

# StarDuell

## SOFTWARE ARCHITECTURE DOCUMENT (SAD)

Nikolai Kloß 1047491

Dennis Lüdeke 1082427

Jan Hendrik Plümer 1078666

Eduard Ljaschenko 1027565

FH-Bielefeld – Campus Minden – Software Engineering

# Inhaltsverzeichnis

<b>1.</b>	<b>EINLEITUNG</b>	<b>5</b>
<b>1.1.</b>	<b>PROJEKTBESCHREIBUNG</b>	<b>5</b>
<b>1.2.</b>	<b>ZWECK DES DOKUMENTS</b>	<b>6</b>
<b>2.</b>	<b>ARCHITEKTONISCHE DARSTELLUNG</b>	<b>7</b>
<b>3.</b>	<b>ZIELE UND RAHMENBEDINGUNGEN DER ARCHITEKTUR</b>	<b>8</b>
<b>3.1.</b>	<b>CLIENT</b>	<b>8</b>
<b>3.2.</b>	<b>SERVER</b>	<b>8</b>
<b>3.3.</b>	<b>PERSISTENZ</b>	<b>8</b>
<b>3.4.</b>	<b>CLIENT-SERVER-KOMMUNIKATION</b>	<b>8</b>
<b>3.5.</b>	<b>BIBLIOTHEKEN/Frameworks</b>	<b>8</b>
<b>3.6.</b>	<b>ENTWICKLUNGSTOOLS</b>	<b>8</b>
<b>4.</b>	<b>USE-CASE-SICHT</b>	<b>9</b>
<b>4.1.</b>	<b>USER STORIES</b>	<b>9</b>
<b>4.2.</b>	<b>ÜBERSICHT</b>	<b>11</b>
<b>4.3.</b>	<b>BESCHREIBUNG</b>	<b>17</b>
<b>4.4.</b>	<b>AKTIVITÄTSDIAGRAMME</b>	<b>24</b>
<b>4.4.1.</b>	<b>PARTIE SPIELEN</b>	<b>24</b>
<b>4.4.2.</b>	<b>PARTIE GEWINNEN</b>	<b>25</b>
<b>4.4.3.</b>	<b>PARTIE VERLIEREN</b>	<b>26</b>
<b>4.4.4.</b>	<b>CLIENTVERWALTUNG</b>	<b>27</b>
<b>4.4.5.</b>	<b>KONTOVERWALTUNG</b>	<b>28</b>
<b>4.4.6.</b>	<b>KONTOANMELDUNG</b>	<b>29</b>
<b>4.4.7.</b>	<b>KONTOREGISTRIERUNG</b>	<b>30</b>
<b>4.4.8.</b>	<b>SYSTEMEINSTELLUNG</b>	<b>32</b>
<b>4.5.</b>	<b>GUI MOCKUPS</b>	<b>33</b>

4.5.1.	HAUPTMENÜ #1	33
4.5.2.	SPIELAUSWAHL #2	34
4.5.3.	EINZELSPIELER STARTEN #3	35
4.5.4.	SPIELANSICHT #4	36
4.5.5.	SERVER ERSTELLEN #5	37
4.5.6.	SERVER BEITRETEN #6	38
4.5.7.	KONTOVERWALTUNG #7	39
4.5.8.	LOGIN ANSICHT #8	40
4.5.9.	REGISTRIERUNG ANSICHT #9	41
4.5.10.	STATISTIK #10	42
4.5.11.	OPTIONEN #11	43
4.6.	GUI NAVIGATIONEN	44
5.	LOGISCHE-SICHT	45
5.1.	ARCHITEKTONISCHE ÜBERSICHT	45
6.	PROZESS-SICHT	46
6.1.	SEQUENZDIAGRAMME	46
7.	DEPLOYMENT-SICHT	47
7.1.	DATENBANK	47
7.2.	VERSIONSKONTROLLE	47
8.	IMPLEMENTATIONS-SICHT	48
8.1.	DOMÄNENKLASSENDIAGRAMM	48
9.	DATEN-SICHT	49
10.	SECURITY-SICHT	50
10.1.	AUTHENTIFIZIERUNG	50
10.2.	DATENBANK	50
11.	GLOSSAR	51



# 1. Einleitung

## 1.1. Projektbeschreibung

Das Ziel ist die Implementierung eines 2-D Spiels als mobile Smartphone Applikation in C# mit dem Unity Framework. Es wird das simple Spielprinzip von „Bomberman“ aufgegriffen. Die App soll die Möglichkeit bieten einen Spieleserver zu hosten oder einen solchen beizutreten. Entweder per Bluetooth oder Wireless LAN. Die Applikation soll kompatibel zu Android und iOS sein. Die Spielgrafik soll simple (evtl. einfach geometrische Formen) und nostalgisch (SNES/8Bit Style) gehalten werden.

### **Was ist Bomberman:**

Das Ziel eines ist es, alle seine Gegner mit seinen Bomben weg zu sprengen. Es wird auf einer Labyrinth-artigen Karte gespielt. Die Karte besteht aus 3 Arten von Blöcken. Eine lässt sich mit Bomben zerstören, die zweite ist nicht zerstörbar und die letzte ist ein begehbare Block. Es existieren Monster auf der Karte, welche bestrebt sind den Bomberman durch Berührung zu besiegen. Es gibt diverse „Power-Ups“ die zufällig und für eine begrenzte Zeit erscheinen, wie stärkere Bomben zum Beispiel. Die Explosion einer Bombe erfolgt zeitversetzt nach der Platzierung und beschränkt sich immer auf eine kreuzartige Form in ihrer Ausführung mit einer festen Breite und Höhe. Sofern ein unbegehbare Block den Weg der Explosion kreuzt, wird das Explosionskreuz hierdurch an der Stelle des Blockes gestoppt. Das Spiel endet, wenn nur noch ein Bomberman übrigbleibt oder keiner mehr aktiv am Spiel teilnimmt.

