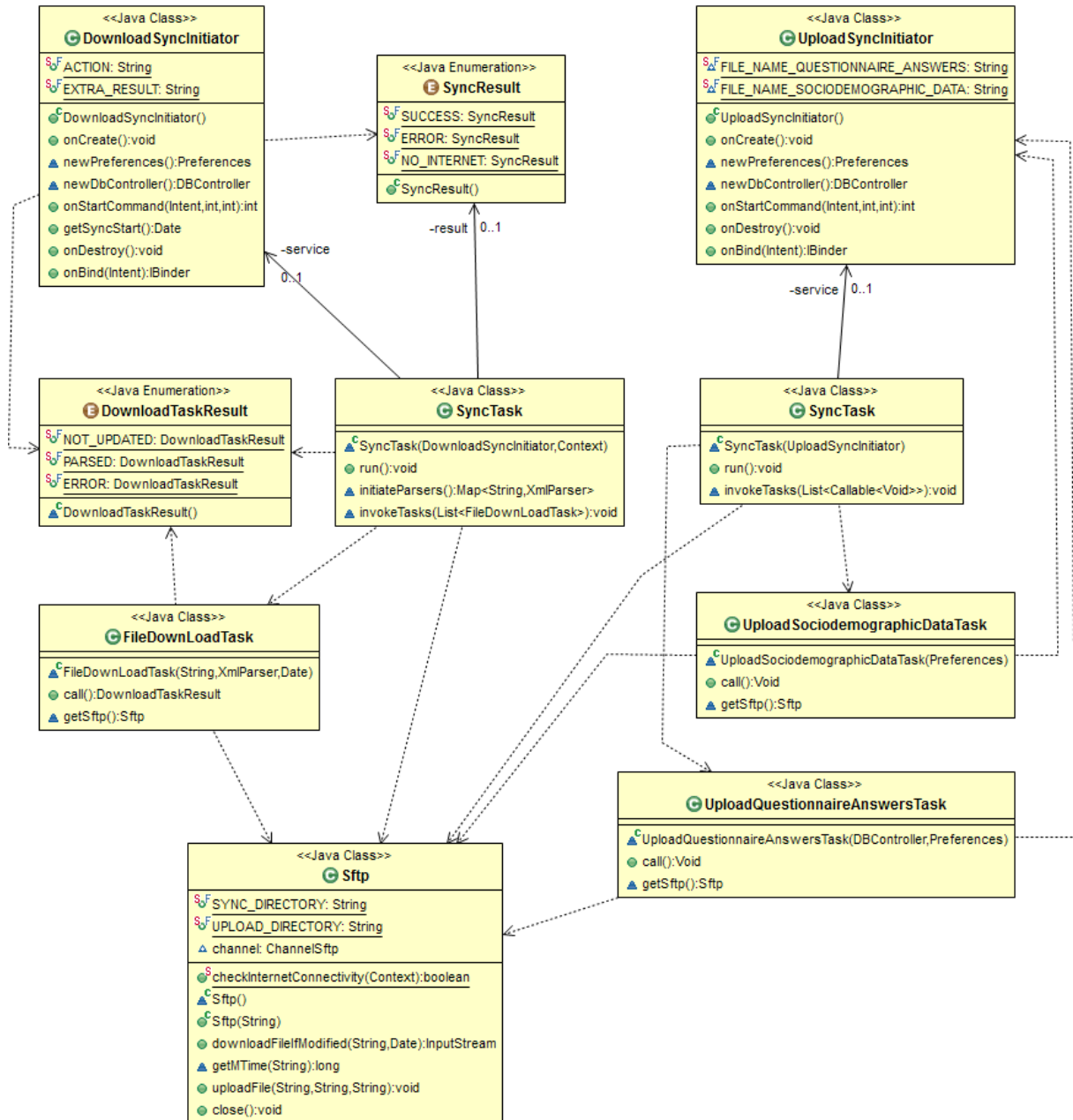


Abbildung 15: Klassendiagramm Package *application.network*

Download-SyncInitiator	<p>Er ist als nicht-gebundener Service realisiert. Dieser Service wird einmal gestartet und wird von Android nicht beendet, solange er läuft. Dies hat zur Folge, dass das Herunterladen der XML-Dateien vom Back-End-Server im Hintergrund geschehen kann, ohne dass die Synchronisation von Android abgebrochen wird, obwohl der Benutzer die App verlässt.</p> <p>Wir entschieden uns gegen einen Content Provider, weil wir keine Erfahrungen diesbezüglich besitzen und uns der Aufwand für unsere Zwecke zu gross schien. Zusätzlich kann der Benutzer so selbst entscheiden, ob er die Daten synchronisieren will oder nicht (Internet-Verbindung ausschalten).</p>
Upload-SyncInitiator	<p>Er ist ebenfalls als ungebundener Service realisiert. Er startet den <i>SyncTask</i> welcher den <i>UploadSociodemographicDataTask</i> und den <i>UploadQuestionnaire-AnswersTask</i> startet. Diese Tasks laden bei Bedarf die entsprechenden Daten auf den Back-End-Server hoch.</p>
SyncTask	<p>Die Klassen <i>SyncTask</i> stellen jeweils einen Worker-Thread dar und werden vom <i>Download-/UploadSyncInitiator</i> gestartet, damit die Synchronisation nicht auf dem GUI-Thread läuft. Sie sind als innere Klassen im <i>Download-/Upload-SyncInitiator</i> realisiert, da sie Unteraufgaben ihrer Elternklasse übernehmen und auf sie nicht von anderen Klassen zugegriffen werden sollte.</p>
Sftp	<p>Diese Klasse regelt die Zugriffe auf den Back-End-Server. Sie stellt Methoden für das Herunter- und Hochladen von Dateien zur Verfügung. Bei der Instanziierung öffnet sie eine Verbindung und wechselt in das entsprechende Verzeichnis.</p>

4.3.6 application.xml

Es enthält eine abstrakte Klasse, die die Grundfunktionalität der XML-Parser zur Verfügung stellt. Für jede mögliche XML-Datei, die geparkt werden muss, gibt es einen spezifischen XML-Parser.

4.3.6.1 Klassenstruktur

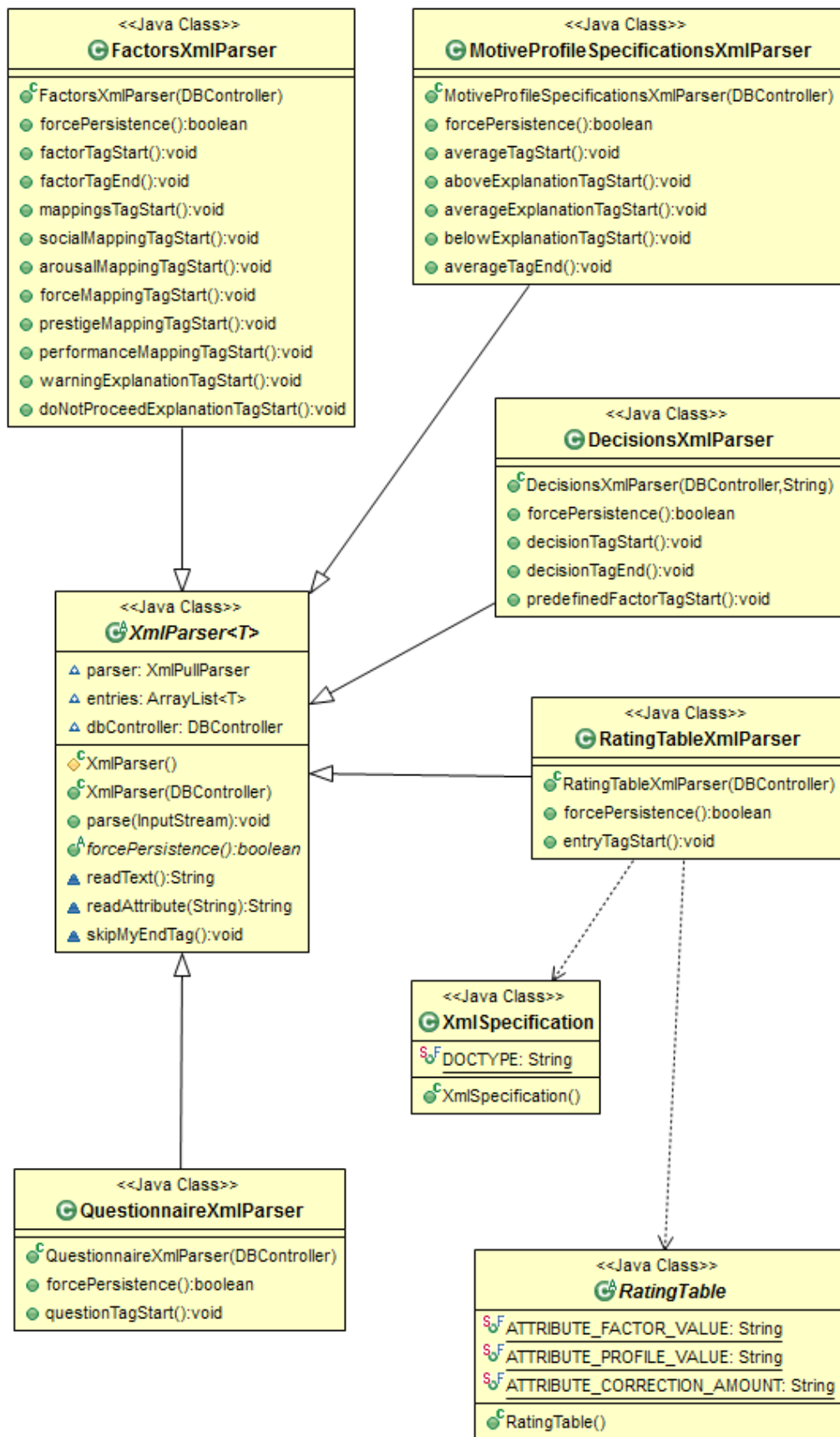


Abbildung 16: Klassendiagramm Package *application.xml*

XmlParser	Grund-Klasse für die spezifischen Parser. Sie stellt drei Methoden zur Verfügung, welche den Aufwand in den spezifischen Parsern minimieren (Attribut-Wert lesen, Tag-Text lesen und End-Tag überspringen).
XmlSpecification	Sie enthält eine Klasse pro mögliche XML-Datei, in denen die Attribute dieser Datei definiert sind. Somit sind diese Informationen zentralisiert. Für die Dateien zum Hochladen sind in den Klassen zusätzlich die benötigten Tags definiert.

4.3.6.2 Wichtige interne Abläufe

Der Parse-Prozess wird durch die Klasse *XmlParser* durchgeführt. Pro Tag-Beginn und -Ende wird mittels Reflection die entsprechende Methode des spezifischen Parsers aufgerufen. Gibt es keine solche Methode, wird der Fehler ignoriert, da der spezifische XML-Parser diesen Tag nicht lesen will, oder die Datei einen Schreibfehler enthält.

4.3.7 view

In diesem Package sind alle Activities und Fragmente der App enthalten.

4.3.7.1 Klassenstruktur

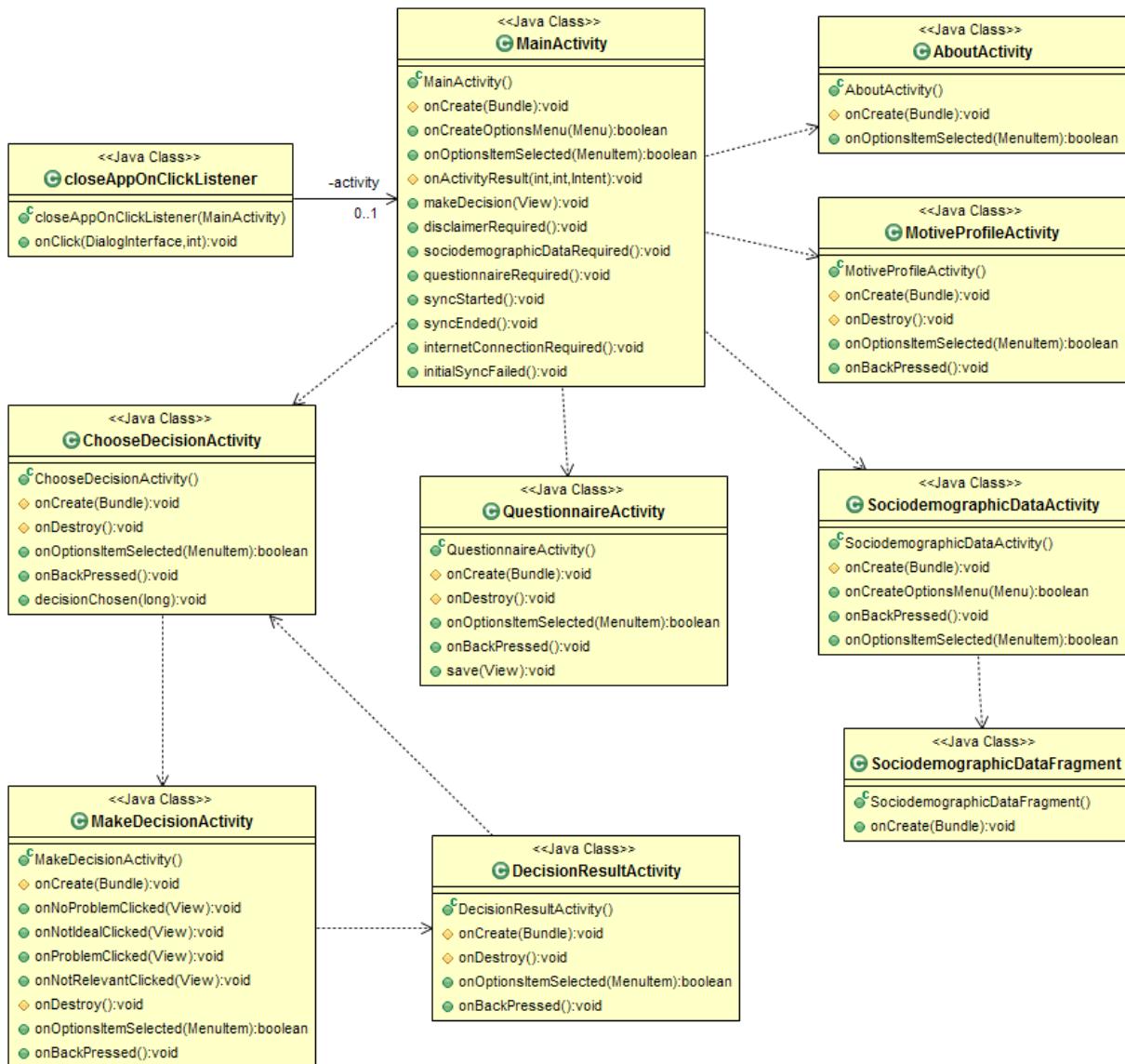


Abbildung 17: Klassendiagramm Package view

CloseAppOnClickListener Wird vom Disclaimer Dialog verwendet, damit bei einer Ablehnung die App geschlossen werden kann. Die *MainActivity* implementiert ihn und übergibt ihn dem Dialog.

