



# Recherchebericht zum Thema Multi User Dungeon

## Inhaltsverzeichnis:

1. Allgemeines
2. Themenrelevante Applikationen
3. IT-Spezifika
4. Written Realms

## 1. Allgemeines

Multi-User-Dungeons (MUDs) haben ihren Ursprung in England Ende der 1970er Jahre und sind textbasierte Rollenspiele. Anfänglich liefen die Spiele auf Großrechnern über Terminals (VT100). In den frühen Jahren des Internets wurden sie über einfache Internetverbindungsprotokolle gespielt.<sup>1</sup> Heutzutage laufen MUDs auf Servern, mit denen sich die Spieler über ihren Browser verbinden und so im Spiel miteinander in Interaktion treten können.

Wie in der Papierversion „Dungeons and Dragons“ geht es auch bei MUDs darum, unter Einsatz der eigenen Fantasie eine Geschichte zu erzählen. Spieler haben die Wahl, nachdem sie sich eingeloggt haben, entweder in eine vorgefertigte Spielwelt einzutreten, wie z.B. Mittel Erde von J. R. R. Tolkien oder sich ihre eigene Welt zu erstellen. In dieser werden die Spieler vor unterschiedlichste Herausforderungen gestellt, wie Rätsel zu lösen oder Schätze zu finden. Um diese Aufgaben erfolgreich zu lösen, ist oft die Zusammenarbeit der Spieler notwendig.

Vor dem Eintritt ins Spiel muss sich jeder Spieler einen Charakter, in Anlehnung an die Vorgaben der Welt, erstellen. Mit diesem Charakter (Spielfigur) kann der Spieler zusammen mit anderen Spielern in einen Dungeon eintreten, welcher aus Räumen besteht.

„Das typische MUD beschreibt dem Spieler den Raum oder das Gebiet, in dem er sich befindet, und listet die Objekte, Spieler und non-player characters (NPCs) in diesem Gebiet, sowie alle Ausgänge auf. Um eine Tätigkeit [/Action] auszuführen, gibt der Spieler einen Textbefehl ein, z. B. Apfel nehmen oder Drache angreifen. Die Bewegung in der Spielumgebung erfolgt in der Regel durch Eingabe der Richtung (oder einer Abkürzung davon), in die sich der Spieler bewegen möchte, z. B. würde die Eingabe von Norden oder einfach n/N dazu führen, dass der Spieler das aktuelle Gebiet über den Weg nach Norden verlässt.“<sup>2</sup>

Neben den Spielern muss auch ein Dungeon Master (DM) dem Dungeon beitreten. Dieser agiert als „Schiedsrichter“ und leitet das Abenteuer, d.h. er bestimmt die Ereignisse (z.B. das Erscheinen von feindlichen NPCs...). Die Spieler entscheiden in Echtzeit mit ihren

---

<sup>1</sup> Vgl. [https://de.wikipedia.org/wiki/Multi\\_User\\_Dungeon](https://de.wikipedia.org/wiki/Multi_User_Dungeon), Abruf 20.03.21

<sup>2</sup> Vgl. <https://en.wikipedia.org/wiki/MUD#Gameplay>, Abruf 20.03.21



Charakteren wie sie auf diese vom DM initialisierten Ereignisse reagieren. Nach erfolgreichem Abschluss eines Ereignisses bekommt normalerweise jeder beteiligte Spieler eine Belohnung in Form von Gegenständen und/oder Erfahrungspunkten. Wenn ein Charakter eine bestimmte Menge an Erfahrungspunkten erreicht hat, steigt er in ein höheres Level auf, was wiederum zur Verbesserung seiner Fähigkeiten führt. Die Spieler können mit ihren Charakteren den Dungeon vor Beendung verlassen und sich zu einem anderen Zeitpunkt wieder einloggen, um an der gleichen Stelle mit ihren Charakteren weiterzuspielen.

