Design-Dokumentation

Inhalt

[Glossar 2](#_Toc36476369)

[Design der Anwendung 3](#_Toc36476370)

[Benutzerführung 3](#_Toc36476371)

[Dokumentation der Anwendung 4](#_Toc36476372)

[UML-Diagramm 5](#_Toc36476373)

[Lastenheft 6](#_Toc36476374)

# Glossar

Im Folgenden werden die Begriffe erläutert, die im weiteren Verlauf dieses Dokuments genutzt werden.

|  |  |
| --- | --- |
| Begriff | Bedeutung |
| MUD, Multi User Dungeon | Ein meist textbasiertes Rollenspiel, in dem mehrere Spielercharaktere ein Abenteuer erleben können. |
| DM, Dungeon Master | Person, die die Geschichte erzählt und die Spielergruppe durch den Spielinhalt leitet. Der DM besitzt nahezu völlige Kontrolle auf das Spielgeschehen, und kann auf den Chat der Spieler, mit Ausnahme des Flüsterchats, zugreifen und daran teilnehmen. |
| Spieler, User | Ein Teilnehmer an einer Session auf einem MUD, bei dem es sich nicht um einen Dungeon Master handelt. Ein Spieler in einer Session kann der Dungeon Master in einer anderen Session sein. |
| Responsive | Die Webseite soll sich dynamisch an verschiedene Browsergrößen anpassen. |
| Accountname | Name des Nutzeraccounts. Im Gegensatz zu den Nicknamen, die für die jeweiligen Sessions gewählt werden, ist der Accountname für andere Nutzer nicht einsehbar. |
| Rasse, Klasse | Beides sind Konfigurationsmöglichkeiten eines Spielercharakters. Sie beeinflussen die Fähigkeiten eines Charakters und seine Attribute. Im Verlauf des Spiels kann ein der Spieler abhängig von seiner Rasse bzw. Klasse zusätzliche Fähigkeiten erhalten. |

# Design der Anwendung

Die Anwendung wird aufgeteilt in ein Frontend und ein Backend. Durch diese Aufteilung wird eine Modularität geschaffen, die später eine größere Flexibilität ermöglicht. So kann beispielsweise ein komplett alternatives Frontend hinzugefügt werden, für eine andere Darstellung. Weiterhin wird aufgrund der Modularität und Wartbarkeit die Benutzung von Microservices geplant. Der Benutzer innerhalb der Anwendung mit *du* angesprochen.

Das Frontend umfasst dabei die Webseite, die der Kunde nutzt um auf die Anwendung zuzugreifen und die Anwendung zu bedienen. Das Frontend wird daher in HTML und JavaScript geschrieben und mithilfe von CSS gestaltet. Es wird das VUE-Framework benutzt.

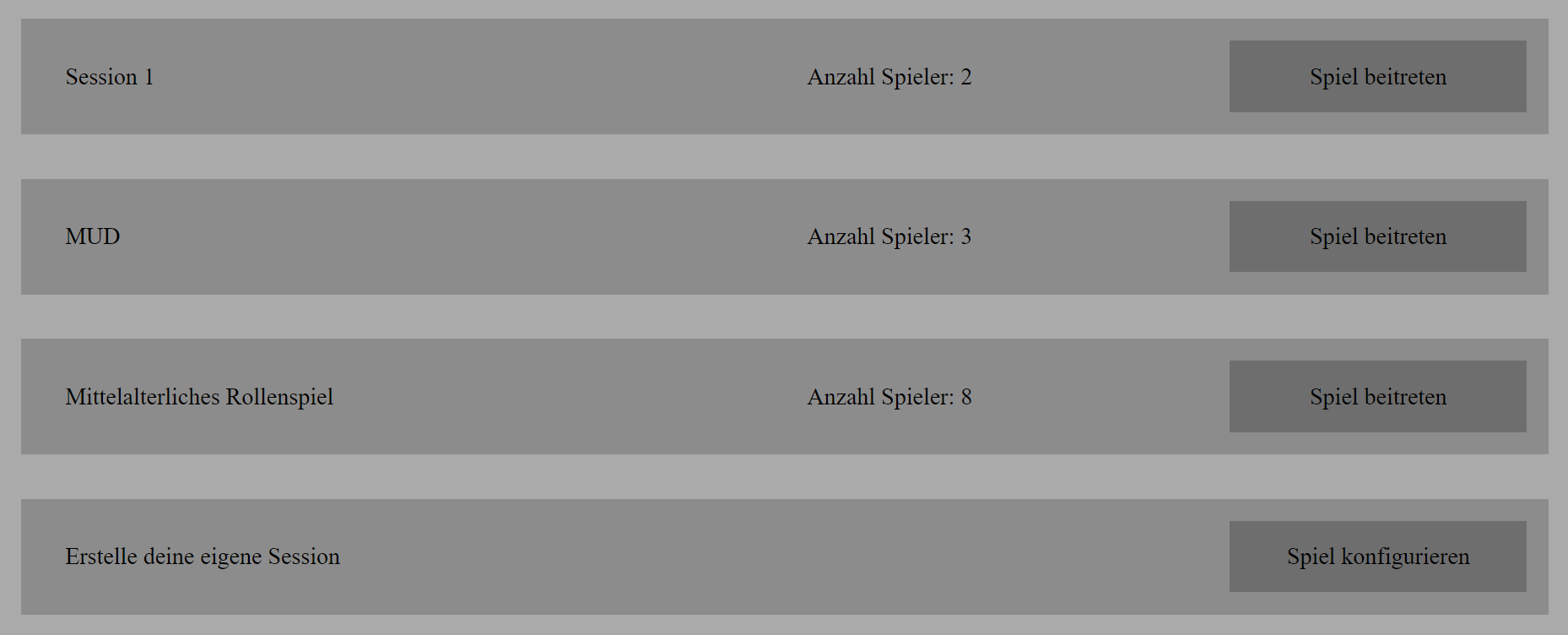
Das Backend beinhaltet die serverseitige Verarbeitung der Daten, die vom Frontend geliefert werden. Es wird in Java geschrieben.