## 1.2. Zweck des Dokuments

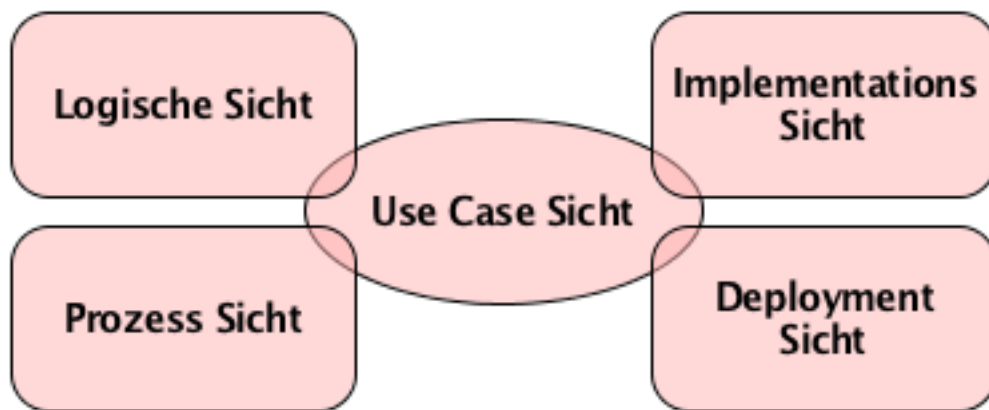
Dieses „Software Architecture Document“, im Weiteren kurz SAD genannt, lehnt sich an die RUP- Struktur an. Das Ziel dieses Dokumentes ist es, einen umfassenden Überblick über die Architektur des Projekts zu vermitteln, wobei unterschiedliche Aspekte mittels verschiedenen architektonischen Sichten („4+1 Architecture View Model“) aufgezeigt werden. Darüber hinaus werden im SAD die im Laufe des Projekts getroffenen die Architektur beeinflussenden Entscheide festgehalten und begründet.

Das Dokument basiert bezüglich Struktur und Form im Wesentlichen auf der Vorlage des „Rational Unified Process“. Es richtet sich an die RUP-Rollen „Designer“, „Software-Architect“, „Implementer“, „Integrator“ und den Betrieb des Projekts und soll einen Überblick über die einzelnen Architektur-Aspekte des Projekts geben.

## 2. Architektonische Darstellung

Die Architektur eines Systems lässt sich nicht in einem einzigen Diagramm darstellen. Man betrachtet sie von mehreren Seiten, um die verschiedenen Aspekte zu illustrieren und um die Komplexität reduzieren zu können. Man spricht dabei von Architektur-Views oder Architektur-Sichten. Eine Architektur-Sicht ist eine vereinfachte Beschreibung - eine Abstraktion - eines Systems, welche die Architektur aus einem bestimmten Blickwinkel betrachtet, nur die für sie relevanten und wichtigen Details darstellt und den ganzen Rest ausblendet. Die einzelnen Sichten müssen sorgfältig aufeinander abgestimmt sein, damit sie zusammen die gesamte Architektur repräsentieren und zueinander widerspruchsfrei sind.

Als Grundlage für die Architektur-Sichten wird hier das 4+1 Modell von RUP/UML verwendet, welches die zentrale Bedeutung der Use-Case-Sicht zum Ausdruck bringt.



Zusätzlich zum 4+1 Modell wird in diesem Dokument eine Data-View und Security- View aufgeführt. Die einzelnen Sichten bestehen aus folgenden Inhalten:

- Use Case-Sicht: Beinhaltet die wesentlichsten funktionalen Anforderungen in Form von geschäftsrelevanten Anwendungsfällen, d.h. explizit aus Sicht der künftigen Anwender, welche eine konkrete Auswirkung auf die Architektur haben.
- Logische Sicht: Beinhaltet die wichtigsten Abstraktionen der Applikation und deren Organisation in Subsysteme und Schichten.
- Prozess-Sicht: Zeigt die Zerlegung der Applikation in aktive Einheiten wie Prozesse und Threads.
- Implementations-Sicht Zeigt die Zuteilung von logischen Subsystemen zu physischen Komponenten, ihre Abhängigkeiten und ihre hierarchische Organisation.
- Deployment-Sicht: Zeigt, auf welchen physischen Geräten (Knoten) die verschiedenen Komponenten installiert werden. Gibt zusätzlich Auskunft über die Natur der Knoten, sowie über deren Kommunikationsverbindungen.
- Daten-Sicht: Zeigt, wie die Datenhaltung organisiert ist

## 3. Ziele und Rahmenbedingungen der Architektur

### 3.1. Client

Der Client dient als Eingabe- und Ausgabemöglichkeit eines Mehrspieler Benutzers, welcher aktiv am Spielgeschehen teilnimmt. Er sendet diverse Informationen, wie z.B. Eingaben, dem Server. Somit hat der Client nichts mit der reinen Spielmechanik zu tun und dient des weiteren als Ausgabeanzeige, damit der Benutzer z.B. sehen kann wo sich seine Spielfigur im Augenblick befindet.

### 3.2. Server

Der Server empfängt die Eingaben der Clients und verwertet diese mit der Spiellogik. Und gibt dann dessen Ergebnisse zurück an die Clients.

### 3.3. Persistenz

Die Spielerrelevanten Informationen wie, Niederlagen, Siege und Höchstpunktzahl werden in einer kleinen SQLite Datenbank festgehalten und verwaltet. Dieses geschieht auf einem externen Server über dem per „REST“ Kommuniziert wird.

### 3.4. Client-Server-Kommunikation

Die Kommunikation zwischen Server und Clients basiert auf Bluetooth und Wireless LAN APIs.