### Themenspezifische Terminologie

Begriffe	Erklärung
MUD	Ist die Abkürzung für Multi-User Dungeon.
(Spiel)Welt	In einer Welt ist vordefiniert welche Charaktere erstellt werden können.
Dungeon	In Welten kann es verschiedene Dungeons geben die der Spieler mit seinem in der Welt erstellten Charakter betreten kann.
Charakter (/Spielfigur)	Der Charakter ist die Figur, die der Spieler im Spiel erstellt und steuert.
Tätigkeit/Action	Eine durch den Spieler gesteuerte Handlung des Charakters.
Dungeon Master (DM)	Ist die Person, die die Funktionen des Dungeons ausführt.
Spieler	Der Spieler ist die Person, die sich einen Account erstellt.

### Mögliche Leistungsmerkmale einer prototypischen Anwendung

In einer prototypischen Anwendung sollte es dem Spieler möglich sein einen Account zu erstellen, und in diesem einen Charakter zu konfigurieren. Zudem sollte es möglich sein mit diesem Charakter in einen prototypischen Dungeon zu gehen. In dem Dungeon sollte es dem Spieler möglich sein, einfache Funktionen, wie das Chatten oder das Ausführen von einfachen Aktionen, wie das Bewegen durch den Dungeon, durchzuführen. Beim Ausloggen soll der Spielstand gespeichert werden und beim erneuten Einloggen geladen werden.