Den Aufbau der Anwendung kann man anhand des UML-Diagrammes erkennen.

Der Funktionsumfang wird im Lastenheft beschrieben.

# Benutzerführung

Der Benutzer kann sich mit E-Mailadresse und Passwort registrieren und anmelden. Aktuell ist ein Testbenutzer mit Benutzername *asd* und Passwort *123* angelegt. Sobald die Anmeldung erfolgt ist kann der Benutzer einer Session beitreten oder eine neue Session erstellen. Bei der Erstellung einer neuen Session kann ein Spiel konfiguriert werden. Unten ist eine Ansicht wie dies im Umfang des Prototyps umgesetzt wird.



# Dokumentation der Anwendung

Die Inline-Dokumentation erfolgt im Quellcode. Hier wird jeweils der Autor des Programmstücks sowie eine Versionsnummer und Datum hinterlegt.

Das Backend in Java wird mit Javadoc dokumentiert (siehe <http://www.oracle.com/technetwork/java/javase/documentation/writingdoccomments-137785.html>). Eine Inline-Dokumentation einer Klasse würde wie folgt aussehen:

*/\*\**

*\* @author Tobias Götz*

*\* @version 1.2*

*\* @date 02.04.2020*

*\* Einzeilige Beschreibung der Funktionalität*

*\*/*

**public** **class** **World** {

*// code follows here*

}

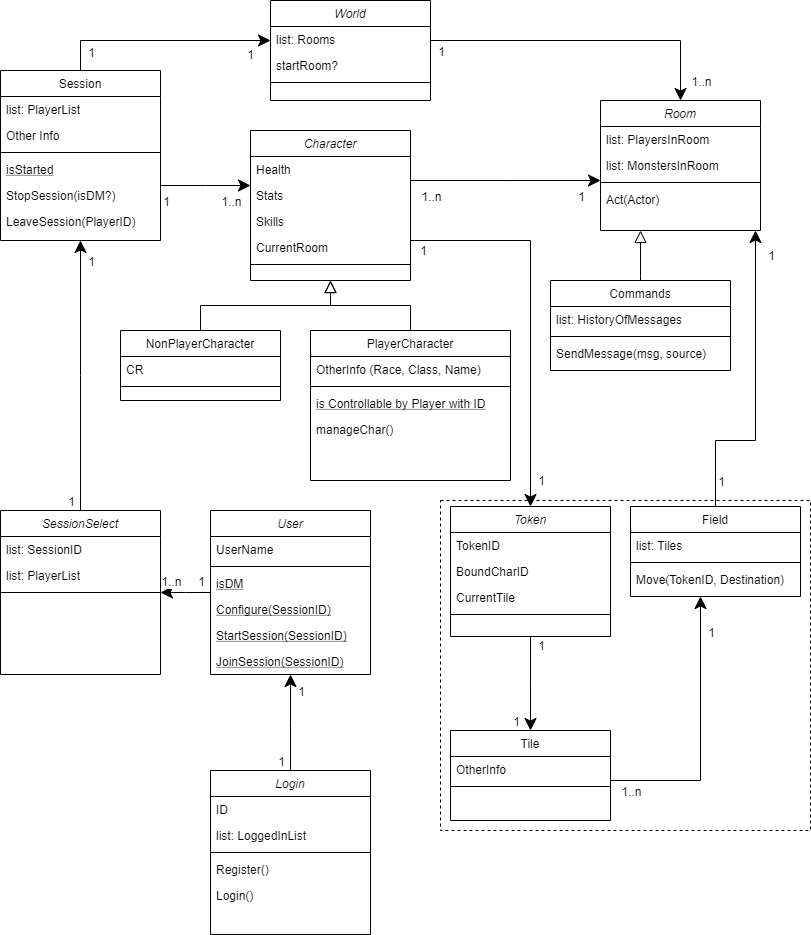
Mithilfe von Javadoc können diese Kommentare zu einer externen Dokumentation exportiert werden. Das Frontend in JavaScript wird in einer möglichst ähnlichen Form dokumentiert. Alles weitere, wie beispielsweise Schnittstellen, wird mit Jira in Confluence dokumentiert. Die Nutzung von Apiary wäre eine mögliche Alternative.

Es wird den Coding Conventions gefolgt, die allgemein anerkannt sind. Für Java sind diese hier zu finden: <https://www.torsten-horn.de/techdocs/java-codingconventions.htm>. Für HTML gelten die folgenden Richtlinien: <https://www.w3schools.com/html/html5_syntax.asp>.

# REST-Schnittstelle

REST bedeutet REpresentational State Transfer. Durch die REST-API können verteilte Systeme (z.B. Server und Client) miteinander kommunizieren. Die Verwendung der REST-API basiert auf bekannte Verfahren, wie HTTP/S, URI, JSON. Der Informationsaustausch zwischen Server und Client ist in diesem Projekt von hoher Bedeutung weshalb diese Schnittstelle gut umgesetzt werden muss. Realisiert wird die REST-Schnittstelle in diesem Projekt mit Jersey und Tomcat. Jersey ist ein Standard um REST-Services in Java zu implementieren. Mit Tomcat wird eine Umgebung bereitgestellt um Webbasierten Java-Code auf einem Webserver auszuführen.

# UML-Diagramm



# Lastenheft

1. **Zielbestimmung**

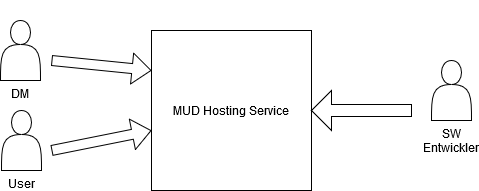
Der Kunde erhält mit dem Produkt eine Software, die zur Installation auf einem Server vorgesehen ist, um Multi User Dungeon Spiele in beliebiger Anzahl parallel zu hosten. Der Server soll je nach benötigter Leistung die Kapazität dynamisch skalieren.

Für die Spielvorbereitung und -durchführung wird ein Dungeon Master benötigt, der die Umgebung manuell konfiguriert.

1. **Produkteinsatz**

Das entwickelte Produkt wird auf dem Markt der Online Spiele vertrieben werden. Es zielt auf die Sparte der Multi User Dungeon Spiele ab. Die Software muss eine Möglichkeit bieten, beliebig viele Spiele parallel hosten zu können und entsprechend der benötigten Serverbeanspruchung Kapazitäten zur Verfügung stellen.

