



SOFTWARE PRAKTIKUM 2015 AM WI2

LERNDUELL

Software-Entwicklungspraktikum (SEP)
Sommersemester 2015

Pflichtenheft

Auftraggeber
Technische Universität Braunschweig
Institut für Wirtschaftsinformatik
Abteilung Informationsmanagement
Prof. Dr. Susanne Robra-Bissantz
Mühlenpfordtstraße 23
38106 Braunschweig

Betreuer: Michael Kallookaran

Auftragnehmer:

Name	E-Mail-Adresse
Christopher Schulz	chri.schulz@tu-braunschweig.de
Ekaterina Bazhenova	e.bazhenova@tu-braunschweig.de
Fabian Sgonina	f.sgonina@tu-braunschweig.de
Frederik Book	f.book@tu-braunschweig.de
Jannis Kottke	j.kottke@tu-braunschweig.de
Jonas Korth	jonas.korth@tu-braunschweig.de
Steffen Döring	steffen.doering@tu-braunschweig.de
Timo Christophers	t.christophers@tu-braunschweig.de

Braunschweig, 13. Mai 2015

Inhaltsverzeichnis

1 Zielbestimmung	5
1.1 Musskriterien	5
1.2 Sollkriterien	6
1.3 Kannkriterien	6
1.4 Abgrenzungskriterien	6
2 Produkteinsatz	7
2.1 Anwendungsbereiche	7
2.2 Zielgruppen	7
2.3 Betriebsbedingungen	7
3 Produktübersicht	9
4 Produktfunktionen	14
4.1 Benutzerfunktionen	14
4.2 Spielfunktionen	15
4.3 Administration des Fragenkatalogs	18
4.4 Administratorfunktionen	19
5 Produktdaten	20
6 Nichtfunktionale Anforderungen	22
6.1 Funktionalität	22
6.2 Sicherheit	22
6.3 Benutzbarkeit	23
6.4 Änderbarkeit	23
6.5 Qualitätsanforderungen	24
7 Benutzeroberfläche/Schnittstellen	25
8 Technische Produktumgebung	30
8.1 Software	30
8.2 Hardware	30
8.3 Produktschnittstellen	30

9 Glossar

31

Abbildungsverzeichnis

3.1	Use-Case-Diagramm <i>User spielt Lernduell</i>	10
3.2	Aktivitätsdiagramm <i>User spielt Lernduell</i>	12
3.3	Aktivitätsdiagramm <i>Spiel zwischen zwei zufällig ausgewählten Usern</i>	13
7.1	Anmeldung $\langle UI10 \rangle$	26
7.2	Frage erstellen $\langle UI10 \rangle$	27
7.3	Spiel $\langle UI10 \rangle$	28
7.4	Statistik $\langle UI10 \rangle$	29

1 Zielbestimmung

Mobile Geräte sind zu einem grundlegenden Teil unseres Lebens geworden. Da Smartphones und Tablets unsere ständigen Begleiter sind, haben sich so viele Bereiche unseres täglichen Lebens stark verändert. Die Menschen sind noch mobiler geworden, wodurch sich ein Großteil ihrer Aktivitäten unterwegs abspielt.

So gewinnt auch das Mobile Learning zunehmend an Bedeutung, denn das Lernen mit portablen Endgeräten oder mobilen Medien ermöglicht flexibles, spontanes Lernen vor jedem Ort aus, egal wo sich der Lernpartner zurzeit befindet.

Die Intention, welche hinter dem Projekt *Lernduell* steckt, ist es, in jeder Situation und an jedem Ort an dem es möglich ist ein Smartphone zu verwenden auch Lernen zu können und das spielend ohne Karteikarten oder sonstige Unterlagen.

Lernduell kann entweder direkt mit Freunden oder gegen zufällig gewählte Gegenspieler gespielt werden. Eine Statistik visualisiert dabei den Lernerfolg und erinnert zudem an Schwachstellen in einem bestimmten Fach. Die Möglichkeit den vorhandenen Fragenkatalog über die App ständig selbst zu erweitern sichert stets genügend aktuelle Prüfungsfragen.

So soll die Android-App, ähnlich wie Smartphones selbst, zu einem praktischen Begleiter für das Studium werden.

1.1 Musskriterien

⟨RM1⟩ Benutzerkonten

Studierende können sich bei der ersten Nutzung von *Lernduell* registrieren oder anmelden.

⟨RM2⟩ Spielervermittlung

Durch den Server, entweder an Freunde oder zufällige Personen.

⟨RM3⟩ Eigene Fragen

Benutzer können selbst Fragen erstellen.

⟨RM4⟩ Partien annehmen oder ablehnen

Fordert ein Spieler (nicht Random) den anderen heraus, hat dieser die Möglichkeit das Spiel anzunehmen oder abzulehnen.

⟨RM5⟩ Flüssiger Spielablauf

Die Partie findet in 3 Runden mit jeweils 3 Fragen statt.

⟨RM6⟩ Statistiken

Dem Nutzer werden die Statistiken der letzten fünf gespielten Spiele zur Verfügung gestellt.

1.2 Sollkriterien

⟨RS1⟩ Übersichtliches und ansprechendes Design

Damit sich die Nutzer schnell zurechtfinden und gerne mit der App lernen, sollte das Design intuitiv und optisch ansprechend sein.

1.3 Kannkriterien

⟨RC1⟩ Semester

Damit die Aktualität der Frage schnell ersichtlich ist, soll es bei der Fragerstellung möglich sein das Semester mit anzugeben.

⟨RC2⟩ Fragen melden

Fällt dem Nutzer auf, dass eine Frage veraltet oder falsch ist, soll er die Möglichkeit haben, diese zu melden, wobei sie anschließend von den Administratoren überprüft wird.

⟨RC3⟩ Kontaktliste

Um Freunde schneller auszuwählen soll es möglich sein diese zu einer Kontaktliste hinzufügen zu können.

⟨RC4⟩ Aufgeben

Möchte der Nutzer das Spiel nicht zu Ende spielen, soll er die Möglichkeit haben aufzugeben.

1.4 Abgrenzungskriterien

⟨RW1⟩ Fragen

Die Fragen müssen einem Studienfach zugehörig sein, Fragen aus anderen Lebensbereichen gehören nicht zu *Lernduell*.

⟨RW2⟩ Unterkategorien

Da manche Fächer schwer einzuteilen sind, werden bei *Lernduell* keine Kategorien genutzt.

2 Produkteinsatz

2.1 Anwendungsbereiche

Lernduell ist ein Spiel, bei dem die Benutzer Fragen aus verschiedenen Fächern beantworten müssen und gegen reale Gegner antreten. Einzige Voraussetzung zur Nutzung des *Lernduells* ist ein Smartphone mit Android-Betriebssystem, welches Zugang zum Internet hat. Das Spiel ist nur für jeden registrierten Nutzer gedacht. Es besteht keine Möglichkeit ohne Gegenspieler zu spielen. Es ist nicht möglich, dass in einer Runde mehr als 2 Personen gegeneinander spielen. Die Android-App ist ein guter Begleiter für das Studium und eignet sich perfekt als Vorbereitung für Klausuren. Die Nutzung der Lern-App ist sehr leicht und intuitiv.

