Dozent: Brunsmann

Modul: Software Engineering

Se: Modulprüfung

War of the Summon

Autoren

Marcel Sander (Matrikelnummer: 1260933) - marcel\_manfred\_friedrich.sander@fh-bielefeld.de  
Clemens Maas (Matrikelnummer: 1260892) – clemens.maas@fh-bielefeld.de

Inhaltsverzeichnis

[Einführung 0](#_Toc98903128)

[Beschreibung 0](#_Toc98903129)

[Ziele 0](#_Toc98903130)

[Anforderungen 1](#_Toc98903131)

[Stakeholder 1](#_Toc98903132)

[Funktionale Anforderungen 1](#_Toc98903133)

[Use-Case Diagramm: Spieler -> Spiel 1](#_Toc98903134)

[Use-Case Diagramm: Spieler -> Website 2](#_Toc98903135)

[Nicht-funktionale Anforderungen 2](#_Toc98903136)

[Grafische Benutzerschnittstelle 3](#_Toc98903137)

[Screenshot: GUI-Mockup 3](#_Toc98903138)

[Screenshot: Startbildschirm 5](#_Toc98903139)

[Screenshot: Spielaufbau 5](#_Toc98903140)

[Screenshot: Spielphase 6](#_Toc98903141)

[Screenshot: Aktionsphase 6](#_Toc98903142)

[Anforderungen im Detail 7](#_Toc98903143)

[Technische Beschreibung 8](#_Toc98903144)

[Systemübersicht 8](#_Toc98903145)

[Schnittstellen 9](#_Toc98903146)

[Ereignisse 9](#_Toc98903147)

[Ereignisse: Client 9](#_Toc98903148)

[Ereignisse: Backend 9](#_Toc98903149)

[Abläufe 10](#_Toc98903150)

[Aktionsdiagramm: Client 10](#_Toc98903151)

[Aktionsdiagramm: Server 11](#_Toc98903152)

[Entwurf 12](#_Toc98903153)

[UML: Client 12](#_Toc98903154)

[UML: API 13](#_Toc98903155)

[UML: Game Server 14](#_Toc98903156)

[UML: SampleClient 14](#_Toc98903157)

[Fehlerbehandlung 15](#_Toc98903158)

[Validierung 15](#_Toc98903159)

[Projektorganisation 15](#_Toc98903160)

[Annahmen 15](#_Toc98903161)

[Risiken 15](#_Toc98903162)

[Verantwortlichkeit 17](#_Toc98903163)

[Grober Projektplan 17](#_Toc98903164)

Anforderungs und Entwurfsspezifikationen

# Einführung

## Beschreibung

War of the Summon ist ein zwei Spieler Online-Game, in dem man mit 12 unterschiedlichen Spielfiguren versucht die gegnerische Burg versucht zu zerstören und so zu gewinnen. Das Spiel fördert die Vorausplanungsfähigkeit und dient der Unterhaltung.

Grafisch ist das Spiel minimalistisch gehalten mit einem zwei-dimensionalem Spielbrett und Figuren.  
Eine Runde kann von 30min. bis zu über einer Stunde dauen.  
Im Spielverlauf bedient man sich Figuren mit verschiedenen Fähigkeiten und Statistiken.

Der Multiplayer selbst wird über einen separaten Server realisiert der die Kommunikation der Clients handled.  
Über eine Webseite kann man sich einen Nutzeraccount erstellen, worüber man am Ende das Spiel spielen kann. Durch das gewinnen von Matches, können sich Nutzer unteranderem auch Punkte dazu verdienen und Ränge erarbeiten, die von einer REST Api mithilfe einer Postgresql Datenbank verwaltet wird und auf der Webseite eingesehen werden kann.

## Ziele

**Anwendungsbereich**  
Freizeit vergnügen, Kompetitiver Wettbewerb mit anderen Nutzern.

**Motivation**  
Die Motivation hinter der Projektidee war die Erstellung eines anspruchsvollen taktischen 2D Spiels, welches man mit Freunden spielen kann.

**Umfang**  
Ein voll funktionales Spiel mit Multiplayer Option und Rangliste zur Vervollständigung des kompetitiven Ansatzes.

**Alleinstellungsmerkale**  
Es gibt kein Spiel mit Aufbau und Spielprinzip wie wir es umgesetzt haben. Der Spielverlauf und die auszuführenden Tätigkeiten sind einzigartig und so noch nicht vorhanden.

**Marktanforderungen**  
Das Spiel muss auf den größten Plattformen lauffähig sein (Windows, Linux, MacOS).  
Unteranderem soll die Performanceanforderung gering genug sein, um den Großteil des Marktes erreichen zu können. Essenziell für den Markt ist es ebenfalls, dass man das Spiel mit Freunden oder zufälligen Gegnern zusammenspielen kann.

**Informationen zur Zielbenutzergruppe**Unsere Zielgruppe besteht aus Personen die Interesse an strategischen Herausforderungen haben und sich auch gerne mit anderen Personen im Können dieser Strategien messen wollen. Es gibt keine besonderen Anforderungen was den Bildungsgrades angeht. Vorkenntnisse aus Spielen wie Schach oder Go sind vorteilhaft, da vergleichbare Denkmuster angesprochen werden. Spieler aus diesem Genre werden einen einfacheren Einstieg haben, aber auch Nutzer, die ganz neu in diesem Gebiet sind werden sich schnell einfinden.

