



**TECHNISCHE UNIVERSITÄT  
KAISERSLAUTERN**

AG Computergrafik & HCI  
apl. Prof. Dr. Achim Ebert

SEP/MP 2020

---

# Exploding Kittens

Pflichtenheft

4. Mai 2020

---

## *Gruppe N*

Siyu Zhang

Yirui Fu

Manuela Kuaguim

Florent Tandjune Tamoyem

Thiemo Lachmund

Alexander Nachtigall

# Inhaltsverzeichnis

<b>Inhaltsverzeichnis</b>	<b>2</b>
<b>1 Projekttreiber</b>	<b>3</b>
1.1 Projektziel . . . . .	3
1.2 Stakeholders . . . . .	3
1.3 Aktuelle Lage . . . . .	4
<b>2 Projektbeschränkungen</b>	<b>5</b>
2.1 Beschränkungen . . . . .	5
2.2 Glossar . . . . .	6
2.3 Relevante Fakten und Annahmen . . . . .	6
<b>3 Funktionale Anforderungen</b>	<b>8</b>
3.1 Systemfunktionen . . . . .	8
3.2 Systemgrenze (Use Case Diagramm) . . . . .	9
3.3 Beschreibungen der Anwendungsfälle . . . . .	9
3.4 Produktdaten . . . . .	12
<b>4 Nicht-funktionale Anforderungen</b>	<b>13</b>
4.1 Softwarearchitektur . . . . .	13
4.2 Benutzerfreundlichkeit . . . . .	13
4.3 Leistungsanforderungen . . . . .	14
4.4 Anforderungen an Einsatzkontext . . . . .	14
4.5 Anforderungen an Wartung und Unterstützung . . . . .	15
4.6 Sicherheitsanforderungen . . . . .	16
4.7 Prüfungsbezogene Anforderungen . . . . .	16
4.8 Kulturelle und politische Anforderungen . . . . .	17
4.9 Rechtliche und standardsbezogene Anforderungen . . . . .	17
4.10 Bedienoberfläche . . . . .	18
<b>5 Systemtestfälle</b>	<b>20</b>
<b>6 Warteraum</b>	<b>25</b>

# Kapitel 1

## Projekttreiber

### 1.1 Projektziel

Im Rahmen des Software-Entwicklungs-Projekts/Modellierungspraktikums 2020 soll ein einfach zu bedienendes Client-Server-System zum Spielen von Exploding Kittens über ein Netzwerk implementiert werden. Die Benutzeroberfläche soll intuitiv bedienbar sein.

### 1.2 Stakeholders

/SH10/ **Name:** Spieler

**Beschreibung:** Menschliche Spieler.

**Ziele/Aufgaben:** Das Spiel zu spielen.

/SH20/ **Name:** Eltern

**Beschreibung:** Eltern minderjähriger Spieler.

**Ziele/Aufgaben:** Um die Spieler zu kümmern, indem Eltern Spielzeit begrenzen wollen und zugriff auf sensible Inhalte begrenzen.

/SH30/ **Name:** Gesetzgeber

**Beschreibung:** Das Amt für Jugend und Familie.

**Ziele/Aufgaben:** Die Rechte der Spieler zu schützen und zu gewähren, indem er Gesetze erstellt.

/SH40/ **Name:** Investoren (nur für Beispielszwecken)

**Beschreibung:** Parteien, die das Finanzmittel für die Entwicklung des Systems bereitstellen.

**Ziele/Aufgaben:** Gewinn zu ermitteln, indem das System an Endverbraucher verkauft wird.

/SH50/ **Name:** Betreuer

**Beschreibung:** HiWis, die SEP/MP Projektgruppen betreuen.

**Ziele/Aufgaben:** Das Entwicklungsprozess zu betreuen, zu überwachen und teilweise zu steuern als auch die Arbeit der Projektgruppen abzunehmen sowie den Studenten im Prozess Hilfe zur Verfügung zu stellen.