2.2 Zielgruppen

Die Lern-App *Lernduell* richtet sich hauptsächlich an Studierende deutscher Universitäten und Hochschulen. Die Applikation ist für die Nutzung durch Studenten ausgelegt, die den Lernstoff wiederholen, ihren Lernfortschritt regelmäßig überprüfen und sich für die Prüfungen vorbereiten möchten. Die Nutzung durch andere interessierte Personen ist jedoch auch denkbar. Die Lernmaterialien in Form eines Quizduells können jederzeit und an jedem Ort bearbeitet werden: auf der Fahrt zur Universität, vor dem Einschlafen oder während Wartezeiten – jeder Zeit kann die App nützliche Lerneinheiten liefern. Das Lernspiel bietet den Studierenden darüber hinaus den Vorteil, dass es das oftmals langweilig wirkende Wiederholen von Lerninhalten in einer für sie angenehmen Weise präsentiert. Somit kann die Lernmotivation gesteigert werden. Soweit keine weiteren Sprachen integriert sind, muss der Benutzer die Verkehrssprache Deutsch zumindest verstehen.

2.3 Betriebsbedingungen

Dieses System soll sich bezüglich der Betriebsbedingungen nicht wesentlich von anderen mobilen Apps unterscheiden. Das Backend des Produktes läuft im Dauerbetrieb auf einem zentralen Server im Institut für Wirtschaftsinformatik, bedarf keiner Aufsicht und läuft vollautomatisch: es besteht keinen Bedarf der ständigen Beobachtung durch den Anwender. Die Betriebszeit des

Servers ist 24 Stunden am Tag, 7 Tage die Woche (24/7). Mit dem Server können sich mehrere Clients gleichzeitig verbinden. Als Standard-Client ist das Android-Smartphone vorgesehen. Dort wird die App mit der visuellen Benutzeroberfläche installiert. Für den Datenaustausch zwischen Client und Server wird eine Internetverbindung benötigt. Für administrative Aufgaben ist ein Endgerät mit lauffähigem Webbrowser erforderlich, um auf die Internetschnittstelle zugreifen zu können.

3 Produktübersicht

Die Produktübersicht verdeutlicht mit Hilfe einiger Use-Cases den Zusammenhang einzelner Akteure und zeigt auf, wie diese miteinander interagieren. In Abbildung 3.1 werden speziell alle Akteure grob dargestellt. Die Kernfunktionalität der App ist das Spiel, in dem sich Studierende gegenseitig über Wissensfragen zu verschiedenen Fächern herausfordern können. Wie im Use-Case-Diagramm User spielt Lernduell, Abb. 7.1, dargestellt, muss sich der Benutzer bei der ersten Nutzung der App registrieren. Nachdem sich der Nutzer registriert hat, wird er automatisch auf seinem Smartphone eingeloggt. Sollte der Nutzer das Gerät wechseln, ist nach dem Installieren der App nur Einloggen erforderlich. Nach dem Einloggen wird entweder ein neues Spiel gestartet oder der Nutzer kann selber eine Frage für das Spiel erstellen. Startet der Nutzer ein neues Spiel muss er zuerst auswählen, in welchem Fach er spielen möchte und ob er gegen einen bestimmten Nutzer oder gegen einen zufälligen Spieler antreten möchte. Im Spiel beantwortet der Nutzer in drei Runden drei Fragen aus dem ausgewählten Fach. Am Ende des Spiels soll eine Statistik angezeigt werden. Möchte der Nutzer eine Frage erstellen, muss er zuerst ein Fach aus einem Dropdown-Menü auswählen. Nach der Wahl des Faches sind die Frage selbst und vier Antwortmöglichkeiten einzugeben. Anschließend kann die Frage mit dem Status „ungeprüft“ an den Server geschickt werden. Der Administrator kann den Fragenkatalog in der Datenbank aktualisieren und die ungeprüfte Frage freischalten. Die Frage wird nur freigeschaltet, falls es keine Fehler gibt (die Frage gehört zum richtigen Fach usw.). Wenn die Frage fehlerhaft ist, wird sie aus der Tabelle der Fragen gelöscht. Zudem verfügt der User über die Option fehlerhafte oder veraltete Fragen zu melden. In diesem Fall wird der Administrator benachrichtigt, die Fragen werden von ihm überprüft und können aus der Datenbank gelöscht werden.

Mit Hilfe des Aktivitätsdiagramms wird der im Use-Case-Diagramm erläuterte Anwendungsfall genauer beschrieben. Der Abbildung 3.2 ist zu entnehmen, dass der Nutzer den Prozess startet, indem er sich anmelden möchte. Falls die Anmeldung erfolgreich ist, ist der Benutzer eingeloggt und wird direkt auf den Homescreen geleitet. Hier hat er drei weitere Möglichkeiten: entweder eine neue Frage zu erstellen oder ein neues Spiel zu starten oder den Spielprozess zu beenden. Falls er sich entscheidet eine Frage zu erstellen, soll er ein Fach auswählen, eine Frage und vier Antwortmöglichkeiten dazu eingeben und schließlich die neue Frage speichern. Danach wird der Benutzer zurück auf den Homescreen geleitet. Falls der Benutzer ein neues Spiel starten möchte, soll er als erstes auswählen, in welchem Fach er spielen möchte. Als nächstes kann er entweder gegen einen bestimmten Nutzer antreten oder „Random“ auswählen und auf eine Einladung zum Spiel gegen einen zufälligen Gegner warten. Das interaktive Spiel sucht einen Gegner aus dem