**Abgrenzung**Unser Programm dient der Unterhaltung und keinem höheren Zweck.

# Anforderungen

## Stakeholder

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Funktion / Relevanz | Name | Kontakt / Verfügbarkeit | Wissen | Interessen / Ziele |
| Frontend Designer | Marcel Sander | marcel\_manfred\_friedrich.sander@fh-bielefeld.de | LibGDX | Erstellung eines interessanten und neuartigen Spielsystems |
| Backend Engineer | Clemens Maas | clemens.maas@fh-bielefeld.de | API / Sockets / Schach Rating von 1300 | Erstellung und kontinuierlichen Optimierung des Backend Systems |

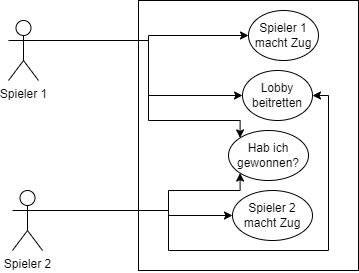
## Funktionale Anforderungen

**Funktionale Gruppen**  
Da wir nur zwei Mitglieder sind, haben wir uns in jeweils dem „Frontend“ und dem „Backend“ eingeteilt und die Aufgaben so komplett getrennt.

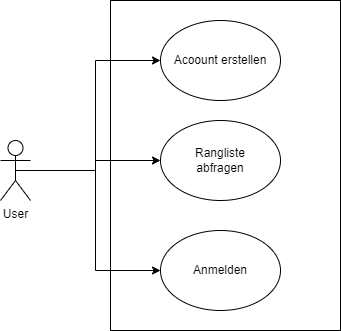
**Definition der Akteure**  
-> User | Der User ist die Person die tatsächlich das Spiel spielt.  
-> Administrator | Die Entwickler des Spiels, die den Spielverlauf überwachen können und die Systeme warten

**Use-Case Diagramme**

### Use-Case Diagramm: Spieler -> Spiel



### Use-Case Diagramm: Spieler -> Website



## Nicht-funktionale Anforderungen

**Rahmenbedingungen**Protokolle: TCP/UDP  
Hardware: Full HD+ Auflösung, Prozessor mit Threading, aktive Internetverbindung (220B/s+)

**Betriebsbedingungen**  
Bedingungen nach eigener Vorgabe:  
Plattform: Windows 7/8/10/11, Linux Kernel 4.9.306-1 oder neuer, MacOS Lion  
Programmiersprache: Java 11 (Server / API), Java 8 (Spiel)

**Mögliche Sicherheitsrisiken und Sicherheitsschwachstellen**  
Datentransfer über TCP Protokoll.  
Das Spiel selber kann über externe Software manipuliert werden und falsche Daten übermitteln.

**Qualitätsmerkmale**

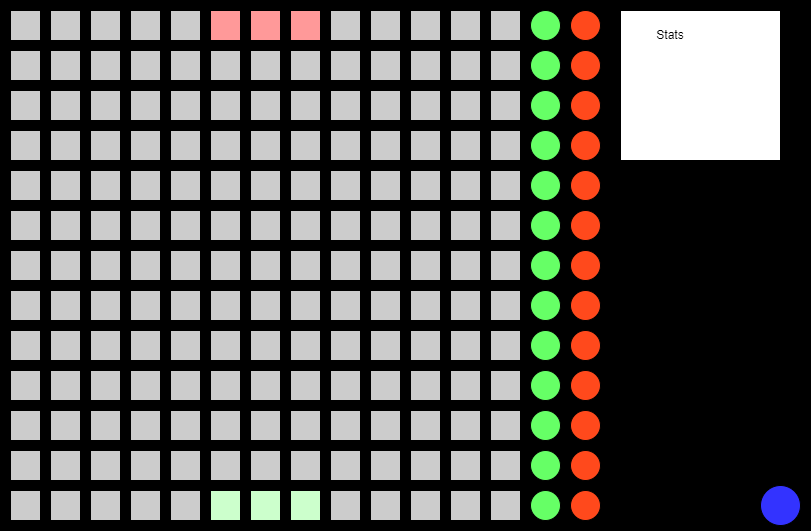
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Qualitätsmerkmale | Sehr gut | Gut | Normale | Nicht relevant |
| Zuverlässigkeit |  |  |  |  |
| Fehler |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

## Grafische Benutzerschnittstelle

Screenshot: GUI-Mockup  
A picture containing background pattern

Description automatically generated  
Shape

Description automatically generated with medium confidence  
[User Story: Grundaufbau](#_top)