/SH60/ **Name:** Prof. Dr. Achim Ebert

**Beschreibung:**

**Ziele/Aufgaben:**

### 1.3 Aktuelle Lage

Aktuell wird das Spiel so gespielt, dass... aber dabei ist das folgende Problematisch... Das Projekt wird den Spielern ermöglichen... Und die Eltern profitieren davon, dass... usw...<sup>1</sup>

---

<sup>1</sup>Hier wird beschrieben, wie die fachlichen Prozesse werden aktuell (also vor der Implementierung) abgewickelt und wieso es wichtig ist, das Projekt umzusetzen. Man kann hier auch auf die Bedürfnisse einzelner Stakeholder eingehen, muss aber nicht.

## Kapitel 2

# Projektbeschränkungen

### 2.1 Beschränkungen

/LB10/ **Name:** Selbstlehrende Bots

**Beschreibung:** Keine Selbstlehrfunktion von Bots wird implementiert.

**Motivation:** Die Funktionalität ist zu aufwändig zu implementieren und passt deshalb nicht in das Zeitbudget.

**Erfüllungskriterium:** Intelligenzalgorithmus von Bots ist so vorprogrammiert, dass sie Entscheidungen nur anhand des vorprogrammierten Wissens sowie des aktuellen Spielstands treffen, ohne dabei frühere Spiele zu berücksichtigen.

/LB20/ **Name:** Anwendungsbereich

**Beschreibung:** Das System ist ausschließlich für den privaten Bereich ausgelegt.

**Motivation:**

**Erfüllungskriterium:**

/LB30/ **Name:** Implementierungssprache

**Beschreibung:** Für die Implementierung ist ausschließlich Java 8 oder höher zu verwenden.

**Motivation:** Das optimiert die Betreuung von SEP/MP und koordiniert die Mitarbeit.

**Erfüllungskriterium:**

/LB40/ **Name:** GUI-Framework

**Beschreibung:** Die GUI ist mit JavaFX zu realisieren.

**Motivation:** Das optimiert die Betreuung von SEP/MP und koordiniert die Mitarbeit.

**Erfüllungskriterium:****/LB50/ Name:** Gitlab**Beschreibung:** Für die Entwicklung ist das vorgegebene GitLab-Repository zu verwenden.**Motivation:** Das optimiert die Betreuung von SEP/MP und koordiniert die Mitarbeit.**Erfüllungskriterium:**

## 2.2 Glossar

Deutsch	Englisch	Bedeutung
Beispiele	Examples	Beispiele aus dem SEP letzter Jahren, welche angepasst werden müssen.
Bot	bot	Spieler, dessen Spielaktionen vom Computer entschieden und durchgeführt werden
Kekse	Cookies	Offiziell keine gültige Maßnahme zur Bestechung der HiWis
Lobby	lobby	Virtueller Raum zum Betreten eines Spielraums
Spiel (Regelwerk)	game	Exploding Kittens
Spieler	player	Teilnehmer am Spielgeschehen
Spielraum	game room	Virtueller Raum, in dem ein Spiel stattfindet
Zug	turn	Zustand in dem ein Spieler eine Spielaktion ausführen muss

## 2.3 Relevante Fakten und Annahmen

Wichtige gekannte Fakten und getroffene Annahmen, die sich auf das Projekt direkt oder indirekt beziehen und dadurch auf die zukünftige Implementierungsentscheidungen Effekt haben können.

**/FA10/ Name:** Keine Fortentwicklung des App nach SEP/MP.**Beschreibung:** Nach Ende des SEP/MP wird das Projekt nicht weiterentwickelt.**Motivation:** Das Entwicklungsteam hat keine Lust darauf.

/FA20/ **Name:** Keine Lizenzen für Spielartefakte.

**Beschreibung:** Weder die TU Kaiserslautern noch das Spielwerk + die Freizeit GmbH gewahren dem Entwicklungsteam die Rechte für die Spielartefakte.

**Motivation:** Rechtliche Vorsorge.

/FA30/ **Name:** Keine bekannte Nachteile von Verwendung von Spielartefakten.

**Beschreibung:** Es ist nicht bekannt, dass die SEP/MP-Teilnehmer der letzten Jahre irgendwelche rechtlichen Probleme dadurch gehabt haben, dass sie die Spielartefakten vom Spielwerk + Freizeit GmbH im Rahmen ihrer SEP/MP eingesetzt haben.

**Motivation:** Rechtliche Vorsorge.

## Kapitel 3

# Funktionale Anforderungen

### 3.1 Systemfunktionen

/LF10/    **Name:** Spielverwaltung

**Beschreibung:** Das System verwaltet das von mehreren Spielern geteilte Spiel in einem Spielraum. Das Spiel erfolgt nach den Spielregeln.