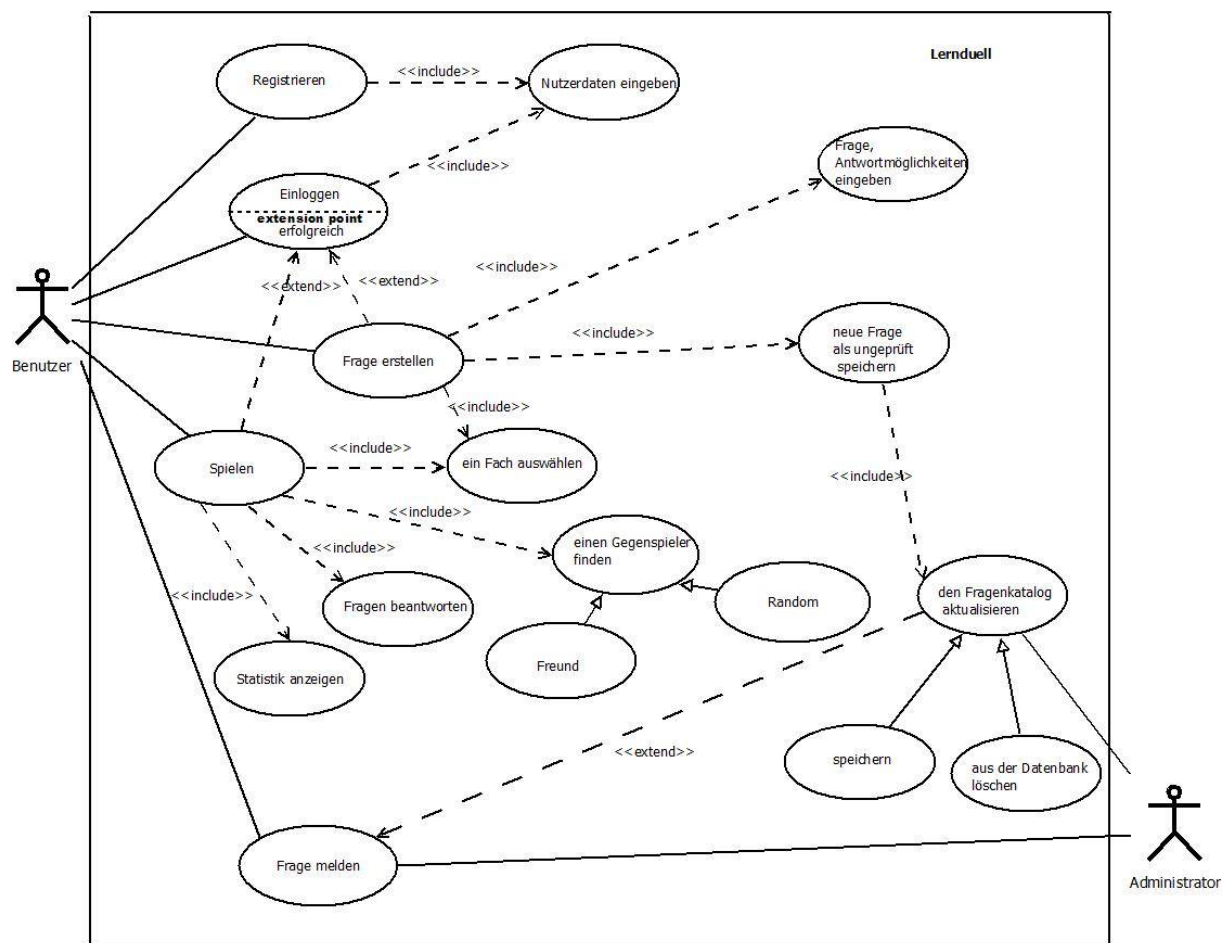
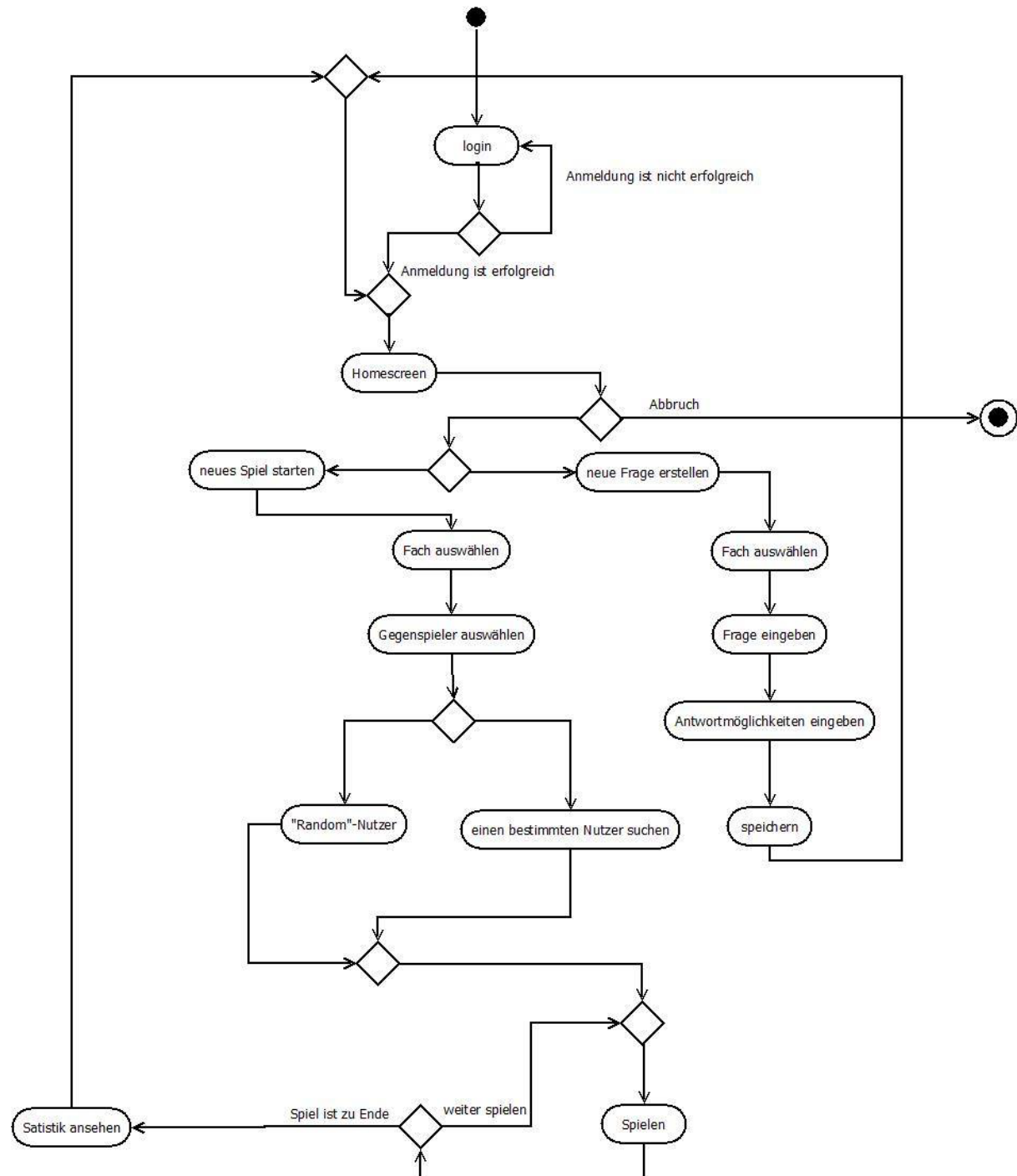


Abbildung 3.1: Use-Case-Diagramm *User spielt Lernduell*

Pool der Mitspieler. Danach startet das Spiel direkt mit einem dreirundigen Duell. Am Ende des Spiels sieht sowohl der User als auch sein Gegner die Ergebnisse mit den richtigen Antworten in einer Übersicht. Danach leitet das System den Spieler zurück auf den Homescreen.