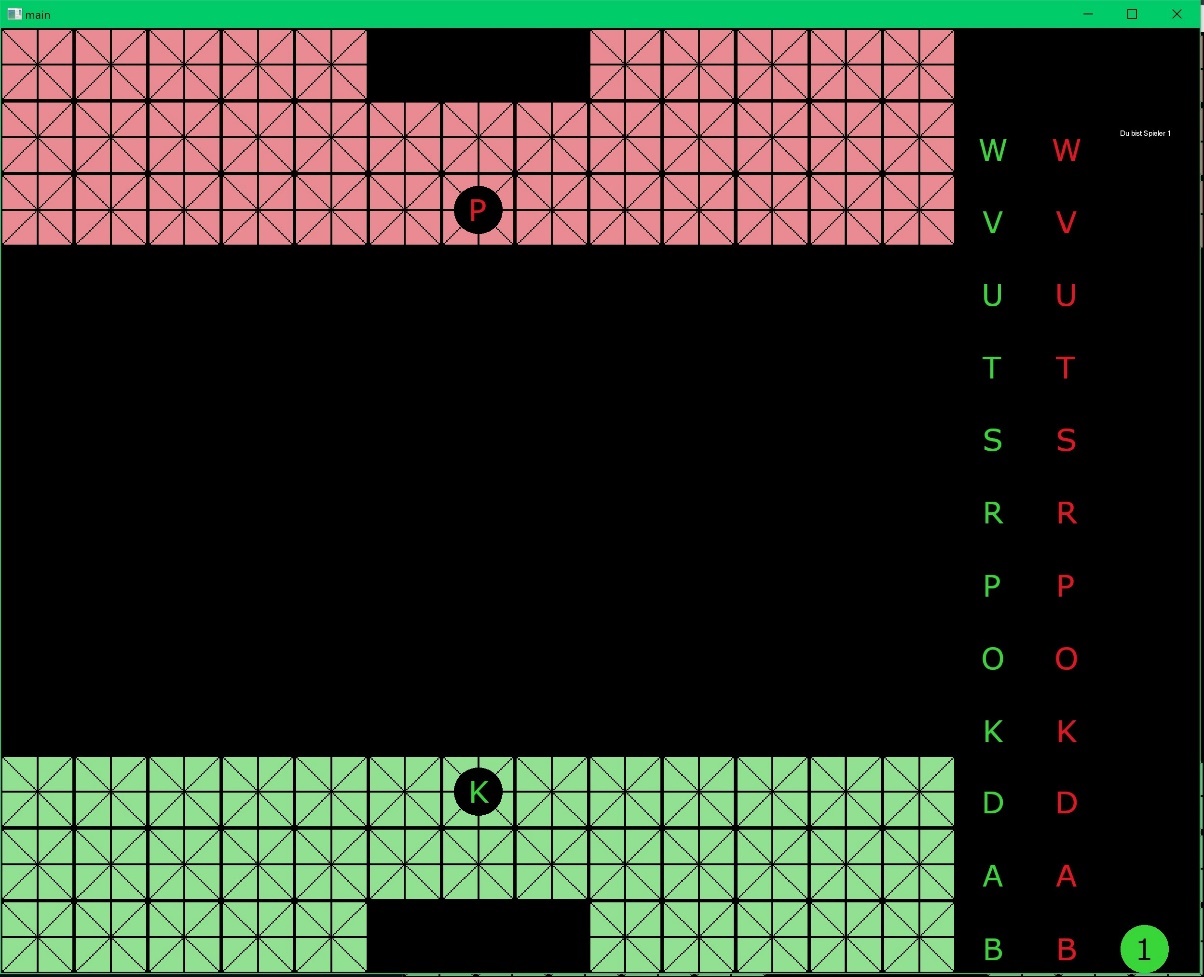


### Screenshot: Startbildschirm

Graphical user interface, application

Description automatically generated

### Screenshot: Spielaufbau

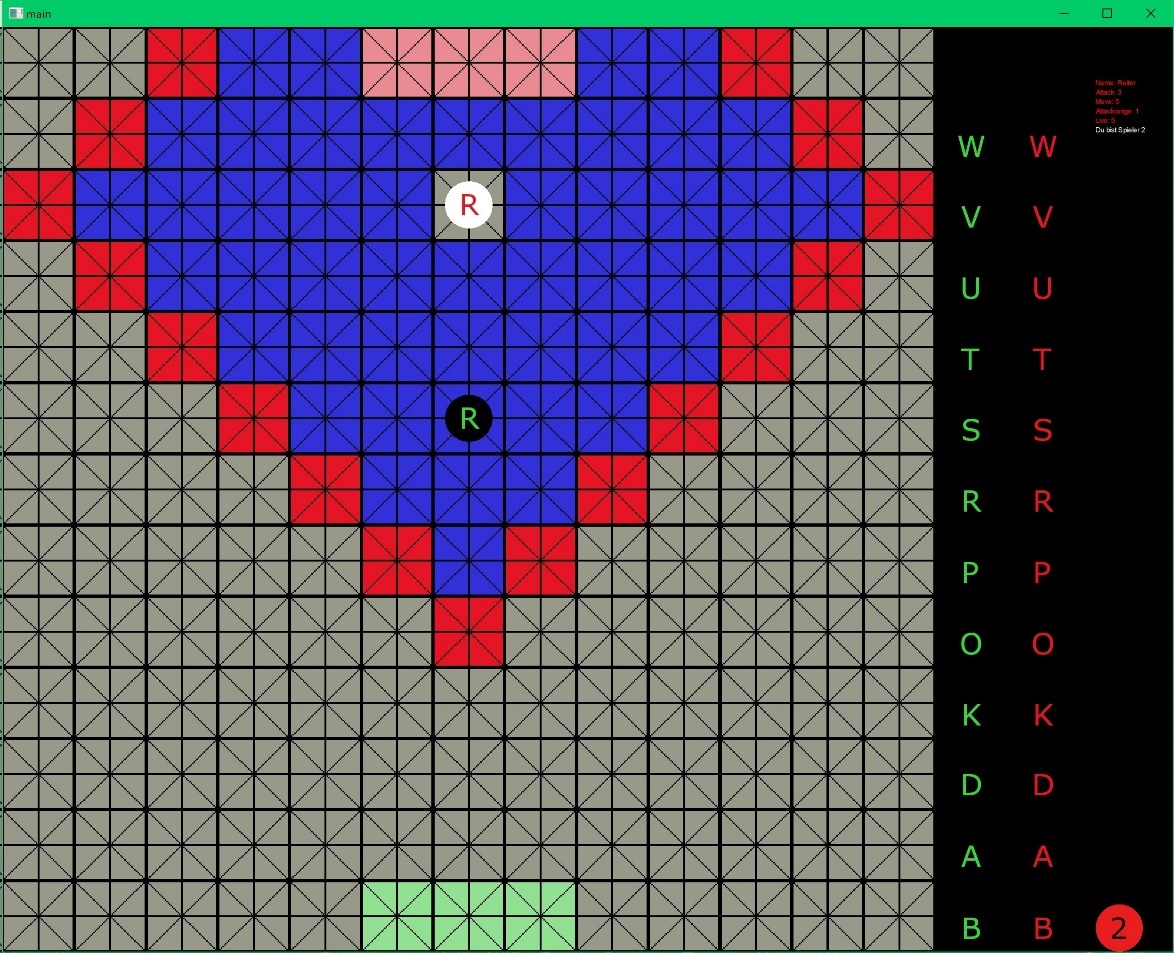


### Screenshot: Spielphase

Chart, scatter chart

Description automatically generated

### Screenshot: Aktionsphase

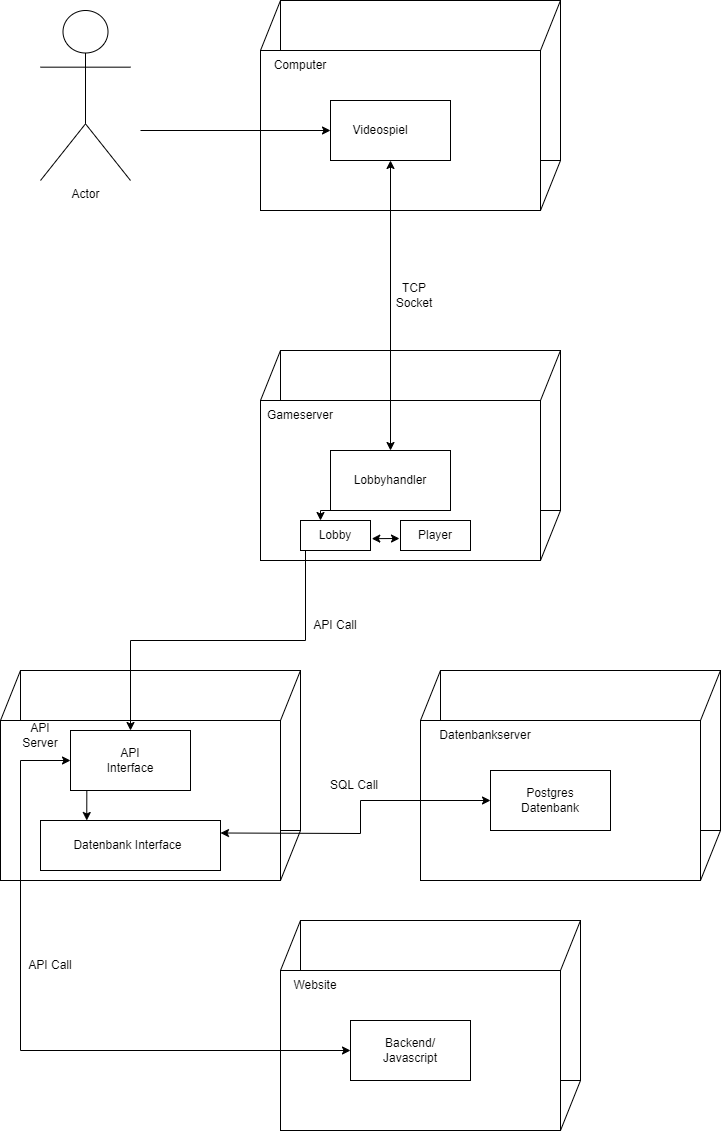


## Anforderungen im Detail

|  |  |  |  |
| --- | --- | --- | --- |
| Als | Möchte ich | So dass | Akzeptanz |
| Spieler | Figuren per Klick auswählen können | Ich Aktionen mit ihr ausführen kann | Figur ändert die Farbe zur Validierung |
| Spieler | Ausgewählte Figuren bewegen können | Ich meine Figuren umpositionieren kann | Figur ändert die Position |
| Spieler | Figuren angreifen können | Die gegnerische Figur leben verliert | Gegnerische Figur verliert Leben |
| Spieler | die Runde beenden können | Wir als Spieler abwechselnd an der Reihe sind | Rundenwechsel nach Zug |
| Spieler | ein eindeutiges Spielergebnis erzeugen können | Ich entweder verlieren oder gewinnen kann | Sieg / Niederlage |
| Spieler | gegen andere Spieler spielen | Ich mich gegen Menschen messen kann | Multiplayer |
| Spieler | ein Erfolgserlebnis nach dem Gewinn einer Runde haben | Ich meinen Account durch Ränge oder XP aufwerten kann | Rangliste |

