

# **Lastenheft**

Gruppe H

19. November 2017

# Inhaltsverzeichnis

|                                                                        |           |
|------------------------------------------------------------------------|-----------|
| <b>1. Introduction</b>                                                 | <b>4</b>  |
| 1.1. (Initial) Problem Statement for the Mobile Decision Lab . . . . . | 4         |
| 1.1.1. Experimenter Perspective . . . . .                              | 5         |
| 1.1.2. Participant Perspective . . . . .                               | 5         |
| <br>                                                                   |           |
| <b>I. User Requirements</b>                                            | <b>8</b>  |
| <b>2. Rollen</b>                                                       | <b>9</b>  |
| 2.1. AdministratorIn . . . . .                                         | 9         |
| 2.2. Client . . . . .                                                  | 9         |
| <b>3. Experimente/Fragen</b>                                           | <b>10</b> |
| <b>4. Chat</b>                                                         | <b>11</b> |
| <b>5. Aktivitätsscore</b>                                              | <b>12</b> |
| <b>6. Datenspeicherung und -verarbeitung</b>                           | <b>13</b> |
| <br>                                                                   |           |
| <b>II. System Requirements</b>                                         | <b>14</b> |
| <b>7. Non-Functional Requirements</b>                                  | <b>15</b> |
| 7.1. Product Requirements . . . . .                                    | 15        |
| 7.2. Usability Requirements . . . . .                                  | 15        |
| 7.3. Efficiency Requirements . . . . .                                 | 15        |
| 7.4. Performance Requirements . . . . .                                | 15        |
| <b>8. Funktionale Anforderungen</b>                                    | <b>16</b> |
| 8.1. Technische Anforderungen . . . . .                                | 16        |
| <br>                                                                   |           |
| <b>III. Szenarios</b>                                                  | <b>20</b> |
| <b>9. Szenario für die Erstellung eines Experiments</b>                | <b>21</b> |
| <b>10. Szenario für die Registrierung als Client</b>                   | <b>22</b> |

Es soll eine datenbankbasierte Android-App entwickelt werden, mit deren Hilfe man analysiert, inwiefern Netzwerk-Topologien individuelle Entscheidungen von sog. Clients beeinflussen. Ein Client ist ein Teilnehmer bzw. Benutzer der App, der bestimmte Fragen beantwortet und Diskussionen mit anderen Clients führt. Diskussionen zwischen Clients sind Chats, über die z.B. auch Bilder oder anderes Informationsmaterial ausgetauscht werden kann. Die andere Gruppe von Benutzern sind die Personen, die diese Experimente durchführen (Admins). Diese sollen dazu in der Lage sein, die zu beantwortenden Fragen zu definieren und diese für bestimmte Clientgruppen freizugeben. Die Freigabe soll dabei für Clients mit bestimmten Merkmalen erfolgen können (z.B. für alle Clients mit Kindern zwischen 20 und 30 Jahren). Zur anschließenden Analyse soll ein filterbarer Datenexport bereitgestellt werden.

# 1. Introduction

## 1.1. (Initial) Problem Statement for the Mobile Decision Lab

Der Nutzen des Systems besteht darin, eine mobile Anwendung für Experimente, betreffend Gruppen-Entscheidungen

The purpose of the system is to have a mobile lab for experiments concerning groups decisions based on different network topologies. The outcome of experiments shall help understand how network structures impact group decisions. The networks are anonymous connections between participants with the possibility to exchange messages between connected participants.

Networks can be constructed, for instance, scale-free or not, or with certain dimensionalities (everyone has exactly one neighbor, exactly two neighbors, etc.) and in a way that they additionally fulfill certain constraints (e.g., a connection between two participants *can only exist* if they share some properties, e.g., they are about the same age or have about the same expertise about the subject matter, or such a connection *cannot exist* if certain properties hold, e.g., neighbors have to be in different age groups or they are not allowed to have the same level of expertise).

The message format can be plain text or more structured, like an integer, or an enumeration type (like “up” or “down” or a chess move or a (legal) sentence like “guilty” or “not guilty”).