4.3.7.2 Wichtige interne Abläufe

Aufgrund der Navigation der *ChooseDecision*-, *MakeDecision*- und *DecisionResultActivity* entsteht die ersichtliche Dreiecks-Beziehung. Für Details zu der Navigation siehe "4.7 Wireframes".

4.3.8 view.helper

Enthält Hilfsklassen für die Activities. Dies beinhaltet den *DialogueHelper*, die ListView-Adapter und deren Listeners, Handlers und Helpers.

4.3.8.1 Klassenstruktur

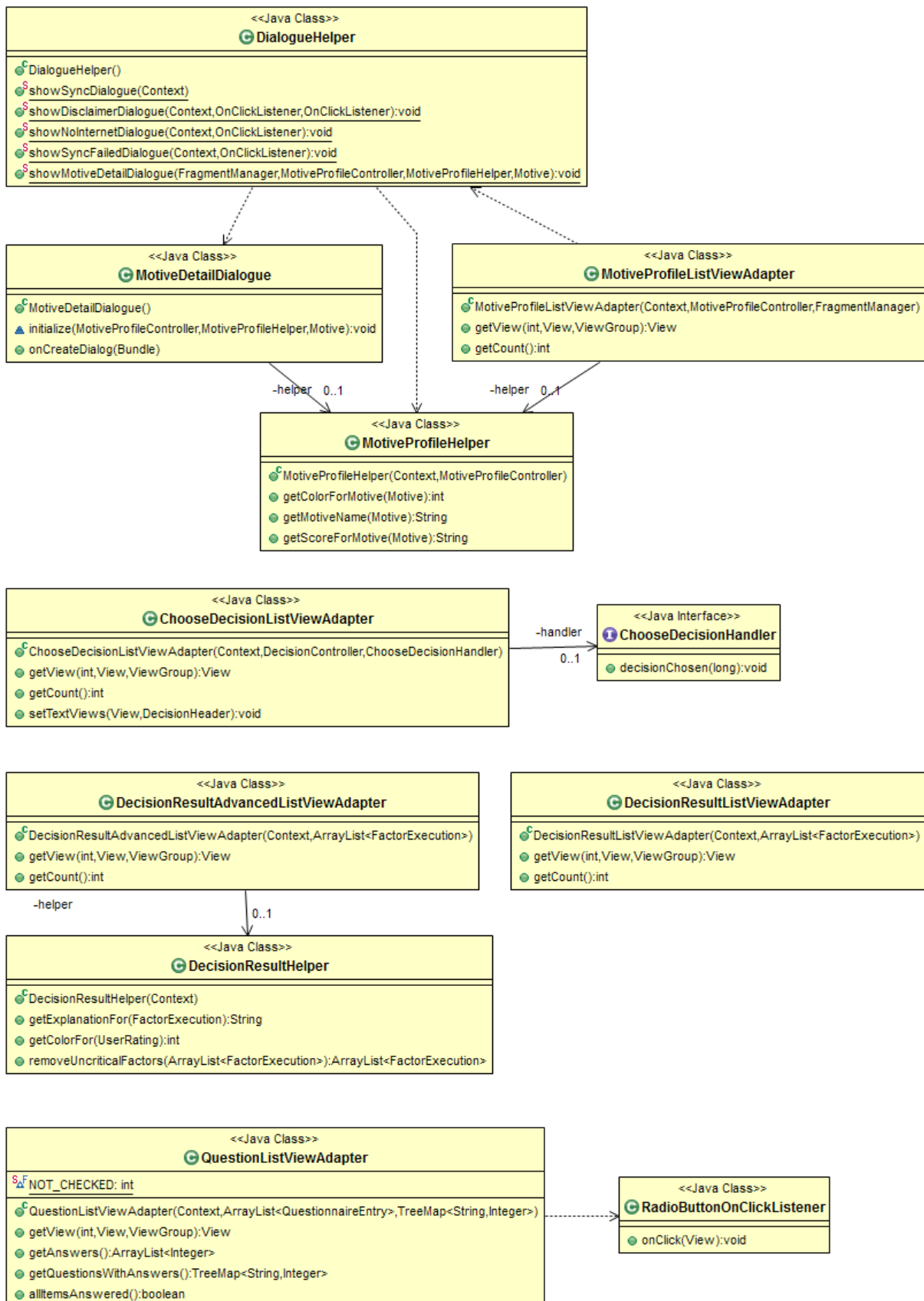


Abbildung 18: Klassendiagramm Package *view.helper*

DialogueHelper	Diese Klasse enthält statische Methoden zum Erstellen/Starten aller Dialoge in der App.
MotiveDetailDialogue	Dies ist der Dialog, um die Detailansicht zum angeklickten Motiv anzuzeigen. Darin wird das erzielte Ranking farblich dargestellt, sowie eine detaillierte Beschreibung zum erreichten Wert angezeigt.
MotiveProfile- ListViewAdapter	Diese Klasse dient zur Unterstützung der Motivprofil Visualisierung. Jedes Motiv wird dabei als eigenes Element in einer ListView dargestellt. Zu jedem Motiv wird das erzielte Ranking farblich angezeigt, sowie eine kurze Beschreibung der Einstufung (Durchschnittlich, über dem Durchschnitt oder unter dem Durchschnitt) angezeigt.
MotiveProfileHelper	Dient zum einheitlichen Zugriff auf die Namen der Motive, sowie den erzielten Motivprofil Werten und den Farbbressourcen (zur Darstellung des Ranking).
ChooseDecision- ListViewAdapter	Klasse zur Generierung der einzelnen Elemente in der Auflistung der vorhandenen Entscheidungssituationen.
DecisionResultAdvanced- ListViewAdapter	Definiert die Darstellung der Elemente in der Liste, welche die Details der Faktor-Bewertungen vom Benutzer im Tab <i>Advanced</i> anzeigt.
DecisionResult- ListViewAdapter	Klasse zur Generierung der einzelnen Elemente in der Auflistung der Faktor-Bewertungen des Benutzers im Tab <i>Result</i> .
Question- ListViewAdapter	Definiert die Darstellung der Elemente in der Auflistung des Motivprofil-Fragebogens. Zusätzlich speichert die Klasse die bereits gegebenen Antworten des Benutzers temporär.

4.3.8.2 Schnittstellen

ChooseDecisionHandler	Wird verwendet, um der Activity mitzuteilen, welche Entscheidungssituation ausgewählt wurde.
-----------------------	--

4.3.9 Wichtige Abläufe

Hier sind wichtige Abläufe beschrieben, welche über mehrere der drei Haupt-Packages *view*, *application* und *data* ablaufen.

4.3.9.1 ListView

Das Android-Framework wiederverwendet Elemente des Typs `ListView` zur Reduktion von benötigtem Speicher. Deshalb speichert der `view.helper.QuestionListViewAdapter`, welche Antworten der Benutzer während dem Ausfüllen bereits gegeben hat. Speichert der Benutzer den Fragebogen, fragt die `view.QuestionnaireActivity` via `application.MotiveProfileController` den Adapter ab, ob alle Fragen beantwortet wurden. Falls ja, weist die Activity den Controller an, die Antworten in der Datenbank zu speichern und startet selbst das Hochladen der Antworten.

4.3.9.2 Synchronisation

4.3.9.2.1 Ablauf

Beim Starten der App wird die *view.MainActivity* gestartet. Diese erstellt zu Beginn einen *application.StartAppController*. Dieser prüft über den aktuellen App-Status, welche Operationen der Benutzer bereits ausgeführt hat und handelt entsprechend.

Sind Synchronisierungen nötig, startet der Controller einen *application.network.DownloadSyncInitiator* und einen *application.network.UploadSyncInitiator*. Danach meldet er mit einem Callback des *application.StartAppControllerHandler* die laufenden Synchronisierungen und registriert sich als Empfänger des Broadcast beim Broadcast-Manager (für das Synchronisierungsergebnis).

Hat der *DownloadSyncInitiator* seine Arbeit erledigt, sendet er einen Broadcast, um dem Controller zu signalisieren, dass die App nun verwendet werden kann (es sind alle nötigen Daten in der Datenbank gespeichert). Auf die Beendigung des *UploadSyncInitiators* muss nicht gewartet werden, da die App trotzdem verwendet werden kann. Er beendet sich selbst, ohne weitere Aktion.

Der Controller meldet die Beendigung wieder mit einem Callback an seinen Handler in die *view*-Schicht und meldet sich vom Broadcast-Manager ab. Die Art des Callbacks ist je nach Synchronisierungsergebnis und App-Status unterschiedlich.

4.3.9.2.2 Ablauf DownloadSyncInitiator

Der *DownloadSyncInitiator* startet einen *SyncTask*, welcher mehrere *FileDownloadTasks* startet. Diese laden die entsprechenden Dateien bei Bedarf herunter und übergeben diese an die spezifischen *XmlParser*.

Die einzelnen XML-Dateien werden nur vom Back-End-Server heruntergeladen, sofern sich etwas am Inhalt der Dateien seit der letzten Synchronisation geändert hat. Es wird das "Zuletzt Geändert" Attribut der Datei mit dem Datum der letzten Synchronisation verglichen.

Wurde die Datei nicht aktualisiert, wird der entsprechende Parser nicht gestartet.

Sind alle nötigen Dateien geparkt, weist der *SyncTask* die *XmlParser* sequentiell an, die Daten in der Datenbank zu speichern. Haben die *XmlParser* gültige Einträge gefunden, übergeben sie diese dem *DBController* zur Speicherung. Das Ergebnis der Persistierung gibt der entsprechende *XmlParser* dem *SyncTask* zurück.

Konnten alle gestarteten Parser ihre Daten speichern, ist die Synchronisation erfolgreich.

4.3.9.2.3 Broadcast Manager

Als Broadcast Manager wird der von Android zur Verfügung gestellte *LocalBroadcastManager* verwendet. Dieser erlaubt es, Meldungen/Broadcasts nur innerhalb der App zu verschicken.

4.3.9.2.4 SFTP-Zugriff

Die SFTP-Zugriffe erfolgen über die Klasse *Sftp*, welche die Library JSch (Java Secure Channel) verwendet. [5]

Die Auswahl von möglichen Libraries die SFTP-Zugriffe unterstützen, ist relativ gering. Unsere Wahl fiel auf JSch, da sie sehr kompakt ist und unter der BSD-Lizenz läuft. [6]

4.3.9.2.5 XML-Parser

Das Auslösen der Datenspeicherung wurde in den *XmlParsern* realisiert, weil der Parser bereits weiss welche Methode des *DBControllers* ausgeführt werden muss. Ein weiterer Vorteil ist, dass die Activity dem Service keine zusätzlichen Argumente übergeben muss. Der Service hat bereits den Context, um eine Instanz des *DBController* für die Parser anzulegen.

4.4 Prozesse und Threads

Bei Android ist das UI single threaded. Alle Netzwerkaktionen müssen auf einem anderen Thread durchgeführt werden, da diese möglicherweise lange dauern können und den UI Thread nicht blockieren sollen.

4.4.1 Synchronisationen

Bei der Synchronisierung, dem Herunterladen der aktuellen Daten, startet der ungebundene Service *DownloadSyncInitiator* einen Worker-Thread (*SyncTask*). Dieser Worker-Thread wiederum startet mehrere Tasks (*FileDownloadTask*) in einem Thread-Pool, damit das Herunterladen mehrerer XML-Dateien optimal parallelisiert werden kann.