# Technische Beschreibung

## Systemübersicht



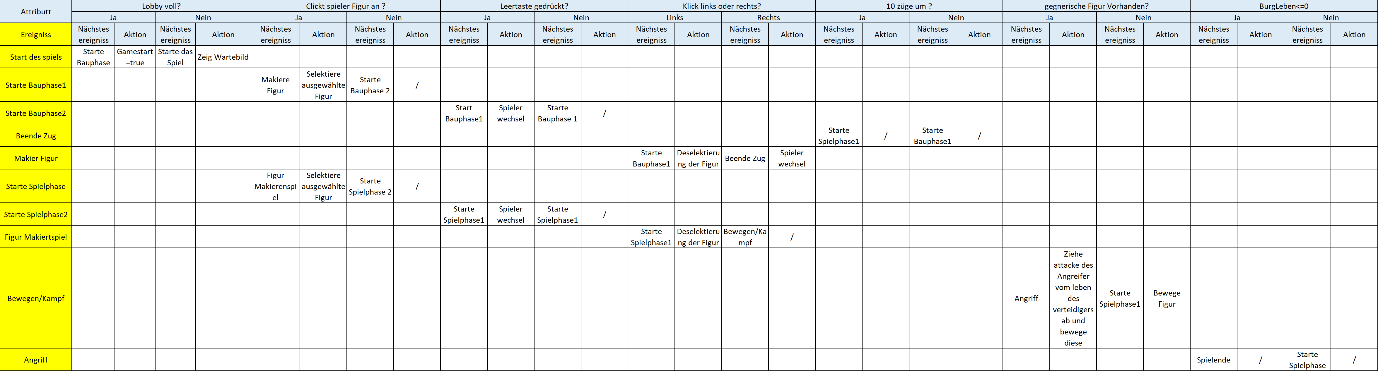
## Schnittstellen

**Externe Schnittstellen**

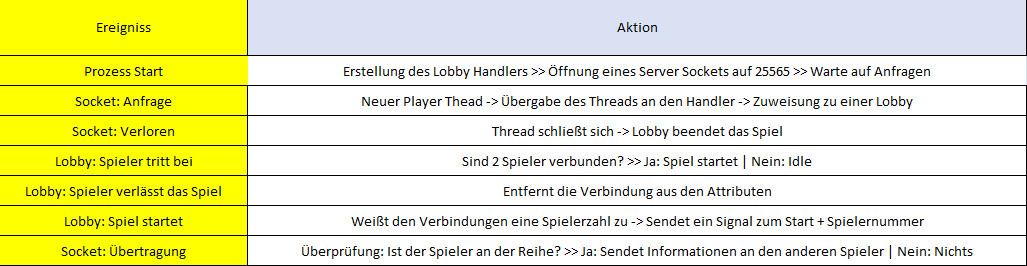
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pfad | Attribute | Response | Key benötigt? | Cross-Origin |
| api/user/newUser | Key, username, password | 200 / 400 | Ja | \* |
| api/user/deleteUser | Key, id | 200 / 400 | Ja | \* |
| api/user/changeEmail | Key, email | 200 / 400 | Ja | \* |
| api/user/changePassword | Key, id, password | 200 / 400 | Ja | \* |
| api/user/login | Key, username, password | 200 / 400 | Ja | \* |
| api/user/allUsers | Key | 200 / 400 | Ja | \* |
| api/ranking/newRanking | Key, id | 200 / 400 | Ja | \* |
| api/ranking/getLevel | Id | 200 / 400 | Nein | \* |
| api/ranking/getRank | Id | 200 / 400 | Nein | \* |
| api/ranking/getTopTen |  | 200 / 400 | Nein | \* |