There are two types of users of the systems: experimenters (admins) or participants (test persons, subjects). Experimenters define the experiments along with all parameters and constraints. Participants can register and decide whether they want to participate in a given experiment. Before the start of the experiment, participants have to answer a few questions related to the task they should solve. (If no such information is requested by the experimenter in the definition of the experiment, this step can be omitted. Answers to these questions are treated confidentially, as all the other information that is entered by participants of the experiments.) Once the experiment has started, they are provided with a task description, which is the problem to be solved and for which a decision has to be reached towards the end of the “game” (i.e., has to be entered by each participant individually), and some supporting information (for instance, some paragraphs of law text for legal problems, or a hint on which supporting information they are allowed to use, e.g., some news website or wikipedia articles).

During one round of deliberation, participants can exchange messages with their (anonymous) neighbors about the problem to be solved, according to their own pace, until the predefined time of the experiment is up or the experimenter decides to close

the experiments manually.

When the time of deliberation has ended, each user is asked to decide about the question he or she was asked to answer in the beginning. The answer will be of a predefined format (not free text).

The nature of the experiment should be such that a “true” outcome is known already, but hard or impossible to access by participants (like a known sentence for a known case in the legal setting), or will be known shortly in the future (like the outcome of a football game or a chess move by a professional chess player in a certain game or the loss or gain of a stock on the stock market on the next day).

At the end of the experiment, the outcome of the decision (for instance, the fraction of participants choosing “guilty” or “not guilty” or the fraction predicting the win or loss of a forthcoming football game) can be communicated to the participants, if the experimenter has chosen this option up-front in the definition of the experiment.

All the results, the individual decisions and the group decisions are stored in a database in anonymized form for future analysis.

An experiment can be defined in a way that it is repeated with the same group of people and the same or, alternatively, a different network topology for a specified number of times. These are parameters specified by the experimenter up-front, before the experiment has started.

The outcome that was known already, but hard to access, or that is only known shortly after the experiment has ended, can also be communicated to the participants, if the experimenter wishes to do so.

#### **1.1.1. Experimenter Perspective**

The experimenter (also known as the admin) can define an experiment (for details see above) and access the results. When the experiment is defined, the questions to be answered by the participants can be specified, the task description, the supporting information, parameters specifying the network type, parameters regarding the repetition of experiments, and the duration of the experiment (predefined or stopped manually).

In the first step, the questions to be asked are defined (e.g., “Are you older than 40?”). The answers serve as background information on the participants, but can also be used as constraints in the automatic and randomized construction of the network. For example, everyone could be connected only to people who are older or younger. Once the network is constructed, a decision problem is sent to the users (along with some material that may help in the decision). The admin can set a time frame for the deliberation or stop the experiment manually at any time. After the time is up or the experiment is stopped manually, and the decisions are entered by all participants, the experimenter has access to the outcomes in anonymized form.

#### **1.1.2. Participant Perspective**

The participant can sign up for the mobile decision lab in an android app. He/she may select an experiment to participate in. The participant then has to answer some

preliminary questions. After that the participant is presented with a decision problem along with additional material. The participant studies the material and exchanges information and opinions with other participants he/she is connected to in the participant network. Information exchange only happens between two participants at a time. The message format may be different from experiment to experiment. In the simplest case, it is just plain text. After some time of deliberation (discussion of neighboring participants by exchanging messages), the participant is asked to make a final decision. After the decision is made, the participant may get some feedback on her answer depending on the experiment setup (the experiment could also be set up in a way that the participant does not receive feedback). Depending on the user's participation in the experiment, he/she may receive a number of points after each question. The participant can compare his/her performance with other participants on a global leaderboard. This scoring system is optional and may be chosen by the experimenter in the specification of the experiment.

# Glossar

**Admin** Der Admin (-istrator) in diesem Dokument ist der Leiter und Verwalter von Experimenten.. 9

**Aktivitätsscore** Der Aktivitätsscore soll die Maßeinheit sein, in der eine eventuelle Beeinflussung von Clients gemessen wird.. 9

**App** Für Clients bereit gestellte Andoid App, die über den Google Play Store/von unserem Laptop aus, weil wir keinen PPlay Store Account haben installiert wird.. 9, 22

**Charakteristik** Charakteristika entsprechen den vom Client angegebenen Profilinformationen. 9

**Chat** Der Chat ist ein sogenanntes Instant-Messaging-Programm wie z.B. WhatsApp. Über den Chat werden die Diskussionen geführt.. 9, 11