/LF20/    **Name:** Zugriffsverwaltung

**Beschreibung:** Das System verwaltet den Zugang zum Spiel anhand Benutzerdaten. Spieler können sich registrieren, anmelden, abmelden sowie ihre Kontos löschen.

/LF30/    **Name:** Verwaltung der Spielräume

**Beschreibung:** Das System verwaltet die Erstellung, Änderung und Löschung der Spielräume.

/LF40/    **Name:** Bestenliste

**Beschreibung:** Die Anzahl der gewonnenen Spiele aller Spieler anzeigen.

/LF50/    **Name:** Intelligente Bots

**Beschreibung:**

/LF60/    **Name:** Chat

**Beschreibung:**



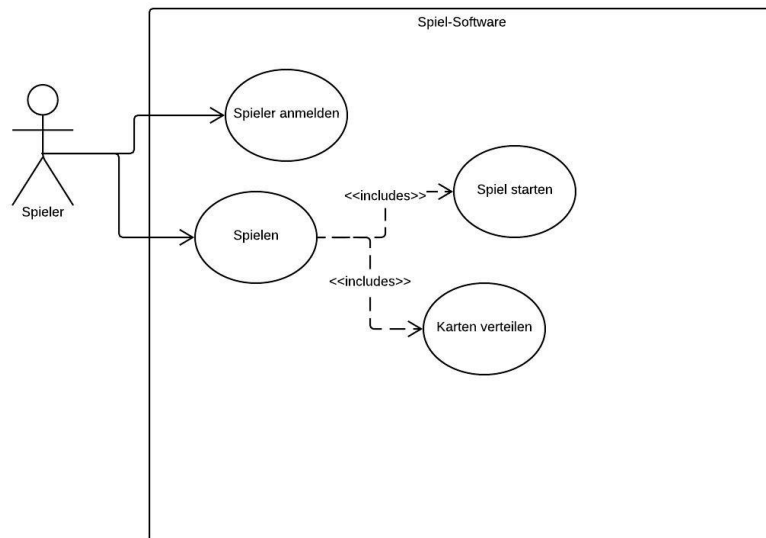


Abbildung 3.1: Beispiel für ein Systemgrenzendiagramm (Use Case Diagramm), das vor Abgabe anzupassen ist.

### 3.2 Systemgrenze (Use Case Diagramm)

Die Systemgrenze wird in der Abbildung 3.1 dargestellt<sup>1</sup>.

### 3.3 Beschreibungen der Anwendungsfälle

Hinweis: Alle Systemfunktionen sind mit Anwendungsfällen zu decken! (Und dieser Hinweis ist zu löschen).

/UC10/ **Name:** Name des Use Cases<sup>2</sup>.

**Ziel:** Ziel und Zweck des Use Cases.

**Akteure:** Akteure (auch benachbarte Systeme können Use Cases anfeuern), die den Use Case aktivieren können.

**Vorbedingungen:** Eigenschaften des Systemzustands, in dem die Aktivierung des Use Cases möglich ist.

**Eingabedaten:** Daten, die für die Ausführung des Use Cases nötig sind. Mit Referenzen auf 3.4.

<sup>1</sup>Weitere Erklärungen und Spezifizierungen, die sich auf Abgrenzungen der Verantwortlichkeiten vom System und weiteren Akteuren/Systemen beziehen, können hier spezifiziert werden.

<sup>2</sup>Dieser Anwendungsfall ist offensichtlich ein/e Beispiel/Anleitung und muss gelöscht werden.

**Beschreibung:** Der Ablauf der Use Cases. Das Handeln von Akteur und System MÜSSEN **nacheinander** abwechselnd erfolgen. PASSIV VERMEIDEN!

Gut:

1. **Der Spieler** sendet das Formular ab.
2. **Das System** prüft die Richtigkeit vom Passwort und fragt Spieler noch ein mal, ob er sich wirklich aus dem System entfernen möchte.
3. **Der Spieler** bestätigt seine Intention.
4. **Das System** entfernt alle Daten des Spielers aus der Datenbank und bewegt Spieler in den Vorraum.