### 3.5. Bibliotheken/Frameworks

Bibliothek	Version	Bemerkung	URL
C#	2.6.5	Implementierungssprache	

### 3.6. Entwicklungstools

Tool	Version	Bemerkung
Unity	5.4.1	Entwicklungsumgebung
MS Office	Office365	Dokumentation
Git	2.11	Versionsverwaltung

## 4. Use-Case-Sicht

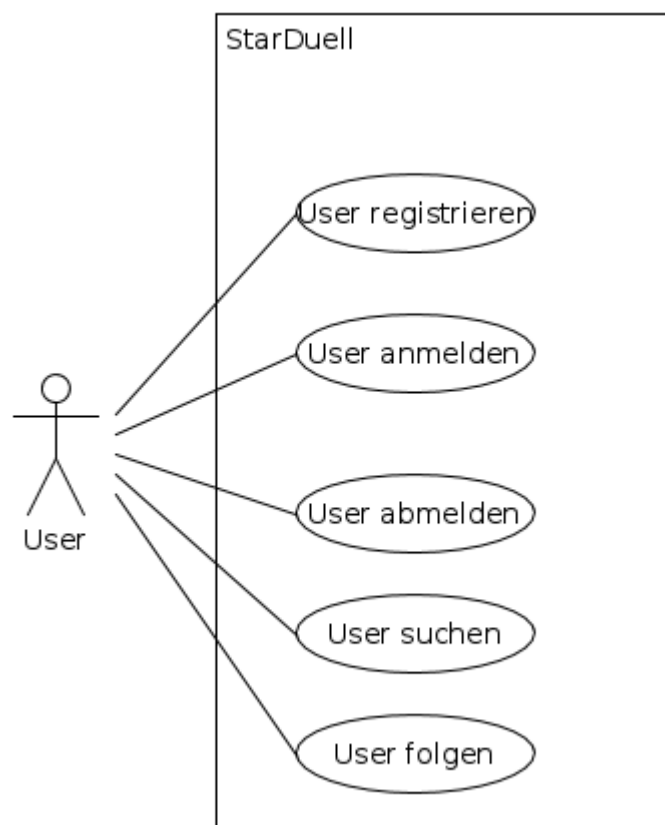
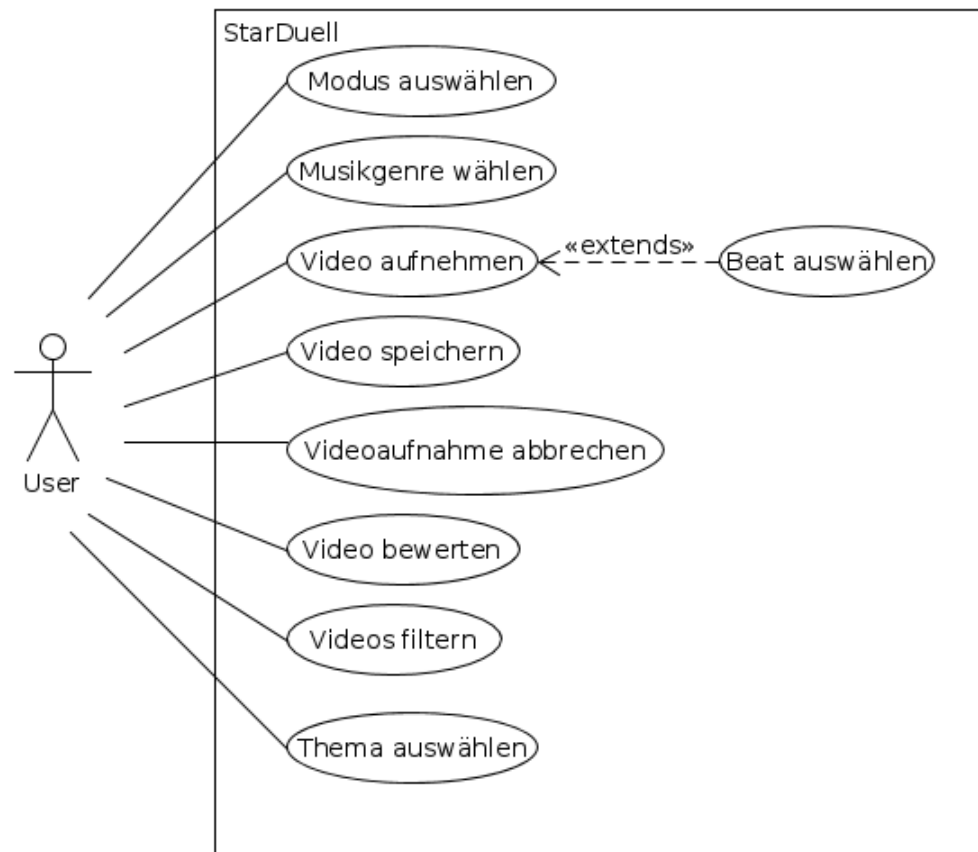
### 4.1. User Stories

Name	In meiner Rolle als	möchte ich	Sodass	Akzeptanzkriterium
User anmelden	Benutzer	mich anmelden	mir Zugriff auf die App gewährt wird	Startmenü öffnet sich
User registrieren	Benutzer	mich registrieren	ein Benutzerkonto angelegt wird	Startmenü öffnet sich
User abmelden	Benutzer	mich abmelden	meine aktuelle Sitzung geschlossen wird	Anmeldebildschirm wird angezeigt
User suchen	Benutzer	User suchen können	Ich ihm folgen kann	Der gewünschte User wird gesucht und gegebenenfalls angezeigt