**Chatverbindung** Die Chatverbindung ist die Kommunikationserlaubnis zwischen zwei Clients.. 9, 11

**Client** Clients sind die Personen, die als ProbandInnen an Experimenten teilnehmen. Die Clients entsprechen den Benutzern. Das sind die Personen, die sich registrieren, Fragen beantworten und diskutieren.. 9, 22

**Diskussion** Diskussionen sind der Austausch zwischen Clients, anhand derer eine eventuelle Beeinflussung gemessen werden kann.. 9, 12

**Experiment** Das Experiment ist der Vorgang in dem Clients eine Frage beantworten, darüber diskutieren, evtl. ihre Antworten ändern und alles vom Admin ausgewertet wird.. 9, 13, 21

**Frage** Die Frage ist die Grundlage eines jeden Experiments.. 9

**Gruppe** Die Gruppe ist eine Einteilung von Clients nach verschiedenen Charakteristika.. 9, 10, 21

**Teil I.**

**User Requirements**



## 2. Rollen

### 2.1. AdministratorIn

**U-REQ 1:** Es soll einen Zugang für Personen, die mit der App Experimente durchführen, geben. Dieser wird als Admin bezeichnet. *Priority: A*

**U-REQ 2:** Ein Admin soll Chatverbindungen zwischen Clients hinzufügen und löschen können. *Priority: A*

**U-REQ 3:** Ein Admin soll eigene Fragen erstellen, bearbeiten, löschen, für Clients freigeben, für Clients beenden und auswerten können. *Priority: A*

**U-REQ 4:** Ein Admin soll anderen Admins eine Berechtigung für seine eigenen Fragen vergeben/entfernen können. *Priority: B*

**U-REQ 5:** Diese Berechtigung umfasst alle administrativen Operationsmöglichkeiten (siehe Punkt 3) des ursprünglichen Eigentümers. *Priority: B*

### 2.2. Client

**U-REQ 6:** Es soll einen Zugang für Personen geben, die als ProbandInnen an Experimenten teilnehmen möchten. Dieser wird als Client bezeichnet. *Priority: A*

**U-REQ 7:** Im Zuge der Registrierung der Clients soll es eine Profil-Seite mit (tw. Pflicht-) Angaben zu Charakteristika geben. *Priority: A*

**U-REQ 8:** Anhang dieser Charakteristika soll eine im Hintergrund ablaufende Zuordnung zu Gruppen stattfinden. *Priority: A*

**U-REQ 9:** Clients sollen sich dazu entscheiden können, an Fragen teilzunehmen. *Priority: A*

**U-REQ 10:** Es soll eine Chatfunktion für die Kommunikation zwischen zwei Clients geben. *Priority: A*

**U-REQ 11:** Die Chatfunktion soll nur mit freigegebenen Clients nutzbar sein. *Priority: A*

**U-REQ 12:** Anhand der abgeschlossenen Experimente und der Diskussionen eines jeden Clients soll im Hintergrund ein Aktivitätsscore errechnet werden. *Priority: B*

**U-REQ 13:** Ein Client soll nur seinen eigenen Aktivitätsscore sehen können. *Priority: C*

### 3. Experimente/Fragen

**U-REQ 14:** Zu jedem Experiment soll genau eine Frage gehören. *Priority: A*

**U-REQ 15:** Zu jedem Experiment soll es zeitliche Parameter geben, wie das vorgesehene Ende des Experiments. *Priority: A*

**U-REQ 16:** Fragen sollen vom Admin ausgewählten Gruppen angeboten werden können. *Priority: B*

**U-REQ 17:** Es sollen Fragen mit verschiedenen auswertbaren Antworttypen (Ja/Nein, Multiple Choice, Bewertung nach Punktesystem bzw. Zahl) möglich sein. *Priority: A*

**U-REQ 18:** Ein Admin kann zu seinen Experimenten Informationsmaterialien hinzufügen, die den Clients direkt oder als Download zur Verfügung stehen. *Priority: A*

## 4. Chat

**U-REQ 19:** Der Chat soll (Text-)Nachrichten in Echtzeit übermitteln. *Priority: A*

**U-REQ 20:** Der Chat soll auch Bilder und Dokumente übermitteln *Priority: B*

**U-REQ 21:** Der Chat soll auch Videos, Audio-Formate und Sprachnachrichten übermitteln. *Priority: C*

**U-REQ 22:** Es soll eine Funktion für Admins geben, die Chatverbindungen zwischen Clients erstellt (Eingabe: Zur Frage angemeldete Clients, deren Charakteristika, sowie die (feste/maximale/flexible) Anzahl der Chatverbindungen pro Client) *Priority: A*