Schlecht:

1. Das Formular wird versandt.
2. Das System prüft die Richtigkeit vom Passwort.
3. Das System fragt Spieler noch ein mal, ob er sich wirklich aus dem System entfernen möchte. Der Spieler bestätigt seine Intention.
4. Das System entfernt alle Daten des Spielers aus der Datenbank.
5. der Spieler wird in den Vorraum bewegt.

**Ausnahmen:** Verhalten des Systems in Ausnahmefällen (wenn etwas nicht ganz wie gedacht geht).

**Ergebnisse und Outputdaten:** Beschreibung des Systemzustands nach einer erfolgreichen Ausführung des Use Cases sowie die auszugebenden Daten.

**Systemfunktionen** Referenz auf die relevante(n) Systemfunktion(en).

/UC20/ **Name:** Spieler anmelden.

**Ziel:** Spieler meldet sich im System an.

**Akteure:** Spieler.

**Vorbedingungen** Spieler ist im Vorraum.

**Eingabedaten:** Zugriffsdaten /LD10/ /LD20/.

**Beschreibung:**

1. Spieler sendet das Formular ab.
2. Das System prüft die Gültigkeit von Zugangsdaten und bewegt

den Spieler in die Lobby.

**Ausnahmen:**

*Passwort oder Benutzername ist falsch:* Das System zeigt eine Fehlermeldung an, anstatt des Schrittes 2.

**Ergebnisse und Outputdaten:** Spieler ist in der Lobby und sieht die Bestenliste.

**Systemfunktionen:** /LF20/.

/UC30/ **Name:** Spieler löschen.

**Ziel:** Spieler entfernt seine Daten aus dem System.

**Akteure:** Spieler.

**Vorbedingungen** Spieler ist im Vorraum.

**Eingabedaten:** Passwort /LD20/.

**Beschreibung:**

1. Spieler sendet das Formular ab.
2. Das System prüft die Richtigkeit des Passworts und fragt Spieler noch ein mal, ob er sich wirklich aus dem System entfernen möchte.
3. Spieler bestätigt seine Intention.
4. Das System entfernt alle Daten des Spielers aus der Datenbank und bewegt Spieler in den Vorraum.

**Ausnahmen:**

*Passwort ist falsch:* Das System zeigt eine Fehlermeldung an, anstatt des Schrittes 2.

*Keine Löschung erwünscht:* Anstatt des Schrittes 4, schließt das System den Dialog.

**Ergebnisse und Outputdaten:** Spieler ist im Vorraum, Spielerkonto wurde gelöscht.

**Systemfunktionen:** /LF20/.

### 3.4 Produktdaten

Hier sollen die Daten genannt werden, die im System verwendet werden.

**/LD10/ Name:** Benutzername\*<sup>3</sup>

**Fachliche Beschreibung:** Benutzername des Spielers

**Relevante Systemfunktionen:** /LF10/, /LF20/

**/LD20/ Name:** Passwort\*

**Fachliche Beschreibung:** Passwort des Spielers

**Relevante Systemfunktionen:** /LF20/

---

<sup>3</sup> “\*” bedeutet hier, dass die Daten in der Datenbank zu speichern sind

## Kapitel 4

# Nicht-funktionale Anforderungen

### 4.1 Softwarearchitektur

/NF10/ **Name:** Client-Server Anwendung

**Beschreibung:** Das verteilte Spiele-System ermöglicht das gemeinsame Spielen von verschiedenen Rechnern aus.

**Motivation:** Aufgabestellung v. SEP/MP.

**Erfüllungskriterium:** Das fertige System besteht aus Client- und Server-Teilen.

/NF20/ **Name:** Plattformunabhängigkeit

**Beschreibung:** Es soll sich um eine plattformunabhängige Anwendung handeln. Zumindest Windows- und Linuxsysteme sind zu unterstützen.

**Motivation:** Aufgabenstellung v. SEP/MP.

**Erfüllungskriterium:**

### 4.2 Benutzerfreundlichkeit

/NF30/ **Name:** Benutzeralter

**Beschreibung:** Das System ist für Benutzer geeignet, die älter als 5 Jahre sind.

**Motivation:** Jüngere Benutzer sind unfähig das Spiel zu spielen.

**Erfüllungskriterium:** In den AGBs steht ein entsprechender Hinweis.

/NF40/ **Name:** Technische Fähigkeiten

**Beschreibung:** Besondere technische Fähigkeiten sind von den Benutzern nicht zu erwarten.

**Motivation:** Auch die Menschen, die kaum etwas von Bedienung bzw. Programmierung von Rechnern verstehen, sollen fähig sein, das System zu verwenden.