User folgen	Benutzer	einem User folgen	Ich seine Videos ansehen und bewerten kann	Der Benutzer steht auf meiner Folgen-Liste
Modus auswählen	Benutzer	nach dem Start der Anwendung einen Modus der Anwendung auswählen	sich dessen Unterkategorie öffnet	die Seite der Unterkategorie hat sich geöffnet und deren Inhalte sind geladen
Musikgenre auswählen	Benutzer	eines der verfügbaren Musik-Genres auswählen	das nächste Eingabefeld angezeigt werden kann	das Auswahlfeld verschwindet und das nächste erscheint
Beat auswählen	Benutzer	eine der verfügbaren Hintergrundmelodien auswählen	die Vorbereitung der Aufnahme abgeschlossen ist	das Auswahlfeld verschwindet und die Aufnahme kann gestartet werden
Video aufnehmen	Benutzer	die Video-Aufnahme mit den ausgewählten Parametern starten	ich aufgenommen werde	Aufnahme startet mit eingestelltem Beat
Video-Aufnahme abbrechen	Benutzer	die laufende Aufnahme abbrechen	ich zum vorherigen Untermenü zurückkehre und die Aufnahme erneut starten kann	Untermenü wird angezeigt und die zuvor eingestellten Parameter bleiben erhalten
Video speichern	Benutzer	die getätigte Aufnahme speichern	diese anderen Benutzern zur Verfügung steht	Aufnahme wurde gespeichert
Video bewerten	Benutzer	ein zuvor angesehenes Video bewerten	meine Bewertung in die auf dem Server gespeicherte einfließt	Bewertung wurde übertragen
Videos filtern	Benutzer	die angezeigten Videos in einer Liste nach Parametern filtern	die angezeigten verfügbaren Videos gefiltert und nach einer neuen Reihenfolge sortiert werden können	Videos werden nach Filter angezeigt

## 4.2. Übersicht



### 4.3. Beschreibung

<b>Name</b>	User registrieren
<b>ID</b>	UC01
<b>Beschreibung</b>	User kann sich mit seiner E-Mail-Adresse und einem selbst definierten Passwort registrieren
<b>Aktoren</b>	User
<b>Auslöser</b>	Auf den „sign up“- Button tippen
<b>Verknüpfung</b>	
<b>Vorbedingung</b>	App geöffnet, E-Mail noch nicht verwendet
<b>Nachbedingung</b>	User ist im System registriert
<b>Typischer Ablauf</b>	<ol style="list-style-type: none"> <li>1. Auf „sign up“- Button getippt <ol style="list-style-type: none"> <li>1.1 E-Mail eingeben</li> <li>1.2 Passwort eingeben</li> </ol> </li> <li>2. Auf „sign up now“- Button getippt</li> </ol>
<b>Alternative Abläufe</b>	<p>2a – E-Mail bereits vergeben</p> <p>2a1Fehlermeldung: „Die E-Mail-Adresse ist bereits vergeben!“</p> <p>2b – Client kann Server nicht erreichen</p> <p>2b1Fehlermeldung: „Server kann momentan nicht erreicht werden!“</p>
<b>GUI Mockup</b>	Login, Signup

<b>Name</b>	User anmelden
<b>ID</b>	UC02
<b>Beschreibung</b>	User meldet sich mit vorhandenen Daten an
<b>Aktoren</b>	User
<b>Auslöser</b>	Auf „sign in“- Button getippt
<b>Verknüpfung</b>	
<b>Vorbedingung</b>	App gestartet, Nutzerdaten im System vorhanden und gültig
<b>Nachbedingung</b>	User ist im System eingeloggt
<b>Typischer Ablauf</b>	<ol style="list-style-type: none"> <li>1. Auf „sign in“- Button getippt <ol style="list-style-type: none"> <li>1.1 E-Mail-Adresse eingeben</li> <li>1.2 Passwort eingeben</li> </ol> </li> <li>2. Auf „sign in now“- Button getippt</li> </ol>
<b>Alternative Abläufe</b>	2a – eingegebene Nutzerdaten ungültig

	2a1Fehlermeldung: „Die eingegebenen Nutzerdaten sind ungültig“
<b>GUI Mockup</b>	Anmelden, Startseite

<b>Name</b>	User abmelden
<b>ID</b>	UC03
<b>Beschreibung</b>	User wird aus dem System ausgeloggt
<b>Aktoren</b>	User
<b>Auslöser</b>	User tippt auf „logout“-Button/Label
<b>Verknüpfung</b>	
<b>Vorbedingung</b>	User ist eingeloggt
<b>Nachbedingung</b>	User befindet sich auf der Startseite der App
<b>Typischer Ablauf</b>	User tippt auf „logout“-Button/Label
<b>Alternative Abläufe</b>	-
<b>GUI Mockup</b>	Archiv, Options, Modusansicht, Profilansicht

<b>Name</b>	User suchen
<b>ID</b>	UC04
<b>Beschreibung</b>	User sucht nach anderen Usern
<b>Aktoren</b>	User
<b>Auslöser</b>	Im „Suchen“- Textfeld wird eine Eingabe getätigt
<b>Verknüpfung</b>	-
<b>Vorbedingung</b>	User ist angemeldet und auf der Startseite
<b>Nachbedingung</b>	User sieht das Ergebnis seiner Suche
<b>Typischer Ablauf</b>	1. Namen ins Suchfeld eingeben
<b>Alternative Abläufe</b>	-
<b>GUI Mockup</b>	Modusansicht

<b>Name</b>	User folgen
<b>ID</b>	UC05
<b>Beschreibung</b>	User kann anderen Usern folgen
<b>Aktoren</b>	User

<b>Auslöser</b>	User tippt auf „folgen“-Button
<b>Verknüpfung</b>	-
<b>Vorbedingung</b>	User ist eingeloggt und befindet sich auf dem Profil eines anderen Users
<b>Nachbedingung</b>	Videos des gefolgten Users werden durch den Filter „Follower“ angezeigt
<b>Typischer Ablauf</b>	1. Auf „folgen“-Button tippen
<b>Alternative Abläufe</b>	1a – User wird bereits gefolgt
<b>GUI Mockup</b>	Profilansicht

<b>Name</b>	Modus auswählen
<b>ID</b>	UC06
<b>Beschreibung</b>	User wählt aus einer Liste den gewünschten Modus
<b>Aktoren</b>	User
<b>Auslöser</b>	User tippt auf die Select Box „Modus wählen“ und wählt Element aus
<b>Verknüpfung</b>	-
<b>Vorbedingung</b>	User befindet sich auf der Startseite
<b>Nachbedingung</b>	User bekommt die Seite des von ihm gewählten Modus angezeigt
<b>Typischer Ablauf</b>	1. Im Startmenü einen Modus auswählen
<b>Alternative Abläufe</b>	-
<b>GUI Mockup</b>	Modusansicht