## 5. Aktivitätsscore

**U-REQ 23:** Jeder Client soll einen eigenen Aktivitätsscore haben. *Priority: A*

**U-REQ 24:** Der Aktivitätsscore soll für Admins auswertbar sein. *Priority: A*

**U-REQ 25:** Der Aktivitätsscore soll anhand folgender Informationen systemseitig errechnet werden: Anzahl Diskussionen, Anzahl beantworteter Fragen und Qualität der Antworten. *Priority: B*

## 6. Datenspeicherung und -verarbeitung

**U-REQ 26:** Die erfassten Daten sollen anonymisiert gespeichert werden. *Priority: A*

**U-REQ 27:** Der Zugriff auf die Datenbank soll auf Admins beschränkt sein. *Priority: A*

**U-REQ 28:** Die Datenbank soll über Filter verfügen, damit die Daten einzelner Experimente getrennt abgerufen werden können. *Priority: B*

**U-REQ 29:** Es soll eine Möglichkeit zum Datenexport geben, um die Daten (mit externen Programmen) auswerten zu können. *Priority: B*

**Teil II.**

**System Requirements**

## **7. Non-Functional Requirements**

### **7.1. Product Requirements**

NF-REQ 1: Beispiel1 *Priority: A*

### **7.2. Usability Requirements**

NF-REQ 2: Beispiel2 *Priority: A*

### **7.3. Efficiency Requirements**

NF-REQ 3: Beispiel3 *Priority: A*

### **7.4. Performance Requirements**

NF-REQ 4: Beispiel4 *Priority: A*

## 8. Funktionale Anforderungen

### 8.1. Technische Anforderungen

**F-REQ 1:** Wir verwenden eine Postgre-SQL Datenbank. *U-REQ 1 Priority: A*

**F-REQ 2:** Benutzer können sich (mit E-Mail-Adresse, Benutzername und Passwort) am System registrieren und ihren Account verwalten. Dazu gibt es eine Benutzer-Tabelle. *U-REQ 27 Priority: A*

**F-REQ 3:** In dieser Tabelle gibt es ein Boolean-Feld über das ein Client sich für den Admin-Status bewerben kann. *U-REQ 1 Priority: B*

**F-REQ 4:** Die Admins stehen ebenfalls in der gleichen Benutzer-Tabelle und sind über ein Boolean-Feld gekennzeichnet. *U-REQ 1 Priority: A*

**F-REQ 5:** SysAdmins haben Zugriff auf die gesamte Datenbank und können (markierte) Clients zu Admins befördern. Sie können den Admin-Status auch wieder entfernen. *U-REQ 2 Priority: A*

**F-REQ 6:** Durch diese Struktur sind Admins auch gleichzeitig Clients. *U-REQ 3 Priority: A*

**F-REQ 7:** Für Admins können in der Datenbank zusätzliche Informationen gespeichert werden (Name, Grund für die Nutzung des Systems, Universität, Studienfach) *U-REQ 4 Priority: A*

**F-REQ 8:** Für jeden Client wird in der Datenbank der Benutzername und ein Passwort gespeichert. *U-REQ 5 Priority: A*

**F-REQ 9:** Die Datenbank ist in der Lage für jede Admin-ID einen Wiederherstellungs-Code zu generieren, über den er ein neues Passwort festlegen kann. *U-REQ 6 Priority: A*

**F-REQ 10:** Die Datenbank speichert die durch Admins erstellte Experimente in einer Tabelle. *U-REQ 7 Priority: A*

**F-REQ 11:** In der Experiment-Tabelle gibt es ein Feld zur Eingabe der Experiment-Beschreibung. *U-REQ 8 Priority: B*

**F-REQ 12:** In der Datenbank existiert eine erweiterbare Tabelle mit Sprachen. *U-REQ 9 Priority: A*

**F-REQ 13:** Der Admin wählt experimentbezogen eine Sprache aus. Diese wird in der Experiment-Tabelle hinterlegt. (Default: Englisch.) *U-REQ 9 Priority: A*