Die Interaktion zwischen dem Nutzer und seinem Gegenspieler, der zufällig ausgewählt wurde kann der Abbildung 3.3 entnommen werden. Hierbei handelt es sich um dieselbe Aktion, lediglich zwischen zwei Benutzer. Zu Beginn melden sich beide Benutzer an. Die Anmeldung ist erfolgreich, die Benutzer werden auf den Homescreen geleitet. Nachdem beide Spieler ein Fach ausgewählt und den „Random“-Button gedrückt haben, wird das neue Spiel erstellt indem die Spieler vom System als Gegner ausgewählt werden. Von der Serverseite werden Fragen verschickt, die von den Spieler beantwortet werden müssen. Wenn das Spiel zu Ende ist, werden die Spielergebnisse in der Datenbank gespeichert. Das System zeigt den Benutzer ihre Ergebnisse an. Falls der Benutzer mehr Fragen als sein Gegner richtig beantwortet hat, wird eine Siegenachrichtigung angezeigt, falls der Gegner gewinnt, eine Niederlageinfo.

Abbildung 3.2: Aktivitätsdiagramm *User spielt Lernduell*

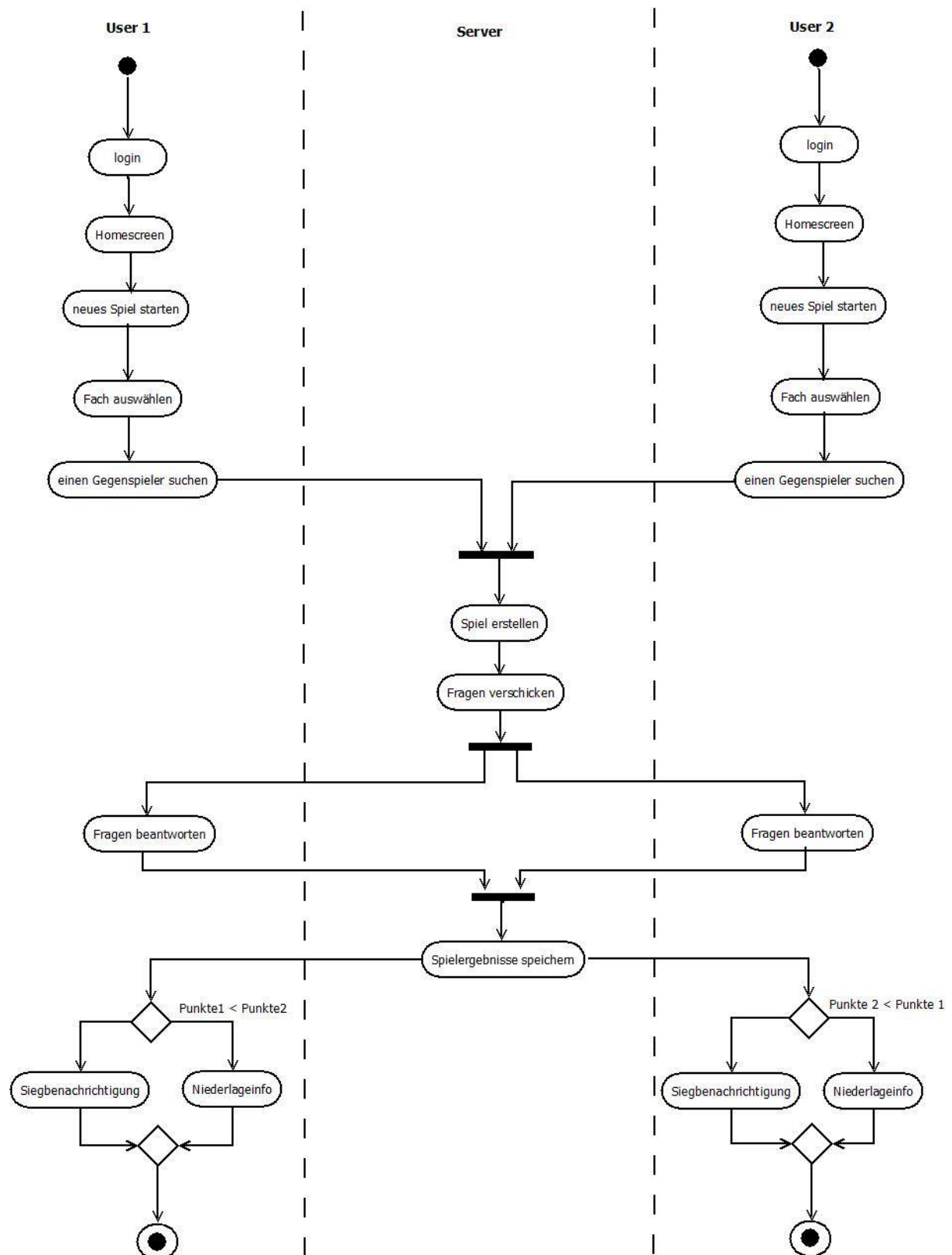


Abbildung 3.3: Aktivitätsdiagramm *Spiel* zwischen zwei zufällig ausgewählten Usern

4 Produktfunktionen

4.1 Benutzerfunktionen

Ein im System registrierter Benutzer kann das System erst nutzen, wenn er angemeldet ist. Falls der Studierende sich schon einmal auf dem Smartphone angemeldet hat, wird er bei jedem Start der App automatisch eingeloggt.

Registrieren $\langle F10 \rangle$

Anwendungsfall: Registrieren des Benutzers

Anforderung: $\langle RM1 \rangle$

Ziel: Der Benutzer registriert sich nachdem die App auf dem Smartphone installiert wurde.

Vorbedingung: Die App wurde auf dem Smartphone installiert, der Benutzer besitzt noch keinen Account.

Nachbedingung Erfolg: Der Benutzer ist registriert, seine Daten (Benutzername, Passwort und die aktuell von ihm besuchte Universität) werden auf dem Server in der Benutzer-Tabelle gespeichert. Mit dem erfolgreichen Abschließen des Registrierungsvorgangs ist der neue Benutzer am System angemeldet.

Nachbedingung Fehlschlag: Die Verbindung zum Server wurde abgebrochen, somit wurde kein neuer Account angelegt.

Akteure: Benutzer Client Datenbank

Beschreibung:

1. Der Benutzer gibt seinen Daten ein: den Benutzernamen, sein Passwort und die Universität, wo er eingeschrieben ist.
2. Die Daten des Nutzers werden in die Datenbank eingepflegt.
3. Nach dem Registrierungsvorgang ist der Benutzer im System angemeldet.

Client Login $\langle F20 \rangle$

Anwendungsfall: Login am Client

Anforderung: $\langle RM1 \rangle$

Ziel: Der Benutzer loggt sich am Client ein, um das Spiel zu starten.

Vorbedingung: Der Benutzer muss einen Account besitzen.

Nachbedingung Erfolg: Der Benutzer ist eingeloggt und kann entweder das Spiel starten oder neue Fragen erstellen.

Nachbedingung Fehlschlag: Die Anmeldung ist fehlgeschlagen, somit ist der Benutzer nicht eingeloggt.

Akteure: Benutzer Client Datenbank

Beschreibung:

1. Der Benutzer gibt seinen Benutzernamen und Passwort ein.
2. Die Benutzerdaten werden mit der Benutzer-Tabelle der Datenbank abgeglichen.
3. Bei Übereinstimmung wird der Benutzer eingeloggt.