**Erfüllungskriterium:**

### 4.3 Leistungsanforderungen

/NF50/ **Name:** Antwortzeit

**Beschreibung:** Maximale Antwortzeit für alle Systemprozesse.

**Motivation:** Das System muss immer brauchbar sein.

**Erfüllungskriterium:** Das System antwortet auf Benutzerhandlungen nie später als in 10 Sekunden.

### 4.4 Anforderungen an Einsatzkontext

#### Anforderungen an physische Umgebung

/NF60/ **Name:** Lauffähigkeit an SCI-Rechnern

**Beschreibung:** Das Produkt muss auf einem eigenem Gerät lauffähig sein, welches zur Präsentation am Ende des SEPs genutzt werden muss. Falls keine eigenen Rechner vorhanden sind, stehen auch die SCI-Terminals zur Verfügung.

**Motivation:** Optimierung von Betreuung und Abnahme des SEP/MP

**Erfüllungskriterium:**

#### Absatz- sowie Installationsbezogene Anforderungen

/NF70/ **Name:** Installationsanleitung

**Beschreibung:** Falls die Installation nicht lediglich das Öffnen einer Datei voraussetzt, muss der genaue Installations- und Startvorgang schriftlich für Benutzer zur Verfügung gestellt werden.

**Motivation:** Spezifikation

**Erfüllungskriterium:**

#### Anforderungen an Versionierung

/NF80/ **Name:** Keine weitere Versionen

**Beschreibung:** Nach Version 1.0 ist keine weitere Entwicklung vorgesehen.

**Motivation:** Das ist nur das SEP/MP, kein Geschäftsprojekt, siehe [/FA10/](#)

**Erfüllungskriterium:**

## 4.5 Anforderungen an Wartung und Unterstützung

### Wartungsanforderungen

**/NF90/ Name:** Beispiel

**Beschreibung:**

**Motivation:**

**Erfüllungskriterium:**

**/NF100/ Name:** Dokumentation

**Beschreibung:** Der Quellcode muss ausführlich dokumentiert werden.

**Motivation:**

**Erfüllungskriterium:** JavaDoc

**/NF110/ Name:** Testen

**Beschreibung:** Der Quellcode außer GUI muss gut getestet werden.

**Motivation:**

**Erfüllungskriterium:** Von Unit-Tests muss mindestens 70% des Quellcodes bedeckt werden. GUI-Klassen sind aus der Anforderung ausgenommen.

### Anforderungen an technische und fachliche Unterstützung

**/NF120/ Name:** Beispiel

**Beschreibung:** Es ist keine technische und fachliche Unterstützung des Systems geplant.

**Motivation:** Siehe [/FA10/](#).

**Erfüllungskriterium:** Nicht anwendbar.

### Anforderungen an technische Kompatibilität

**/NF130/ Name:** Beispiel

**Beschreibung:**

**Motivation:**

**Erfüllungskriterium:**

## 4.6 Sicherheitsanforderungen

### Zugang

/NF140/ **Name:** Beispiel

**Beschreibung:**

**Motivation:**

**Erfüllungskriterium:**

### Integrität

/NF150/ **Name:** Beispiel

**Beschreibung:**

**Motivation:**

**Erfüllungskriterium:**

### Datenschutz/Privatsphäre

/NF160/ **Name:** Beispiel

**Beschreibung:**

**Motivation:**

**Erfüllungskriterium:**

### Virenschutz

/NF170/ **Name:** Beispiel

**Beschreibung:**

**Motivation:**

**Erfüllungskriterium:**

## 4.7 Prüfungsbezogene Anforderungen

Anforderungen, die sich auf die Prüfung/Audit vom System von SEP/MP-Tutoren oder von weiteren Instanzen beziehen.

/NF180/ **Name:** Formate der Systemdokumentation

**Beschreibung:** Systemdokumentation muss in 2 Formen geführt werden (wenn anwendbar): Die Ausgangsdateien (L<sup>A</sup>T<sub>E</sub>X, Dateien der Diagrammerstellungssoftware, Dateien der Grafiksoftware usw.) und PDFs.