- F-REQ 14:** Eine zu den Experimenten verlinkte Tabelle speichert "preliminary questions". *U-REQ 10 Priority: A*
- F-REQ 15:** In der Experiment-Tabelle kann in einem Feld vom Admin die eigentliche Frage hinterlegt werden. *U-REQ 11 Priority: A*
- F-REQ 16:** In der Experiment-Tabelle kann in einem Feld vom Admin der Antworttyp eingetragen werden (Multiple-Choice, Datum, Zahl, Boolean) *U-REQ 11 Priority: A*
- F-REQ 17:** In der Experiment-Tabelle kann in einem Feld (eins je Antworttyp) vom Admin die korrekte angegeben werden. *U-REQ 12 Priority: A*
- F-REQ 18:** Admins können zu Ihren Fragen Dokumente in bestimmten Formaten (PDF, JPG, PNG, ...) hochladen. *U-REQ 13 Priority: A*
- F-REQ 19:** In der Datenbank gibt es eine Sequenz-Tabelle, in der eine Abfolge von Experimenten gespeichert werden kann. *U-REQ 14 Priority: C*
- F-REQ 20:** In der Datenbank gespeicherte Fragen können von Admins kopiert werden, wobei Parameter geändert werden können (preliminary Questions, Dokumente, Beschreibung, ...) *U-REQ 15 Priority: A*
- F-REQ 21:** Es gibt in der Datenbank eine Compound-Tabelle, die die Netzwerk-Topologie speichert (Clientverbindungen) *U-REQ 16 Priority: A*
- F-REQ 22:** In der Compound-Tabelle gibt es mehrere Boolean-Felder für das Nachrichtenformat, das vom Admin festgelegt wird. *U-REQ 17 Priority: A*
- F-REQ 23:** In der Experiment-Tabelle gibt es Felder den Start- und Endzeitpunkt. (vom Admin festgelegt) *U-REQ 18 Priority: A*
- F-REQ 24:** In der Experiment-Tabelle gibt es ein Feld für die Bedenkzeit (vom Admin festgelegt). *U-REQ 19 Priority: A*
- F-REQ 25:** In der Experiment-Tabelle gibt es ein Boolean-Feld für die Markierung von inaktiven Experimenten (vom Admin festgelegt) *U-REQ 20 Priority: A*
- F-REQ 26:** In der Benutzer-Tabelle existiert ein Feld für den Aktivitätsscore. *U-REQ 21 Priority: B*
- F-REQ 27:** In der Experiment-Tabelle existiert ein Boolean-Feld, das angibt, ob diese Frage den Aktivitätsscore beeinflusst (vom Admin festgelegt). *U-REQ 21 Priority: B*
- F-REQ 28:** In der Experiment-Tabelle muss vom Admin festgelegt werden können, ob das Ergebnis nur ihm, öffentlich, oder nur bestimmten Usern zur Verfügung steht. *U-REQ 22 Priority: A*
- F-REQ 29:** Die Datenbank speichert die Antworten in einer Antwort-Tabelle, in der die User-IDs der beantwortenden Personen verschlüsselt sind. *U-REQ 23 Priority: A*
- F-REQ 30:** Die Datenbank ist in der Lage einen Export bestimmter Daten zur Verfügung zu stellen. Admins können diese herunterladen. *U-REQ 24 Priority: C*
- F-REQ 31:** Siehe F-REQ 30 *U-REQ 25 Priority: A*