<b>Name</b>	Musikgenre wählen
<b>ID</b>	UC07
<b>Beschreibung</b>	User wählt aus einer Liste sein gewünschtes Musikgenre
<b>Aktoren</b>	User
<b>Auslöser</b>	User tippt auf die Select Box „Musikgenre wählen“ und wählt Element aus
<b>Verknüpfung</b>	-
<b>Vorbedingung</b>	User befindet sich auf der Vorbereitungsseite des Modus Aufnehmen oder Duell
<b>Nachbedingung</b>	User bekommt eine Auswahl an Instrumentals gemäß seines gewählten Musikgenres
<b>Typischer Ablauf</b>	1. In Vorbereitung auf „Musikgenre wählen“-tippen
<b>Alternative Abläufe</b>	-
<b>GUI Mockup</b>	Options

<b>Name</b>	Video aufnehmen
<b>ID</b>	UC08
<b>Beschreibung</b>	User startet eine Videoaufnahme
<b>Aktoren</b>	User
<b>Auslöser</b>	User tippt in der Vorbereitungsseite auf den „Start“-Button
<b>Verknüpfung</b>	-
<b>Vorbedingung</b>	In der Vorbereitungsseite wurde Genre, Zeitlimit, Thema und Instrumental gewählt
<b>Nachbedingung</b>	Videoaufnahme wird gemäß den Einstellungen gestartet
<b>Typischer Ablauf</b>	<ol style="list-style-type: none"> <li>1. Einstellungen wählen <ol style="list-style-type: none"> <li>1.2 Musikgenre auswählen</li> <li>1.3 Thema auswählen</li> <li>1.4 Instrumental auswählen</li> <li>1.5 Zeitlimit auswählen</li> </ol> </li> <li>2. Auf „Start“- Button tippen</li> </ol>
<b>Alternative Abläufe</b>	<p>2a – Keine Auswahl für Einstellungen erfolgt</p> <p>2a1Fehlermeldung: „Bitte treffen Sie für die jeweiligen Optionen eine Wahl“</p>
<b>GUI Mockup</b>	Options, aufnehmen

<b>Name</b>	Beat auswählen
<b>ID</b>	UC09
<b>Beschreibung</b>	User wählt gemäß, seines gewählten Musikgenres, einen Beat aus
<b>Aktoren</b>	User
<b>Auslöser</b>	User tippt auf die Select Box „Beat wählen“ und wählt ein Element aus
<b>Verknüpfung</b>	UC06
<b>Vorbedingung</b>	Der User hat ein Musikgenre gewählt
<b>Nachbedingung</b>	Der User kann den gewählten Beat hören
<b>Typischer Ablauf</b>	<ol style="list-style-type: none"> <li>2. User wählt ein Musikgenre</li> <li>3. User wählt einen Beat</li> <li>4. Gewählter Beat wird zum Anhören zur Verfügung gestellt</li> </ol>
<b>Alternative Abläufe</b>	<p>4a – Es wurde keine Auswahl für das Musikgenre getätigt</p> <p>2a1Fehlermeldung: „Bitte wählen Sie zuerst ein Musikgenre“</p>
<b>GUI Mockup</b>	Options

<b>Name</b>	Video speichern
-------------	-----------------

<b>ID</b>	UC10
<b>Beschreibung</b>	Das aufgenommene Video wird in der Datenbank gespeichert und verfügbar für die Bewertung gemacht
<b>Aktoren</b>	User
<b>Auslöser</b>	Zeitlimit einer Videoaufnahme ist abgelaufen und User bestätigt die Speicherung des Videos mit dem „speichern“-Button
<b>Verknüpfung</b>	-
<b>Vorbedingung</b>	Videoaufnahme abgeschlossen und auf „speichern“-Button tippen
<b>Nachbedingung</b>	Video wird in der Datenbank gespeichert und für alle User sichtbar
<b>Typischer Ablauf</b>	1. Videoaufnahme starten 2. Auf „speichern“-Button tippen
<b>Alternative Abläufe</b>	2a - Videoaufnahme wurde abgebrochen
<b>GUI Mockup</b>	Aufnahme

<b>Name</b>	Videoaufnahme abbrechen
<b>ID</b>	UC11
<b>Beschreibung</b>	User bricht die Videoaufnahme ab
<b>Aktoren</b>	User
<b>Auslöser</b>	User tippt auf Button
<b>Verknüpfung</b>	-
<b>Vorbedingung</b>	User hat eine Videoaufnahme gestartet
<b>Nachbedingung</b>	Die Aufzeichnung wird verworfen
<b>Typischer Ablauf</b>	Videoaufnahme gestartet Auf Button zum Abbrechen tippen
<b>Alternative Abläufe</b>	2a - Zeitlimit bereits abgelaufen
<b>GUI Mockup</b>	Aufnahme

<b>Name</b>	Video bewerten
<b>ID</b>	UC12
<b>Beschreibung</b>	Der User bekommt im Modus „Bewerten“ per Zufall ein Video, das er bewerten kann
<b>Aktoren</b>	User
<b>Auslöser</b>	Auf der Startseite auf den „Bewerten“-Button tippen
<b>Verknüpfung</b>	-

<b>Vorbedingung</b>	- User ist angemeldet - noch nicht vom User bewertete Videos in der Datenbank vorhanden
<b>Nachbedingung</b>	Bewertung des bewerteten Videos wird aktualisiert
<b>Typischer Ablauf</b>	„Bewerten“- Modus wählen Video abspielen Video bewerten
<b>Alternative Abläufe</b>	2a – Keine Videos zum bewerten vorhanden
<b>GUI Mockup</b>	Bewertung