Erweiterung: Der Benutzer kann seinen Benutzernamen und sein Passwort nicht ändern. Falls das Passwort vergessen wurde, muss der Benutzer sich neu registrieren. Im Folgenden sei der Benutzer stets am System angemeldet.

4.2 Spielfunktionen

Der angemeldete Benutzer ist in erster Linie ein Spieler. Ein Spieler kann gegen einen anderen Spieler nur im 2-Spieler-Modus antreten, der Einzelspieler-Modus ist nicht vorgesehen.

Eröffnung eines Spieles gegen einen zufälligen Nutzer $\langle F30 \rangle$

Anwendungsfall: Der Benutzer eröffnet ein neues Spiel gegen einen zufälligen Nutzer.

Anforderung: $\langle RM2 \rangle$

Ziel: Zwei von dem Spiel zufällig ausgewählten Spieler treten gegeneinander an.

Vorbedingung: Beide Spieler sind am Client eingeloggt, haben das gleiche Fach aus einem Dropdown-Menü ausgewählt und suchen einen zufälligen Spieler als Gegner.

Nachbedingung Erfolg: Das Spiel hat beide Benutzer benachrichtigt, dass der Gegner gefunden wurden, oder dass es kein Gegenspieler gefunden wurde, falls kein anderer Benutzer im gleichen Fach spielen möchte.

Nachbedingung Fehlschlag: Die Suche fehlgeschlagen oder die Verbindung zum Server wurde abgebrochen, somit kein erfolgreicher Spielstart.

Akteure: Benutzer Client Datenbank

Beschreibung:

1. Der Benutzer wählt ein Fach aus, in welchem er spielen möchte.
2. Der Benutzer wählt die Funktion „Random“ aus und wartet auf eine Einladung.

3. Der zufällige Spieler muss ebenfalls im gleichen Fach nach den „Random“-Gegenspielern suchen.
4. Wurde ein Gegenspieler gefunden, wird das neue Spiel erstellt.

Herausfordern eines Gegners $\langle F40 \rangle$

Anwendungsfall: Der Benutzer fordert einen bestimmten Gegner heraus.

Anforderung: $\langle RM2 \rangle$

Ziel: Der Benutzer eröffnet ein neues Spiel gegen einen bestimmten Benutzer.

Vorbedingung: Der Spieler ist am Client eingeloggt, hat ein Fach aus einem Dropdown-Menü ausgewählt.

Nachbedingung Erfolg: Der Gegenspieler bekommt eine Einladung zum Spiel.

Nachbedingung Fehlschlag: Die Einladung fehlgeschlagen oder Verbindung zum Server wurde abgebrochen, somit kein erfolgreicher Spielstart.

Akteure: Benutzer Client Datenbank

Beschreibung:

1. Der Benutzer wählt ein Fach aus, in welchem er spielen möchte.
2. Der Benutzer wählt einen bestimmten Benutzer als Gegner und sendet an ihn eine Einladung zum Spiel.

Annahme einer Herausforderung $\langle F50 \rangle$

Anwendungsfall: Der Benutzer nimmt die Herausforderung an.

Anforderung: $\langle RM4 \rangle$

Ziel: Der Benutzer startet ein Spiel gegen einen bestimmten Benutzer.

Vorbedingung: Der Spieler ist eingeloggt.

Nachbedingung Erfolg: Der Spieler nimmt die Herausforderung an und das Spiel startet.

Nachbedingung Fehlschlag: Die Annahme der Herausforderung fehlgeschlagen oder Verbindung zum Server wurde abgebrochen, somit kein erfolgreicher Spielstart.

Akteure: Benutzer Client Datenbank

Beschreibung:

1. Der Benutzer ist angemeldet und bekommt eine Einladung zum Spiel in einem bestimmten Fach. Er nimmt die Herausforderung an.

Ablehnen einer Herausforderung $\langle F60 \rangle$

Anwendungsfall: Der Benutzer lehnt die Herausforderung ab.

Anforderung: $\langle RM4 \rangle$

Ziel: Der Benutzer lehnt die Einladung zum Spiel ab.

Vorbedingung: Der Spieler ist eingeloggt.

Nachbedingung Erfolg: Der Spieler lehnt die Herausforderung ab, somit kann kein neues Spiel erstellt werden.

Nachbedingung Fehlschlag: Das Ablehnen schlägt fehl.

Akteure: Benutzer Client Datenbank

Beschreibung:

1. Der Benutzer ist online und bekommt eine Einladung zum Spiel in einem bestimmten Fach. Er lehnt die Herausforderung ab.

Aufgaben $\langle F70 \rangle$

Anwendungsfall: Der Benutzer gibt auf.

Anforderung: $\langle RM5 \rangle$

Ziel: Der Spieler kann zu jedem Zeitpunkt aufgeben.

Vorbedingung: Der Spieler ist am Client eingeloggt und spielt gegen einen anderen Benutzer.

Nachbedingung Erfolg: Der Gegenspieler hat aufgegeben und das Spiel wurde beendet.

Nachbedingung Fehlschlag: Die Verbindung zum Server wurde abgebrochen.

Akteure: Benutzer Client Datenbank

Beschreibung:

1. Das Spiel ist beendet, wenn der Benutzer aufgegeben hat.
2. Der Gegner hat gewonnen.

Statistiken $\langle F80 \rangle$

Anwendungsfall: Spielstatistiken werden angezeigt.

Anforderung: $\langle RM6 \rangle$

Ziel: Dem Benutzer werden seine Spielstatistiken angezeigt.

Vorbedingung: Der Spieler ist eingeloggt und das Spiel wurde beendet.

Nachbedingung Erfolg: Der Benutzer kann seine Ergebnisse zu den letzten 5 Spielen ansehen.

Nachbedingung Fehlschlag: Die Verbindung zum Server wurde abgebrochen, somit können keine Spielergebnisse angezeigt werden.

Akteure: Benutzer Client Datenbank

Beschreibung:

1. Nachdem das Spiel beendet wurde zeigt das System dem Benutzer seine Ergebnisse an.
2. Es wird angezeigt ob er gewonnen oder verloren hat sowie die Anzahl der richtig und falsch beantworteten Fragen.

3. Der Benutzer kann die Ergebnisse der fünf letzten Spiele ansehen.

4.3 Administration des Fragenkatalogs

Erstellen einer neuen Frage $\langle F90 \rangle$

Anwendungsfall: Der Benutzer erstellt neue Frage.

Anforderung: $\langle RM3 \rangle$

Ziel: Der Benutzer erstellt eine Frage und vier Antwortmöglichkeiten dazu, die im digitalen Fragenkatalog gespeichert werden.

Vorbedingung: Der Spieler ist am Client eingeloggt.

Nachbedingung Erfolg: Die neu erstellte Frage wurde in der Datenbank gespeichert.

Nachbedingung Fehlschlag: Die Verbindung zum Server wurde abgebrochen, somit wurde die Frage nicht gespeichert.