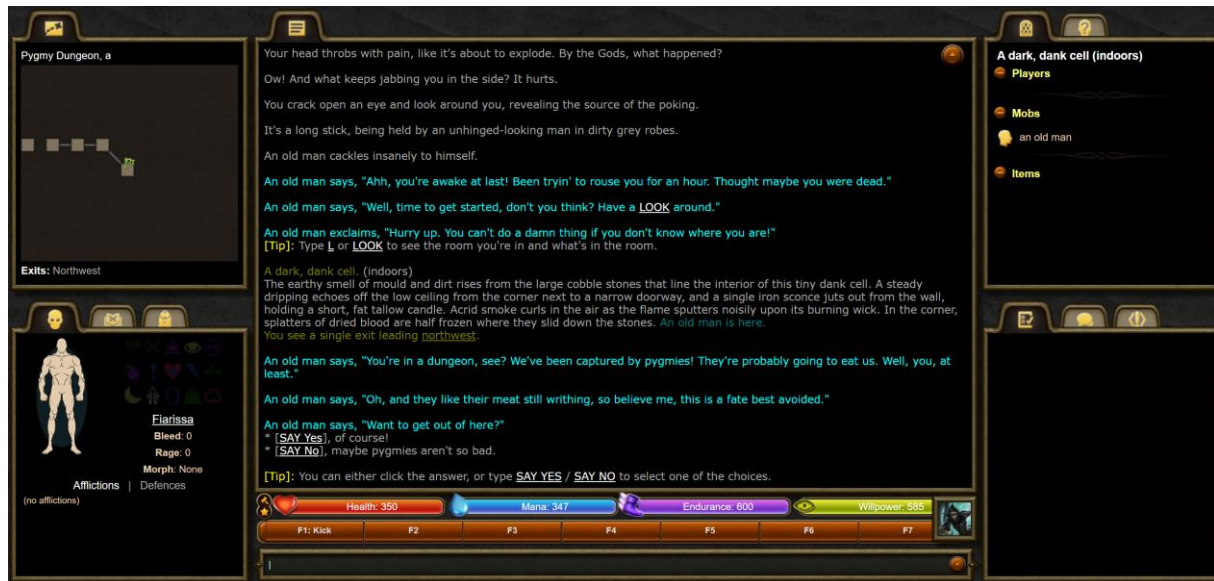
## 2. Themenrelevante Applikationen

### **Achaea:**

<https://www.achaea.com/game-features/>

### Leistungsmerkmale:

Achaea ist eines des populärsten MUD Spiele bis zur heutigen Zeit. Achaea ist ein klassischer Multi User Dungeon in einer Web Umgebung, welcher einen mit textbasierten Ereignissen durch eine Welt führt. Man kann einen Charakter erstellen und zwischen 13 unterschiedlichen Rassen auswählen. Dabei sind manche Rassen an ein bestimmtes Geschlecht gebunden. Hat man eine beliebige Rasse ausgewählt hat man die Wahl zwischen 17 einzigartigen Klassen, von Krieger bis hin zu einem Alchemisten. Danach sucht man sich ein Startgebiet in Form einer Stadt aus und wird in die Geschichte hineingeführt durch eine Art Tutorial. Man gibt dabei Kommandos in eine Textbox ein und interagiert somit mit der Welt. Dabei hat man immer eine Übersicht über den aktuellen Dungeon und sieht die unterschiedlichen Räume, welche man Besuchen kann. Ein weiterer wichtiger Bestandteil ist das Inventar, in welchem der Spieler seine gesammelten Gegenstände lagert.



(Quelle: play.achaea.com)

Im späteren Verlauf des MUD kann der Charakter unterschiedliche Aktionen vornehmen. Beispielsweise kann der Charakter ein Haus oder einen Clan anführen, oder ein Kapitän von einem Schiff werden. Eine weitere Funktion des Spiels ist es Berufe zu erlernen welche dem Charakter Möglichkeiten geben einzigartige Gegenstände herzustellen. Ebenfalls gibt es Spieler-gegen-Spieler-Kämpfe, in welchen Charaktere des MUD gegeneinander antreten können.

Es gibt ebenfalls sogenannte Mikrotransaktionen mit welchen sich der Spieler durch Echtgeld, Währung für das Spiel kaufen kann.

### Technologien:

Das Spiel basiert auf der Rapture Engine. Die Rapture Engine ist ein Framework was dafür entwickelt wurde, um Multi User Dungeon Spiele zu entwerfen.  
(siehe: <https://www.ironrealms.com/rapture/manual/files/Overview-txt.html>)

### **Written Realms:**

<https://writtenrealms.com/about>

### Leistungsmerkmale:

Written Realms ist ein MUD-Konfigurator in welchem eigene Dungeons per UI entworfen und veröffentlicht werden können. Die Dungeons werden demnach von der Community für andere Spieler bereitgestellt. Sobald man sich für einen der Community-Dungeons entschieden hat, kann man einen Charakter erstellen. Die Basisauswahlmöglichkeiten sind Name, Geschlecht, Klasse und Rasse. Ist der Charakter erstellt wird man in den Dungeon weitergeleitet und kann nun die Geschichte spielen.

Weitere Merkmale für diesen Konfigurator sind unter anderem ein Quest System, in welchem man kleinere Aufgaben innerhalb eines Dungeons löst und dafür Belohnungen wie Gegenstände und Erfahrung erhält. Jeder Charakter hat ein Levelsystem, was bedeutet, dass ein Charakter stärker wird, wenn er Monster besiegt oder Aufgaben erfüllt. Der Ablauf des



Dungeons und der Geschichte wird rein von dem Ersteller bestimmt. Der Ersteller ist dafür zuständig Interaktivität durch Gegenstände oder das Platzieren von Monstern herzustellen.

Eine Besonderheit von Writen Realms ist das responsive Design. Hierdurch kann man ebenfalls auf allen Geräten mit einer Internetverbindung die Dungeons erstellen und spielen.

#### Technologien:

Für das Backend wird Python als Programmiersprache verwendet. Die Live Game Data wird in einer Redis Datenbank abgespeichert. Dies beinhaltet Daten wie Lebenspunkte und Gegenstände von Spielern. Erstellte Dungeons und statische Informationen werden in einer Postgres Datenbank festgehalten, welche mit dem Django Framework und der RESTful API abgefragt werden können. Für die Datenübertragung wird das WebSockets Protokoll verwendet. Das Frontend besteht aus Vue.js, Backbone und Sass.