<b>Name</b>	Videos filtern
<b>ID</b>	UC13
<b>Beschreibung</b>	Der User kann die Videos im Archiv mittels bestimmter Filter anordnen lassen
<b>Aktoren</b>	User
<b>Auslöser</b>	Der User hat in der „Filter“ Select Box ein Element ausgewählt
<b>Verknüpfung</b>	-
<b>Vorbedingung</b>	„Archiv“-Modus aus der Startseite wurde gewählt
<b>Nachbedingung</b>	Liste der Videos werden nach dem gewählten Filter aufgelistet
<b>Typischer Ablauf</b>	Archivmodus wählen Filter wählen
<b>Alternative Abläufe</b>	-
<b>GUI Mockup</b>	Archiv

<b>Name</b>	Thema auswählen
<b>ID</b>	UC14
<b>Beschreibung</b>	User wählt ein Thema für seine Videoaufzeichnung aus
<b>Aktoren</b>	User
<b>Auslöser</b>	User tippt auf die „Thema wählen“-Select Box
<b>Verknüpfung</b>	-
<b>Vorbedingung</b>	User befindet sich auf der Vorbereitungsseite einer Videoaufzeichnung
<b>Nachbedingung</b>	Video wird mit Thema markiert
<b>Typischer Ablauf</b>	<ol style="list-style-type: none"> <li>1. Auf „Thema wählen“-Select Box tippen</li> <li>1.2. Thema wählen</li> <li>1.3. „Eigenes Thema“ wählen <ol style="list-style-type: none"> <li>1.3.1. Thema eingeben</li> </ol> </li> </ol>