## Ereignisse

### Ereignisse: Client

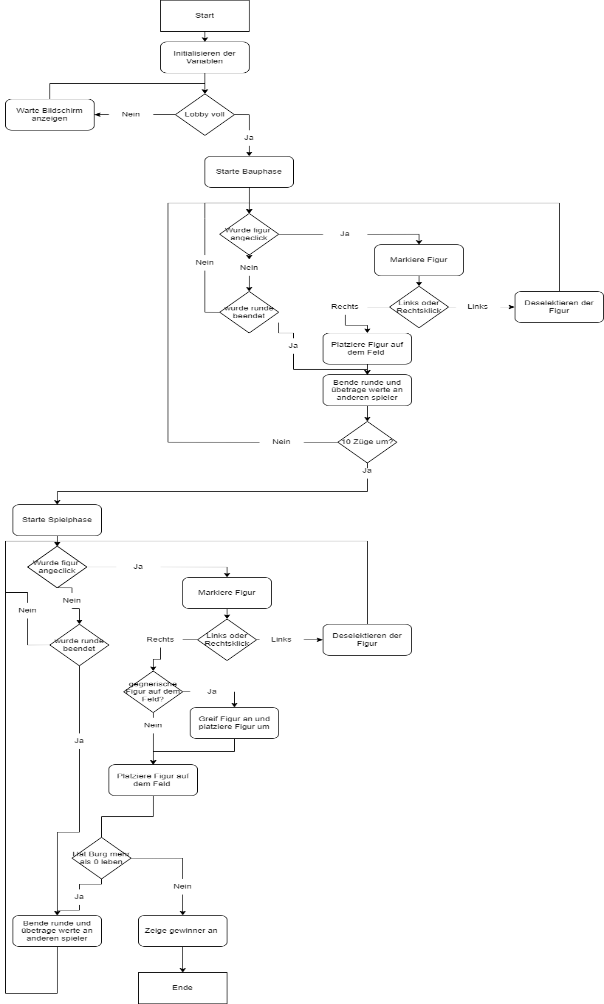


### Ereignisse: Backend

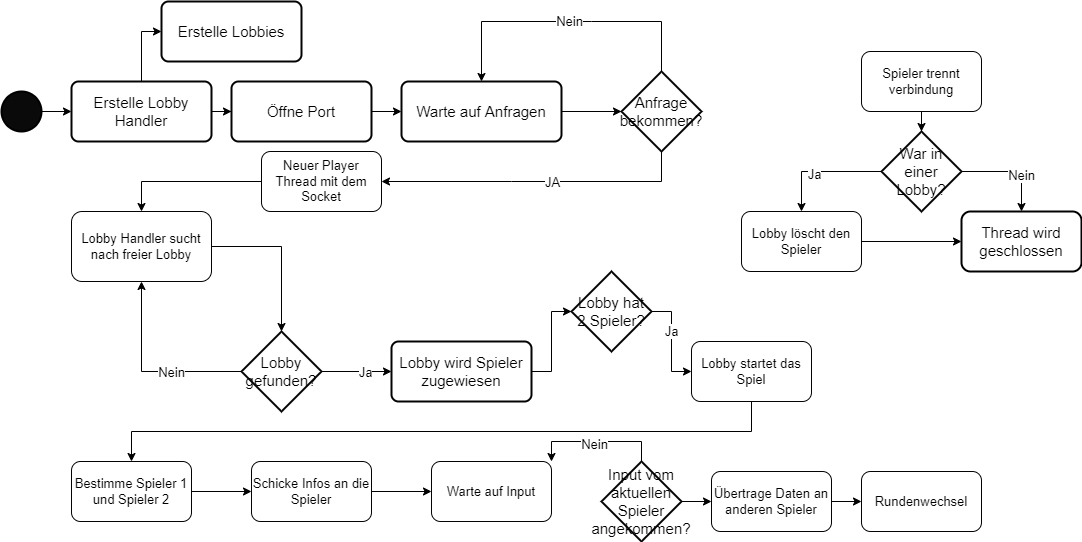


## Abläufe

### Aktionsdiagramm: Client



### Aktionsdiagramm: Server

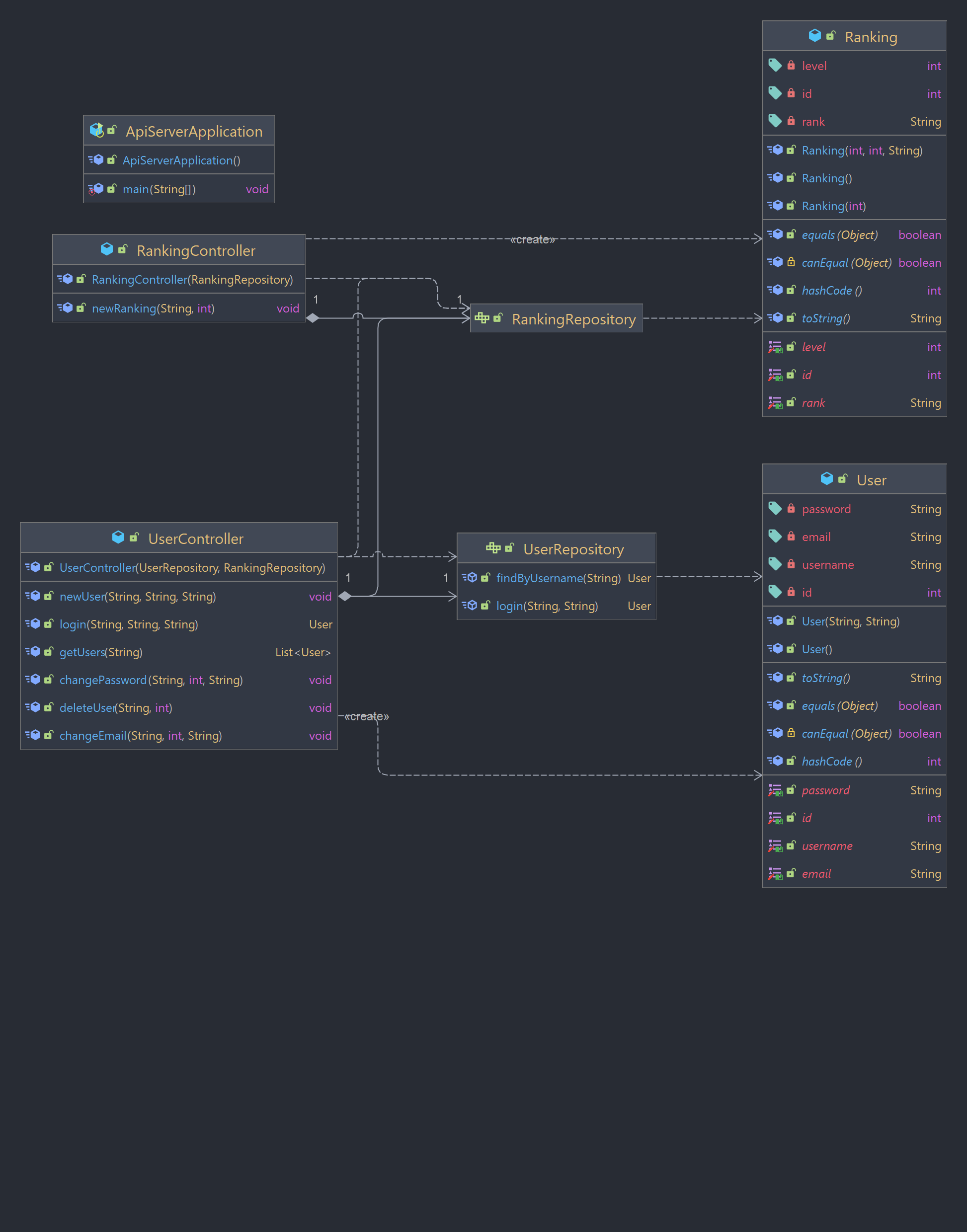


## Entwurf

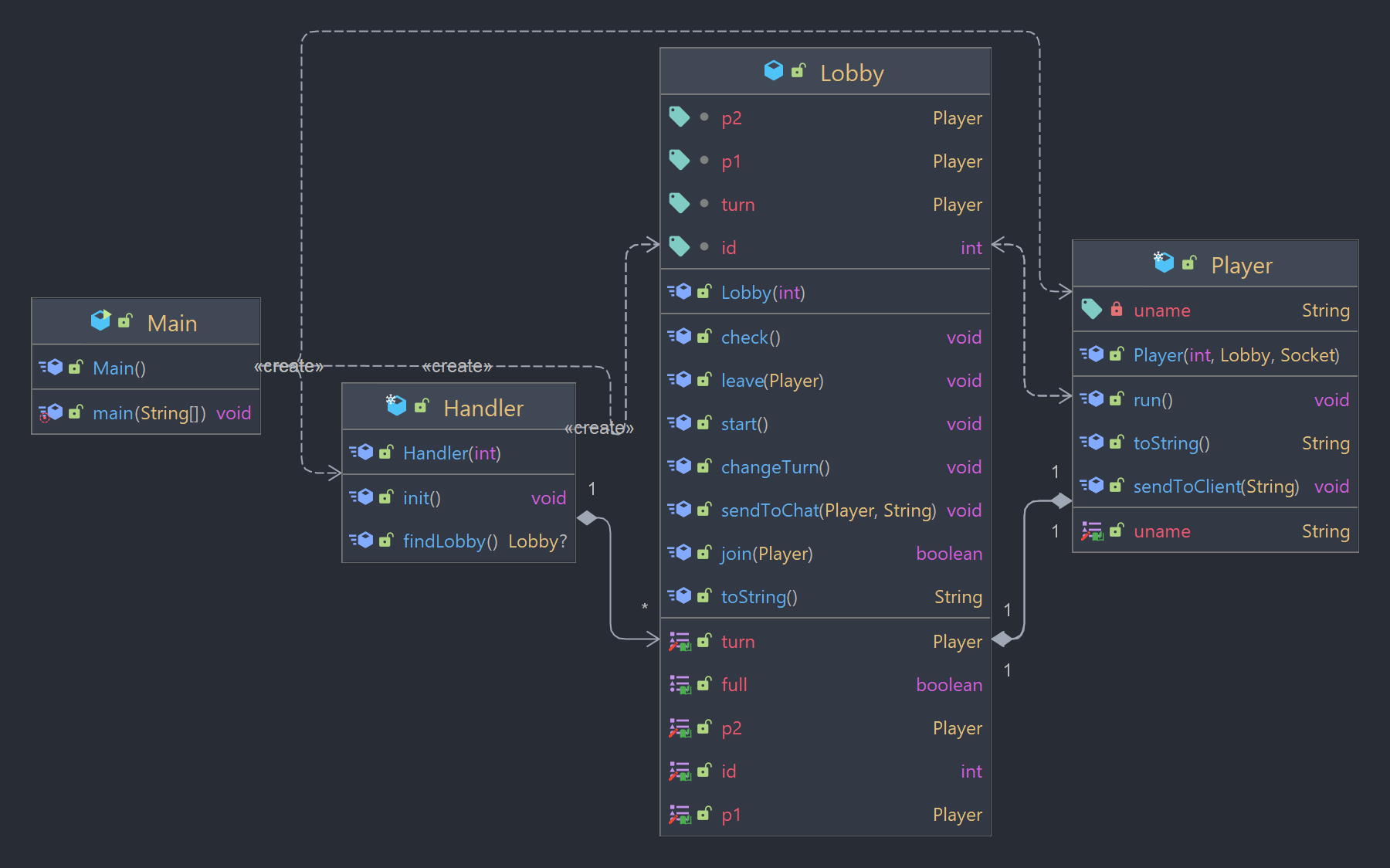
### UML: Client



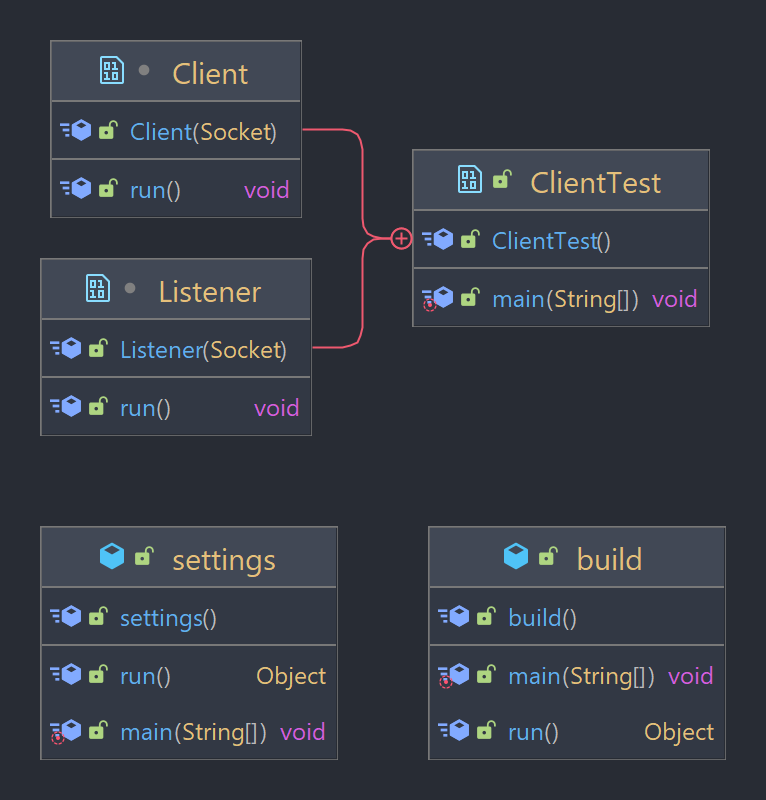
### UML: API



### UML: Game Server



### UML: SampleClient



## Fehlerbehandlung

|  |  |  |  |
| --- | --- | --- | --- |
| Baustein | Exception | Klasse | Behandelt |
| Game Server | IOException | Player | Ja |
| Game Server | SocketException | Player | Ja |
| Game Server | IOException | Main | Ja |
| Game Server | SocketException | Main | Ja |
| API Server | \*Exception | ApiServerApplication | Ja |
| Client | IOException | Listener | Ja |