#### **Evennia:**

<https://www.evennia.com/>  
<https://www.evennia.com/docs/latest/index.html>

#### Leistungsmerkmale:

Evennia ist ein MUD Konfigurator, welcher ein Grundgerüst für das Erstellen von MUDs bereitstellt. Hier werden Code Teile bereitgestellt, welche modular erweiterbar sind und für das eigene Spiel angepasst werden können. Es liefert aber auch Grundfunktionen wie das Erstellen von Dungeons und Items, interagieren mit anderen Spielern und das Erstellen von Monstern/Charakteren.

#### Technologien:

Wie bei Writen Realms wird für das Backend als Programmiersprache Python verwendet. Der Webclient basiert auf HTML5 und kommuniziert mit einem Webserver über WebSockets. Verwendet man einen klassischen MUD-Client, so werden die Protokolle Telnet oder SSH benutzt. Für die Sicherung der Daten werden mehrere Datenbanken unterstützt. Hierzu zählen unter anderem Sqlite3, PostgreSQL oder MySQL.

#### **DikuEdit:**

<http://galacticsoft.github.io/DikuEdit/>

#### Leistungsmerkmale:

DikuEdit bietet die Erstellung von Dungeons an, welche man mit Hilfe von einem grafischen Editor genau definieren kann. Neben der Erstellung von Räumen, sind auch Gegenstände und Non Player Characters zur freien Konfiguration verfügbar. Man hat in diesem Editor nahezu endlose Möglichkeiten wie man seinen MUD aufbauen will, da jeder einzelne Aspekt erstellbar und modifizierbar ist. Trotz der grafischen Darstellung der Räume ist die Erzählung der MUDs textbasiert. Die erstellten Dungeons sind im selben Programm spielbar.

Hierbei ist vor allem der Umfang der Konfiguration interessant.

#### Technologien:



Das Produkt wurde in C++ geschrieben und verwendet Microsoft Foundation Classes für die vereinfachte Entwicklung einer Objektorientierten Software in Zusammenarbeit mit einer grafischen Oberfläche.

### 3. IT-Spezifika

Damit ein MUD erfolgreich umgesetzt werden kann werden verschiedene Technologien und Applikationen benötigt. Nur ein reibungsloses Zusammenspielen der Komponenten garantiert eine erfolgreiche Entwicklung und später eine positive Nutzererfahrung.

#### Programmiersprache/Hochsprache:

Bei der Entwicklung eines MUD kommen viele Entwickler nicht um die Programmiersprache Python drum herum. Viele voran gegangenen MUDs wurden schon in Python entwickelt und dementsprechend ist die Dokumentation im Internet sehr ausführlich. Python ist eine Hochsprache welche dynamisch typisiert und ist mit dem Ziel, eine gute Lesbarkeit zu haben, entwickelt worden. Die Programmiersprache für Front-/Backend bildet die Schnittstelle zwischen Mensch und Maschine und sollte somit sehr sorgfältig ausgewählt werden. Bekannte Vertreter der MUD-Szene wie Writen Realms und Evennia wurden und werden immer noch in Python entwickelt.

Eine Alternative zu Python wäre Java, eine objektorientierte Programmiersprache welche vom MUD-Client Mudslinger genutzt wird.

