



# SOFTWARE PRAKTIKUM 2015 AM WI2

## LERNDUELL

Software-Entwicklungspraktikum (SEP)  
Sommersemester 2015

## Fachentwurf

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, 3. Juni 2015

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>4</b>
1.1	Projektdetails . . . . .	6
<b>2</b>	<b>Analyse der Produktfunktionen</b>	<b>8</b>
2.1	Analyse von Funktionalität <b>F10</b> : Registrierung eines Benutzers . . . . .	8
2.2	Analyse von Funktionalität <b>F20</b> : Client Login . . . . .	8
2.3	Analyse von Funktionalität <b>F30</b> : Spiel gegen einen zufälligen Benutzer . . . . .	10
2.4	Analyse von Funktionalität <b>F40</b> : Herausfordern eines Gegners . . . . .	11
2.5	Analyse von Funktionalität <b>F50</b> : Herausforderung eines Gegners annehmen . . . . .	12
2.6	Analyse von Funktionalität <b>F60</b> : Herausforderung eines Gegners ablehnen . . . . .	12
2.7	Analyse von Funktionalität <b>F70</b> : Spiel abbrechen . . . . .	14
2.8	Analyse von Funktionalität <b>F80</b> : Abrufen der Statistik . . . . .	15
2.9	Analyse von Funktionalität <b>F90</b> : Erstellen einer neuen Frage . . . . .	15
2.10	Analyse von Funktionalität <b>F100</b> : Melden einer veralteten oder fehlerhaften Frage . . . . .	16
2.11	Analyse von Funktionalität <b>F110</b> : Aktualisieren des digitalen Fragenkataloges . . . . .	19
2.12	Analyse von Funktionalität <b>F120</b> : Spielen . . . . .	19
<b>3</b>	<b>Datenmodell</b>	<b>21</b>
3.1	Diagramm . . . . .	21
3.2	Erläuterung . . . . .	23
<b>4</b>	<b>Konfiguration</b>	<b>28</b>
<b>5</b>	<b>Glossar</b>	<b>29</b>

## Abbildungsverzeichnis

1.1	Aktivitätsdiagramm <i>User spielt Lernduell</i> . . . . .	5
1.2	Aktivitätsdiagramm <i>Spiel zwischen zwei zufällig ausgewählten Usern</i> . . . . .	7
2.1	Sequenzdiagramm <b>F10</b> <i>Registrierung eines Benutzers</i> . . . . .	9
2.2	Sequenzdiagramm <b>F20</b> <i>Client Login</i> . . . . .	10
2.3	Sequenzdiagramm <b>F30</b> <i>Spiel gegen einen zufälligen Benutzer</i> . . . . .	11
2.4	Sequenzdiagramm <b>F40</b> <i>Herausfordern eines Gegners</i> . . . . .	12
2.5	Sequenzdiagramm <b>F50</b> <i>Herausforderung eines Gegners annehmen</i> . . . . .	13
2.6	Sequenzdiagramm <b>F60</b> <i>Herausforderung eines Gegners ablehnen</i> . . . . .	14
2.7	Sequenzdiagramm <b>F70</b> <i>Spiel abbrechen</i> . . . . .	15
2.8	Sequenzdiagramm <b>F80</b> <i>Abrufen der Statistik</i> . . . . .	16
2.9	Sequenzdiagramm <b>F90</b> <i>Erstellen einer neuen Frage</i> . . . . .	17
2.10	Sequenzdiagramm <b>F100</b> <i>Melden einer veralteten oder fehlerhaften Frage</i> . . . . .	18
2.11	Sequenzdiagramm <b>F110</b> <i>Aktualisieren des digitalen Fragenkataloges</i> . . . . .	19
2.12	Sequenzdiagramm <b>F120</b> <i>Spielen</i> . . . . .	20
3.1	Klassendiagramm des Datenmodells von <i>Lernduell</i> . . . . .	22

# 1 Einleitung

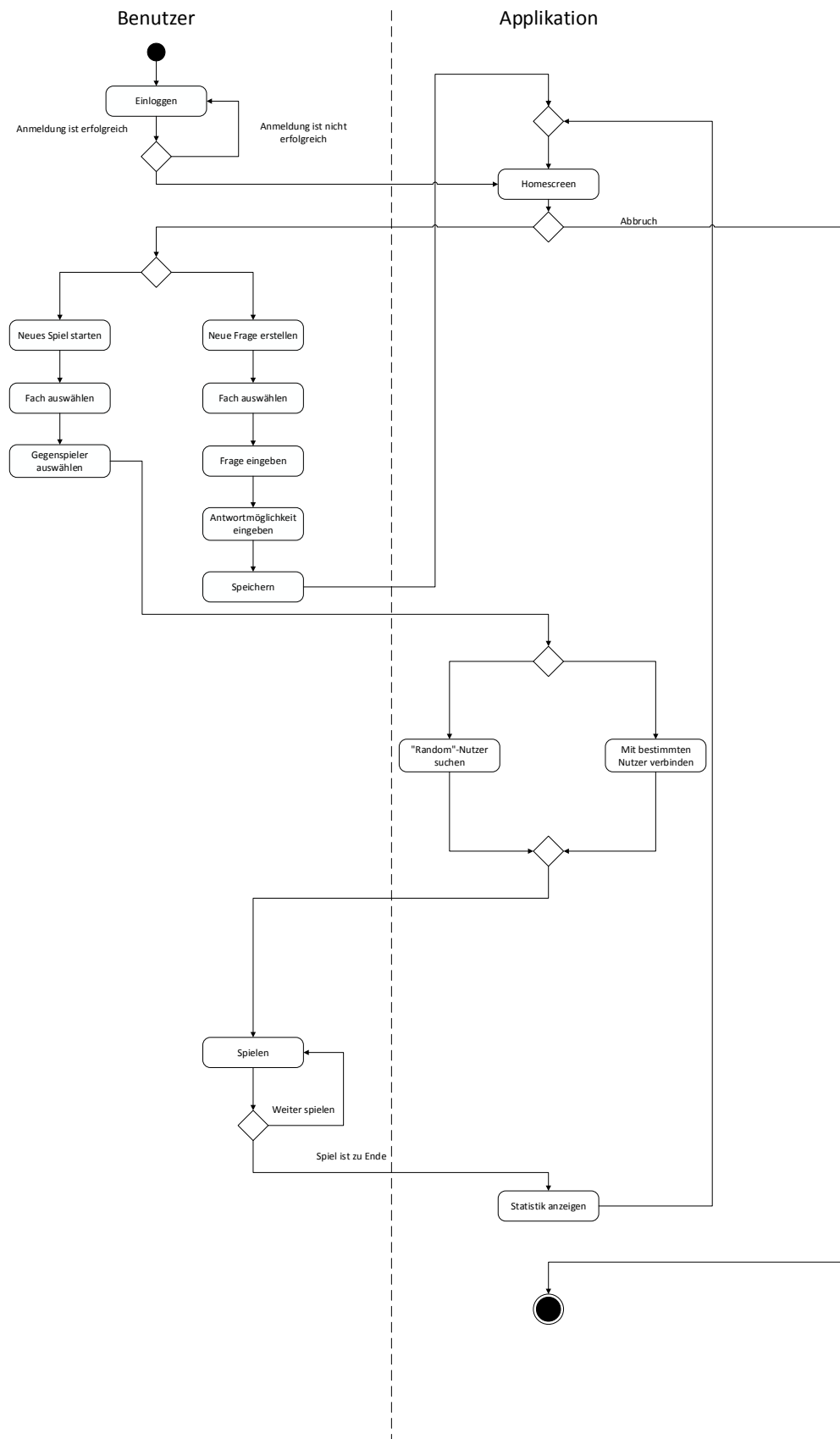
Aufgabe des vorliegenden Dokuments ist die fachliche Modellierung des Systems. Dabei soll mit Hilfe von verschiedenen Diagrammen ein ungefährer Überblick über das Verhalten und die anfallenden sowie verwendeten Daten des Projekts *Lernduell* verschafft werden.

*Lernduell* ist ein Spiel, bei dem Studenten Fragen aus verschiedenen Fächern beantworten müssen und gegen reale Gegner antreten.

Der zum Spielen benötigte Client in Form einer Android Applikation arbeitet im Zusammenhang mit einem zentralen Server, welcher für die Speicherung der Spiel- und Spielerdaten, sowie das Match-Making zuständig ist.

Mit Hilfe des Aktivitätsdiagramms 1.1 wird die Struktur von *Lernduell* genauer beschrieben:

Nach dem starten der Applikation möchte der Nutzer sich anmelden. Falls die Anmeldung erfolgreich ist, ist der Benutzer eingeloggt und wird direkt auf den Homescreen geleitet. Ist die Anmeldung nicht erfolgreich (wenn der User z.B. seine Daten falsch eingegeben, oder sich noch nicht registriert hat) muss er die Anmeldung erneut durchführen. Auf dem Homescreen hat der Nutzer drei weitere Möglichkeiten: Entweder eine neue Frage zu erstellen, ein neues Spiel zu starten oder den Spielprozess zu beenden. Falls er sich dazu entscheidet eine Frage zu erstellen, muss er zunächst ein Fach auswählen und dazu eine Frage sowie vier Antwortmöglichkeiten eingeben und abschließend die neue Frage speichern. Danach wird der Benutzer zurück auf den Homescreen geleitet. Möchte der Nutzer ein neues Spiel starten, muss 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. In diesem Fall sucht *Lernduell* einen Gegner aus dem 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. Abschließend leitet das System den Spieler wieder zurück auf den Homescreen der Applikation.