- F-REQ 32:** Admins sind über das System in der Lage, SysAdmins zu kontaktieren (z.B. per Kontaktformular). *U-REQ 26 Priority: A*
- F-REQ 33:** Es gibt in der Benutzer-Tabelle einen Unique-Key mit Benutzername bzw. E-Mail Adresse. *U-REQ 28 Priority: A*
- F-REQ 34:** Bei der Registrierung wird ein Benutzer aufgefordert, die Nutzungsbedingungen zu akzeptieren. *U-REQ 29 Priority: A*
- F-REQ 35:** Die Datenbank ist in der Lage für jede Benutzer-ID einen Wiederherstellungscode zu generieren, über den er ein neues Passwort festlegen kann. *U-REQ 30 Priority: A*
- F-REQ 36:** Das System stellt ein (Video-)Tutorial für neue Benutzer bereit. *U-REQ 31 Priority: D*
- F-REQ 37:** Benutzer können das Tutorial überspringen bzw. beenden. *U-REQ 32 Priority: D*
- F-REQ 38:** Es gibt eine API-Schnittstelle zu anderen Datenbanken (z.B. Facebook), über die Benutzer-Account exportiert bzw. importiert werden können. *U-REQ 33 Priority: D*
- F-REQ 39:** Es gibt für Clients eine filterbare Übersicht aller Experimente mit deren Beschreibungen. *U-REQ 34 Priority: A*
- F-REQ 40:** Das System stellt Clients eine Übersicht der Experimente bereit, an denen sie teilnehmen können. *U-REQ 35 Priority: A*
- F-REQ 41:** Das System stellt Clients eine Übersicht der Experimente bereit, an denen sie aktuell teilnehmen. *U-REQ 36 Priority: A*
- F-REQ 42:** In der Experiment-Übersicht für Clients soll der Start- und Endzeitpunkt bzw. Status (aktiv / inaktiv). *U-REQ 37 Priority: B*
- F-REQ 43:** Das System akzeptiert Antworten auf "preliminary Questions" nur von am Experiment teilnehmenden Benutzern. *U-REQ 38 Priority: A*
- F-REQ 44:** Das System zeigt jedem teilnehmenden Client den Wert der verbleibenden Bedenkzeit an. *U-REQ 39 Priority: A*
- F-REQ 45:** Das System zeigt den teilnehmenden Clients die eigentliche Frage nur während der Bedenkzeit an. *U-REQ 40 Priority: A*
- F-REQ 46:** Das System ermöglicht teilnehmenden Clients nach der Bedenkzeit die Eingabe der Antworten. *U-REQ 41 Priority: A*
- F-REQ 47:** Das System sperrt die Antworteingabe nach dem Endzeitpunkt des Experiments. *U-REQ 42 Priority: A*
- F-REQ 48:** Das System zeigt jedem teilnehmenden Client nach der Bedenkzeit die verbleibende Antwortzeit bis zum Endzeitpunkt an. *U-REQ 43 Priority: B*
- F-REQ 49:** Teilnehmende Clients sollen in der Lage sein, die hochgeladenen Doku-

mente herunterzuladen. *U-REQ 44 Priority: A*

**F-REQ 50:** Teilnehmende Clients sollen (falls vom Admin so definiert) in der Lage sein, nach dem Endzeitpunkt die korrekte Antwort einer Frage zu sehen. *U-REQ 45 Priority: B*

**F-REQ 51:** Das System soll über verschiedene Einstellungen verfügen. *U-REQ 46 Priority: B*

**F-REQ 52:** Diese Einstellungen sollen individuell von Benutzern konfiguriert werden können. *U-REQ 47 Priority: D*

**F-REQ 53:** Clients sind über das System in der Lage, SysAdmins zu kontaktieren (z.B. per Kontaktformular). *U-REQ 48 Priority: C*

**F-REQ 54:** In den individuellen Einstellungen können Clients ihre bevorzugte Sprache konfigurieren. *U-REQ 49 Priority: A*

**F-REQ 55:** Wenn eine Clientverbindung im System besteht, können die beteiligten Clients gegenseitig ihre Benutzernamen sehen. *U-REQ 50 Priority: A*

**F-REQ 56:** Wenn eine Clientverbindung im System besteht, können beteiligte Clients während der Bedenkzeit Nachrichten austauschen. *U-REQ 51 Priority: A*

**F-REQ 57:** Clients sind über das System in der Lage, Admins zu kontaktieren (z.B. per Kontaktformular). *U-REQ 52 Priority: B*

**F-REQ 58:** In der Compound-Tabelle existiert ein Boolean-Feld, über das die Clientverbindung stummgeschaltet werden kann. *U-REQ 53 Priority: B*

**F-REQ 59:** Das System stellt eine geeignete Filterfunktion für unangemessene Ausdrücken in Chats zur Verfügung. *U-REQ 54 Priority: B*

**F-REQ 60:** Das System ermöglicht die gleichzeitige Auslieferung einer Nachricht an mehrere Clients. *U-REQ 55 Priority: B*

**F-REQ 61:** Das System stellt eine globale Rangliste aller Clients mit deren Aktivitätsscore zu Verfügung (in absteigender Reihenfolge). *U-REQ 56 Priority: C*