1. **Produktübersicht**



1. **Produktfunktionen**

|  |  |
| --- | --- |
| /LF10/ | Geschäftsprozess: Spiel konfigurieren |
|  | Akteur: Dungeon Master |
|  | Beschreibung: Das Spiel soll nach Wünschen des Dungeon Masters gestaltet werden. Sowohl das Thema des Spiels, die Räumlichkeiten du Konfigurationen der Avatare in Klassen, Rassen und die Ausstattung mit Equipment sollen möglich sein. Die Konfigurationen dienen der Klassifizierung der Avatare sowie der Steuerung ihrer Fähigkeiten. Die Raumanzahl und -zuordnung kann konfiguriert werden. Die Räume können mit Objekten und Nicht-Spieler-Charakteren dekoriert werden, zusätzlich können dem Raum Aktionen zugewiesen werden. |
| /LF20/ | Geschäftsprozess: Registrierung auf Plattform |
|  | Akteur: Spieler, Dungeon Master |
|  | Beschreibung: Der Spieler / Dungeon Master muss sich auf der Plattform registrieren können. Hierzu werden folgende Informationen benötigt:   * Vor- und Nachname * Valide E-Mail-Adresse * Accountname   Nach erfolgter Registrierung erhält der Spieler / Dungeon Master eine E-Mail mit einem Bestätigungslink. Wenn er auf diesen klickt, öffnet sich eine neue Seite, auf der ein Passwort vergeben werden kann. |
| /LF30/ | Geschäftsprozess: Anmeldung auf Plattform |
|  | Akteur: User |
|  | Beschreibung: Der User meldet sich auf der Plattform an. Er kann jetzt entweder an einem Spiel teilnehmen, dass ihm in der Übersicht gezeigt wird und agiert als Spieler oder selbst ein Spiel gemäß /LF10/ kreieren, womit er die Rolle als Dungeon Master einnimmt. |
| /LF40/ | Geschäftsprozess: Avatar konfigurieren |
|  | Akteur: Spieler |
|  | Beschreibung: Der Spieler kann nach Beitritt eines Spiels seinen Avatar im Rahmen des vom DM erlaubten Maßes konfigurieren. Ein Spieler kann sich auch mehrere Avatare konfigurieren, mit denen er am Spiel teilnehmen kann.  Der Spieler kann an mehreren Spielen gleichzeitig teilnehmen, muss allerdings für jedes Spiel einen anderen Avatar haben. |
| /LF50/ | Geschäftsprozess: Anmeldung bzw. Abmeldung eines Spiels |
|  | Akteur: Spieler |
|  | Beschreibung: Der Spieler kann sich zu einem verfügbaren Spiel anmelden und sich von einem Spiel, bei dem er angemeldet ist, wieder abmelden. Bei Anmeldung zu einem Spiel erhält er die Möglichkeit, seinen Avatar im Rahmen der vom DM zugelassenen Möglichkeiten zu konfigurieren und kann den Startraum betreten. Meldet der Spieler sich wieder ab, werden seine Daten gespeichert und bei der nächsten Anmeldung bereitgestellt. |
| /LF60/ | Geschäftsprozess: Spielverlauf |
|  | Akteur: Dungeon Master, Spieler |
|  | Beschreibung: Im Spielverlauf kann der Spieler Aktionen, die vom Dungeon Master definiert wurden, durchführen. Er kann auf diese mit Interaktion reagieren oder sich einer anderen Aktion zuwenden. Während des Spielverlaufes können Spieler (als Gruppe oder einzeln) mit dem Dungeon Master interagieren. Der Spieler kann Räume (nach Definiton des Dungeon Masters) betreten. |
| /LF70/ | Geschäftsprozess: Mit anderen Spielern chatten |