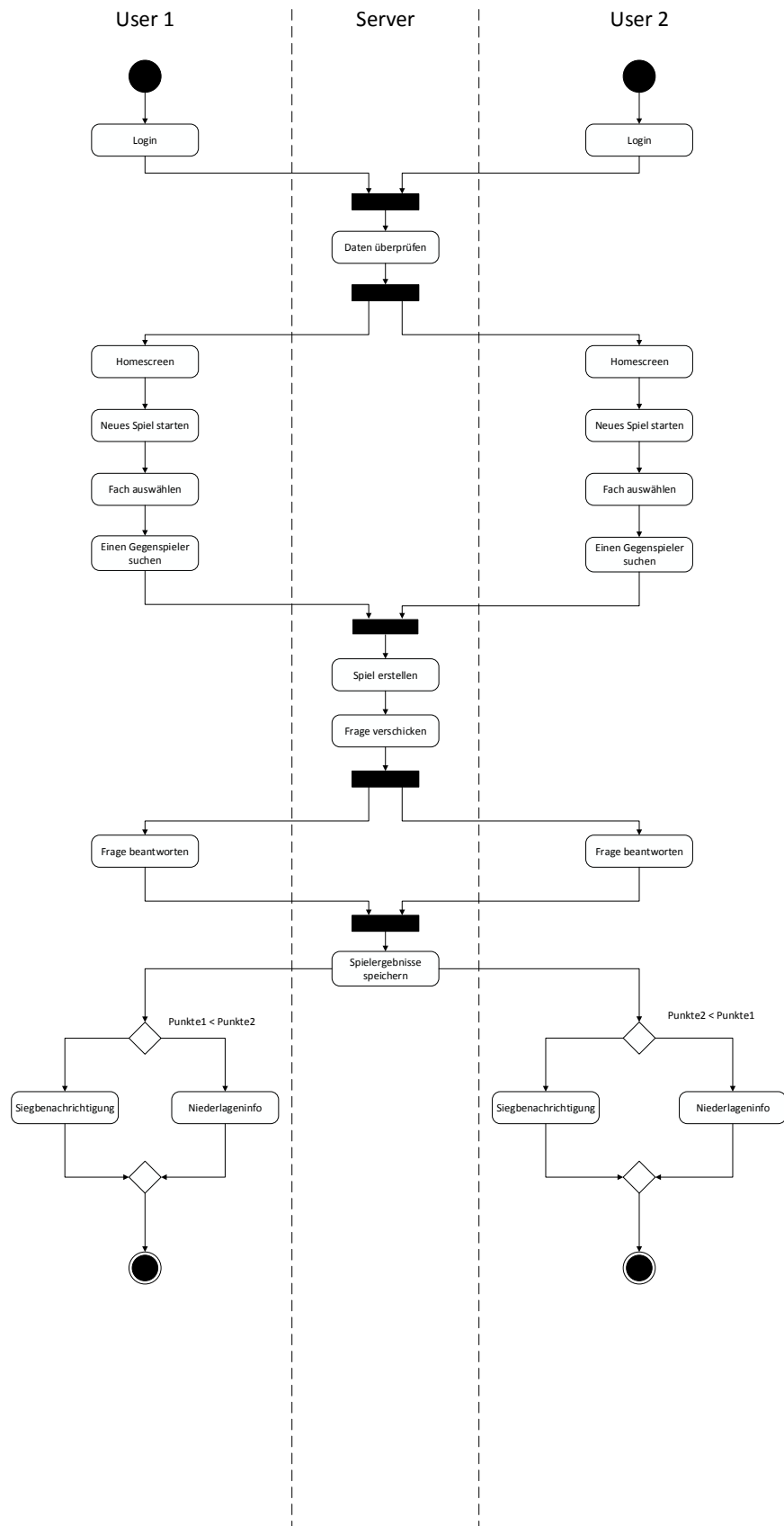
Abbildung 1.1: Aktivitätsdiagramm *User spielt Lernduell*

## 1.1 Projektdetails

Um die in der Einleitung gezeigte Struktur von Lernduell noch detaillierter aufzuzeigen, wird in diesem Abschnitt noch genauer auf den Spielprozess eingegangen und anhand eines Aktivitätsdiagramms visualisiert.

Als Beispiel dient die Interaktion zwischen einem Nutzer und seinem zufällig ausgewählten Gegenspieler, welche der Abbildung 1.2 entnommen werden kann:

Zu Beginn melden sich beide Benutzer an, wobei die Anmeldedaten vom Client an den Server gesendet und überprüft werden. Da die Daten beider Nutzer korrekt sind, wird der Nutzer eingeloggt auf den Homescreen geleitet. Nachdem beide Spieler ein Fach ausgewählt und den „Random“-Button gedrückt haben, wird vom Server ein neues Spiel erstellt indem die Spieler vom System als Gegner ausgewählt werden. Von der Serverseite werden Fragen verschickt, die von den Spielern beantwortet werden müssen. Wenn das Spiel zu Ende ist, werden die Spielergebnisse in der Datenbank des Servers gespeichert. Das System zeigt den Nutzern ihre Ergebnisse an. Falls der Benutzer mehr Fragen als sein Gegner richtig beantwortet hat, wird eine Siegnachrichtigung angezeigt, falls der Gegner gewinnt, eine Niederlageninfo.

Abbildung 1.2: Aktivitätsdiagramm *Spiel zwischen zwei zufällig ausgewählten Usern*

## 2 Analyse der Produktfunktionen

In diesem Kapitel werden die im Pflichtenheft vorgestellten Produktfunktionen analysiert und in Sequenzdiagrammen dargestellt. So wird gut erkennbar, welche Nachrichten die einzelnen Komponenten von *Lernduell* austauschen und welche Funktionen aufgerufen werden um die jeweilige Funktion zu realisieren. Dies stellt insgesamt die Basis dar, um später eine geeignete Architektur festlegen zu können.

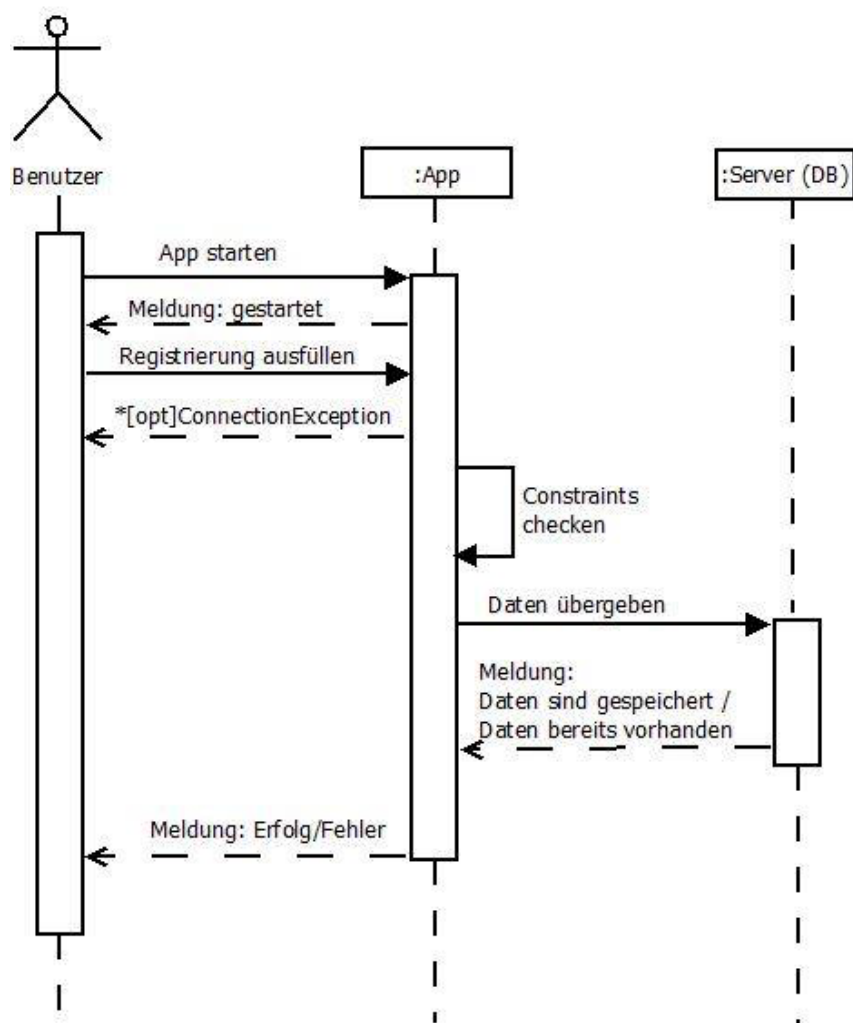
### 2.1 Analyse von Funktionalität F10: Registrierung eines Benutzers

Die Funktion *Registrieren eines Benutzers* ist notwendig um die App nutzen zu können, muss dafür aber auch nur einmal pro Benutzer durchgeführt werden. Denn mit dieser Funktion erstellt der Benutzer ein Nutzerkonto. Im folgenden Sequenzdiagramm in Abbildung 2.1 werden nun die Komponenten dargestellt, die an diesem Prozess beteiligt sind. Um sich zu registrieren, muss ein Benutzer zuerst die Applikation starten. Als nächstes gibt der Benutzer auf dem Mobilgerät einen Benutzernamen und ein Passwort an. Diese Daten, sowie aus der ComboBox ausgewählte Universität, werden dann an den Server übermittelt, welcher die Daten in die Benutzertabelle der Datenbank einträgt. Dabei kann eine *ConnectionsException* ausgelöst werden, wenn es Verbindungsprobleme geben sollten. Das Ergebnis des Servers kann entweder eine Erfolgsnachricht sein, oder ein Fehler, falls der Benutzername in der Datenbank vorhanden ist. Im letzteren Fall wird eine *NameAlreadyUsedException* erstellt.