**Motivation:** Optimierung der SEP/MP-Betreuung.

**Erfüllungskriterium:** Siehe Beschreibung.



## 4.8 Kulturelle und politische Anforderungen

/NF190/ **Name:** Systemsprache

**Beschreibung:** Die Interfacesprache ist Deutsch.

**Motivation:** Synchronisation des Verständnisses von Teammitgliedern mit unterschiedlichen kulturellen Hintergründen.

**Erfüllungskriterium:**

## 4.9 Rechtliche und standardsbezogene Anforderungen

/NF200/ **Name:** Nicht rechtliche Anforderungen

**Beschreibung:** Keine relevanten rechtlichen Anforderungen bekannt.

**Motivation:** Siehe [/FA10/](#).

**Erfüllungskriterium:** Nicht anwendbar.

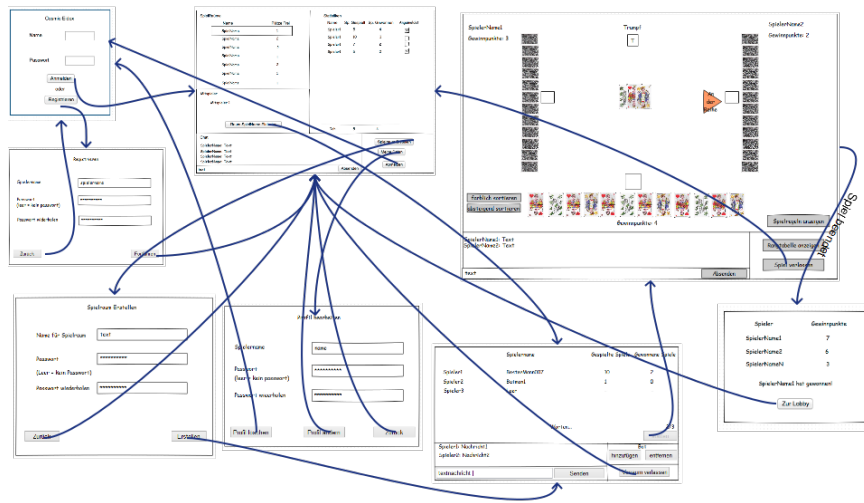


Abbildung 4.1: Beispiel zur Darstellung der Zusammenhänge zwischen GUI-Ansichten.

## 4.10 Bedienoberfläche

Hier sollen die Skizzen/Prototypen von Bedienoberflächen dargestellt werden, als auch die Zusammenhänge zwischen denen (wie gelingt man von einem zu dem anderen Fenster/Ansicht). Ein Beispiel für Bildereinbau in LaTeX ist die Abbildung 4.1.<sup>1</sup>

/GUI10/ **Name:** Vorräum-Interface

**Beschreibung:** Interface für Anmeldung

**Relevante Systemfunktionen:** /LF20/

**Abbildungen:** 4.2

/GUI20/ **Name:** Zusammenhänge

**Beschreibung:** Zusammenhänge zwischen GUI-Ansichten

**Relevante Systemfunktionen:** Alle

**Abbildungen:** 4.1

<sup>1</sup>Bevor Sie mit den Skizzen anfangen, überlegen Sie sich, welche virtuelle Räume im System zu haben sind und dann halte Sie die Namen der GUI-Fenstern mit diesen konstant.



The image shows a GUI mockup for a system named "Cosmic Eidex". It features a central white rectangular area with a solid blue border, set against a light gray background with a dashed gray border. Inside the white area, the title "Cosmic Eidex" is centered at the top. Below the title, there are two input fields: one labeled "Name" and one labeled "Passwort". Each label is positioned to the left of its corresponding input box. Below the input fields, there are two buttons: "Anmelden" and "Registrieren", both with a light gray gradient and rounded corners. The word "oder" is centered between these two buttons.

Cosmic Eidex

Name

Passwort

Anmelden

oder

Registrieren

Abbildung 4.2: Beispiel für ein GUI-Mockup. Dieser Text ist auch nur ein Beispiel :)

## Kapitel 5

# Systemtestfälle

Hier sollen verschiedene Szenarien beschrieben werden, mithilfe deren Sie später Systemtests ausführen und die erwarteten Ergebnisse darstellen.