Akteure: Benutzer Client Datenbank

Beschreibung:

1. Der Benutzer wählt ein Fach aus.
2. Als nächstes gibt er seine Frage und vier Antwortmöglichkeiten ein, wobei eine davon richtig und drei falsch sind.
3. Der Benutzer speichert die Frage und die Daten werden an den Server geschickt.

Melden einer veralteten oder fehlerhaften Frage $\langle F100 \rangle$

Anwendungsfall: Der Benutzer meldet die veraltete oder fehlerhafte Frage.

Anforderung: $\langle RC2 \rangle$

Ziel: Der Administrator wird vom Benutzer benachrichtigt, dass die Frage fehlerhaft oder veraltet ist.

Vorbedingung: Der Spieler beantwortet eine Frage, dabei fällt ihm auf, dass die Frage veraltet oder fehlerhaft ist.

Nachbedingung Erfolg: Die Frage wurde vom Benutzer gemeldet.

Nachbedingung Fehlschlag: Die Verbindung zum Server wurde abgebrochen, somit wurde keine Nachricht an Administrator geschickt.

Akteure: Benutzer Client Datenbank

Beschreibung:

1. Der Benutzer spielt die Runde und beantwortet Fragen, dabei fällt ihm auf, dass eine Frage veraltet oder fehlerhaft ist.

2. Der Benutzer meldet die Frage.
3. Der Administrator wird benachrichtigt und kann die Frage überprüfen.

4.4 Administratorfunktionen

Der Administrator verfügt über alle Benutzerfunktionen, und kann darüber hinaus die Eigenschaften des Systems konfigurieren. Zudem kann der Administrator die von den Benutzern neu erstellten Fragen kontrollieren und freischalten.

Den digitalen Fragenkatalog aktualisieren $\langle F110 \rangle$

Anwendungsfall: Der Administrator kontrolliert neu eingegangene Fragen.

Anforderung: $\langle RC2 \rangle$

Ziel: Die Fragen-Tabelle in der Datenbank wird aktualisiert, neu erstellte Fragen werden freigeschaltet.

Vorbedingung: Der Administrator ist eingeloggt. In der Datenbank sind neu erstellte, noch nicht freigeschaltete Fragen gespeichert.

Nachbedingung Erfolg: Die neu erstellten Fragen sind vom Administrator freigeschaltet.

Nachbedingung Fehlschlag: Die Verbindung zum Server wurde abgebrochen, somit keine Fragen wurden freigeschaltet.

Akteure: Benutzer Client Datenbank

Beschreibung:

1. Der Administrator löscht die fehlerhafte Frage und schaltet die fehlerfreie Frage frei.

5 Produktdaten

Auf dem Server wird eine Datenbank erstellt, wo die Spielerprofile, der Fragenkatalog, Fächerdaten, Statistiken der gespielten Spiele, Daten zu den Universitäten gespeichert werden.

Übersicht der Datensätze:

Nutzerdaten $\langle D10 \rangle$

Alle Informationen zu einem Benutzer:

- Benutzername,
- Passwort (verschlüsselt),
- Universität/Hochschule

Daten zu den Universitäten/Hochschulen $\langle D20 \rangle$

Alle Informationen zur angegebenen Universität oder der Hochschule:

- Uni-ID,
- Name der Universität/Hochschule

Fächerdaten $\langle D30 \rangle$

Informationen zu den Fächern verschiedener Universitäten und Hochschulen:

- Fach-ID,
- Bezeichnung des Faches

UniFach $\langle D40 \rangle$

Verknüpfung von Fach und Uni:

- ID,
- Fach,
- Universität

Daten zu den Fragen $\langle D50 \rangle$

Fragenkatalog für das Spiel:

- Frage-ID,
- Benutzer-ID: verweist auf den Benutzer, der die Frage erstellt hat,

- UniFach-ID,
- Status der Frage (0 = vom Administrator noch nicht freigeschaltet, 1 = vom Administrator freigeschaltet),
- Korrekte Antwortmöglichkeit,
- Falsche Antwortmöglichkeit 1,
- Falsche Antwortmöglichkeit 2,
- Falsche Antwortmöglichkeit 3

Spieldaten $\langle D60 \rangle$

- Spiel-ID,
- UniFach-ID,
- Benutzer-ID1,
- Benutzer-ID2,
- Bereitschaft zum Spiel (User 1) (0 = der Benutzer ist nicht bereit zu spielen, 1= bereit zu spielen),
- Bereitschaft zum Spiel (User 2) (0 = der Benutzer ist nicht bereit zu spielen, 1= bereit zu spielen),
- die Runden

Daten zu den Runden $\langle D70 \rangle$

- Runde-ID,
- Zu jeder Frage jeder Runde werden die Antworten der Benutzer gespeichert (0= falsch beantwortet, 1= richtig beantwortet),
- die Fragen

6 Nichtfunktionale Anforderungen

6.1 Funktionalität

Produktqualität	sehr gut	gut	normal	nicht relevant
Angemessenheit		x		
Richtigkeit	x			
Interoperabilität		x		
Ordnungsmäßigkeit			x	

Lernduell muss alle vom Kunden geforderten Funktionen in entsprechendem Umfang realisieren, wobei die Richtigkeit jedoch *sehr gut* sein sollte, da falsche Fragen oder Antworten nicht zum Lernerfolg beitragen, sondern das Gegenteil erzielen.

6.2 Sicherheit

Produktqualität	sehr gut	gut	normal	nicht relevant
Zuverlässigkeit		x		
Reife	x			
Fehlertoleranz			x	
Wiederherstellbarkeit		x		

Lernduell sollte von Anfang an eine *sehr gute* Reife besitzen, denn bei einer unausgereiften Lern-App würden die Studenten eher wieder zu herkömmlichen Methoden greifen. Weiterhin sollte *Lernduell* zuverlässig funktionieren, damit ein flüssiger Spiel-, und Lernbetrieb möglich ist. Eine *gute* Wiederherstellbarkeit ist vor allem für den Fragenkatalog wichtig.

6.3 Benutzbarkeit

Produktqualität	sehr gut	gut	normal	nicht relevant
Verständlichkeit		x		
Erlernbarkeit	x			
Bedienbarkeit	x			
Effizienz			x	
Zeitverhalten	x			
Verbrauchsverhalten			x	

Eine *sehr gute* Erlernbarkeit und Bedienbarkeit sind für *Lernduell* essenziell, da es keine Einführung in die App gibt, muss diese schnell erlernbar und intuitiv bedienbar sein, um ein gutes Lern-, und Spielgefühl zu erzeugen. Weiterhin ist ein *sehr gutes* Zeitverhalten für einen optimalen Spiel-, und Lernfluss unabdingbar.