### 2.2 Analyse von Funktionalität F20: Client Login

In dem Sequenzdiagramm in Abbildung 2.2 wird die Funktionalität *Client Login* näher dargestellt. Diese Funktion wird benötigt, damit der Nutzer auf sein vorher erstelltes Nutzerkonto zugreifen kann, um die Funktionen von *Lernduell* verwenden zu können. In der Abbildung gibt der Benutzer als Erstes die benötigten Daten, also seinen Benutzernamen und sein Passwort, in der Lernduell-GUI an. Diese Daten wiederum werden an den Server übermittelt und mit der Benutzerdaten in der Datenbank abgeglichen. Wenn der Benutzer bereits in der Datenbank registriert ist, loggt er sich erfolgreich ein. Wenn keine entsprechenden Benutzerdaten in der Datenbank gefunden wurden, wird ein Fehler ausgegeben und die Eingabe beginnt von vorne. Ist der Login korrekt, wird mithilfe des Session Managements eine Session ID generiert und der



Abbildung 2.1: Sequenzdiagramm **F10** *Registrierung eines Benutzers*

App als Antwort übergeben. Ist der Login inkorrekt wird keine ID generiert und eine negative Session ID zurückgeliefert.

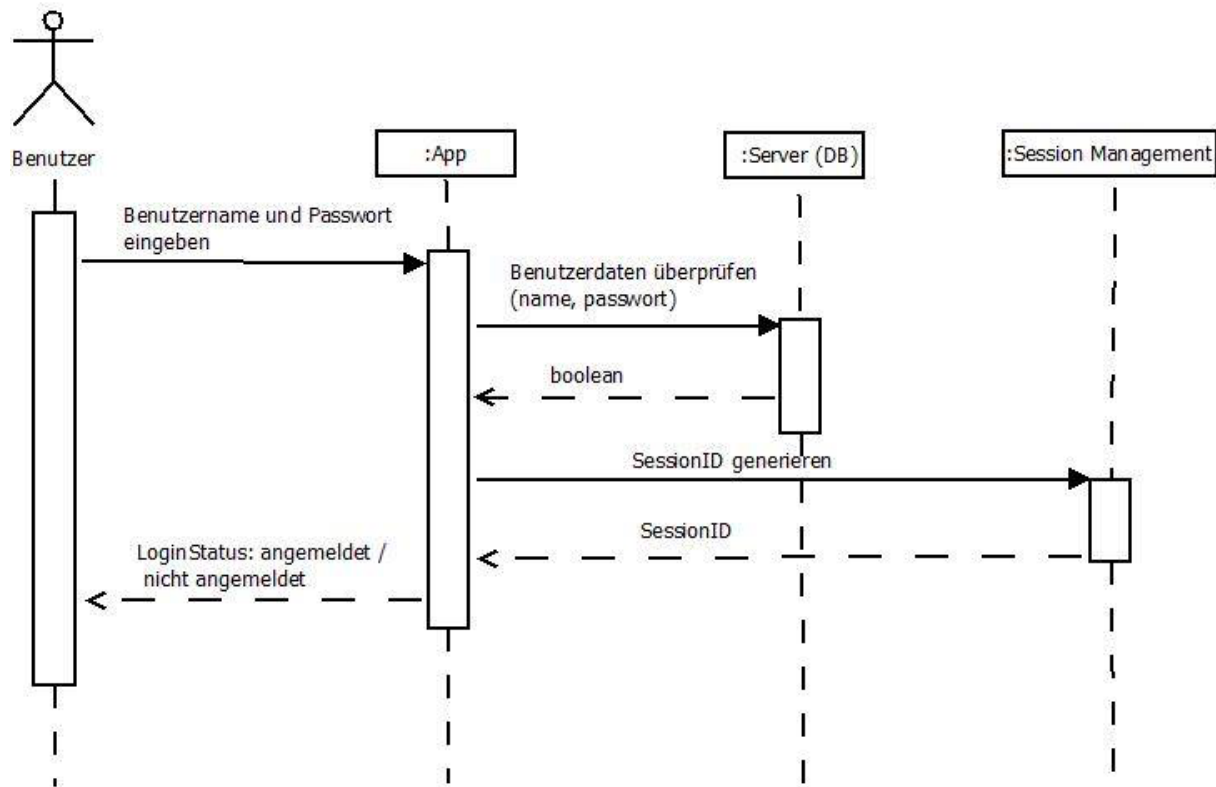


Abbildung 2.2: Sequenzdiagramm **F20** *Client Login*

## 2.3 Analyse von Funktionalität **F30**: Spiel gegen einen zufälligen Benutzer

*Spiel gegen einen zufälligen Benutzer* ist eine Funktionalität, die die Hauptaufgabe von Lernduell erweitert, um dem Nutzer mehr Vielfalt zu bieten. In dem folgenden Sequenzdiagramm in Abbildung 2.3 wird die Funktion bildlich dargestellt. Zunächst ist in der Abbildung dargestellt, wie der Nutzer in der Lernduell-GUI ein neues Spiel startet. Die App zeigt das Dropdown-Menü mit den Fächern. Der Benutzer wählt aus der Liste ein Fach aus und klickt auf den Button „Random-Gegner“. In der Datenbank wird das neue Spiel mit Spielerdaten erzeugt und der Server startet einen zufälligen Gegner zu suchen. Solange dem Benutzer kein Gegner zugeordnet wurde, kann er nur eine Runde spielen. Nachdem die letzte Frage der ersten Runde beantwortet wurde, soll der Benutzer auf den Gegner warten. Das System wählt den Gegenspieler aus, nachdem er sich angemeldet, das gleiche Fach ausgewählt und auf den Button „Random-Gegner“ geklickt hat. Die Lernduell-GUI wird dann vom Server benachrichtigt. Die App leitet den Benutzer in die

nächste Runde und seinen Gegner – in die erste Runde weiter.