|  | Akteur: Spieler, Dungeon Master |
|  | Beschreibung: Jeder Spieler im Raum sowie der Dungeon Master können über einen öffentlichen Chat kommunizieren. Jeder Spieler, der im gleichen Raum ist, kann die Unterhaltung sehen. Zusätzlich können Spieler sich privat ansprechen, also „flüstern“, wenn der Flüsterpartner im gleichen Raum befindlich ist.  Der Dungeon Master kann von jedem Raum aus am Chat teilnehmen und angeflüstert werden. Zusätzlich kann der Dungeon Master den Namen eines Monsters/ anderen Spielers annehmen, um in seinem Namen zu reden. |
| /LF80/ | Geschäftsprozess: Mit anderen Spielern flüstern |
|  | Akteur: Spieler, Dungeon Master |
|  | Beschreibung: Jeder Spieler kann einen anderen Spieler oder den Dungeon Master anflüstern, was bedeutet, dass die Nachricht nur von dem Empfänger gesehen werden kann. Der Dungeon Master kann Namen von Monstern/ anderen Spielern verwenden, um ihre Rolle beim Flüstern einzunehmen. |
| /LF90/ | Geschäftsprozess: Spielgeschehen steuern |
|  | Akteur: Dungeon Master |
|  | Beschreibung: Der Dungeon Master kann aktiv in das Spielgeschehen eingreifen und es steuern. Dazu zählen   * Aktionen in jedem Raum und zu jeder Zeit ausführen * Nicht Spieler Charaktere steuern * Schwierigkeitsgrad senken oder erhöhen * Spieler aus dem Spiel entfernen * Räume oder Nicht Spieler Charaktere hinzufügen/ entfernen.   Dem Dungeon Master ist es ausdrücklich nicht erlaubt, den Spielern während des laufenden Spielbetriebs andere Rassen zuzuteilen. |
| /LF100/ | Geschäftsprozess: Spiel bewerten |
|  | Akteur: Spieler |
|  | Beschreibung: Ist ein Spiel erfolgreich beendet worden, kann das Spiel bewertet werden. Der Dungeon Master erhält damit eine Punktzahl, die anderen Spielern als Empfehlung für die Teilnahme an seinen Spielen werten kann. |
| /LF110/ | Geschäftsprozess: Konfliktlösung |
|  | Akteur: Spieler, Dungeon Master |
|  | Beschreibung: Kommen während eines aktiven Spieles Konflikte auch, durch die der Spielverlauf ins Stocken gerät, kann entweder der Dungeon Master den zu wählenden Weg vorgeben oder er kann eine Entscheidung in der Gruppe fordern. Per Mehrheitsbeschluss wird das weitere Vorgehen beschlossen und das Spiel entsprechend fortgesetzt. |
| /LF120/ | Geschäftsprozess: Abtritt der Game Master Rolle im Spiel |
|  | Akteur: Spieler, Dungeon Master, Spieler |
|  | Beschreibung: Ein Spiel kann nur gespielt werden, wenn ein Dungeon Master als Game Master anwesend ist. Sollte der Dungeon Master aus dem Spiel austreten, aber die Mitspieler wollen trotzdem weiterspielen, kann er einem anderen Spieler die Rolle des Game Masters zuordnen, der für ihn die Leitung des Spiels übernimmt. |
| /LF130/ | Geschäftsprozess: Teilnahme an privatem Chat |
|  | Akteur: Spieler, Dungeon Master, Spieler |
|  | Beschreibung: Ein Spieler kann einen privaten Chat mit mehreren anderen Spielern starten, die sich im gleichen Raum befinden. Der Dungeon Master kann dem Chat nicht beitreten oder ihn einsehen, aber er kann sehen, welcher Spieler in diesem privaten Chat ist. |