Beim Hochladen von Benutzerdaten wird ebenfalls ein ungebundener Service (*UploadSyncInitiator*) gestartet, welcher einen Worker-Thread (*SyncTask*) startet. Dieser startet die beiden Upload-Tasks *UploadSociodemographicDataTask* und *UploadQuestionnaireAnswersTask*, welche bei Bedarf die entsprechenden Daten hochladen.

4.4.2 Datenspeicherung

Wie in Kapitel 4.3.9.2 beschrieben, wird das Speichern der heruntergeladenen Daten in der Datenbank sequentiell auf dem *SyncTask* Worker-Thread erledigt. Dieser Ansatz wurde gewählt, damit keine Race-Conditions, bzw. Update Probleme, in der Datenbank beim Aktualisieren der Daten auftreten. Es wird ein Synchronisierungs-Dialog angezeigt, bis alle Daten in der Datenbank gespeichert wurden. Somit kann keine andere Klasse als der *SyncTask* Worker-Thread während der Synchronisation auf die Datenbank zugreifen.

4.5 Datenspeicherung

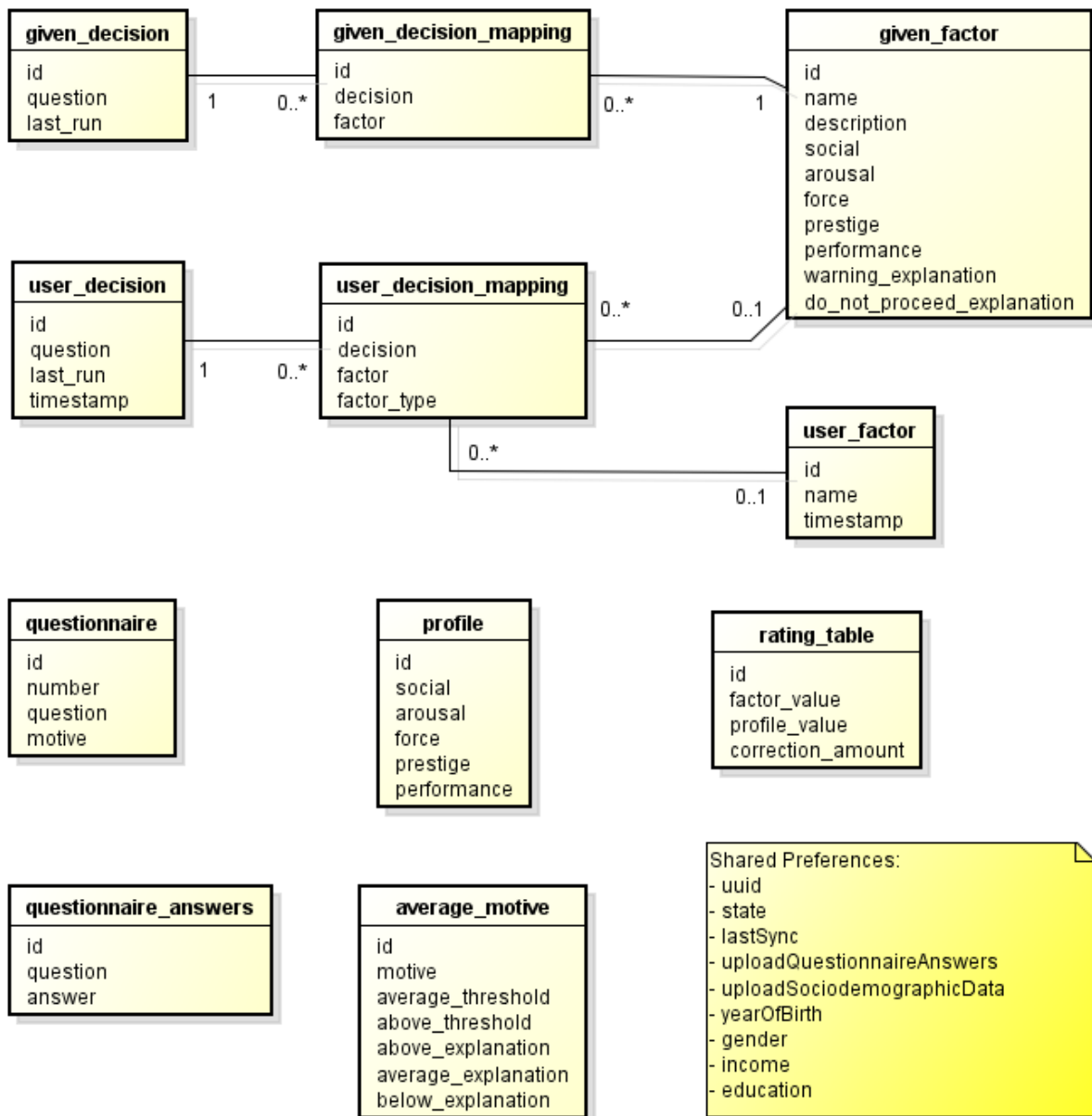


Abbildung 19: Datenbankdiagramm

given_decision	Speichert die vorgegebenen Entscheidungssituationen.
given_factor	Speichert die vorgegebenen Faktoren mit deren Mapping der Motive und der anzuzeigenden Detail-Beschreibungen.
given_decision_mapping	Löst die n:m-Beziehung auf, da ein Faktor zu mehreren Entscheidungssituationen gehören kann.
user_decision	Wird noch nicht verwendet. Würde die Entscheidungssituationen speichern, die der Benutzer erstellt.
user_decision_mapping	Wird noch nicht verwendet. Würde die Faktoren, vorgegebene und benutzer-erstellte, speichern, die der Benutzer einer Entscheidungssituation anfügt. Das Attribut <i>factor_type</i> wird zur Unterscheidung zwischen vorgegebenen und benutzer-erstellten Faktoren benötigt.

user_factor	Wird noch nicht verwendet. Diese würde die Faktoren speichern, die der Benutzer erstellt.
questionnaire	Speichert die Fragen des Motivprofil-Fragebogens, sortiert nach Nummer.
questionnaire_answers	Speichert die Motivprofil-Fragebogen Antworten des Benutzers. Es wird lediglich die Frage und die Antwort gespeichert (ohne Nummer).
profile	Speichert die Motivprofil Werte des Benutzers, welche aufgrund der Antworten zum Motivprofil-Fragebogen berechnet wurden.
rating_table	Speichert die Ratingtabelle der Entscheidungsregel. (Welcher Faktor hat welchen Korrekturbetrag bei welchem Profil)
average_motive	Speichert die Motivprofil Beschreibungen, d.h. die Schwellwerte für den Durchschnittsbereich und die anzuzeigenden Erläuterungen zu den Einstufungen.

4.5.1 Wichtige Konzepte

Die benutzer-erstellten Entscheidungssituationen und Entscheidungsfaktoren werden in separaten Tabellen gehalten und nicht zu den vorgegebenen Entscheidungssituationen bzw. Entscheidungsfaktoren eingefügt. Dies, um Konflikte der IDs zu verhindern, wenn neue Entscheidungsfaktoren oder Entscheidungssituationen vom Server herunter geladen werden.

4.6 Datenaustausch-Format

Der Datenaustausch zwischen den Android Geräten und dem Back-End-Server erfolgt über **XML-Files**. Dies erlaubt die Konfiguration der vorgegebenen Daten (z.B. Entscheidungssituationen) und stellt keinen grossen Zusatz-Aufwand dar, sodass der Hauptfokus auf der Android App bleibt.

In den folgenden Unterpunkten werden die Formate für den Austausch der einzelnen Informationen zwischen den Android Geräten und dem Back-End-Server beschrieben.

4.6.1 Motive

Bei manchen Dateien wird das Motiv verlangt (Motivprofil-Fragebogen und Motivprofil Beschreibungen). Es wurden englische Begriffe gewählt, welche die Verwechslungsgefahr möglichst gering halten sollen. Folgende Werte sind gültig:

Begriff	Erläuterung
social	Soziale Sicherheit, Abhängigkeit
arousal	Neugier, Tatendrang, Unternehmenslust
force	Macht
prestige	Geltung
performance	Leistung

Tabelle 3: Liste der gültigen Motive

Ist beim Attribut *motive* ein ungültiger Wert – nicht einer dieser fünf Werte – gesetzt, wird der Eintrag ignoriert. Bei den Motivprofil Beschreibungen kann das dazu führen, dass die Beschreibungen nicht aktualisiert werden.

4.6.2 Entscheidungssituationen

Root-Element: <decisions>

- | | | |
|-------------------------------|---------------------|------------------------|
| • Entscheidung: | <decision> | Element |
| – Frage (Name): | question | Attribut |
| – gegebener Faktor: | <predefinedFactor> | Sub-Element |
| • Identifikation: | id | Attribut |
| – benutzer-erstellter Faktor: | <userdefinedFactor> | Optionales Sub-Element |
| • Identifikation: | id | Attribut |

Beispiel:

```
<decisions>
  <decision question="Should I buy a fancy sports car?">
    <predefinedFactor id="1" />
    <predefinedFactor id="4" />
    <userdefinedFactor id="5" />
  </decision>
</decisions>
```

4.6.2.1 Erklärungen

Die Tags *userdefinedFactor* sind nur vorhanden, wenn ein neuer Faktor vom Client zum Back-End-Server gesendet wird. Die Unterscheidung ist nötig, da die beiden Faktor-Arten separat nummeriert werden.

4.6.3 Entscheidungsfaktoren

Root-Element: <factors>

- | | | |
|-------------------------------|---------------------------|------------------------|
| • Entscheidungsfaktor: | <factor> | Element |
| – Identifikation: | id | Attribut |
| – Name des Faktors: | name | Attribut |
| – Beschreibung: | description | Optionales Attribut |
| – Erläuterungen: | <explanations> | Optionales Sub-Element |
| • Erläuterung Warnung: | <warningExplanation> | Optionales Sub-Element |
| • Erläuterung Negativ: | <doNotProceedExplanation> | Optionales Sub-Element |
| – Mappings: | <mappings> | Optionales Sub-Element |
| • Mapping soziale Sicherheit: | <socialMapping> | Optionales Sub-Element |
| • Mapping Neugier: | <arousalMapping> | Optionales Sub-Element |
| • Mapping Macht: | <forceMapping> | Optionales Sub-Element |
| • Mapping Geltung: | <prestigeMapping> | Optionales Sub-Element |
| • Mapping Leistung: | <performanceMapping> | Optionales Sub-Element |

Beispiel:

```
<factors>
  <factor id="1" name="money" description="Do you have enough money for
  another activity?">
    <explanations>
      <warningExplanation>you might be short with it</warningExplanation>
      <doNotProceedExplanation>you have too less</doNotProceedExplanation>
    </explanations>
  </factor>
  <factor id="2" name="work">
    <mappings>
      <socialMapping>0.2</socialMapping>
      <arousalMapping>0</arousalMapping>
      <forceMapping>-0.1</forceMapping>
      <prestigeMapping>0.3</prestigeMapping>
      <performanceMapping>0</performanceMapping>
    </mappings>
  </factor>
</factors>
```

4.6.3.1 Erklärungen

Die Erklärungen (Warnung oder Negativ) sind nur in den vorgegebenen Faktoren, d.h. in Faktoren, die vom Back-End-Server zu den Clients geschickt werden, vorhanden.

Die Gewichtungs-Attribute sind nur in vorgegebenen Faktoren, d.h. Faktoren, die vom Back-End-Server an die Clients geschickt werden, vorhanden. Bei Faktoren, die von Benutzern erstellt wurden, wird ein Default-Mapping verwendet.

4.6.4 Motivprofil-Fragebogen

Root-Element: <questionnaire>

- | | | |
|------------------------------|------------|----------|
| • Frage: | <question> | Element |
| – Frage-Nummer/-Reihenfolge: | number | Attribut |
| – Motiv Zugehörigkeit: | motive | Attribut |

Beispiel:

```
<questionnaire>
  <question number="1" motive="arousal">Question-Text</question>
  <question number="2" motive="social">Question-Text</question>
  ...
</questionnaire>
```

4.6.4.1 Erklärungen

Der Fragebogen zum Erstellen des Motivprofils kann auf den Clients aktualisiert werden, indem das entsprechende File vom Back-End-Server heruntergeladen wird.

Eine Frage wird einem der 5 Motive zugeordnet. Jedem Motiv werden aber mehrere Fragen zugeordnet, um daraus einen aussagekräftigen Faktor zu bilden. Da die Reihenfolge des Fragebogens relevant ist, wird diese ebenfalls als Attribut gesetzt.