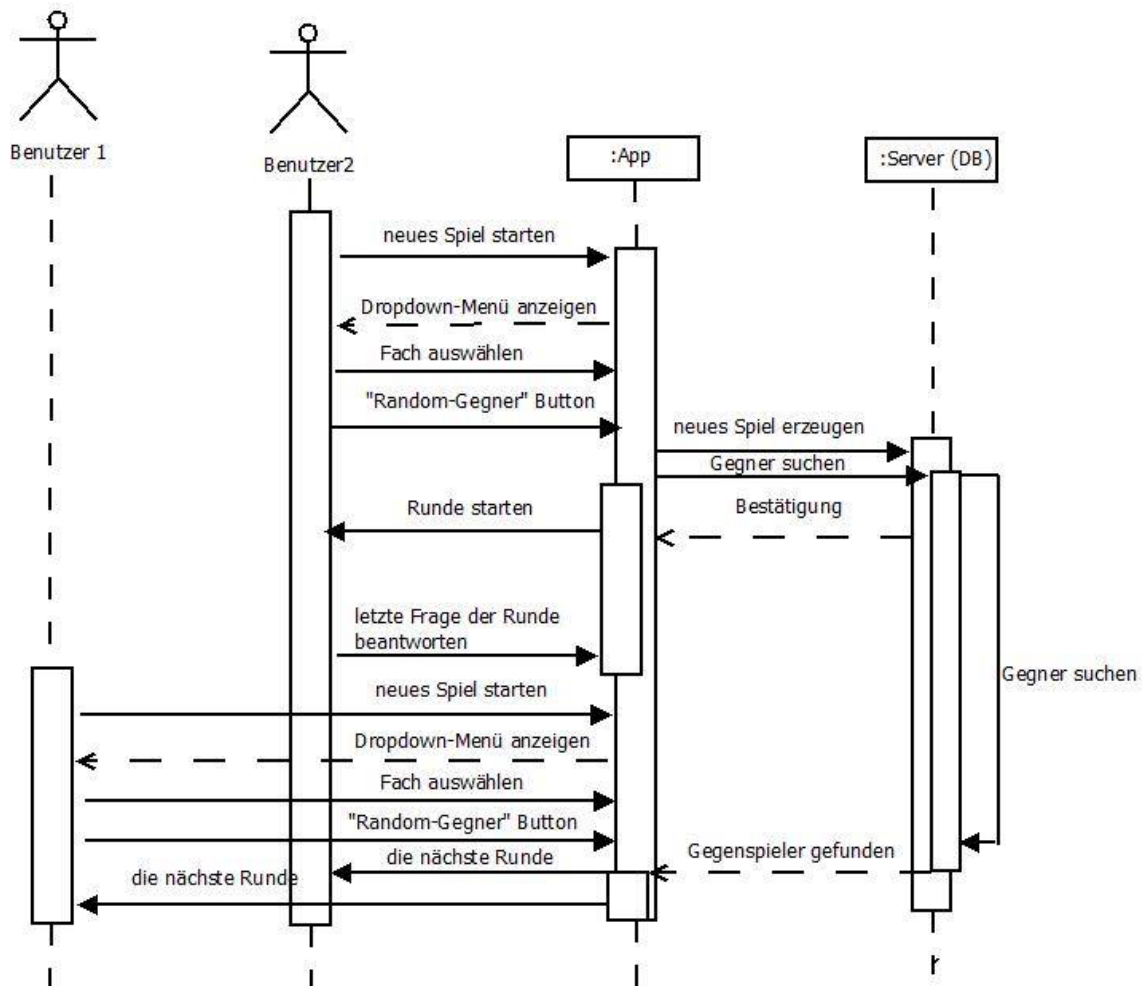
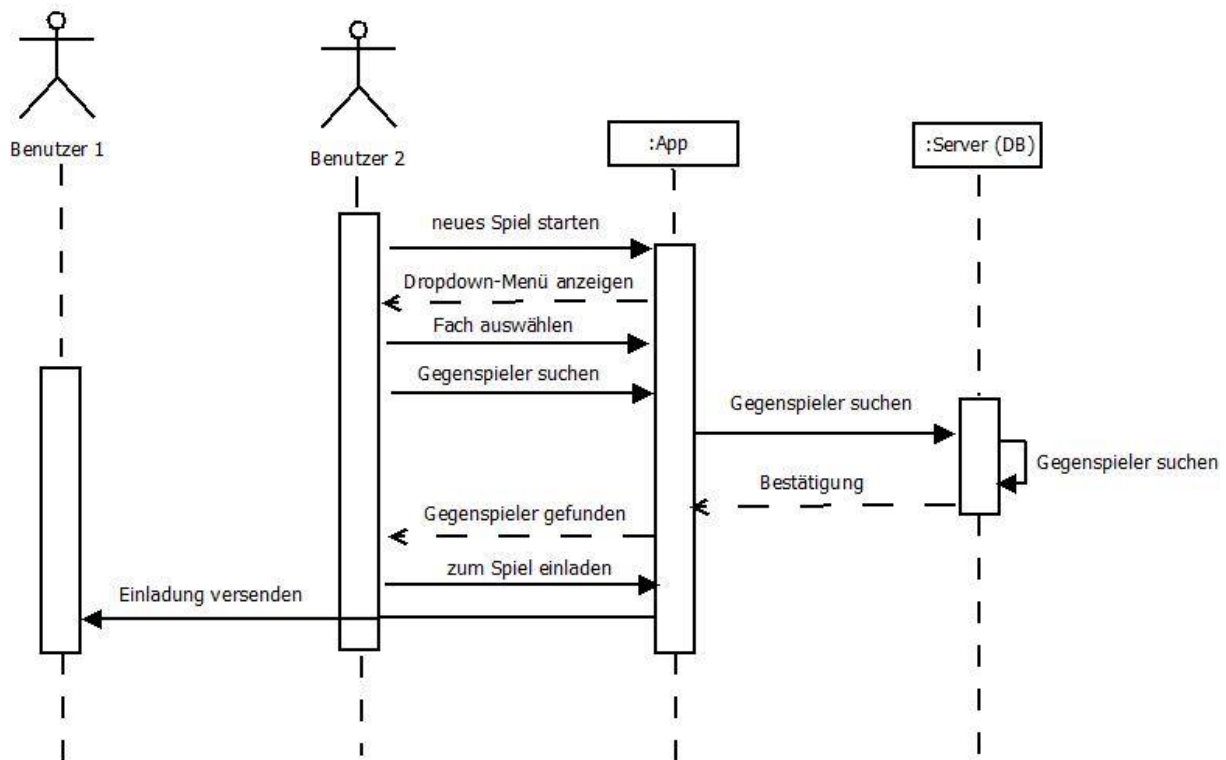


Abbildung 2.3: Sequenzdiagramm **F30** *Spiel gegen einen zufälligen Benutzer*

## 2.4 Analyse von Funktionalität F40: Herausfordern eines Gegners

Die Funktion *Herausfordern eines Gegners* bietet die Möglichkeit, einen bestimmten Benutzer zum Spiel einladen. Dies wird in Abbildung 2.4 in dem Sequenzdiagramm verdeutlicht. Der bereits angemeldete Benutzer startet ein neues Spiel und fordert einen bestimmten Gegner heraus. Dafür wählt er ein Fach aus, gibt den Benutzernamen des Gegners an und klickt auf den Button „Suche“. Die App sendet die Anfrage an den Server und der Server startet die Suche. Nachdem der Benutzer gefunden wurde, sendet der Server eine Bestätigung an die App, die als nächstes den Benutzer darüber informiert. Der Benutzer lädt den Gegner zum Spiel ein, die App versendet die Einladung.

Abbildung 2.4: Sequenzdiagramm **F40** Herausfordern eines Gegners

## 2.5 Analyse von Funktionalität **F50**: Herausforderung eines Gegners annehmen

Das Sequenzdiagramm in der Abbildung 2.5 zeigt die Interaktion des Spielers beim Annehmen einer Herausforderung. Der Benutzer ist angemeldet und erhält von einem anderen Benutzer die Einladung zum Spiel in einem bestimmten Fach. Um das Spiel zu starten, klickt er auf den Button „Herausforderung annehmen“. Sein Gegner wird über Lernduell-GUI informiert, dass Herausforderung angenommen wurde. In der Datenbank wird automatisch ein neues Spiel erzeugt, dabei werden Spiel ID, Benutzer ID1, Benutzer ID2 in die Tabelle „Spiele“ eingetragen. Das System startet die erste Spielrunde und informiert beide Benutzer darüber über GUI.

## 2.6 Analyse von Funktionalität **F60**: Herausforderung eines Gegners ablehnen

Der Benutzer kann die erhaltene Herausforderung ablehnen. Im Sequenzdiagramm, in der Abbildung 2.6, wird näher auf die Funktion eingegangen. Nachdem der Benutzer eine Einladung zum Spiel in einem bestimmten Fach erhalten hat, klickt er auf den Button „Herausforderung

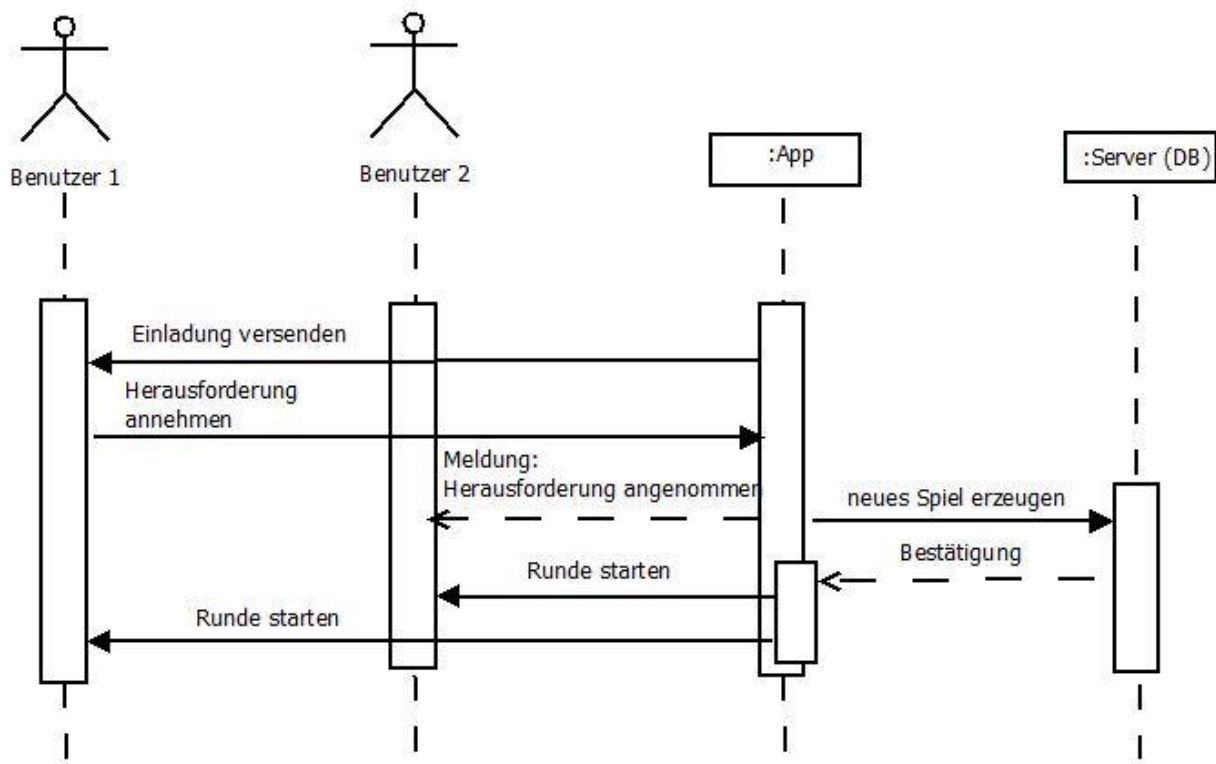


Abbildung 2.5: Sequenzdiagramm **F50** *Herausforderung eines Gegners annehmen*

ablehnen“. Sein Gegner wird darüber über Lernduell-GUI informiert. Anschließend wird dem Benutzer das Hauptmenü von *Lernduell* angezeigt.