| Client | IOException | Client | Ja |

## Validierung

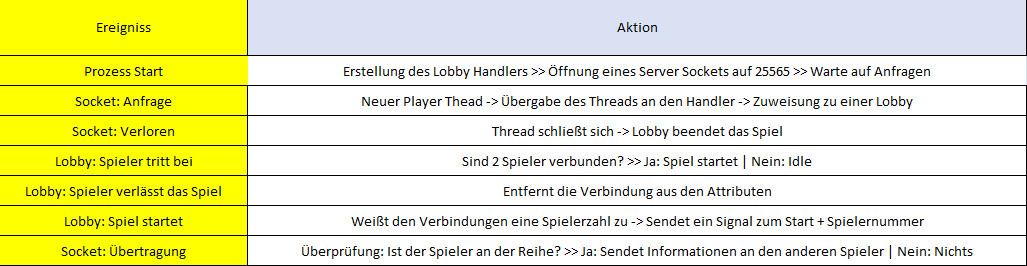
Können im Code oder dem Repository betrachtet werden.

# Projektorganisation

## Annahmen

**Verwendete Technologien**  
Sprachen: Java, JavaScript, HTML, CSS, SQL  
Frameworks: LibGDX, Spring Boot, Bootstrap  
Build Tools: Gradle  
Pipeline: TeamCity  
Tools: IntelliJ Ultimate, DataGrip, VIM, GitHub, Git, Powershell, Bootstrap Studio

**Repositories**  
Client: [Marckfx/War-of-the-summon](https://github.com/Marckfx/War-of-the-summon)  
Backend: [DubskySteam/GameServer](https://github.com/DubskySteam/GameServer)

**Einschränkungen**-> Zeitliche Begrenzung (fixen Abgabetermin)  
-> Kein System mit Mac Basis vorhanden, daher keine realen Tests möglich

## Risiken

* Nicht Einhaltung des Zeitplans
* Schlechte Zusammenarbeit
* Absprung von Gruppenmitgliedern
* Softwaretechnische Probleme
* Langwieriger Verlust der Internetverbindung (Tatsächlich ein dagewesenes Problem)
* Fehlende Kenntnisse

## Verantwortlichkeit

|  |  |
| --- | --- |
| Softwarebaustein | Person |
| Grafikdesign | Marcel Sander |
| Spiellogik | Marcel Sander |
| Spieldesign | Marcel Sander |
| Game Server | Clemens Maas |
| API Server | Clemens Maas |
| Datenbank | Clemens Maas |

#### Rollenzuordnung

|  |  |
| --- | --- |
| Rolle | Person |
| Frontend-Entwickler | Softwarearchitekt | Marcel Sander |
| Backend-Entwickler | Softwarearchitekt | Clemens Maas |

## Grober Projektplan

**Backend**: [Kanban Board](https://github.com/DubskySteam/GameServer/projects)

**Frontend**: [Kanban Board](https://github.com/Marckfx/War-of-the-summon/projects/1)