**/TF10/ Name:** Spieler anmelden.

**Motivation:** Testet, ob die Anmeldung im System korrekt funktioniert.

**Szenarien:**

1. *Zugriffsdaten sind vorhanden und richtig*  
⇒ Spieler wird in die Lobby bewegt.
2. *Benutzername ist registriert, Passwort ist falsch*  
⇒ Fehlermeldung wird angezeigt.
3. *Benutzername ist nicht registriert*  
⇒ Fehlermeldung wird angezeigt.

**Relevante Systemfunktionen:** /LF20/

**Relevante Use Cases:** /UC30/

**/TF20/ Name:** Spieler registrieren.

**Motivation:** Testet, ob die Registrierung in dem System korrekt funktioniert.

**Szenarien:**

1. *Benutzername sind vorhanden und richtig*  
⇒ Ein Account mit den eingetragenen Daten wird erzeugt und der Spieler wird in die Lobby bewegt.
2. *Benutzername ist schon genutzt*  
⇒ Fehlermeldung wird angezeigt.
3. *Email ist schon genutzt*  
⇒ Fehlermeldung wird angezeigt.

**Relevante Systemfunktionen:** /LF20/

**Relevante Use Cases:** /UC30/

/TF30/ **Name:** Account bearbeiten.

**Motivation:** Testet, ob das Bearbeiten des Accounts in dem System korrekt funktioniert.

**Scenarien:**

1. *Neuer Benutzername nicht vorhanden*  
⇒ Benutzername wird erfolgreich geändert.
2. *Neuer Benutzername bereits vorhanden*  
⇒ Fehlermeldung wird angezeigt.
3. *Neues Passwort bearbeiten, altes richtig*  
⇒ Passwort wird erfolgreich geändert.
4. *Neues Passwort bearbeiten, altes falsch*  
⇒ Fehlermeldung wird angezeigt.
5. *Profil Bild bearbeiten*  
⇒ Profil Bild wird erfolgreich geändert.
6. *Profil Bild bearbeiten funktioniert nicht*  
⇒ Fehlermeldung wird angezeigt.

**Relevante Systemfunktionen:** /LF20/

**Relevante Use Cases:** /UC30/

/TF40/ **Name:** Account löschen.

**Motivation:** Testet, ob das Löschen des Accounts in dem System korrekt funktioniert.

**Scenarien:**

1. *Passwort richtig*  
⇒ Spieler wird in den Anmeldebildschirm bewegt, Account wird gelöscht.
2. *Passwort falsch*  
⇒ Fehlermeldung wird angezeigt.

**Relevante Systemfunktionen:** /LF20/

**Relevante Use Cases:** /UC30/

/TF50/ **Name:** Spielraum erstellen, bearbeiten oder löschen.

**Motivation:** Testet, ob das Erstellen, das Bearbeiten oder das Löschen eines Spielraums im System korrekt funktioniert.

**Scenarien:**

1. *Spielraum erstellt*  
⇒ Spieler wird im erstellte Spielraum bewegt.

2. *Spielraum erstellt funktioniert nicht*  
⇒ Fehlermeldung wird angezeigt und kein Spielraum wird erstellt.
3. *Spielraum bearbeiten*  
⇒ Spielraum Eigenschaften werden geändert.
4. *Spielraum bearbeiten funktioniert nicht*  
⇒ Fehlermeldung wird angezeigt.
5. *Spielraum löschen*  
⇒ Spieler soll das Löschen bestätigen.  
⇒ Spieler wird im Spiel Menü bewegt.  
⇒ Spielraum wird gelöscht.
6. *Spielraum löschen funktioniert nicht*  
⇒ Fehlermeldung wird angezeigt.  
⇒ Spielraum wird nicht gelöst.

**Relevante Systemfunktionen:** /LF20/

**Relevante Use Cases:** /UC30/

/TF60/ **Name:** Bots.