#### Datenbank:

Eine Datenbank ist für ein MUD unerlässlich. Sie garantiert, dass sämtlichen Charaktere, Objekte, Umgebungen, also alle Eigenschaften, die ein Spieler/Dungeon besitzt permanent zu speichern und für mehrere Clients gleichzeitig zur Verfügung zu stellen. Für ein MUD stehen hier zwei verschiedene Datenbankstrukturen zur Verfügung. Eine relationale SQL-Datenbank kann hier zum Beispiel in Kombination mit dem Django-Webframework benutzt werden sofern das entsprechende Backend vorhanden ist. Für das Datenbanksystem wie MySQL besitzt Django eine integrierte objektrelationale Abbildung. Ein relationales Datenbanksystem speichert Daten anhand von Tupeln innerhalb einer Tabelle sowie anhand von wohldefinierten Relationen zwischen den Tabellen. Eine andere Möglichkeit wäre die Nutzung eines In-Memory Datenbanksystems wie Redis. Dieses speichert die Daten nicht relational in Tabellen, sondern als sog. Key-Value Pairs direkt im Arbeitsspeicher. Somit kann schneller auf die Daten zugegriffen werden, allerdings muss auch entsprechend viel Arbeitsspeicher zur Verfügung stehen und die Daten werden weniger effizient abgespeichert als in relationalen Tabellen. Außerdem muss zusätzlich für Persistenz der Daten gesorgt werden. Wenn Arbeitsspeicher nicht mehr unter Strom steht gehen die Daten verloren. Hierfür müssen Backupbatterien für den Arbeitsspeicher verwendet werden oder Snapshots mit entsprechenden Änderungsprotokollen. SQL-Lite.



### Kommunikationsprotokolle:

Um eine reibungslose Kommunikation zwischen Client und Server zu gewährleisten werden Kommunikationsprotokolle benötigt, die den Datentransfer koordinieren. Bekannte Vertreter wie Mudslinger benutzen hier Telnet, ein altes Client/Server-Protokoll welches über die TCP-Ebene kommuniziert. Telnet wird native von den gängigsten Betriebssystemen unterstützt allerdings kommuniziert das Protokoll unverschlüsselt. Selbst das Kennwort für die Verbindungsinitialisierung wird im Klartext übertragen. Ein alternatives Netzwerkprotokoll hierfür wäre SSH (Secure Shell). Ein kryptografisches Netzwerkprotokoll welches ebenfalls über die TCP-Ebene kommuniziert. Das MUD-Template Evennia ist zum Beispiel inzwischen von Telnet auf SSH umgestiegen. Eine weitere Möglichkeit für die Verbindung zwischen Client und Server wäre die Verwendung von Websockets. Diese wurden speziell für Webanwendung entwickelt, da eine asynchrone Datenübertragung zwischen Client und Server möglich ist. Der Client hat die Möglichkeit nach einer Verbindung zum Server dauerhaft zu öffnen die der Server nach Belieben nutzen kann um den Client mit Daten zu versorgen. Des Weiteren unterstützen Websockets im URI-Schema „wss“ verschlüsselte Verbindungen. Für einen MUD würde sich die ZeroMQ-Bibliothek als Websocket anbieten das sie speziell dafür ausgelegt wurde auf mehreren Clients gleichzeitig ausgeführt zu werden.

### Webserver:

Ein Webserver ist für die Koordination der Dateien zwischen Client und Datenbank zuständig. Er empfängt ein Request vom Client in Form eines „POST“ oder „GET“ und bearbeitet diesen und gibt die Antwort in Form von HTML und Dateien wie PNG etc. zurück. Ein Webserver hat die Möglichkeit eine Website an mehrere Clients gleichzeitig auszuliefern. Die Anzahl an Clients hängt hier von den Kapazitäten des Webserver ab.