	1.3.2. Auf „bestätigen“- tippen
<b>Alternative Abläufe</b>	-
<b>GUI Mockup</b>	Options

## 4.4. Aktivitätsdiagramme

## 4.5. GUI Mockups

### 4.5.1. Startseite



## 4.5.2. Registrieren



### 4.5.3. Anmelden



#### 4.5.4. Modusansicht



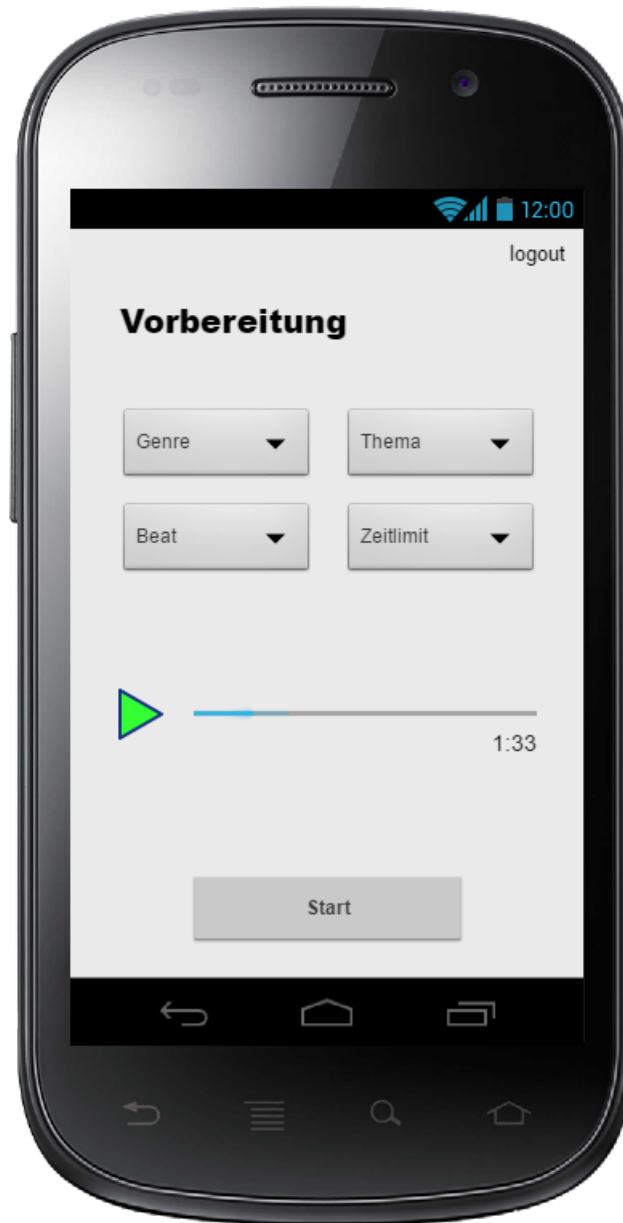
#### 4.5.5. Archiv



#### 4.5.6. Profilansicht



#### 4.5.7. Options

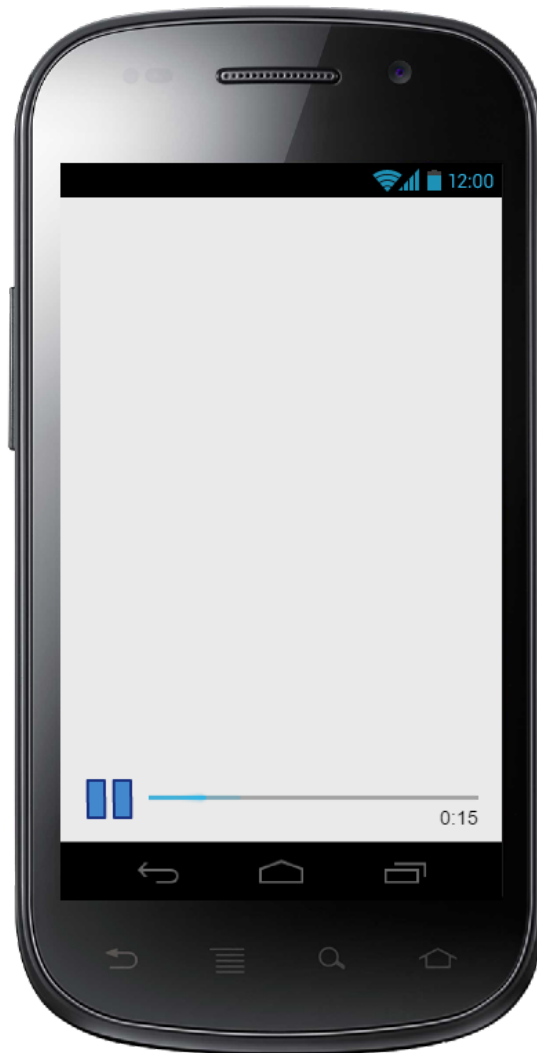




#### 4.5.8. Aufnahme



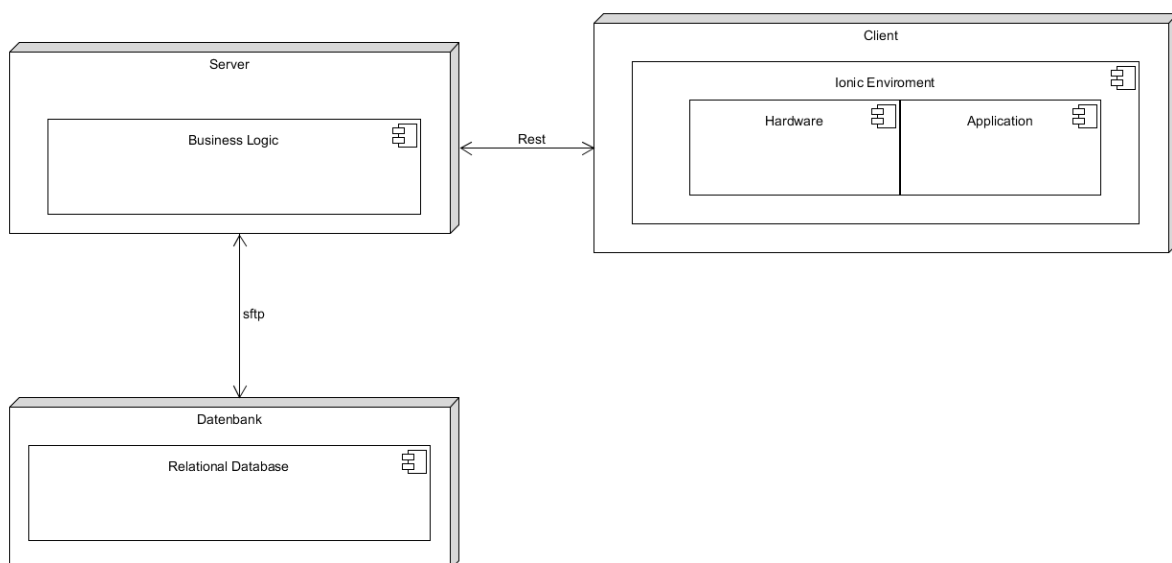
#### 4.5.9. Bewertung



## 4.6. GUI Navigationen

## 5. Logische-Sicht

### 5.1. Architekturmodell Hardware raus



## 6. Prozess-Sicht

### 6.1. Sequenzdiagramme

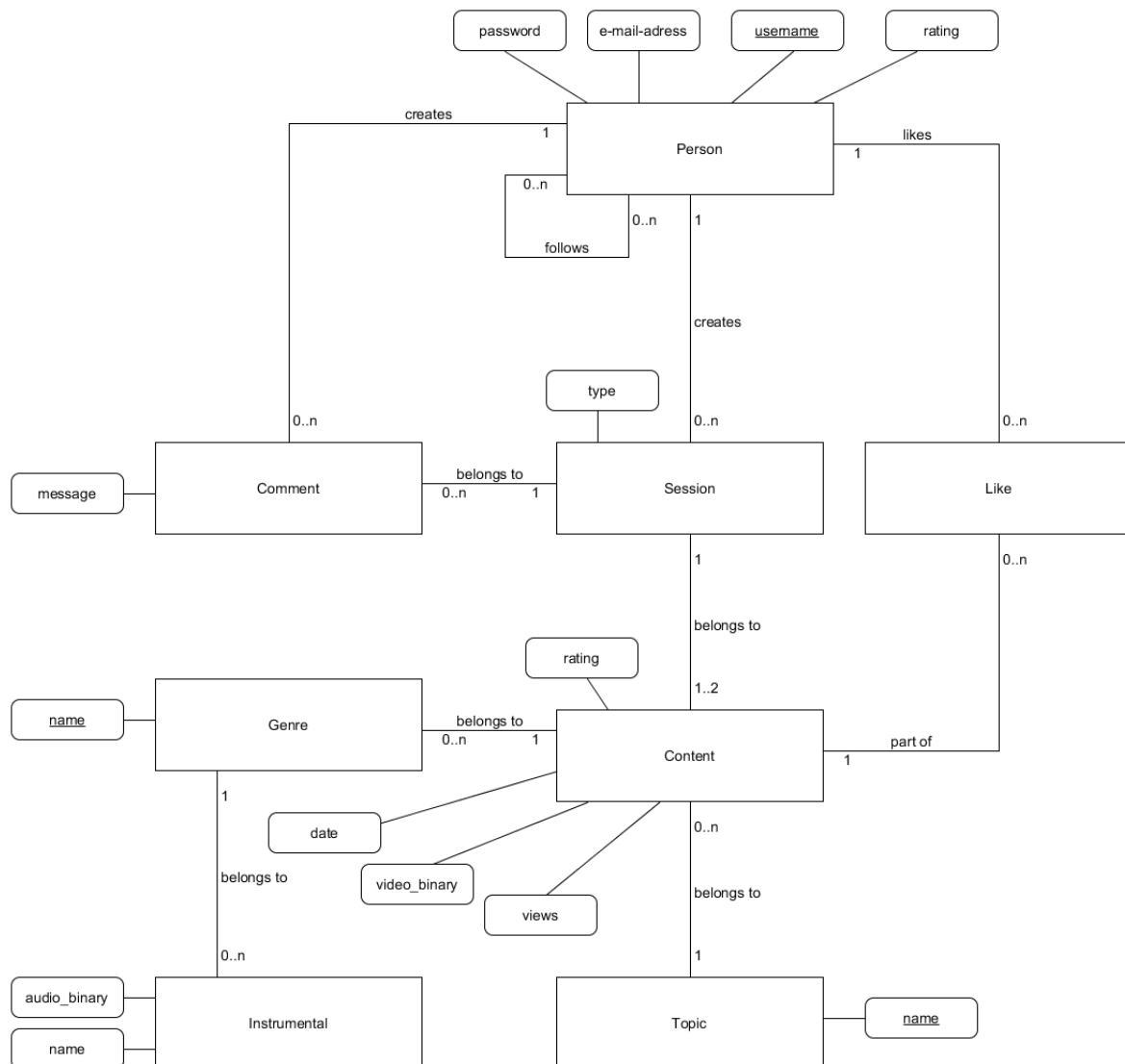
## 7. Deployment-Sicht

### 7.1. Datenbank

Für das „Star-Duell“ Projekt wird ein relationales Datenbanksystem verwendet. Ein MySQL Server, welcher auf Linux-Ubuntu 16.04 LTS Basis läuft, verwaltet alle Applikation relevanten Daten.

Der Server steht ausschließlich in Kontakt zum Webserver und niemals direkt zum Anwender.

*Konzeptuelles Datenbankschema:*



## 7.2. Versionskontrolle

Basierend auf GitHub wird die Versionsverwaltung realisiert. Hierzu wird dort ein Repository eingerichtet.

## 8. Implementations-Sicht

### 8.1. Domänenklassendiagramm

## 9. Daten-Sicht

Innerhalb der relationalen Datenbank werden Informationen zur Kontoverwaltung und die spielerischen Errungenschaften des Anwenders gehalten.

## 10. Security-Sicht

### 10.1. Authentifizierung

Mit Hilfe eines Passwortes und einer Email-Adresse ist ein Spieler in der Lage sich zu Authentifizieren.

### 10.2. Datenbank

Das Passwort des Anwenders wird nicht im Klartext, sondern verschlüsselt (Salt) in der Datenbank gehalten.

## 11. Glossar

Begriff	Definition/Erklärung
Block	Elementarer Bestandteil der <b>Karte</b> auf der gespielt wird. Wird in verschiedene Typen unterteilt: <i>begehrbar</i> , nicht <i>begehrbar</i> in Kombination mit den Eigenschaften <i>zerstörbar</i> und <i>unzerstörbar</i> . Ein <b>Block</b> kann verschiedene <b>Texturen</b> besitzen.
Bluetooth	Eine <i>drahtlose</i> Kommunikationsschnittstelle, die dazu dient unterschiedliche Geräte kabellos zu verbinden.
Bombe	Der <b>Spieler</b> platziert eine <b>Bombe</b> um <b>Blöcke</b> /Hindernisse zu zerstören oder seine Gegner zu besiegen. Eine Bombe erzeugt einen