4.6.5 Motivprofil Beschreibungen

Root-Element: <motiveProfileSpecifications>

- Durchschnitt: <average> Element
 - Motiv Zugehörigkeit: motive Attribut
 - Grenze "Durchschnitt": averageThreshold Attribut
 - Grenze "über Durchschnitt": aboveAverageThreshold Attribut
 - Erläuterung über Durchschnitt: <aboveExplanation> Sub-Element
 - Erläuterung Durchschnitt: <averageExplanation> Sub-Element
 - Erläuterung unter Durchschnitt: <belowExplanation> Sub-Element

Beispiel:

```
<motiveProfileSpecifications>
  <average motive="social" averageThreshold="2.0"
    aboveAverageThreshold="4.0">
    <aboveExplanation>Being close with people close to you is important to
    you.</aboveExplanation>
    <averageExplanation> ... </averageExplanation>
    <belowExplanation> ... </belowExplanation>
  </average>
  ...
</motiveProfileSpecifications>
```

4.6.5.1 Erklärungen

Die Beschreibung der erzielten Werte im Motivprofil kann auf den Clients aktualisiert werden, indem die entsprechende Datei vom Back-End-Server heruntergeladen wird.

Zu jedem Motiv gibt es 3 verschiedene Beschreibungen: Über dem Durchschnitt, durchschnittlich und unter dem Durchschnitt. Der erreichte Wert im Motivprofil wird mit den Grenzwerten, die in der Datei auf dem Server hinterlegt sind, verglichen. Danach kann die entsprechende Beschreibung angezeigt werden.

4.6.6 Ratingtabelle

Root-Element: <ratingTable>

- Tabelleneintrag: <entry> Element
 - Faktor-Wert: factorValue Attribut
 - Motivprofil-Wert: profileValue Attribut
 - Korrekturbetrag: correctionAmount Attribut

Beispiel:

```
<ratingTable>
  <entry factorValue="1" profileValue="1" correctionAmount="-0.4"/>
  <entry factorValue="1" profileValue="2" correctionAmount="-0.3"/>
  <entry factorValue="1" profileValue="3" correctionAmount="0"/>
  <entry factorValue="1" profileValue="4" correctionAmount="0.1"/>
  <entry factorValue="1" profileValue="5" correctionAmount="0.3"/>
  ...
</ratingTable>
```

4.6.6.1 Erklärungen

Die Tabelle zum Bestimmen (Lookup) des Korrekturbetrags für die Entscheidungsregel kann aktualisiert werden, indem das Ratingtabellen-File vom Back-End-Server heruntergeladen wird.

4.7 Wireframes

The wireframe for MainActivity shows a title bar with the text "Decisio" and a hamburger menu icon. Below the title bar, there are two main action items: "Make decision" with a question mark icon and "Create decision" with a plus sign icon.

The wireframe for SociodemographicDataActivity features a back navigation arrow and the title "Sociodemographic Data". It contains four form fields: "Gender" with radio buttons for "Male" (selected) and "Female"; "Age" with a text input field containing "28"; "Income" with a dropdown menu showing "2000.- ... 3000.-"; and "Education" with a dropdown menu showing "University".

The wireframe for QuestionnaireActivity has a back navigation arrow and the title "Motivation profile". It displays four questionnaire items, each with five radio buttons for a 1-5 scale: "In a team I like to take the lead.", "I wish i were a famous person.", "In a team I like to take the lead.", and "It is important to me to be successful in life."

Abbildung 20: MainActivity

Bietet die Hauptnavigation in der App an.

Optionen in der Action Bar:

- Show motive profile: Zeigt das Motivprofil des Benutzers an.
- Edit motive profile: Öffnet den Motivprofil-Fragebogen, um allfällige Änderungen anzubringen.
- Edit personal information: Einsehen und bearbeiten der soziodemographischen Daten.
- About: Copyright- und Lizenz-Infos.

Abbildung 21: SociodemographicDataActivity

Zum Ausfüllen der anonymen, soziodemografischen Daten.

Für das Einkommen und den Abschluss werden nur vordefinierte Werte zugelassen, enthalten jedoch eine Option "no answer".

Die Back-Navigation führt zurück in die MainActivity.

Abbildung 22: QuestionnaireActivity

Dient zum Ausfüllen des Motivprofil-Fragebogens zur Erstellung des eigenen Motivprofils.

Die Back-Navigation führt zurück in die MainActivity.



Abbildung 23: ChooseDecisionActivity

Activity zum Ausführen von Entscheidungen. Hier werden die vorhandenen Entscheidungssituationen aufgelistet. Wird eine Entscheidungssituation ausgewählt, wird die *MakeDecisionActivity* geöffnet.

Zum Erstellen einer neuen Entscheidung, kann der Plus-Button in der Action Bar verwendet werden.

Die Back-Navigation führt zurück in die *MainActivity*.

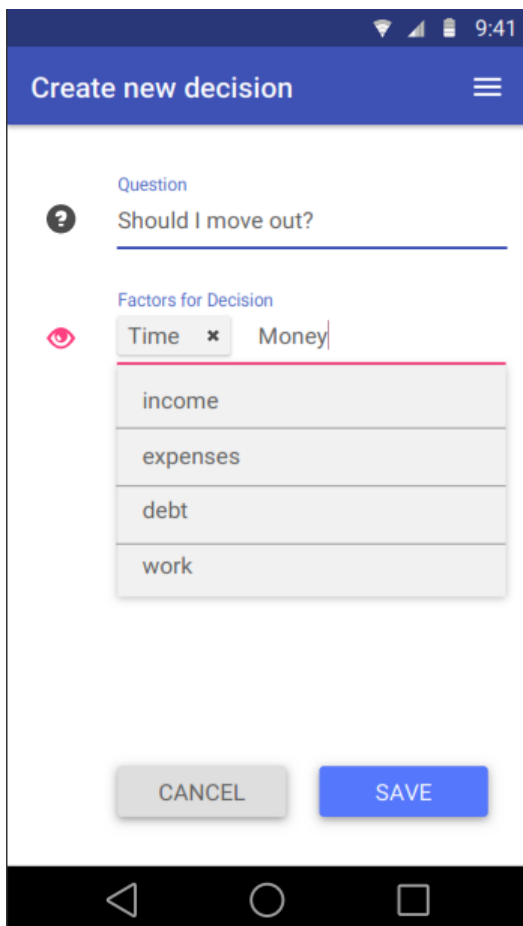


Abbildung 24: CreateDecisionActivity

Dient zum Erstellen einer neuen Entscheidung. Dabei können auch direkt neue Entscheidungsfaktoren erstellt werden.

Dieses Design stammt aus der Vorarbeit der ZHAW [2]. Das Textfeld zur Eingabe der Entscheidungsfaktoren verfügt über Auto-Vervollständigung, welches eine Liste mit passenden Vorschlägen im Bereich unterhalb des Textfeldes anzeigt.

Save-Button: Erstellt und speichert die neue Entscheidung.

Cancel-Button: Bricht die Aktion ab und navigiert zurück zur *MainActivity*.

Die Back-Navigation führt zurück in die *MainActivity*.

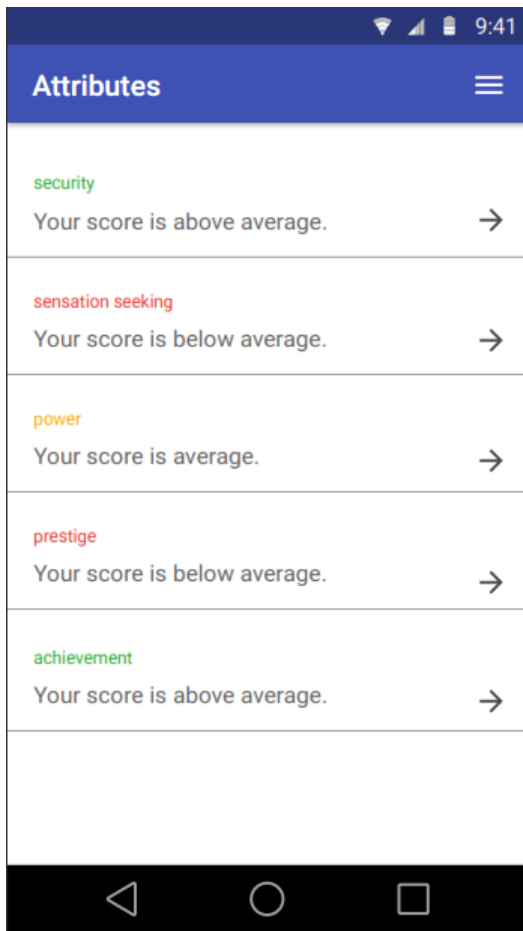


Abbildung 25: MotiveProfileActivity

Wird zur Visualisierung des Motivprofils verwendet. Darin kann der Benutzer die einzelnen Motive öffnen, um eine Detailbeschreibung zum erzielten Wert zu erhalten.

Dieses Design stammt aus der Vorarbeit der ZHAW [2]. Den Titel Attributes in der Action Bar werden wir zu "Motive profile" umbenennen und zusätzlich eine Back-Navigation, die zur *MainActivity* führt, einbauen.

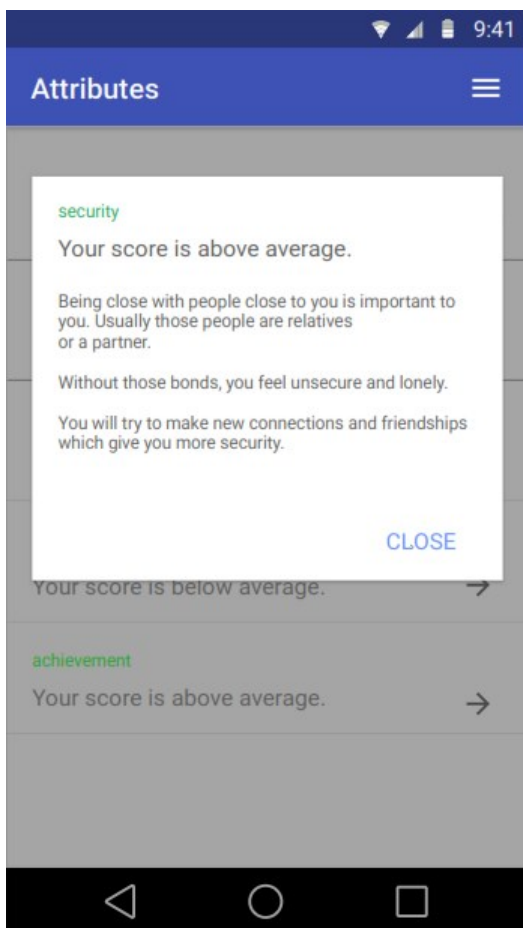


Abbildung 26: MotiveDetailDialogue

Wird zur Detailbeschreibung des erzielten Motivprofil-Wertes gebraucht.

Dieses Design stammt aus der Vorarbeit der ZHAW [2].

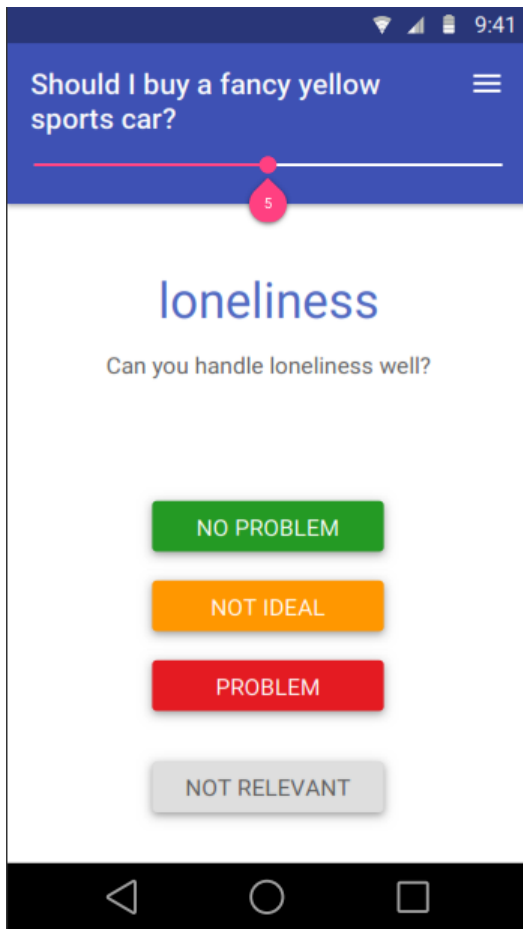


Abbildung 27: MakeDecisionActivity

Dient zur Einstufung (Rating) der zur Entscheidungssituation gehörenden Entscheidungsfaktoren.

Sind alle Faktoren bewertet, wird die *DecisionResultActivity* angezeigt.

Dieses Design stammt aus der Vorarbeit der ZHAW [2]. Zu diesem Design wird es aber auch noch eine Back-Navigation geben, die zur *ChooseDecisionActivity* zurück führt und die aktuelle Entscheidung abbricht.