**Motivation:** Testet, ob das Hinzufügen bzw. das Entfernen der Bots korrekt funktioniert.

**Scenarien:**

1. *Bots hinzufügen, Platz vorhanden*  
⇒ Bots werden im Spielraum bewegt.
2. *Bots hinzufügen, kein Platz vorhanden*  
⇒ Fehlermeldung wird angezeigt.
3. *Bots entfernen, Bot im Raum*  
⇒ Bots werden entfernt.
4. *Bots entfernen, kein Bot im Raum*  
⇒ Fehlermeldung wird angezeigt.

**Relevante Systemfunktionen:** /LF20/

**Relevante Use Cases:** /UC30/

/TF70/ **Name:** Spiel.

**Motivation:** Testet, ob das Starten bzw. das Ende des Spiels korrekt funktioniert.

**Scenarien:**

1. *Spiel starten, genug Spieler*  
⇒ Spiel starten.  
⇒ Spieler werden im Spielraum bewegt.

2. *Spiel starten, nicht genug Spieler*  
⇒ Fehlermeldung wird angezeigt.
3. *Spiel beenden funktioniert*  
⇒ Spiel wird beendet.  
⇒ Spieler werden in Lobby bewegt.
4. *Spiel beenden funktioniert nicht*  
⇒ Fehlermeldung wird angezeigt.

**Relevante Systemfunktionen:** /LF20/

**Relevante Use Cases:** /UC30/

/TF80/ **Name:** Bestenliste.

**Motivation:** Testet, ob das Anzeigen der Bestenliste korrekt funktioniert.

**Scenarien:**

1. *Bestenliste anzeigen, mindestens ein Spiel abgeschlossen*  
⇒ Bestenliste anzeigen.
2. *Bestenliste anzeigen, bisher niemand gespielt*  
⇒ Fehlermeldung wird angezeigt.

**Relevante Systemfunktionen:** /LF20/

**Relevante Use Cases:** /UC30/

/TF90/ **Name:** Chatroom.

**Motivation:** Testet, ob das Empfangen bzw. das Senden von Nachrichten korrekt funktioniert.

**Scenarien:**

1. *Nachricht schicken, mit Inhalt*  
⇒ Nachricht wird verschickt und erscheint im Chatfenster aller Spieler.
2. *Nachricht schicken, kein Inhalt*  
⇒ Fehlermeldung wird angezeigt.  
⇒ Keine Nachricht wird verschickt.

**Relevante Systemfunktionen:** /LF20/

**Relevante Use Cases:** /UC30/

/TF100/ **Name:** :Spieler einladen.

**Motivation:** Testet, ob man Spieler einladen kann.

**Scenarien:**

1. *Spieler wird zum Spiel eingeladen, Spieler existiert*  
⇒ Spiel starten mit Freunden.

2. *Spieler wird zum Spiel eingeladen, Spieler existiert nicht*  
⇒ Fehlermeldung wird angezeigt.

**Relevante Systemfunktionen:** /LF20/

**Relevante Use Cases:** /UC30/

/TF110/ **Name:** Spieler Punkte.

**Motivation:** Testet, ob die Spielers Punkte übertragen werden.

**Szenarien:**

1. *Das Spiel ist fertig. Kein Spielabbruch.*  
⇒ Verdiente Punkte werden zu den vorherigen Punkte addiert.
2. *Das Spiel ist fertig. Spielabbruch.*  
⇒ Keine Punkte werden zu den vorherigen Punkte addiert.

**Relevante Systemfunktionen:** /LF20/

**Relevante Use Cases:** /UC30/



## Kapitel 6

# Warteraum

Hier werden Anforderungen spezifiziert die den sogenannten “Warteraum” darstellen. Hier gehören alle Anforderungen, die “Wünschkriterien” sind, das heißt, sie sind zwar erwünscht, aber werden nur dann in aktuelle Anforderungen übernommen, wenn dafür genügendes Zeitbudget vorhanden ist und werden am wahrscheinlichsten in der Zukunft (und nicht jetzt) implementiert (oder in den kommenden Sprints beim SCRUM-Prozessmodell).

**/WR10/ Name:** Hintergrundmusik

**Beschreibung:** Für die Spieler soll eine Auswahl zur Verfügung stehen, mit der die Hintergrundmusik beim Spielen ausgewählt werden kann.

**Motivation:** Höhere Zufriedenheit der Benutzer.

**Erfüllungskriterium:** Spieler können zu jedem Zeitpunkt (außer im Vorraum) die Musik ausschalten oder ein anderes Lied auswählen.