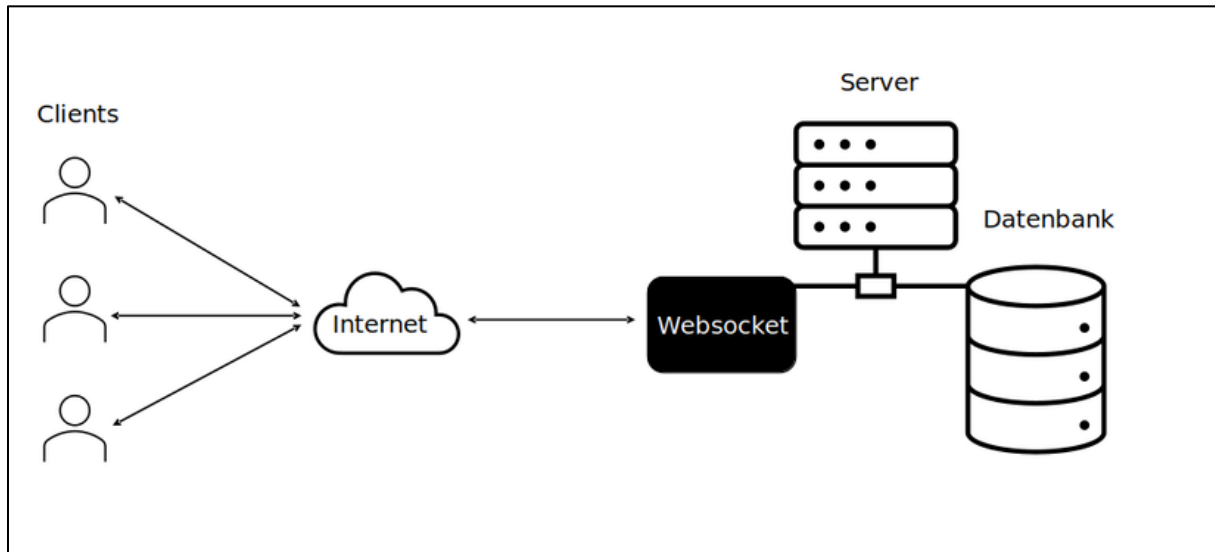
### Webframework:

Das Webframework ist die Software auf dem Webserver die die Entwicklung der Webanwendungen ermöglicht. Solche Webframeworks stellen Grundstrukturen zur Verfügung, die das Entwickeln einfacher schlanker und effektiver machen. Strukturen können wiederverwendet werden und mit wenig Aufwand können neue Strukturen erzeugt werden. Eins der bekanntesten Frameworks ist Angular.

### Frontend:

Als Frontend wird das auf dem Client laufende Programm genannt. Es stellt bei Client-Server Anwendungen die Schnittstelle um mit dem Server zu kommunizieren. Das Programm wird in Form von HTML und CSS im Browser als Website angezeigt. Zur optischen Aufwertung der Seite und um Interaktionen besser gestalten zu können werden häufig JavaScript-Bibliotheken verwendet, die Funktionen bereitstellen. Bekannte Vertreter sind hier JQuery.js und Vue.js.





*Beispielhafter Kommunikationsfluss zwischen Clients und Server mit Websocket als Kommunikationsprotokoll für dauerhaften Datenaustausch.*

## 4. Written Realms

Das Multi-User-Dungeon Written Realms qualifiziert sich insbesondere durch seine saubere und klar definierte Oberfläche, welche auch auf mobilen Geräten äußerst optimiert wirkt. Vor allem dies lässt diese Applikation von anderen auf dem Markt herausstechen.

Written Relams besitzt aber auch die typischen Funktionen eines MUDs. Dazu gehören eine beispielbare Karte, zu bekämpfende Gegner, eine Chatfunktion zur Interaktion untereinander und die Charaktererstellung mit dazugehörigem Levelsystem.

Zunächst ist es dem Nutzer möglich auch ohne Authentifizierung ein Tutorial zu spielen, welches ihm die Grundkenntnisse eines MUDs näherbringt. Ein Dungeon besteht bei Written Realms aus einer Map, welche der Nutzer durch Verwendung eines Terminals erkunden kann.

Dieses Terminal hat einen Textausgabe- und einen Texteingabebereich. Durch diese kann der Nutzer mit dem Spielgeschehen interagieren. Entweder indem ihm die Welt durch Text beschrieben wird oder indem er durch vordefinierte Befehle Aktionen ausführt. Wenn der Nutzer nun beim Erkunden des Tutorial-Dungeons auf ein Monster trifft erscheint ein Kampfsystem, welches vollautomatisch einen Kampf ausführt. Die Austragung eines Kampfes geschieht allerdings in einem separaten PopUp, welches durch den Nutzer mit dem Button „Fliehen“ beendet werden kann.