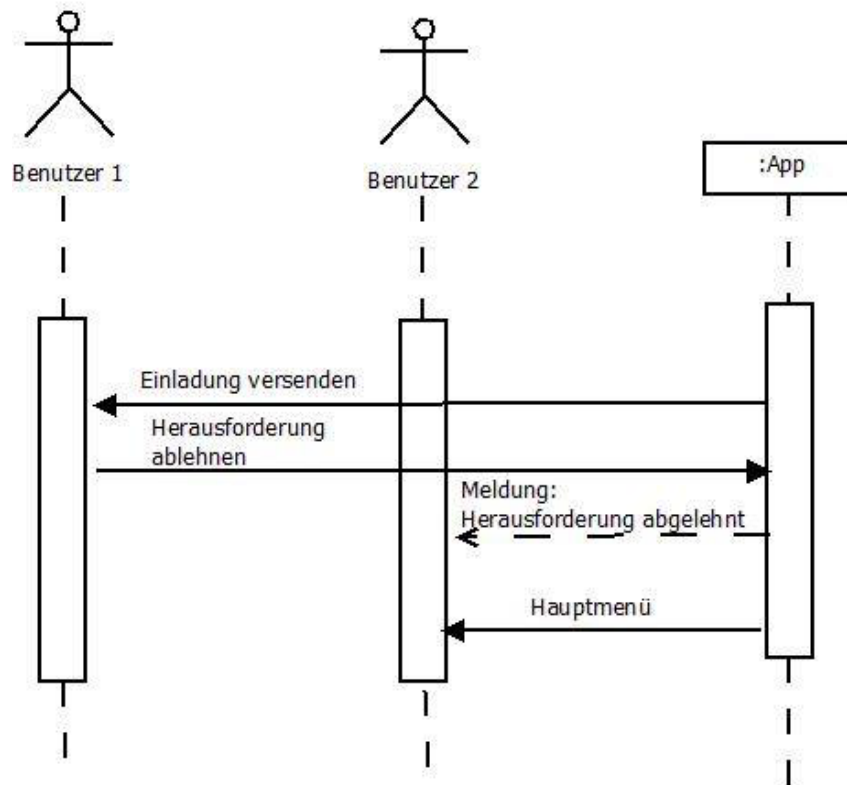
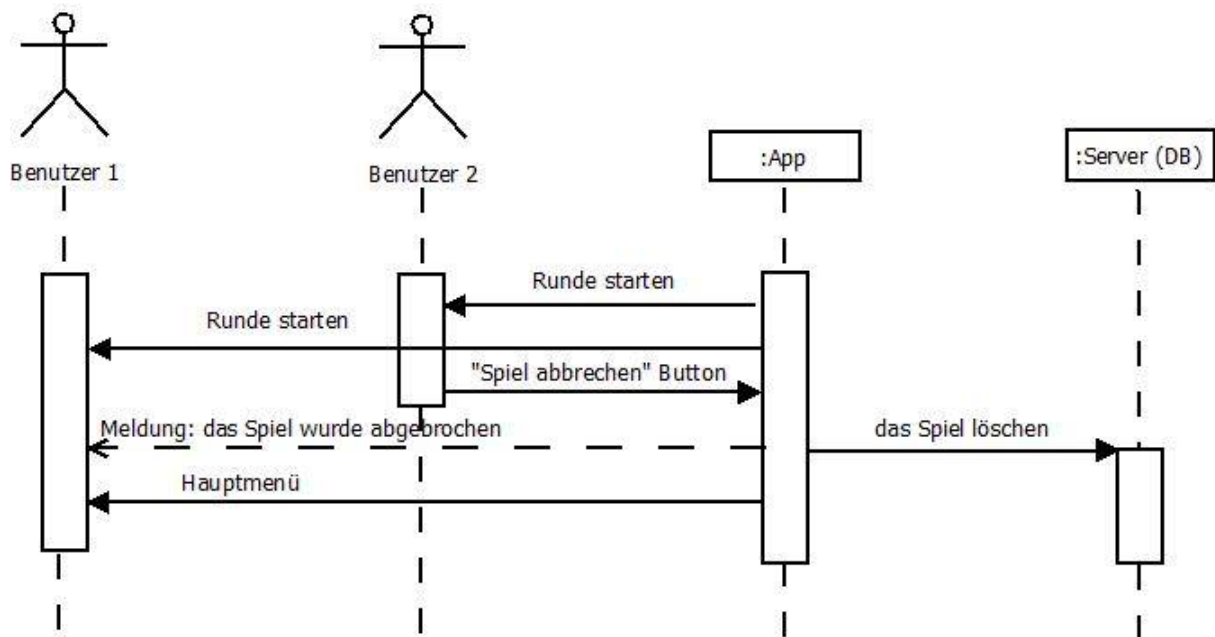


Abbildung 2.6: Sequenzdiagramm **F60** Herausforderung eines Gegners ablehnen

## 2.7 Analyse von Funktionalität F70: Spiel abbrechen

*Spiel abbrechen* ist eine Funktion, welche *Lernduell* erweitert. Sie soll dem Nutzer die Möglichkeit geben, das bereits angefangene Spiel gegen einen anderen Benutzer abzubrechen. Im Sequenzdiagramm in Abbildung 2.7 wird diese Funktion bildlich dargestellt. In der Abbildung wird dargestellt, dass der Benutzer auf den Button „Spiel abbrechen“ in der *Lernduell*-GUI klickt, welche anschließend dem Gegenspieler eine Nachricht anzeigt, dass das Spiel vom Gegner abgebrochen wurde. Gleichzeitig wird der Server benachrichtigt, dass das angefangene Spiel aus der Datenbank gelöscht werden soll. Die App-GUI springt automatisch in das Hauptmenü.

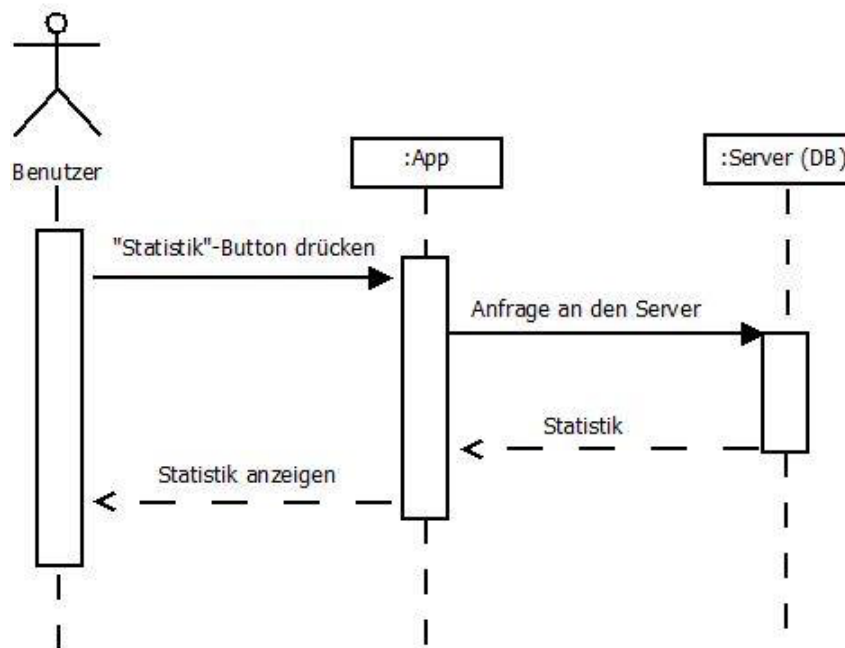
Abbildung 2.7: Sequenzdiagramm **F70** *Spiel abbrechen*

## 2.8 Analyse von Funktionalität F80: Abrufen der Statistik

Im Sequenzdiagramm in Abbildung 2.8 wird die Funktionalität *Abrufen der Statistik* genauer dargestellt. Diese Funktion wird benötigt, damit der Nutzer seine Spielergebnisse ansehen kann. In der Abbildung wird dargestellt, dass der Benutzer auf den „Statistik“-Button in der *Lernduell*-GUI klickt, wenn das Spiel beendet wurde. Die App sendet eine Anfrage an den Server, welcher anschließend dem Benutzer die Ergebnisse seiner letzten fünf gespielten Spiele anzeigt.