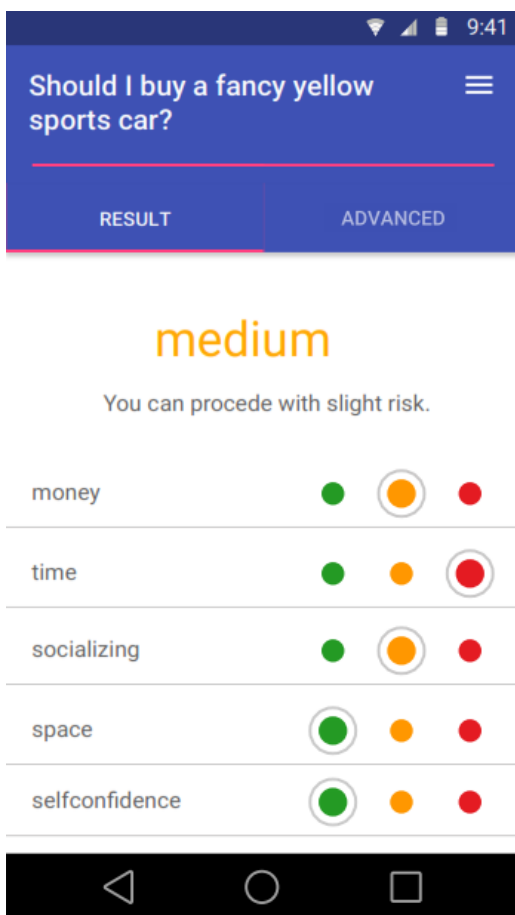


Abbildung 28: DecisionResultActivity

Gibt dem Benutzer die berechnete Handlungsempfehlung und bietet eine Übersicht zu den ausgefüllten Entscheidungsfaktoren an.

Dieses Design stammt aus der Vorarbeit der ZHAW [2].

Zudem wird eine Back-Navigation in der Action Bar, wie in den anderen Activities, eingefügt. Sie führt zur *ChooseDecisionActivity* zurück.

4.8 Systemtest

4.8.1 Systemtestspezifikation

4.8.1.1 Vorbereitung und allgemeine Vorbedingungen

Die App muss vor dem Test auf einem Gerät (kein Emulator) installiert worden sein. Zudem muss das Gerät Zugang zum Internet haben.

4.8.1.2 Test Cases

4.8.1.2.1 App starten

Vorbedingungen: App wurde neu auf dem Gerät installiert und besitzt noch keine Daten im Speicher.

Schritt	Beschreibung	Erwartetes Resultat
1	Starten Sie die App.	System öffnet die App und zeigt den Disclaimer Text in einem Dialog an.
2	Lehnen Sie den Disclaimer Text ab.	App wird geschlossen.
3	Starten Sie die App erneut.	System öffnet die App und zeigt den Disclaimer Text erneut in einem Dialog an.
4	Schliessen Sie die App und schalten Sie jegliche Internetverbindungsmöglichkeiten (WLAN, mobile Daten, ...) ab. Starten Sie die App ein weiteres Mal.	System schliesst und öffnet die App. Disclaimer Text wird in einem Dialog angezeigt.
5	Akzeptieren Sie den Disclaimer Text.	System zeigt eine Informationsmeldung an, dass einmal eine Internetverbindung benötigt wird und schliesst die App.
6	Schalten Sie eine Internetverbindungsmöglichkeit ein und starten Sie die App.	System startet die App lädt die neuesten Einstellungen vom Server herunter. Der Disclaimer Text wird nicht angezeigt. System fordert den Benutzer zur Eingabe der Soziodemografischen Daten auf.
7	Führen Sie den Test Case "Soziodemografische Daten erfassen" aus.	
8	Schliessen Sie die App und schalten Sie jegliche Internetverbindungsmöglichkeiten (WLAN, mobile Daten, ...) ab. Starten Sie die App ein weiteres Mal.	System schliesst und öffnet die App. System fordert den Benutzer direkt zum Ausfüllen des Motivprofil-Fragebogens auf.
9	Führen Sie den Test Case "Motivprofil erstellen" aus.	System zeigt nun die Startseite der App an.
10	Schliessen Sie die App und starten Sie die App ein weiteres Mal.	System zeigt direkt die Startseite der App an.

4.8.1.2.2 Soziodemografische Daten erfassen

Vorbedingungen:

- Test Case "App starten" wurde bis Schritt 7, jedoch noch nicht weiter ausgeführt.
- Für diesen Test Case ist der Zugriff auf den Back-End-Server erforderlich.
- Das Gerät hat eine Internetverbindungsmöglichkeit eingeschaltet.

Schritt	Beschreibung	Erwartetes Resultat
1	Starten Sie die App, falls noch nicht erledigt.	System fordert den Benutzer zum Eingeben der Soziodemografischen Daten auf.
2	Geben Sie als Geschlecht 'Männlich' und als Jahrgang '1980' ein. Versuchen Sie die Eingaben zu speichern.	System zeigt Meldung an, dass noch nicht alle Daten eingegeben wurden. Back-Navigation nicht möglich.
3	Wählen Sie als Abschluss die erste Option. Versuchen Sie die Eingaben zu speichern.	System zeigt Meldung an, dass noch nicht alle Daten eingegeben wurden. Back-Navigation nicht möglich.
4	Wählen Sie als Einkommensklasse die zweite Option. Versuchen Sie die Eingaben zu speichern.	System akzeptiert die Eingaben und verlässt die Soziodemografischen Daten.
5	Öffnen Sie die zum Benutzer gehörende Datei der soziodemografischen Daten auf dem Back-End-Server.	Datei der soziodemografischen Daten des Benutzers enthält die soeben eingegebenen Daten.

4.8.1.2.3 Motivprofil erstellen

Vorbedingungen:

- Test Case "App starten" wurde bis Schritt 9, jedoch noch nicht weiter ausgeführt.
- Für Schritt 3 sind Informationen über den Fragebogen auf dem Server notwendig.
- Das Gerät hat eine Internetverbindungsmöglichkeit eingeschaltet.

Schritt	Beschreibung	Erwartetes Resultat
1	Starten Sie die App, falls noch nicht erledigt.	System fordert den Benutzer zum Ausfüllen des Motivprofil-Fragebogens auf.
2	Beantworten Sie einige, jedoch noch nicht alle Fragen des Fragebogens und versuchen Sie die Eingaben zu speichern.	System zeigt Meldung an, dass noch nicht alle Fragen beantwortet wurden. Back-Navigation nicht möglich.
3	Beantworten Sie alle Fragen des Fragebogens wie folgt: <ul style="list-style-type: none">- Alle Fragen zu Motiven 'performance' und 'arousal' mit Wert: 1- Alle Fragen zu Motiven 'social' und 'force' mit Wert: 5- Alle Fragen zu Motiv 'prestige' mit einem Wert im Durchschnittsbereich für 'prestige'. Versuchen Sie die Eingaben zu speichern.	System akzeptiert die Eingaben und zeigt das daraus erstellte Motivprofil an.
4	Überprüfen Sie die Werte des Motivprofils.	<ul style="list-style-type: none">- Die Motive 'performance' und 'arousal' sind als Rot und unter dem Durchschnitt angezeigt.- Die Motive 'social' und 'force' sind als Grün und über dem Durchschnitt angezeigt.- Das Motiv 'prestige' ist als Orange und durchschnittlich angezeigt.
5	Akzeptieren Sie das Motivprofil.	System zeigt Startseite der App an.
6	Öffnen Sie die zum Benutzer gehörende Datei des Motivprofils auf dem Back-End-Server.	Datei des Motivprofils des Benutzers enthält die in Schritt 3 eingegebenen Einstufungen.

4.8.1.2.4 Entscheidung fällen

Vorbedingungen:

- Disclaimer akzeptiert, Soziodemografische Daten eingegeben und Motivprofil erstellt.
- Für Schritt 5 sind Informationen über die Entscheidungssituation auf dem Server notwendig.

Schritt	Beschreibung	Erwartetes Resultat
1	Starten Sie die App, falls noch nicht erledigt.	System öffnet die Startseite der App.
2	Wählen Sie die Option, um eine Entscheidung zu fällen.	System zeigt den Katalog der vorhandenen Entscheidungssituationen an.
3	Wählen Sie eine vorgegebene Entscheidungssituation aus.	System zeigt Seite zur Einstufung des ersten Entscheidungsfaktors an.
4	Stufen Sie den Faktor als 'Orange' ein.	System zeigt Seite zur Einstufung des nächsten Entscheidungsfaktors an.
5	Stufen Sie die nächsten Faktoren wie folgt ein: <ul style="list-style-type: none">- Einen Faktor als 'Grün'.- 2 Faktoren als 'Rot'.- Einen Faktor als 'Nicht relevant'.- Alle nachfolgenden Faktoren als 'Orange'.	System führt den Benutzer durch die zur Entscheidungssituation gehörenden Entscheidungsfaktoren. Die Reihenfolge entspricht der Reihenfolge in der Konfigurationsdatei.
6	Überprüfen Sie die Handlungsempfehlung und die Zusammenfassung der Faktoreinstufungen.	System zeigt negative Handlungsempfehlung an. Die entsprechenden Faktoren werden korrekt angezeigt (Grün/Orange/Rot). Der als 'Nicht relevant' eingestufte Faktor erscheint nicht in der Zusammenfassung.
7	Führen Sie eine Back-Navigation durch.	System zeigt Katalog der vorhandenen Entscheidungssituationen an. Die Einstufungen der Entscheidungsfaktoren können nicht verändert werden.
8	Überprüfen Sie das Datum, wann die soeben ausgeführte Entscheidung das letzte Mal ausgeführt wurde.	Das Datum der letzten Ausführung wurde auf das heutige Datum gesetzt.

4.8.1.2.5 Neue Entscheidung erfassen

Vorbedingungen: Disclaimer akzeptiert, Soziodemografische Daten eingegeben und Motivprofil erstellt.

Schritt	Beschreibung	Erwartetes Resultat
1	Starten Sie die App, falls noch nicht erledigt.	System öffnet die Startseite der App.
2	Wählen Sie die Option, um eine neue Entscheidung zu erfassen.	System öffnet Seite, um neue Entscheidung zu erfassen.
3	Geben Sie als Frage 'Should I buy new clothes' ein.	System zeigt die zur Auswahl stehenden Faktoren mit einer "Auto-Completion" Funktion an.
4	Wählen Sie mindestens 3 Faktoren mit der "Auto-Completion" Funktion aus.	System zeigt die ausgewählten Faktoren in der Auswahlreihenfolge an.
5	Versuchen Sie die erfasste Entscheidung zu speichern.	System speichert die neu erfasste Entscheidungssituation.
6	Führen Sie die neu erstellte Entscheidung durch.	System zeigt die neu erfasste Entscheidungssituation im Katalog der vorhandenen Entscheidungssituationen an. System führt den Benutzer durch die zur Entscheidungssituation gehörenden Entscheidungsfaktoren. Die Reihenfolge entspricht der Auswahlreihenfolge.

4.8.1.2.6 Neuen Entscheidungsfaktor erfassen

Vorbedingungen: Disclaimer akzeptiert, Soziodemografische Daten eingegeben und Motivprofil erstellt.

Schritt	Beschreibung	Erwartetes Resultat
1	Starten Sie die App, falls noch nicht erledigt.	System öffnet die Startseite der App.
2	Wählen Sie die Option, um eine neue Entscheidung zu erfassen.	System öffnet Seite, um neue Entscheidung zu erfassen.
3	Geben Sie als Frage 'Should I get out to drink a beer' ein.	System zeigt die zur Auswahl stehenden Faktoren mit einer "Auto-Completion" Funktion an.
4	Geben Sie als Faktor 'loving beer' ein.	System zeigt den Faktor als neuen Faktor für die Entscheidungssituation an.
5	Speichern Sie die Entscheidungssituation.	System speichert die Entscheidungssituation.
6	Starten Sie die neu erstellte Entscheidung.	System zeigt die neu erfasste Entscheidungssituation im Katalog der vorhandenen Entscheidungssituationen an. System zeigt dem Benutzer den neu erstellten Entscheidungsfaktor zur Einschätzung an.
7	Stufen sie den neu erstellten Entscheidungsfaktor als 'Orange' ein.	System zeigt Handlungsempfehlung und Orange-Einstufung des Faktors an. Als Detail-Erläuterung wird der Standardtext für Orange-Einstufungen angezeigt.
8	Führen Sie die Entscheidung erneut durch. Stufen Sie den Faktor als 'Rot' ein.	System zeigt Handlungsempfehlung und Rot-Einstufung des Faktors an. Als Detail-Erläuterung wird der Standardtext für Rot-Einstufungen angezeigt.

4.8.1.2.7 CRUD vorgegebene Entscheidung

Vorbedingungen:

- Disclaimer akzeptiert, Soziodemografische Daten eingegeben und Motivprofil erstellt.
- Für diesen Test Case ist der Zugriff auf den Back-End-Server erforderlich.
- Das Gerät hat eine Internetverbindungsmöglichkeit eingeschaltet.