6.4 Änderbarkeit

Produktqualität	sehr gut	gut	normal	nicht relevant
Analysierbarkeit			x	
Modifizierbarkeit			x	
Stabilität		x		
Prüfbarkeit		x		
Übertragbarkeit	x			
Anpassbarkeit		x		
Installierbarkeit	x			
Konformität			x	
Austauschbarkeit				x

Da es sich bei *Lernduell* um eine Android-App handelt, muss diese auf einer Vielzahl von Android-Smartphones lauffähig sein, weswegen eine *sehr gute* Übertragbarkeit und Installierbarkeit sehr wichtig sind. Da unter Android-Studio eine große Menge von Emulatoren verschiedenster Android Geräte vorhanden sind, ist *Lernduell* gut prüfbar. Weiterhin sollte die Stabilität nicht durch Änderungen an der Applikation beeinträchtigt werden.

6.5 Qualitätsanforderungen

- $\langle Q10 \rangle$ Alle spielbaren Fragen müssen auf Richtigkeit überprüft werden
- $\langle Q20 \rangle$ Um eine geringe Versagenshäufigkeit zu gewährleisten muss das Produkt ausgereift ausgeliefert werden
- $\langle Q30 \rangle$ Das Produkt muss anwenderfreundlich sein (intuitive Bedienbarkeit, leichter Einstieg ohne Einführung)
- $\langle Q40 \rangle$ Serverseitige Antwortzeiten dürfen den Spielfluss nicht behindern
- $\langle Q50 \rangle$ Das Produkt soll (Android-)Smartphone unabhängig sein
- $\langle Q60 \rangle$ Das Produkt soll leicht installierbar sein

7 Benutzeroberfläche/Schnittstellen

Folgende Rollen sind zu unterscheiden:

Rolle	Rechte	Benutzeroberfläche
Nutzer	$\langle F10 \rangle$ $\langle F20 \rangle$ $\langle F20 \rangle$ $\langle F30 \rangle$ $\langle F40 \rangle$ $\langle F50 \rangle$ $\langle F60 \rangle$ $\langle F70 \rangle$ $\langle F80 \rangle$	Android-App
Administrator	$\langle F90 \rangle$ $\langle F100 \rangle$ $\langle F110 \rangle$	Android Studio, phpMyAdmin, Android-App

Hinweis: Bei den folgenden Abbildungen handelt es sich nicht um die endgültige Darstellung der Benutzeroberfläche.

Anmeldung $\langle UI10 \rangle$

Bei der ersten Nutzung können sich die Studierenden entweder anmelden (sofern schon ein Benutzerkonto vorhanden ist) oder registrieren. Siehe Abbildung 7.1.

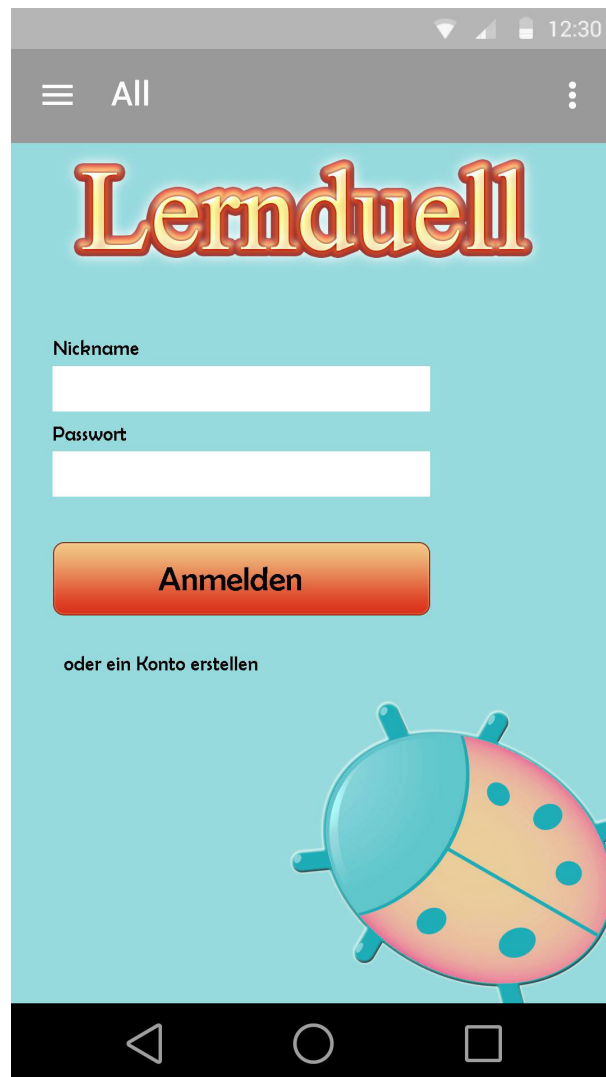


Abbildung 7.1: Anmeldung $\langle UI10 \rangle$

Frage erstellen $\langle UI20 \rangle$

Um eine Frage zu erstellen wählt der Nutzer ein Fach aus einem Dropdown-Menü aus, steht das gewünschte Fach noch nicht zur Verfügung, ist es möglich dieses selbst hinzuzufügen. Nach der Wahl des Fachs, ist die Frage selbst und vier Antwortmöglichkeiten einzugeben. Eine dieser Fragen muss als richtig markiert werden, die restlichen drei als falsch. Anschließend kann die fertige Frage abgesendet werden. Siehe Abbildung 7.2.



Abbildung 7.2: Frage erstellen $\langle UI10 \rangle$

Spiel $\langle UI30 \rangle$

Jedes Spiel besteht aus 3 Runden mit jeweils 3 Fragen. Die Buttons haben verschiedene Farben: grün - richtig, rot - falsch beantwortet, gelb - die Antwort des Gegenspielers. Siehe Abbildung 7.3.

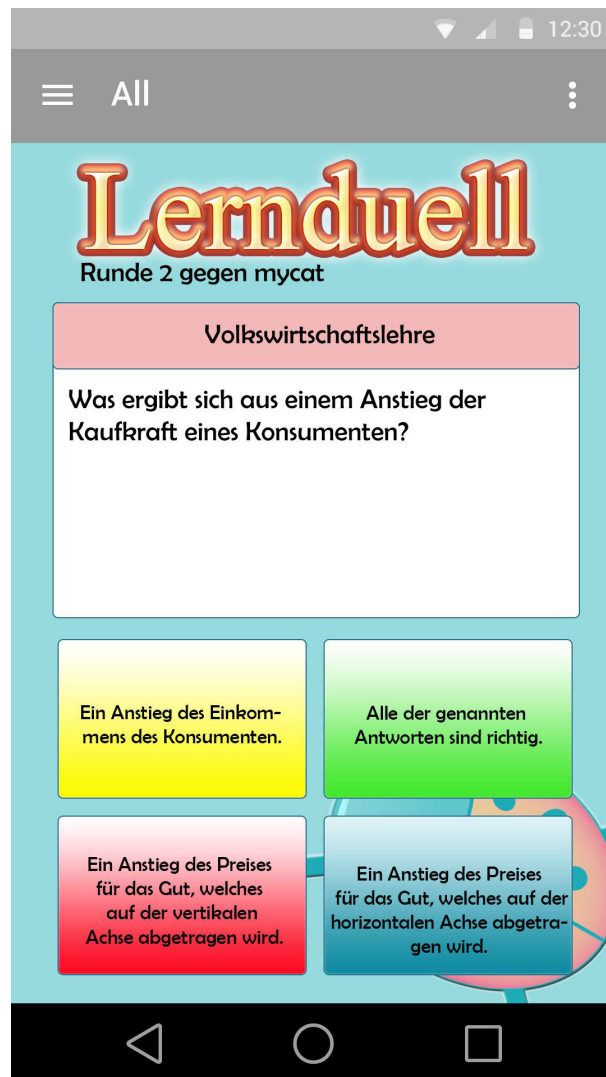


Abbildung 7.3: Spiel $\langle UI10 \rangle$

Statistik $\langle UI40 \rangle$

Dem Nutzer werden jeweils die letzten fünf gespielten Spiele (Richtigkeit der beantworteten Fragen von Spieler und Gegenspieler) zur Verfügung gestellt. Siehe Abbildung 7.4.