## 2.9 Analyse von Funktionalität F90: Erstellen einer neuen Frage

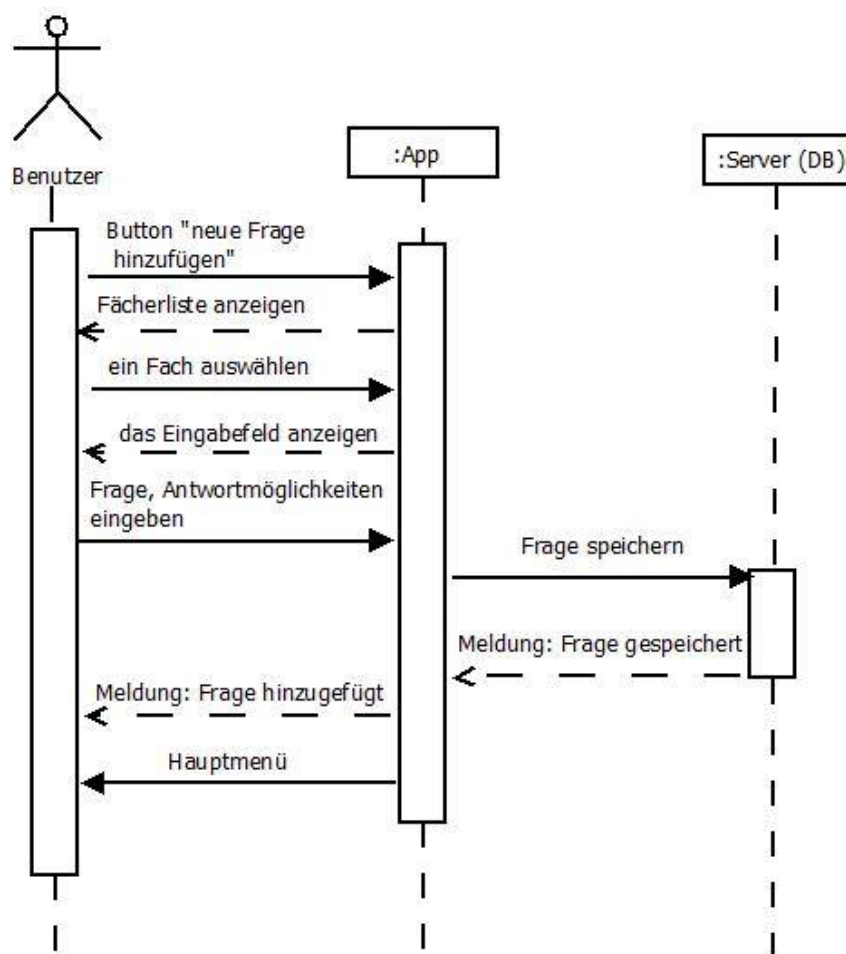
Der Benutzer hat die Möglichkeit den Fragenkatalog durch neue Fragen zu erweitern. Diese Funktion ist in Abbildung 2.9 dargestellt. Dazu klickt er auf den Button „Neue Frage hinzufügen“. Die App zeigt die Fächer im Dropdown-Menü. Nachdem der Benutzer ein Fach aus der Liste ausgewählt hat, wird das Eingabefeld von der App angezeigt, hier kann der Benutzer seine Frage, vier Antwortmöglichkeiten dazu angeben und korrekte Antwort markieren. Die App überträgt Informationen an die Datenbank und der Server sendet eine Bestätigung, dass die Daten erfolgreich gespeichert wurden. Die App-GUI benachrichtigt den Benutzer und springt automatisch in das Hauptmenü.

Abbildung 2.8: Sequenzdiagramm **F80** *Abrufen der Statistik*

## 2.10 Analyse von Funktionalität F100: Melden einer veralteten oder fehlerhaften Frage

*Lernduell* bietet dem Nutzer die Möglichkeit, veraltete oder fehlerhafte Fragen aus dem Fragenkatalog zu melden. Im folgenden Sequenzdiagramm in Abbildung 2.10 wird diese Funktion bildlich erläutert. Ein angemeldeter Spieler findet während eines laufenden Spiels eine fehlerhafte oder veraltete Frage. Im nächsten Schritt beantwortet er die Frage und klickt auf den Button „Frage markieren“. Die Applikation erhält Informationen vom Benutzer, welche Frage markiert wurde, und sendet dementsprechend die Frage an den Server, welcher sie in der Datenbank temporär speichert. Wenn das Spiel beendet wurde, fordert das System aus der Datenbank Informationen zu temporär gespeicherten Fragen an und leitet den Benutzer zurück zu diesen Fragen. Der Benutzer meldet die Frage, der Server speichert die gemeldete Frage. Anschließend wird dem Benutzer das Hauptmenü angezeigt. Die gemeldete Frage wird von dem Server an Administrator übermittelt. Der Administrator prüft die Frage und, falls die fehlerhaft ist, entfernt er die Frage aus dem Fragenkatalog.



Abbildung 2.9: Sequenzdiagramm **F90** *Erstellen einer neuen Frage*

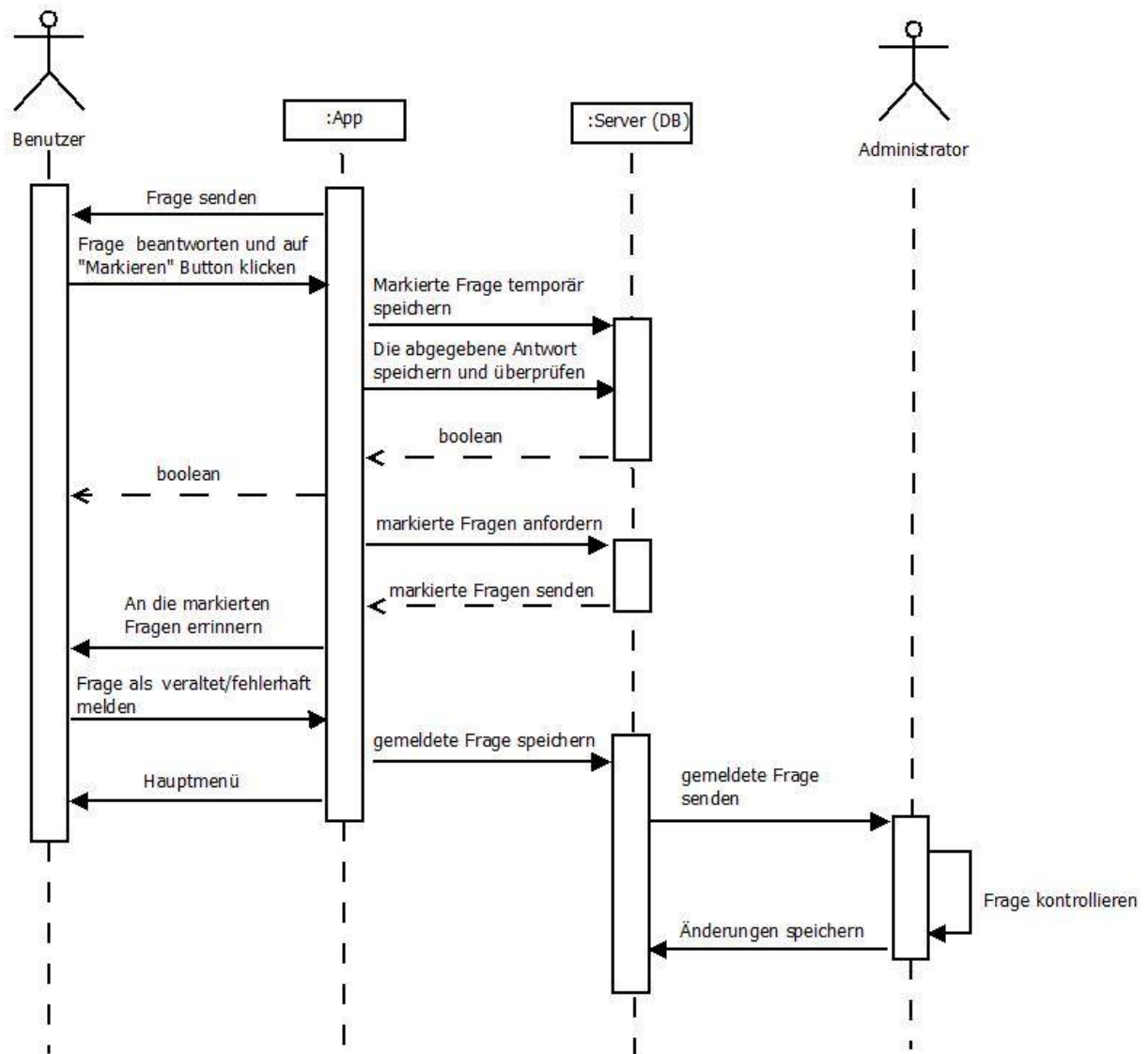


Abbildung 2.10: Sequenzdiagramm **F100** *Melden einer veralteten oder fehlerhaften Frage*

## 2.11 Analyse von Funktionalität F110: Aktualisieren des digitalen Fragenkataloges

Im Sequenzdiagramm in Abbildung 2.11 wird die Administratorfunktion dargestellt. Nachdem neu erstellte Fragen in der Datenbank gespeichert wurden, benachrichtigt der Server den Administrator, welcher die Fragen kontrolliert. Der Administrator löscht fehlerhafte Fragen und schaltet fehlerfreie Fragen frei.