Schritt	Beschreibung	Erwartetes Resultat
1	Ändern Sie die Konfigurationsdatei auf dem Back-End-Server, um eine neue vorgegebene Entscheidung hinzuzufügen: "Should I buy myself a birthday gift?". Geben Sie als Faktoren einige vordefinierte Faktoren an.	Entscheidungssituation in Konfigurationsdatei auf dem Back-End-Server hinzugefügt.
2	Starten Sie die App.	Die neuesten Daten werden vom Back-End-Server heruntergeladen.
3	Wählen Sie die Option, um eine Entscheidung zu fällen. Starten Sie die neu erstellte vorgegebene Entscheidungssituation und überprüfen Sie, ob die richtigen Entscheidungsfaktoren zur Einschätzung angezeigt werden.	Die erstellte vorgegebene Entscheidungssituation wird im Katalog der vorhandenen Entscheidungssituationen angezeigt. Die Entscheidung enthält die definierten Entscheidungsfaktoren.
4	Öffnen Sie die Konfigurationsdatei auf dem Back-End-Server, um die neue vorgegebene Entscheidungssituation zu ändern: - Ändern Sie die Frage auf: "Should I buy myself a Christmas gift?" - Entfernen Sie einen oder mehrere Entscheidungsfaktoren.	Entscheidungssituation in Konfigurationsdatei auf dem Back-End-Server geändert.
5	Schliessen Sie die App und Starten Sie sie erneut.	Die neuesten Daten werden vom Back-End-Server heruntergeladen.
6	Wählen Sie die Option, um eine Entscheidung zu fällen. Starten Sie die soeben geänderte vorgegebene Entscheidungssituation und überprüfen Sie, ob die entfernten Entscheidungsfaktoren nicht mehr zur Einschätzung angezeigt werden.	Die geänderte vorgegebene Entscheidungssituation enthält den geänderten Frage Text. Die entfernten Entscheidungsfaktoren werden nicht mehr zur Einschätzung angezeigt.
7	Öffnen Sie die Konfigurationsdatei auf dem Back-End-Server und löschen Sie die neue vorgegebene Entscheidungssituation.	Entscheidungssituation in Konfigurationsdatei auf dem Back-End-Server gelöscht.
8	Schliessen Sie die App und Starten Sie sie erneut.	Die neuesten Daten werden vom Back-End-Server heruntergeladen.
9	Wählen Sie die Option, um eine Entscheidung zu fällen.	Die soeben entfernte vorgegebene Entscheidungssituation wird nicht mehr im Katalog der vorhandenen Entscheidungssituationen angezeigt.

4.8.1.2.8 CRUD vorgegebener Entscheidungsfaktor

Vorbedingungen:

- Disclaimer akzeptiert, Soziodemografische Daten eingegeben und Motivprofil erstellt.
- Für diesen Test Case ist der Zugriff auf den Back-End-Server erforderlich.
- Das Gerät hat eine Internetverbindungsmöglichkeit eingeschaltet.

Schritt	Beschreibung	Erwartetes Resultat
1	Ändern Sie die Konfigurationsdatei auf dem Back-End-Server, um einen neuen vorgegebenen Entscheidungsfaktor hinzuzufügen: "soccer" - Setzen Sie als Beschreibung: "Do you like soccer?" - Setzen Sie als Detailerklärung für Orange: "You might not like soccer enough." - Setzen Sie als Detailerklärung für Rot: "You don't like soccer enough." - Setzen Sie als Mapping 0 für alle Motive.	Neuen Entscheidungsfaktor in Konfigurationsdatei auf Back-End-Server hinzugefügt.
2	Fügen Sie eine neue Entscheidungssituation in die vorgegebenen Entscheidungssituationen auf dem Back-End-Server ein und setzen Sie den obigen Faktor als dessen einzigen Entscheidungsfaktor. - Setzen Sie als Fragetext: "Should I go out to watch a soccer match?"	Neue Entscheidungssituation mit dem neuen Entscheidungsfaktor auf Back-End-Server hinzugefügt.
3	Starten Sie die App.	Die neuesten Daten werden vom Back-End-Server heruntergeladen.
4	Starten Sie die neu erstellte Entscheidungssituation.	Der neu erstellte Entscheidungsfaktor wird zur Einstufung angezeigt.
5	Überprüfen Sie den Namen und die Beschreibung des Entscheidungsfaktors.	Der Name und die Beschreibung entsprechen der Konfiguration auf dem Server.
6	Stufen Sie den Faktor als Orange ein. Überprüfen Sie die Detailerklärung.	System zeigt positive Handlungsempfehlung an. Die Detailerklärung entspricht der Konfiguration auf dem Server.
7	Führen Sie die Entscheidungssituation erneut durch, aber Stufen Sie den Faktor dieses Mal als Rot ein. Überprüfen Sie die Detailerklärung.	System zeigt negative Handlungsempfehlung an. Die Detailerklärung entspricht der Konfiguration auf dem Server.
8	Ändern Sie die Beschreibung des Faktors auf dem Back-End-Server auf: "Do you like watching a soccer match?"	Beschreibung des Faktors auf dem Back-End-Server geändert.
9	Schliessen Sie die App und starten Sie sie erneut.	Die neuesten Daten werden vom Back-End-Server heruntergeladen.

Schritt	Beschreibung	Erwartetes Resultat
10	Starten Sie die Entscheidungssituation erneut. Überprüfen Sie die Beschreibung des Faktors.	Beschreibung des Faktors entspricht der geänderten Konfiguration auf dem Server.
11	Löschen Sie den Faktor wieder aus der Konfigurationsdatei auf dem Back-End-Server. Fügen Sie einen vorgegebenen Faktor in die in Schritt 2 erstellte Entscheidungssituation ein.	Faktor aus Konfigurationsdatei auf dem Back-End-Server gelöscht. Vorgegebenen Faktor in Entscheidung definiert.
12	Schliessen Sie die App und starten Sie sie erneut.	Die neuesten Daten werden vom Back-End-Server heruntergeladen.
13	Starten Sie die Entscheidungssituation erneut. Überprüfen Sie den einzustufenden Faktor.	Der definierte vorgegebene Entscheidungsfaktor wird zur Einstufung angezeigt.
14	Löschen Sie die in Schritt 2 erstellte Entscheidungssituation auf dem Back-End-Server.	Entscheidungssituation aus der Konfigurationsdatei auf dem Back-End-Server gelöscht.

4.8.1.2.9 CRUD Benutzer-erstellte Entscheidungen

Vorbedingungen:

- Disclaimer akzeptiert, Soziodemografische Daten eingegeben und Motivprofil erstellt.
- Für diesen Test Case ist der Zugriff auf den Back-End-Server erforderlich.
- Das Gerät hat eine Internetverbindungsmöglichkeit eingeschaltet.

Schritt	Beschreibung	Erwartetes Resultat
1	Starten Sie die App.	System öffnet die Startseite der App.
2	Erstellen Sie eine neue Entscheidungssituation in der App und notieren Sie sich den Fragetext und die ausgewählten Entscheidungsfaktoren.	Neue Entscheidungssituation in der App erfasst.
3	Öffnen Sie die zum Benutzer gehörende Datei der erstellten Entscheidungssituationen auf dem Back-End-Server.	Datei der erstellten Entscheidungssituationen des Benutzers enthält die soeben erstellte Entscheidungssituation.
4	Überprüfen Sie, ob der Fragetext und die ausgewählten Entscheidungsfaktoren korrekt in der Datei eingetragen wurden.	Fragetext und Entscheidungsfaktoren sind korrekt in der Datei eingetragen.

4.8.1.2.10 CRUD Benutzer-erstellte Entscheidungsfaktoren

Vorbedingungen:

- Disclaimer akzeptiert, Soziodemografische Daten eingegeben und Motivprofil erstellt.
- Für diesen Test Case ist der Zugriff auf den Back-End-Server erforderlich.
- Das Gerät hat eine Internetverbindungsmöglichkeit eingeschaltet.

Schritt	Beschreibung	Erwartetes Resultat
1	Starten Sie die App.	System öffnet die Startseite der App.
2	Erstellen Sie eine neue Entscheidungssituation und erstellen Sie darin einen neuen Entscheidungsfaktor. Notieren Sie sich den Fragetext und den Namen des neu erstellten Entscheidungsfaktors.	Neue Entscheidungssituation und neuen Entscheidungsfaktor in der App erfasst.
3	Öffnen Sie die zum Benutzer gehörende Datei der erstellten Entscheidungsfaktoren auf dem Back-End-Server.	Datei der erstellten Entscheidungsfaktoren des Benutzers enthält den soeben erstellten Entscheidungsfaktor mit korrektem Namen.
4	Öffnen Sie die zum Benutzer gehörende Datei der erstellten Entscheidungssituationen auf dem Back-End-Server.	Datei der erstellten Entscheidungssituationen des Benutzers enthält die soeben erstellte Entscheidungssituation.
5	Überprüfen Sie, ob der Fragetext und der neu erstellte Entscheidungsfaktor korrekt in der Datei eingetragen wurden.	Fragetext und der neu erstellte Entscheidungsfaktor sind korrekt in der Datei eingetragen.

4.8.1.2.11 Motivprofil-Fragebogen bearbeiten

Vorbedingungen:

- Disclaimer akzeptiert, Soziodemografische Daten eingegeben und Motivprofil erstellt.
- Für diesen Test Case ist der Zugriff auf den Back-End-Server erforderlich.
- Das Gerät hat eine Internetverbindungsmöglichkeit eingeschaltet.

Schritt	Beschreibung	Erwartetes Resultat
1	Öffnen Sie die Konfigurationsdatei auf dem Back-End-Server, um eine neue Frage im Motivprofil-Fragebogen zu erstellen. Fügen Sie die Frage: "I enjoy going out with friends to have a drink.", als letzte Frage im Fragebogen ein und definieren Sie "social" als Motiv.	Frage für Motivprofil-Fragebogen in Konfigurationsdatei auf dem Back-End-Server hinzugefügt.
2	Starten Sie die App.	Die neuesten Daten werden vom Back-End-Server heruntergeladen.
3	Wählen Sie die Option, um Ihr Motivprofil zu bearbeiten. Überprüfen Sie, ob die neu erstellte Frage als letzte Frage im Fragebogen erscheint.	System öffnet den Motivprofil-Fragebogen. Neu erstellte Frage erscheint als letzte Frage im Motivprofil-Fragebogen. Frage enthält noch keine Einstufung vom Benutzer.
4	Versuchen Sie ohne Einstufung auf die Frage das Motivprofil zu speichern.	Motivprofil kann nicht erstellt werden, da nicht alle Fragen ausgefüllt sind.
5	Geben Sie eine Einstufung zu der Frage an und speichern Sie das Motivprofil.	Motivprofil erstellt und gespeichert.
6	Öffnen Sie den Motivprofil-Fragebogen erneut.	Die Einstufung zur neu erstellten Frage wird korrekt angezeigt.
7	Bearbeiten Sie die neu erstellte Frage auf dem Back-End-Server: - Ändern Sie die Frage auf: "I enjoy going out with friends to have fun." - Ändern Sie die Nummer, dass die Frage die zweithöchste Nummer hat. Belassen Sie die Frage an letzter Stelle in der Datei.	Frage für Motivprofil-Fragebogen in Konfigurationsdatei auf dem Back-End-Server geändert.
8	Schliessen Sie die App und starten Sie sie erneut.	Die neuesten Daten werden vom Back-End-Server heruntergeladen.
9	Öffnen Sie den Motivprofil-Fragebogen.	Die bearbeitete Frage erscheint an zweitletzter Stelle und enthält den definierten Fragetext, jedoch keine Einstufung vom Benutzer.
10	Öffnen Sie die Konfigurationsdatei auf dem Back-End-Server und löschen Sie die neu erstellte Frage wieder aus dem Motivprofil-Fragebogen.	Frage für Motivprofil-Fragebogen aus Konfigurationsdatei auf dem Back-End-Server gelöscht.
11	Schliessen Sie die App und starten Sie sie erneut.	Die neuesten Daten werden vom Back-End-Server heruntergeladen.
12	Öffnen Sie den Motivprofil-Fragebogen.	Die Frage erscheint nicht mehr im Motivprofil-Fragebogen.

4.8.1.2.12 Soziodemografische Daten bearbeiten

Vorbedingungen:

- Disclaimer akzeptiert, Soziodemografische Daten eingegeben und Motivprofil erstellt.
- Für diesen Test Case ist der Zugriff auf den Back-End-Server erforderlich.
- Das Gerät hat eine Internetverbindungsmöglichkeit eingeschaltet.

Schritt	Beschreibung	Erwartetes Resultat
1	Starten Sie die App.	System öffnet die Startseite der App.
2	Wählen Sie die Option, um ihre soziodemografischen Daten zu bearbeiten.	System öffnet die Seite der soziodemografischen Daten.
3	Ändern Sie alle Daten und notieren Sie sich die geänderten Werte.	Soziodemografische Daten geändert.
4	Speichern Sie die Daten.	Soziodemografische Daten gespeichert. System öffnet die Startseite der App.
5	Wählen Sie die Option, um ihre soziodemografischen Daten zu bearbeiten.	System öffnet die Seite der soziodemografischen Daten.
6	Überprüfen Sie, ob Ihre geänderten Werte erfolgreich gespeichert und geladen wurden.	Soziodemografische Daten entsprechen den geänderten Werten.
7	Öffnen Sie die zum Benutzer gehörende Datei der soziodemografischen Daten auf dem Back-End-Server.	Datei der soziodemografischen Daten des Benutzers enthält die soeben geänderten Daten.

4.8.1.2.13 Motivprofil bearbeiten

Vorbedingungen:

- Disclaimer akzeptiert, Soziodemografische Daten eingegeben und Motivprofil erstellt.
- Für Schritt 3 sind Informationen über den Fragebogen auf dem Server notwendig.
- Für diesen Test Case ist der Zugriff auf den Back-End-Server erforderlich.
- Das Gerät hat eine Internetverbindungsmöglichkeit eingeschaltet.