**F-REQ 62:** Das System zeigt Clients ihre Position in der globalen Rangliste an. *U-REQ 57 Priority: C*

**F-REQ 63:** Das System soll eine Weboberfläche zur Verfügung stellen. *U-REQ 58 Priority: A*

**F-REQ 64:** Das System soll über eine mobile App benutzbar sein. *U-REQ 59 Priority: A*

**F-REQ 65:** Das System soll ethischen Leitlinien unterliegen. *U-REQ 60 Priority: A*

**F-REQ 66:** Die Datenbank erfüllt zeitgemäße Sicherheitsstandards *U-REQ 61 Priority: A*

## **Teil III.**

# **Szenarios**

## 9. Szenario für die Erstellung eines Experiments

**INITIALE SITUATION:** *Ein Admin möchte ein Experiment erstellen: Wie lange muss ein verurteilter Täter in Haft? Dazu lädt er eine PDF-Datei mit Informationen zum Tathergang und dem Urteil hoch und ordnet eine Gruppe dieser Frage zu.*

**NORMAL:**

- *Der Admin benutzt das Experiment-Erstellungs-Interface und gibt die Frage, die Antwortmöglichkeit als Kommazahl und eine Fragebeschreibung (z.B. Bitte zugeordnetes PDF beachten. Kommazahlen als Antwort sind erlaubt.) ein.*
- *Der Admin gibt außerdem einen Zeitraum für das Experiment, sowie die Dauer für Clients, eine Antwort abgeben zu können, an.*
- *Der Admin lädt das Info-PDF hoch.*
- *Der Admin sucht eine Gruppe aus und ordnet diese zu. (z.B. Jura-Studenten)*

**WAS SCHIEF GEHEN KANN:**

- *Die Eingaben (Fragetext und Fragebeschreibung) könnten zu lang sein.*
- *Das PDF könnte zu groß oder im falschen Format sein.*
- ...

**ANDERE AKTIVITÄTEN:** *Eingabefelder in ihrer Länge beschränken. Checks einbauen, die das PDF auf Größe (z.B. 5 MB) und auf das Format prüfen.*

**ZUSTAND DES SYSTEMS BEI ERFOLG:** *User ist eingeloggt. Die Frage ist erstellt und wird mit PDF und allen weiteren vom Admin eingegebenen Infos den ausgewählten Clients zur Teilnahme bereitgestellt.*

## 10. Szenario für die Registrierung als Client

**INITIALE SITUATION:** *Eine Person möchte sich über die App als Client registrieren.*

**NORMAL:**

- *Die Person installiert die App über den Google Play Store.*
- *Sie startet die App, die sich mit dem Server verbindet und einen Registrierungsbildschirm anzeigt.*
- *Sie wird aufgefordert die für die Registrierung erforderlichen Daten einzugeben. Sie gibt diese ein und bestätigt die Übermittlung an den Server.*
- *Der Server speichert die Eingaben in der Datenbank und meldet die erfolgreiche Registrierung zurück.*

**WAS SCHIEF GEHEN KANN:**

- *Die App ist für das verwendete OS nicht verfügbar.*
- *Die App ist zwar verfügbar, läuft aber auf dem verwendeten System nicht.*
- *Die App läuft unbrauchbar langsam.*
- *Es ist keine Verbindung zum Server möglich.*
- *Der Server ist überlastet.*
- *Die Internetverbindung wird während der Kommunikation mit dem Server unterbrochen.*
- *Ein frei wählbarer Bezeichner, der als Schlüssel für die Datenbank verwendet wird, ist schon belegt.*
- *Der User scheitert an der Eingabe der Daten.*
- *Die Übermittlung der Daten ist unvollständig/wird abgebrochen.*
- *Der Server verarbeitet die Eingabedaten fehlerhaft.*
- *Die erfolgreiche Registrierung mit Eintrag in die Datenbank kann nicht übermittelt werden, weil die Internetverbindung zwischenzeitlich abgebrochen ist.*
- *Die App wurde zwischenzeitlich beendet, die erfolgreiche Registrierung kann nicht übermittelt werden.*

**ANDERE AKTIVITÄTEN:** *Eine andere Person versucht den gleichen Schlüssel zur Registrierung zu benutzen, andere Zugriffe auf den Server und die Datenbank, Datenbankeinträge könnten abgefragt werden, bevor sie fertig sind.*

**ZUSTAND DES SYSTEMS BEI ERFOLG:** *Neuer User ist eingeloggt, Eintrag in der Datenbank ist angelegt, Server und Client sind bereit zur Kommunikation.*