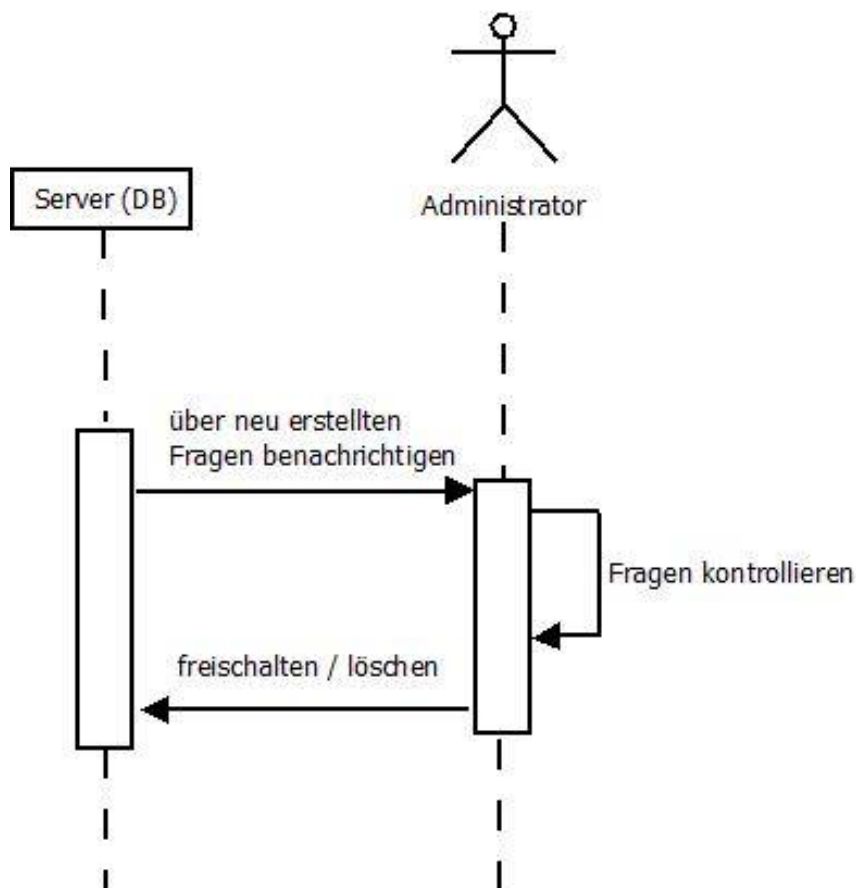


Abbildung 2.11: Sequenzdiagramm **F110** *Aktualisieren des digitalen Fragenkataloges*

## 2.12 Analyse von Funktionalität F120: Spielen

*Spielen* ist eine Kernfunktion der Anwendung *Lernduell*. Diese Funktion bietet dem Nutzer die Möglichkeit die Fragen aus verschiedenen Fächern in Spielform zu beantworten. Die Funktion und die beteiligten Komponenten werden im Sequenzdiagramm in Abbildung 2.12 dargestellt. Nachdem eine Runde gestartet wurde, fordert die App eine Frage aus der Datenbank. Nun über-

gibt das System die ausgewählte Frage an die *Lerduell*-GUI. Diese wiederum ist dafür zuständig, die Frage dem Nutzer anzuzeigen. Gleichzeitig startet Countdown Timer, der 20 Sekunden zählen soll. Beantwortet der Benutzer die Frage, soll der Timer stoppen. Die App leitet die abgegebene Antwort an den Server weiter, welcher die Antwort überprüft und in die Spieltabelle einträgt, ob die Antwort falsch oder richtig beantwortet wurde. Das Ergebnis wird an die App weitergegeben, von wo aus es dann dem Benutzer angezeigt wird.

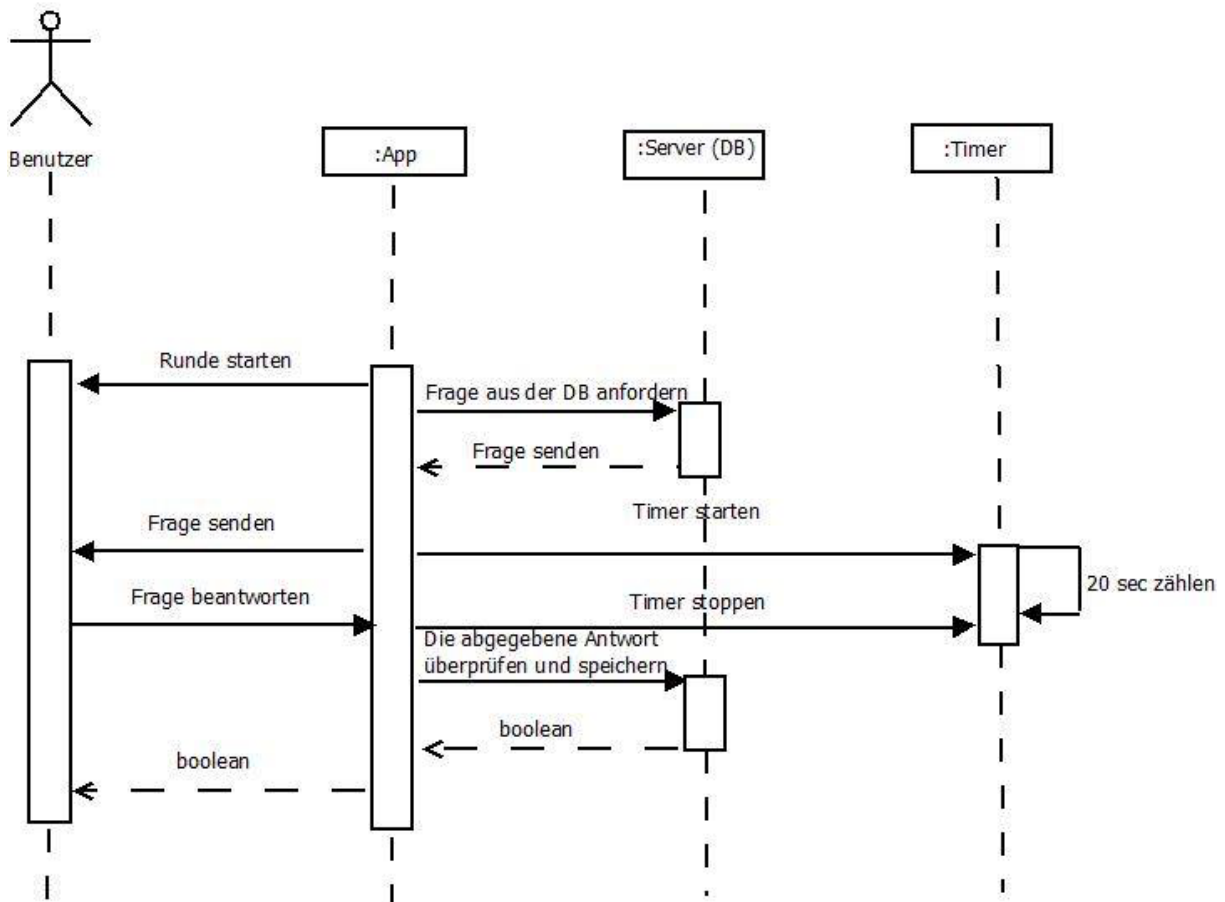


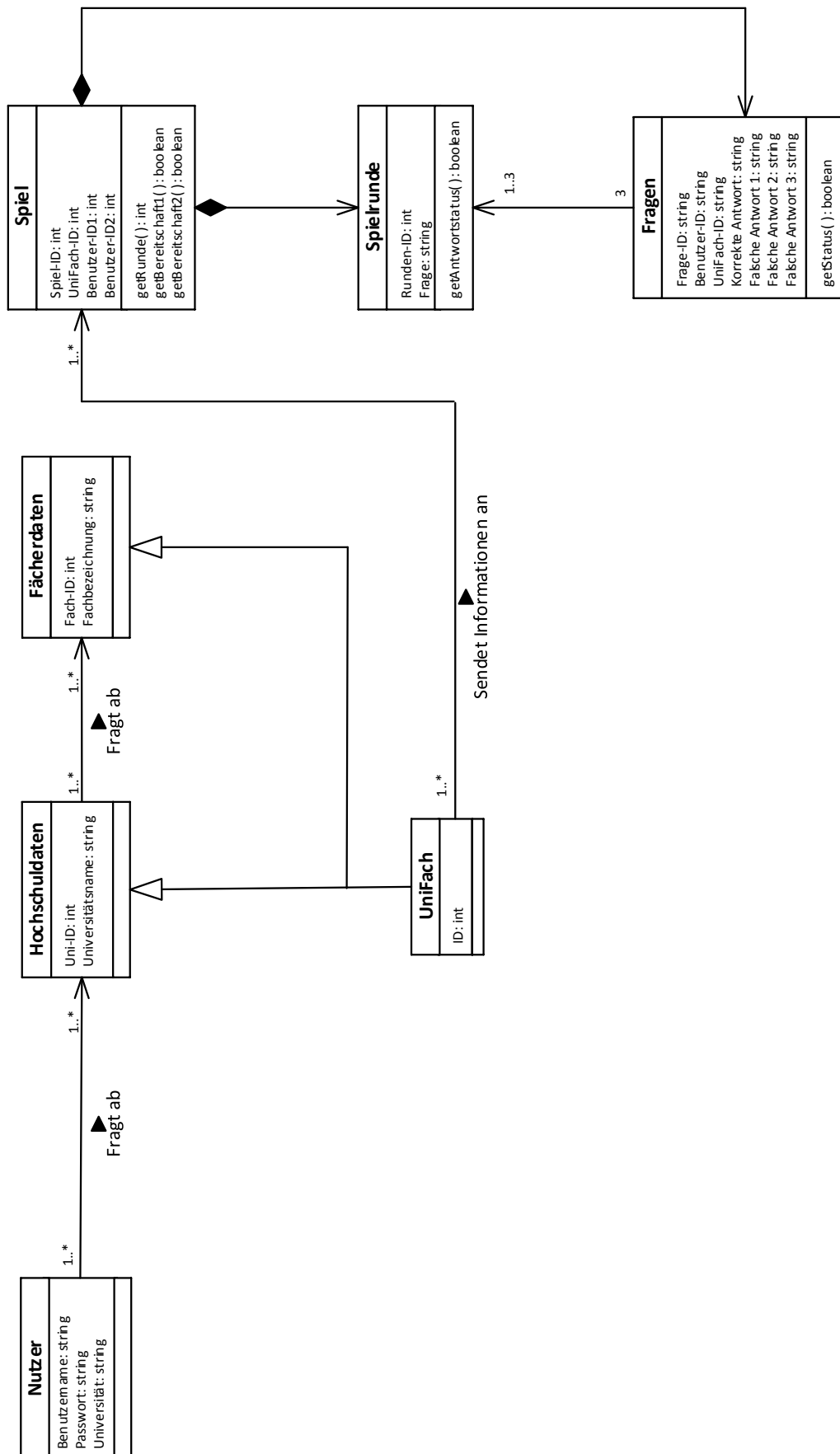
Abbildung 2.12: Sequenzdiagramm **F120 Spielen**