Schritt	Beschreibung	Erwartetes Resultat
1	Starten Sie die App.	System öffnet die Startseite der App.
2	Wählen Sie die Option, um Ihr Motivprofil zu bearbeiten.	System öffnet den Motivprofil-Fragebogen. Der Fragebogen enthält die letzten Einstufungen des Benutzers.
3	Ändern Sie die Einstufungen aller Fragen zum Motiv "performance" auf 5 und speichern Sie das Motivprofil.	Einstufungen geändert und gespeichert.
4	Wählen Sie erneut die Option, um ihr Motivprofil zu bearbeiten.	System öffnet den Motivprofil-Fragebogen.
5	Überprüfen Sie, ob Ihre geänderten Werte erfolgreich gespeichert und geladen wurden.	Die Einstufungen der Fragen entsprechen den geänderten Werten.
6	Wählen Sie die Option, um Ihr Motivprofil anzuzeigen.	System öffnet Motivprofil Ansicht.
7	Überprüfen Sie, dass Ihr Wert für "performance" auf Grün und über dem Durchschnitt angezeigt wird.	Wert für "performance" wird als Grün und über dem Durchschnitt angezeigt.
8	Öffnen Sie die zum Benutzer gehörende Datei des Motivprofils auf dem Back-End-Server.	Datei des Motivprofils des Benutzers enthält die in Schritt 3 eingegebenen Einstufungen.

4.8.1.2.14 Erläuterungen ändern

Vorbedingungen:

- Disclaimer akzeptiert, Soziodemografische Daten eingegeben und Motivprofil erstellt.
- Für diesen Test Case ist der Zugriff auf den Back-End-Server erforderlich.
- Erstellen Sie eine Kopie der Datei zur Bearbeitung der Erläuterungen des Motivprofils, die sich momentan auf dem Back-End-Server befindet, um sie nach diesem Testfall wiederherstellen zu können.
- Das Gerät hat eine Internetverbindungsmöglichkeit eingeschaltet.

Schritt	Beschreibung	Erwartetes Resultat
1	<p>Öffnen Sie die Konfigurationsdatei auf dem Back-End-Server, um die Erläuterungen zu den Motiven im Motivprofil zu bearbeiten:</p> <p>Ändern Sie die "unter dem Durchschnitt" Erläuterung für das Motiv "force" geringfügig (z.B. ein Wort einfügen).</p> <p>Ändern Sie auch die Schwellwerte für das Motiv "force" auf 6.0 und 6.5.</p>	Erläuterung und Schwellwerte in Konfigurationsdatei auf dem Back-End-Server geändert.
2	Starten Sie die App.	Die neuesten Daten werden vom Back-End-Server heruntergeladen.
3	Wählen Sie die Option, um Ihr Motivprofil anzuzeigen.	System öffnet die Motivprofil Ansicht.
4	Überprüfen Sie, dass Ihr Wert für "force" als Rot und unter dem Durchschnitt angezeigt wird.	Motivprofil Wert für "force" wird als Rot und unter dem Durchschnitt angezeigt.
5	Öffnen Sie die Detailansicht für das Motiv "force" und überprüfen Sie den angezeigten Detailtext.	System öffnet Detailansicht für "force". Detailtext für "force" entspricht der geänderten Erläuterung auf dem Back-End-Server.
6	Stellen Sie die Ursprungsdatei auf dem Back-End-Server wieder her.	Konfigurationsdatei zur Bearbeitung der Erläuterungen zum Motivprofil auf dem Back-End-Server wiederhergestellt.

4.8.1.2.15 Entscheidungsregel bearbeiten

Vorbedingungen:

- Disclaimer akzeptiert, Soziodemografische Daten eingegeben und Motivprofil erstellt.
- Für diesen Test Case ist der Zugriff auf den Back-End-Server erforderlich.
- Erstellen Sie eine Kopie der momentanen Dateien auf dem Back-End-Server: Ratingtabelle und Mapping der Entscheidungsfaktoren.
- Das Gerät hat eine Internetverbindungsmöglichkeit eingeschaltet.

Schritt	Beschreibung	Erwartetes Resultat
1	Wählen Sie eine vorgegebene Entscheidungssituation und zwei dafür ausschlaggebende vorgegebene Entscheidungsfaktoren aus. Ändern Sie das sich auf dem Server befindliche Mapping dieser Faktoren für alle Motive auf 0.	Mapping der vorgegebenen Entscheidungsfaktoren auf dem Back-End-Server geändert.
2	Öffnen Sie die Konfigurationsdatei auf dem Back-End-Server, um die Ratingtabelle zu bearbeiten: Ändern Sie alle Korrekturwerte, welche in der Faktor Spalte 0 aufweisen, auf den Wert "-0.2".	Korrekturwerte in der Ratingtabelle auf dem Back-End-Server geändert.
3	Starten Sie die App.	Die neuesten Daten werden vom Back-End-Server heruntergeladen.
4	Führen Sie die von Ihnen gewählte vorgegebene Entscheidungssituation aus: - Stufen Sie die von Ihnen ausgewählten Entscheidungsfaktoren als "Orange" ein. - Stufen Sie alle restlichen Faktoren als "Grün" ein.	System zeigt trotz zwei Orange eingestuften Faktoren eine positive Handlungsempfehlung (ohne Warnung) an.
5	Öffnen Sie erneut die Konfigurationsdatei auf dem Back-End-Server, um die Ratingtabelle zu bearbeiten: Ändern Sie alle Korrekturwerte, welche in der Faktor Spalte 0 aufweisen, auf den Wert "0.2".	Korrekturwerte in der Ratingtabelle auf dem Back-End-Server geändert.
6	Schliessen Sie die App und starten Sie sie erneut.	Die neuesten Daten werden vom Back-End-Server heruntergeladen.
7	Führen Sie die von Ihnen gewählte vorgegebene Entscheidungssituation aus: - Stufen Sie die von Ihnen ausgewählten Entscheidungsfaktoren als "Orange" ein. - Stufen Sie alle restlichen Faktoren auf "Grün" ein.	System zeigt mit zwei Orange eingestuften Faktoren eine negative Handlungsempfehlung an.
8	Stellen Sie die Ursprungsdateien auf dem Back-End-Server wieder her.	Konfigurationsdateien, um Ratingtabelle und das Mapping der Faktoren zu ändern, auf dem Back-End-Server wiederhergestellt.

4.8.2 Systemtest Protokoll: Release End of Construction 2

Testdatum: 09.12.2014

Getestete Version: Revision 167

Verwendete Geräte: Huawei honor U8860 mit Android 4.0.3

4.8.2.1 Test Case Durchführungen

4.8.2.1.1 Test Case: App starten

Schritt	Status	Fehler/Unschönheit
1	Bestanden	
2	Fehlgeschlagen	Wenn auf Bereich ausserhalb des Dialogs geklickt wird, oder der Back-Button gedrückt wird, wird die App nicht geschlossen und könnte trotzdem verwendet werden: <i>SysTestBug01</i>
3	Bestanden	
4	Bestanden	
5	Fehlgeschlagen	Wenn auf Bereich ausserhalb des Dialogs geklickt wird, oder der Back-Button gedrückt wird, wird die App nicht geschlossen und könnte trotzdem verwendet werden: <i>SysTestBug01</i>
6	Bestanden	
7	Bestanden	
8	Bestanden	
9	Bestanden	
10	Bestanden	

4.8.2.1.2 Test Case: Soziodemografische Daten erfassen

Schritt	Status	Fehler/Unschönheit
1	Bestanden	
2	Bestanden	
3	Bestanden	
4	Bestanden	
5	Bestanden	

4.8.2.1.3 Test Case: Motivprofil erstellen

Schritt	Status	Fehler/Unschönheit
1	Bestanden	
2	Bestanden	
3	Bestanden	
4	Bestanden	
5	Bestanden	
6	Bestanden	Auffälligkeit: Die Fragen werden Alphabetisch geordnet.

4.8.2.1.4 Test Case: Entscheidung fällen

Schritt	Status	Fehler/Unschönheit
1	Bestanden	
2	Bestanden	
3	Bestanden	
4	Bestanden	
5	Bestanden	
6	Bestanden	
7	Bestanden	
8	Bestanden	

4.8.2.1.5 Test Case: Neue Entscheidung erfassen

Dieser Test Case konnte nicht ausgeführt werden, da der (Priorität 2) Use Case "UC5: Neue Entscheidung erfassen" aus zeitlichen Gründen nicht umgesetzt werden konnte.

4.8.2.1.6 Test Case: Neuen Entscheidungsfaktor erfassen

Dieser Test Case konnte nicht ausgeführt werden, da der (Priorität 2) Use Case "UC5.1: Neuen Entscheidungsfaktor erfassen" aus zeitlichen Gründen nicht umgesetzt werden konnte.

4.8.2.1.7 Test Case: CRUD vorgegebene Entscheidung

Schritt	Status	Fehler/Unschönheit
1	Bestanden	
2	Bestanden	
3	Fehlgeschlagen	Die neu erstellte Entscheidung erscheint nicht im Client: <i>SysTestBug02</i>
4	Bestanden	
5	Bestanden	
6	Fehlgeschlagen	Die Entscheidung erscheint nicht im Client: <i>SysTestBug02</i>
7	Bestanden	
8	Bestanden	
9	Bestanden	

4.8.2.1.8 Test Case: CRUD vorgegebener Entscheidungsfaktor

Dieser Test Case wurde nicht vollständig ausgeführt, da auch hier der im Systemtest aufgetauchte Fehler *SysTestBug02* auftritt.

4.8.2.1.9 Test Case: CRUD Benutzer-erstellte Entscheidungen

Dieser Test Case konnte nicht ausgeführt werden, da der (Priorität 2) Use Case "UC5: Neue Entscheidung erfassen" aus zeitlichen Gründen nicht umgesetzt werden konnte.

4.8.2.1.10 Test Case: CRUD Benutzer-erstellte Entscheidungsfaktoren

Dieser Test Case konnte nicht ausgeführt werden, da der (Priorität 2) Use Case "UC5.1: Neuen Entscheidungsfaktor erfassen" aus zeitlichen Gründen nicht umgesetzt werden konnte.

4.8.2.1.11 Test Case: Motivprofil-Fragebogen bearbeiten

Schritt	Status	Fehler/Unschönheit
1	Bestanden	
2	Bestanden	
3	Fehlgeschlagen	Die neu erstellte Frage erscheint nicht im Fragebogen: <i>SysTestBug02</i>
4	Nicht möglich	Da Schritt 3 fehlgeschlagen.
5	Nicht möglich	Da Schritt 3 fehlgeschlagen.
6	Nicht möglich	Da Schritt 3 fehlgeschlagen.
7	Bestanden	
8	Bestanden	
9	Fehlgeschlagen	Die Frage erscheint nicht im Fragebogen: <i>SysTestBug02</i>
10	Bestanden	
11	Bestanden	
12	Bestanden	

4.8.2.1.12 Test Case: Soziodemografische Daten bearbeiten

Schritt	Status	Fehler/Unschönheit
1	Bestanden	
2	Bestanden	
3	Bestanden	
4	Bestanden	
5	Bestanden	
6	Bestanden	
7	Bestanden	

4.8.2.1.13 Test Case: Motivprofil bearbeiten

Schritt	Status	Fehler/Unschönheit
1	Bestanden	
2	Bestanden	
3	Bestanden	
4	Bestanden	
5	Bestanden	
6	Bestanden	
7	Bestanden	
8	Bestanden	

4.8.2.1.14 Test Case: Erläuterungen ändern

Schritt	Status	Fehler/Unschönheit
1	Bestanden	
2	Bestanden	
3	Bestanden	
4	Fehlgeschlagen	Motiv "force" wird als Grün und über dem Durchschnitt angezeigt: <i>SysTestBug02</i>
5	Fehlgeschlagen	Detailtext hat sich nicht geändert: <i>SysTestBug02</i>
6	Bestanden	

4.8.2.1.15 Test Case: Entscheidungsregel bearbeiten

Dieser Test Case wurde nicht vollständig ausgeführt, da auch hier der im Systemtest aufgetauchte Fehler *SysTestBug02* auftritt.

4.8.2.2 Verbesserungsmöglichkeiten

4.8.2.2.1 Bekannte Einschränkungen

Während der Projektdauer konnten nicht alle als Priorität 2 eingestuften Use Cases umgesetzt werden. Dies sind die Use Cases "UC5: Neue Entscheidung erfassen" und "UC5.1: Neuen Entscheidungsfaktor erfassen".

Alle als Priorität 1 eingestuften Use Cases konnten umgesetzt und getestet werden.

4.8.2.2.2 Gefundene Fehler

- *SysTestBug01*: Die Dialoge zum akzeptieren des Disclaimers und zur Information, dass eine initiale Internetverbindung benötigt wird, können umgangen werden, indem auf ein Bereich ausserhalb des Dialogs geklickt wird, oder der Back-Button gedrückt wird. Dadurch wird die App nicht geschlossen und könnte trotzdem verwendet werden.
 - Grund: Die beiden Dialoge reagieren nur auf Knopfdrücke, jedoch nicht auf die Cancel-Events, die z.B. über das Drücken des Back-Buttons ausgelöst werden.
- *SysTestBug02*: Sobald alle Daten einmal vom Back-End-Server heruntergeladen wurden, werden zukünftig keine aktualisierten Daten mehr vom Back-End-Server heruntergeladen.
 - Grund: Die für die SFTP-Zugriffe verwendete JSch-Library liefert die Anzahl vergangener Sekunden seit dem 01.01.1970 als Datum-Attribut der Dateien. Im Code wird dieser Wert mit den Anzahl vergangenen Millisekunden seit dem 01.01.1970 der letzten Synchronisierung verglichen.