1. **Produktdaten**

|  |  |
| --- | --- |
| /LD10/ | Dungeon Master Daten |
| /LD20/ | Spielerdaten (Laden der Daten im Sekundenbereich) |
| /LD30/ | Konfigurationsdaten |
| /LD40/ | Spielstanddaten |
| /LD50/ | Spieldaten |

1. **Produktleistungen**

|  |  |
| --- | --- |
| /LL10/ | Die Funktion /LF10/ muss gewährleisten, dass eine Beschreibung des Spiels hinterlegt wird |
| /LL20/ | Die Latenzzeiten zum Laden der Daten darf nicht länger als 2-3 Sekunden sein |
| /LL30/ | Die Spiele müssen nicht für mobile Endgeräte im Sinne einer App konzipiert werden, aber soll responsive sein. |
| /LL40/ | Eine Möglichkeit zur Verifikation soll angeboten werden, damit Spieler erst ab einem gewissen Alter Zutritt zum Server erhalten. Die Altersverifikation wird nach State of the Art implementiert. |
| /LL50/ | Eine Begrenzung der Spieleranzahl pro Spiel darf über vom Dungeon Master festgelegt werden. Die Software muss erkennen, wenn der Server aufgrund zu hoher Belastung zu langsam wird und entsprechend Ressourcen zur Verfügung stellen. |
| /LL60/ | Die Spieler erhalten die Möglichkeit, ihren Avatar/ Icons hochzuladen, sodass andere Spieler sie sehen können. |
| /LL70/ | DM kann in seinen eigenen Spielen auch als Spieler auftreten. |

1. **Qualitätsanforderungen**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Produktqualität | Sehr gut | gut | normal | Nicht relevant |
| Skalierbarkeit |  |  | X |  |
| Performance |  | X |  |  |
| Funktionalität |  |  | X |  |
| Benutzbarkeit | X |  |  |  |
| Zuverlässigkeit |  | X |  |  |

1. **Ergänzungen**

* Es wird erwartet, dass die Serverbelastung gegen Nachmittag/ Abend/ Nacht und am Wochenende zunimmt, da die Spieler untertags arbeiten.