	Feuerschweif. Sie wird immer auf der aktuellen Position des Spielers durch Knopfdruck platziert und ist somit auch nur auf begehbaren Blöcken platzierbar.
Bot	Computer gesteuerte <b>Gegner</b> einer Partie, welche <i>aktiv</i> am Spielgeschehen teilhaben und bemüht sind alle <b>Spieler</b> durch Positionskollision zu besiegen.
Client	Teil der Server-Client Architektur. Verbindet sich mit einem Server.
Einkesseln	
Einzelspieler	Ein Spielmodus der weitere Spieler ausschließt. So ist es einem Spieler hier nur möglich gegen Bot 's anzutreten.
Feuerschweif	Entsteht durch die Explosion einer Bombe. Hat eine variable höhe und breite als Reichweite und wird mit einer Textur für alle Spieler sichtbar. Es zerstört Spieler, Bots und zerstörbare Blöcke bei Kollision mit diesen.
Gegner	Aus der Sicht eines einzelnen Spielers ist jeder Bot und jedes nicht Mitglied seins Teams sein Gegner. Voraussetzung zum Sieg ist es alle Gegner zu zerstören.
Highscore	Eine Datenbank, welche die Leistungen und Errungenschaften des Anwenders festhält.
Karte	Auf der Karte wird eine Partie ausgetragen. Sie besteht aus Blöcken und Texturen.
Loch	Wenn auf einem begehbaren Block eine spezielle „Loch Bombe“ (nur durch PowerUps erhältlich) explodiert, dann entsteht an diesem Block ein Loch und er ist von nun an unbegehrbar. Das erzeugen von Löchern kann zu Einkesseln führen.
Mehrspieler	In diesem Modus ist es dem Spieler möglich eine Partie mit Bots und anderen Spielern auszutragen. Dazu kann er als Server andere Spieler teilnehmen lassen oder als Client einem Server beitreten.
Partie	Die Partie beginnt mit der Karte und den darauf platzierten Blöcken, einem Spieler und seinen Gegnern. Sie endet wenn es ein Spieler schafft

	alle seine Gegner zu besiegen.
PowerUp	Erscheint von Zeit zu Zeit zufällig irgendwo auf einem begehbaren Block, für eine begrenzte Zeit. Kann durch den Spieler aufgenommen werden und verleiht ihm für eine begrenzte Zeit diverse Fähigkeiten/Vorteile.
Schwierigkeitsgrad	Definiert die Stärke und Anzahl der Bots.
Server	Teil der Server-Client Architektur. Ein Spieler ist es möglich im Mehrspieler ein Server zu eröffnen mit dem sich Clients verbinden können.
Spieler	Menschlicher Anwender der Applikation. Besitzt eine Figur mit der er am Spielgeschehen innerhalb einer Partie interagieren kann.
Wireless LAN (WLAN)	Eine <i>drahtlose</i> Kommunikationsschnittstelle, die dazu dient unterschiedliche Geräte kabellos zu verbinden.