4.8.2.2.3 Mögliche Verbesserungen

- *SysTestBug01*: Bei den beiden Dialogen sollte auch auf die Cancel-Events reagiert werden, sodass bei Eintritt eines solchen Events die App ebenfalls geschlossen wird.
- *SysTestBug02*: Bei dem Vergleich der Datumswerte, ob die jeweilige Datei heruntergeladen werden sollte, muss eine Umrechnung des Datumswerts der Datei erfolgen.

4.9 Unit Tests

Das User Interface ist ohne Test-Framework schwer zu testen. Die unteren Schichten jedoch nicht, deshalb wurden diese teilweise mit dem Test Driven Development Ansatz entwickelt.

Gesamthaft wurden 208 Unit Tests geschrieben.

5. Schlussfolgerungen

5.1 Ergebnisse

Die entwickelte App läuft stabil, weist jedoch noch einen kleinen Usability Fehler und einen gravierenderen Fehler in der Synchronisation der Daten auf. Optisch entspricht sie dem umzusetzenden Design recht gut, weist aber in den nicht vorher evaluierten Bereichen (z.B. der Startseite) noch Defizite auf.

Auf die ursprünglich gewünschte Back-End-Applikation mussten wir leider verzichten. Dies besprachen wir aber bereits in der Startphase des Projekts mit den Praxispartnern und einigten uns dabei auf den Einsatz von XML-Dateien auf einem SFTP-Server.

Die gewünschte Muss-Funktionalität und ein Teil der optionalen Funktionen konnten umgesetzt werden. Während der Projektzeit aufgetauchte Änderungswünsche konnten praktisch alle realisiert werden.

5.2 Schlussfolgerungen

Die (soweit möglich) präzise Planung und Definition der Anforderungsspezifikation half uns im späteren Projektverlauf. Es konnten jedoch trotzdem nicht alle Unklarheiten und Missverständnisse aus dem Weg geräumt werden.

Die implementierte Lösung der XML-Parser via Reflection stellte sich als eine gute Lösung heraus, da die spätere Entwicklung der spezifischen XML-Parser und die durchgeführten Refactorings jeweils sehr wenig Zeit in Anspruch nahmen.

Durch die Wahl des ungebundenen Service zur Datensynchronisierung wussten wir stets genau, in welchen Teilen die aufgetretenen Fehler ihren Ursprung hatten. Beim Einsatz eines Content Providers wäre dies allenfalls eher schwieriger gewesen.

Zu Beginn des Projektes hatten wir vor, Continuous Integration mithilfe von Jenkins konstant einzusetzen. Im Verlaufe der Arbeit haben wir Jenkins aber vernachlässigt. Die Builds waren jeweils erfolglos, aufgrund von Konfigurationsfehlern in Jenkins. Trotzdem schlichen sich keine Fehler ein, die mit Jenkins hätten vermieden werden können, da wir Vieles direkt mit Unit-Tests abdeckten und Android Studio die Lint Issues direkt im Code anzeigt. Deshalb werden wir auf Continuous Integration mit Jenkins in der Folgearbeit verzichten.

5.3 Ausblick

Die App ist eine gute Basis für eine Weiterentwicklung zur Marktreife. Sie besitzt aber noch Verbesserungspotential. Der erste Punkt wird sicher sein, dass die gefundenen Fehler behoben werden. Weiter kann das schlichte Design verbessert werden. Grundlage dafür wird in der Folgearbeit ein Design eines Experten bilden.

Da noch nicht alle gewünschten Funktionen umgesetzt sind, ist dies in der Folgearbeit ebenfalls, in Absprache mit den Praxispartnern, noch zu erledigen.

II. Anhang

6. Build Anleitung

1. Installieren Sie das Java SE 8 Development Kit.
2. Legen Sie eine Umgebungsvariable `JAVA_HOME` an. Der Wert soll auf das Installationsverzeichnis des Java Development Kits zeigen (z.B. `C:\Program Files\Java\jdk1.8.0_25`).
3. Installieren Sie Android Studio.
4. In Android Studio ist der Android SDK Manager enthalten. Öffnen Sie diesen.
5. Wählen Sie im SDK Manager mindestens folgende Komponenten zur Installation aus:
 - Tools
 - Android SDK Tools
 - Android SDK Platform-tools
 - Android SDK Build-tools, Version 20
 - Android 4.0.3:
 - SDK Platform
 - Intel x86 Atom System Image
 - Extras
 - Android Support Repository
 - Android Support Library
6. Installieren Sie die ausgewählten Komponenten.
7. Importieren Sie das Projekt in Android Studio.
8. Falls keine Run Configuration existiert, erstellen Sie folgende:
 - Typ: *Android Application*
 - Wählen Sie im Tab *General*, unter Module die Option *app* aus.

Siehe Tabelle 4 für die verwendeten Versionsnummern.

7. Werkzeuge/Tools

Name	Version
Android Studio (Beta)	0.8.14
Android SDK Tools	23.0.5
Android SDK Platform-tools	21.0.0
Android SDK Build-tools	20.0.0
Android 4.0.3: SDK Platform	5
Android 4.0.3: Intel x86 Atom System Image	1
Android Support Repository	8
Android Support Library	21.0.2
Java SE Development Kit 8	1.8.0_25 (64-bit)

Tabelle 4: Liste der verwendeten Software mit deren Version

8. Glossar

Begriff	Beschreibung
(Android-)App	Die Android App, die den Benutzer unterstützt Entscheidungen zu fällen. (Das eigentliche Produkt.)
(Back-End-) Applikation	Die Applikation auf dem Server zur Bearbeitung der vorhandenen und benutzer-erstellten Entscheidungen. Zusätzlich soll der Motivprofil-Fragebogen angepasst werden können.
benutzer-erstellte Entscheidung (-ssituation)	Eine Entscheidung, die ein Benutzer erfasst hat, inklusive deren Entscheidungs-Faktoren.
CRUD	Bearbeiten: Create, Read, Update, Delete
Entscheidungsfaktor	Eine Entscheidung wird anhand mehrerer Faktoren gefällt. Ein Faktor ist eine messbare Eigenschaft des Benutzers (z.B. besitzt er viel oder wenig Geld)
Entscheidungsregel	Anhand dieser Regel wird die Entscheidung gefällt. Sie behandelt die als orange eingestufteten Faktoren. Liegt die Summe der Gewichtungen dieser Faktoren über den definierten Schwellwerten, wird die Entscheidung „positiv mit Warnung“ oder „negativ“.
Gewichtung	Ein Entscheidungsfaktor besitzt pro Motivationssystem einen Korrekturbetrag. Diese werden summiert und zu 1 addiert und fließen so als Gewichtung für den Entscheidungsfaktor in die definitive Entscheidungsregel ein.
Korrekturbetrag	Der Korrekturbetrag wird aus der Ratingtabelle bestimmt. Dieser Betrag bezieht sich auf ein Motivationssystem.
Ratingtabelle	Die Ratingtabelle bestimmt die einzelnen Korrekturbeträge.
Mapping	Als Mapping wird die Bestimmung der einzelnen Korrekturbeträge pro Motivationssystem via Ratingtabelle bezeichnet.
Motivprofil	Das Motivprofil definiert, wie sich der Benutzer in den unterschiedlichen Motivationssystemen einschätzt.
Motivationssystem	Die fünf Motivationssysteme gemäss dem Zürcher Modell der sozialen Motivation [1]: Soziale Sicherheit, Erregung, Macht, Geltung, Leistung. (social, arousal, force, prestige, performance)
Motiv	siehe Motivationssystem
Rating	Als Rating werden die vom Benutzer angegebenen Werte im Motivprofil-Fragebogen bezeichnet.
Schwellwerte der Entscheidungsregel	Diese zwei Werte definieren, ab wann das Resultat einer Entscheidung „positiv mit Warnung“ oder „negativ“ ist. Siehe auch Entscheidungsregel
vorgegebene Entscheidung (-ssituation)	Diese Entscheidungen werden vorgefertigt mit der App geliefert, inklusive Entscheidungs-Faktoren.

9. Abbildungsverzeichnis

Abbildung 1: Use Case Diagramm	12
Abbildung 2: Beispiel Abbildung aus der Vorarbeit, der ZHAW Studenten [2]	18
Abbildung 3: Auszug der Ratingtabelle	19
Abbildung 4: Domain Modell.....	23
Abbildung 5: SSD: Erstes Starten der App	25
Abbildung 6: SSD: Weiteres Starten der App	26
Abbildung 7: Ablauf "UC4: Entscheidung fällen"	27
Abbildung 8: Systemübersicht: Überblick der beteiligten Systeme und Benutzer	28
Abbildung 9: Logische Architekturschichten	30
Abbildung 10: Klassendiagramm Package <i>data</i>	31
Abbildung 11: Klassendiagramm Package <i>data.database</i>	32
Abbildung 12: Klassendiagramm Package <i>data.domain</i>	34
Abbildung 13: Klassendiagramm Package <i>application</i>	36
Abbildung 14: Zustandsdiagramm <i>StartAppController</i>	37
Abbildung 15: Klassendiagramm Package <i>application.network</i>	38
Abbildung 16: Klassendiagramm Package <i>application.xml</i>	40
Abbildung 17: Klassendiagramm Package <i>view</i>	42
Abbildung 18: Klassendiagramm Package <i>view.helper</i>	43
Abbildung 19: Datenbankdiagramm	47
Abbildung 20: <i>MainActivity</i>	52
Abbildung 21: <i>SociodemographicDataActivity</i>	52
Abbildung 22: <i>QuestionnaireActivity</i>	52
Abbildung 23: <i>ChooseDecisionActivity</i>	53
Abbildung 24: <i>CreateDecisionActivity</i>	53
Abbildung 25: <i>MotiveProfileActivity</i>	54
Abbildung 26: <i>MotiveDetailDialogue</i>	54
Abbildung 27: <i>MakeDecisionActivity</i>	55
Abbildung 28: <i>DecisionResultActivity</i>	55

10. Tabellenverzeichnis

Tabelle 1: Change Requests aller Beteiligten	21
Tabelle 2: Konfigurationen	29
Tabelle 3: Liste der gültigen Motive	48
Tabelle 4: Liste der verwendeten Software mit deren Version	78

11. Literaturverzeichnis

- [1] Wikipedia, «Zürcher Modell der sozialen Motivation,» [Online]. Available: http://de.wikipedia.org/wiki/Z%C3%BCrcher_Modell_der_sozialen_Motivation. [Zugriff am 19 December 2014].
- [2] ZHAW, «Decisio Material Redesign,» [Online]. Available: <https://docs.google.com/file/d/0B8lloEpKcQ0bMHN5WnhmUFE5cE0>. [Zugriff am 19 December 2014].
- [3] Google Inc., «Material Design Specification,» [Online]. Available: <http://www.google.com/design/spec/material-design/introduction.html>. [Zugriff am 19 December 2014].
- [4] Google Inc., «Holo Themes | Android Developers,» [Online]. Available: <http://developer.android.com/design/style/themes.html>. [Zugriff am 19 December 2014].
- [5] JCraft Inc., «JSch - Java Secure Channel,» [Online]. Available: <http://www.jcraft.com/jsch/>. [Zugriff am 19 December 2014].
- [6] JCraft Inc., «JSch - Lizenz,» [Online]. Available: <http://www.jcraft.com/jsch/LICENSE.txt>. [Zugriff am 19 December 2014].
- [7] R. Horber und S. Kaufmann, «Projektblog,» [Online]. Available: <http://sinv-56028.edu.hsr.ch/wordpress/>. [Zugriff am 19 December 2014].
- [8] «Redmine,» [Online]. Available: <http://www.redmine.org>. [Zugriff am 19 December 2014].

11.1 Weitere Literatur

Für "best practice" Umsetzungen unter Android: <http://developer.android.com/guide/index.html>

Für allgemeine Lösungen von Android-Problemen, die nicht unter developer.android.com behandelt wurden: <http://www.stackoverflow.com>

Für die Realisierung der Ratingtabelle, des Motivprofil-Fragebogens und die vorgegebenen Entscheidungssituationen mit deren Faktoren erhielten wir Arbeitsdokumente der Praxispartner, welche nicht öffentlich zugänglich sind.

11.2 Lizenzen

BSD Lizenz von JSch – Java Secure Channel:

```
JSch 0.0.* was released under the GNU LGPL license. Later, we have switched over to a BSD-style license.
```

```
-----  
Copyright (c) 2002-2014 Atsuhiko Yamanaka, JCraft, Inc.  
All rights reserved.
```

```
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
```

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The names of the authors may not be used to endorse or promote products derived from this software without specific prior written permission.

```
THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL JCRAFT, INC. OR ANY CONTRIBUTORS TO THIS SOFTWARE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```