## 3 Datenmodell

In diesem Kapitel wird anhand eines Klassendiagramms dargestellt, welche Datensätze dauerhaft auf dem Server gespeichert werden. Dies umfasst neben den Nutzer- und Universitätsdaten auch die Spielrelevanten Informationen vor, während und nach einer Spiel-Runde von *Lernduell*. Darüber hinaus wird auf die Beziehungen und die Kommunikation zwischen den einzelnen Klassen eingegangen.

### 3.1 Diagramm

Die Abbildung 3.1 zeigt das Datenmodell von *Lernduell* als Klassendiagramm, welches in Abschnitt 3.2 noch näher erläutert wird.

Abbildung 3.1: Klassendiagramm des Datenmodells von *Lernduell*

## 3.2 Erläuterung

### Nutzer $\langle E10 \rangle$

Beziehung	Kardinalität	Erwartete Datenmenge	Beschreibung
Hochschuldaten $\langle E20 \rangle$	1..*	Wenige Kilobyte	Das Nutzerprofil bezieht die Daten der Hochschule und speichert Sie im Profil.

### Hochschuldaten $\langle E20 \rangle$

Beziehung	Kardinalität	Erwartete Datenmenge	Beschreibung
Nutzer $\langle E10 \rangle$	1..*	Wenige Kilobyte	Sendet Daten an die Klasse Nutzer, sobald Sie abgefragt werden.
Fächerdaten $\langle E30 \rangle$	1..*	Wenige Kilobyte	Bezieht die Daten der jeweiligen angebotenen Fächer der Hochschule.
UniFach $\langle E40 \rangle$	-	Wenige Kilobyte	Vererbt seine Daten an die Klasse UniFach, welches die Verknüpfung zwischen den Hochschuldaten und den Fächerdaten darstellt.

**Fächerdaten**  $\langle E30 \rangle$ 

Beziehung	Kardinalität	Erwartete Datenmenge	Beschreibung
Hochschuldaten $\langle E20 \rangle$	1..*	Wenige Kilobyte	Sendet Daten an die Klasse Hochschuldaten, sobald Sie abgefragt werden.
UniFach $\langle E40 \rangle$	-	Wenige Kilobyte	Vererbt die Daten an die Klasse UniFach, welches die Verknüpfung zwischen den Hochschuldaten und den Fächerdaten darstellt.

**UniFach**  $\langle E40 \rangle$ 

Beziehung	Kardinalität	Erwartete Datenmenge	Beschreibung
Hochschuldaten $\langle E20 \rangle$	-	Wenige Kilobyte	Erbt aus der Kategorie Hochschuldaten, da diese Klasse die Verknüpfung zwischen den Hochschuldaten und den Fächerdaten darstellt.
Fächerdaten $\langle E30 \rangle$	-	Wenige Kilobyte	Erbt aus der Kategorie Fächerdaten, da diese Klasse die Verknüpfung zwischen den Hochschuldaten und den Fächerdaten darstellt.



Spiel $\langle E40 \rangle$	1..*	Wenige Kilobyte	Sendet Informationen aus der Klasse UniFach an die Klasse Spiel, wodurch bestimmt wird, welcher Mitspieler geeignet wäre und wie der mögliche Fragenpool für die entsprechende Runde aussieht.
-----------------------------	------	-----------------	--

**Fragen  $\langle E50 \rangle$** 

Beziehung	Kardinalität	Erwartete Datenmenge	Beschreibung
Spiel $\langle E60 \rangle$	-	Wenige Kilobyte	Es handelt sich um eine Komposition. Ohne das Spiel gibt es die Klasse Fragen nicht. Die Fragen für die Klasse Spiel werden hier gespeichert.
Spielrunde $\langle E70 \rangle$	3	Wenige Kilobyte	Die Klasse Fragen sendet die relevanten Spiel-Fragen vor jeder Spielrunde eines laufenden Spiels an die Klasse Spielrunde.

**Spiel**  $\langle E60 \rangle$ 

Beziehung	Kardinalität	Erwartete Datenmenge	Beschreibung
UniFach $\langle E40 \rangle$	1..*	Wenige Kilobyte	Bekommt Informationen von der Klasse UniFach, um einen möglichen Mitspieler und die Fragen zu Bestimmen.
Fragen $\langle E50 \rangle$	1	Wenige Kilobyte	Die Klasse Fragen ist ein existenzabhängiger Teil der Klasse Spiel, in welcher der gesamte Fragenpool für das Spiel gespeichert wird.
Spielrunde $\langle E70 \rangle$	1	Wenige Kilobyte	Die Klasse Spielrunde ist ein existenzabhängiger Teil der Klasse Spiel, in welchem die Daten des laufenden Spiels gespeichert werden.

**Spielrunde**  $\langle E70 \rangle$ 

Beziehung	Kardinalität	Erwartete Datenmenge	Beschreibung
Fragen $\langle E50 \rangle$	1..3	Wenige Kilobyte	Bezieht die Fragen für die jeweilige Spielrunde aus dieser Klasse.
Spiel $\langle E60 \rangle$	-	Wenige Kilobyte	Es handelt sich um eine Komposition. Ohne das Spiel gibt es die Klasse Spielrunde nicht. In der Klasse Spiel wird der Verlauf aller Spiele gespeichert, während in Spielrunde die jeweiligen Runden Daten gesichert werden.

## 4 Konfiguration

Dieses Kapitel geht genauer auf die Hard-, und Software Konfiguration von *Lernduell* ein.

Für den Spielbetrieb von *Lernduell* ist ein Web-Server nötig, welcher während der Entwicklung mit dem Betriebssystem Debian betrieben wurde. Für den Endkunden ist keine besondere Konfiguration nötig. Es muss lediglich ein reibungsloser Betrieb von MySQL als Datenbankmanagementsystem und phpMyAdmin für das Webinterface gewährleistet sein, wobei weder MySQL noch phpMyAdmin nennenswerte Anforderungen an die Hardware des Servers stellen.

Als Client ist ein Android Smartphone mit mindestens der Version 4.0.3 (IceCreamSandwich) nötig.

Zum derzeitigen Entwicklungsstand hat *Lernduell* eine config-Datei, welche sich auf dem Webserver befindet.

Config.php dient der Authentifizierung der anderen php-Skripte um auf die Datenbank zuzugreifen. Die Datei definiert also, unter welcher Adresse der Server zu erreichen ist auf dem die Datenbank liegt (MYSQL\_HOST), den Benutzernamen um sich einzuloggen (MYSQL\_BENUTZER), das entsprechende Passwort (MYSQL\_KENNWORT) und das zu verwendende Datenbankschema (MYSQL\_DATENBANK).

## 5 Glossar

**Android** Betriebssystem und Software-Plattform für mobile Geräte.

**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 oder Applikation** Anwendungssoftware

**Exception** Ausnahmebehandlung

**Combobox** Datenfeld in einer grafischen Benutzeroberfläche (GUI) einer Computer-Software, über das der Benutzer eine Auswahl aus vorgegebenen Möglichkeiten treffen oder alternativ eigene Eingaben tätigen kann.

**GUI** Grafische Benutzeroberfläche

**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.

**Session-ID** Wird bei Anwendungen auf zustandslosen Protokollen als Identifikationsmerkmal verwendet, um mehrere zusammengehörige Anfragen eines Benutzers zu erkennen und einer Sit-

zung zuzuordnen.

**Countdown** die getaktete (kurz vor dem Ende im Sekudentakt) Bekanntgabe der bis zum Eintreten eines bestimmten Ereignisses noch fehlenden Zeitspanne.