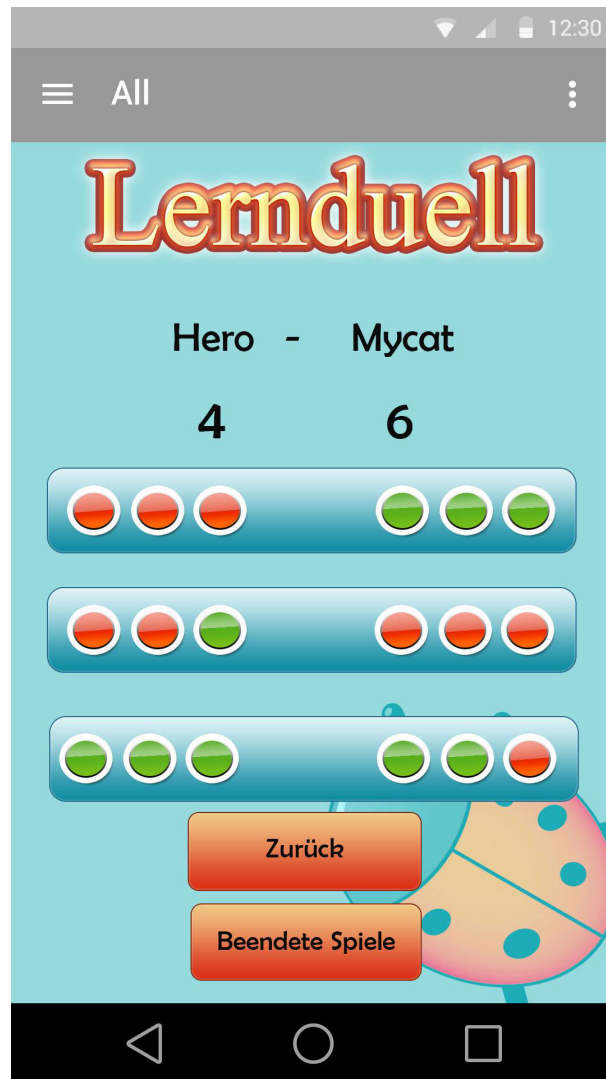


Abbildung 7.4: Statistik $\langle UI10 \rangle$

Bedienbarkeit $\langle UI50 \rangle$

Die Android-App soll intuitiv und leicht bedienbar sein, eine Einführung, oder Einarbeitungszeit sind nicht nötig.

Schnittstellen:

Die Bedienung erfolgt ausschließlich über den Touchscreen des Smartphones.

8 Technische Produktumgebung

8.1 Software

Um *Lernduell* nutzen zu können ist ein Android Smartphone mit mindestens der Version 4.0.3 (IceCreamSandwich) nötig. Für den Spielbetrieb ist außerdem ein Web-Server (momentan mit dem Betriebssystem Debian) nötig. Zur Entwicklung von *Lernduell* wurde *Android Studio* verwendet. Serverseitig wird das Webinterface *phpMyAdmin* genutzt, um eine Datenbank und dessen Managementsystem bereitzustellen. Als Datenbankmanagementsystem kommt *MySQL* zum Einsatz.

8.2 Hardware

Lernduell kann auf jedem Android Smartphone mit Internetverbindung verwendet werden. Weiterhin ist ein Server für u.a. den Fragenkatalog und das Match-Making nötig.

8.3 Produktschnittstellen

Zur Authentifizierung wird HTTP Basic Authentication genutzt, dabei werden der Nutzernamen und das Passwort Base64 codiert und an den Server gesendet. Für die Speicherung der gesendeten Daten vom Mobilgerät zum Webservice wird eine MySQL Datenbank verwendet.

9 Glossar

Android Betriebssystem und Software-Plattform für mobile Geräte.

Java Objektorientierte Programmiersprache.

Server Computerprogramm oder Computer für den Zugriff auf eine zentrale Ressource oder Dienst in einem Netzwerk.

MySQL Relationales Datenbankverwaltungssystem.

PHP „PHP: Hypertext Preprocessor“, Skriptsprache mit einer an C und Perl angelehnten Syntax, die hauptsächlich zur Erstellung dynamischer Webseiten oder Webanwendungen verwendet wird.

Smartphone Mobiltelefon (umgangssprachlich Handy), das mehr Computer-Funktionalität und -konnektivität als ein herkömmliches fortschrittliches Mobiltelefon zur Verfügung stellt.

Tablet der Tabletcomputer ist ein tragbarer, flacher Computer in besonders leichter Ausführung mit einem Touchscreen, aber, anders als bei Notebooks, ohne ausklappbare mechanische Tastatur.

App Anwendungssoftware

Backend Das Backend bezeichnet den Teil der Software, der für die Funktionalität und Logik des Programms – im Hintergrund – zuständig ist. Das Backend wird vom Anwender nicht “gesehen”.

Client Client bezeichnet ein Computerprogramm, das auf dem Endgerät eines Netzwerks ausgeführt wird und mit einem Zentralrechner Server kommuniziert. Man nennt auch ein Endgerät selbst, das Dienste von einem Server abrufen, Client.

Account Ein Benutzerkonto, kurz Nutzerkonto ist eine Zugangsberechtigung zu einem zugangsbeschränkten IT-System.

ID Kurzform für: Identifikator. Ein Identifikator ist ein mit einer bestimmten Identität verknüpft Merkmal zur eindeutigen Identifizierung des tragenden Objekts.

Debian Debian ist ein seit 1993 gemeinschaftlich entwickeltes freies Betriebssystem.

phpMyAdmin phpMyAdmin ist eine freie PHP-Applikation zur Administration von MySQL-Datenbanken.

HTTP Basic Authentication HTTP-Authentifizierung ist ein Verfahren, mit dem sich der Nutzer eines Webbrowsers gegenüber dem Webserver bzw. einer Webanwendung als Benutzer authentifizieren kann, um danach für weitere Zugriffe autorisiert zu sein.

Base64 Base64 beschreibt ein Verfahren zur Kodierung von 8-Bit-Binärdaten.