Written Realms größte Stärke im Vergleich zu dem aktuellen Marktangebot ist die Ausführlichkeit ihres Dungeon Konfigurators, welcher durch eine Registrierung auf der Seite freigeschaltet wird. In diesem ist es möglich sich ein eigenes Dungeon zu erstellen. Dies ist realisiert durch eine frei konfigurierbare Karte und durch das Einsetzen von Monstern, Items oder Spezial Interaktionen, welche außerhalb des eigentlichen Spielgeschehens stattfinden.



Zusätzlich ist es dem Dungeonersteller möglich Quests in Form von Aufträgen an den Spieler zu stellen, welcher dann nach erfolgreichem Abschluss des Auftrags eine Belohnung erhält.

Durch genau diese Funktion macht es Writen Realms allen Spielern möglich eine eigene Geschichte zu erstellen und diese auch im Communitybereich zu veröffentlichen wo anschließend jeder andere Spieler freien Zugang zu diesem Dungeon erhält.

Im Folgenden werden einige UML-Diagramme bereitgestellt, welche die Use Cases von Writen Realms genauer beschreiben.

Siehe **UseCase\_WritenRealms\_Übersicht.pdf**

Zum einen hat der User die Möglichkeit ein Dungeon zu spielen und zum anderen ein Dungeon zu erstellen. Diese beiden Optionen werden in den folgenden Diagrammen weiter ausgeführt. In diesem Diagramm ist der Prozess, welcher um das erstellen und spielen herum geschieht genauer erfasst.

Zunächst betrachten wir den Punkt des Spielens. Hierfür müssten zunächst die relevanten Daten eines Dungeons durch den Server geladen und gegebenenfalls visualisiert werden. Zu den zuladenden Inhalten zählen beispielsweise die relevanten Items oder die Monster, welche in dem Dungeon verwendet werden. Zu visualisieren ist beispielsweise die Karte auf welcher der User sich befindet.

Wenn der Nutzer nun schon einen Charakter innerhalb dieses Dungeons besitzt können alle für diesen Charakter relevanten Daten geladen werden. Zu diesen zählen unter anderem die Items des Charakters und der letzte Standort dessen. In dem Fall, dass der Nutzer noch keinen Charakter besitzt oder er einen neuen spielen möchte muss ein neuer erstellt werden. Sobald dies geschehen ist müssen die entsprechenden Daten innerhalb der Datenbank des Servers festgehalten werden.

Falls der Nutzer nun ein Dungeon erstellen möchte müssen innerhalb der dafür vorgesehenen Funktion auch Daten gespeichert und in Datenbank geschrieben werden.

Siehe **UseCase\_WritenRealms\_CreateDungeon.pdf**

Wenn nun ein Spieler ein Dungeon erstellen möchte kann man dies grob in 2 Hauptaspekte unterteilen. Einerseits die Erstellung einer Karte und andererseits die Erstellung von Entitys.

Bei der Erstellung einer Karte ist es notwendig zumindest einen Raum zu erstellen. Dieser Raum kann wahlweise eine Beschreibung erhalten und einer Zone zugewiesen werden. Zonen sind ein Tool von Writen Realms zur Verwaltung von verschiedenen Räumen.

Wenn nun Entitys erstellt werden sollen klassifizieren sich diese in Monster und in Items. Diese beiden haben jeweils gewisse Untereigenschaften, welche definiert werden müssen. Nachdem ein Entity erstellt wurde kann es einem Raum zugewiesen werden indem man dann mit ihm interagieren kann.

Sobald ein Entity innerhalb eines Dugeons erstellt wurde wird dieses zusätzlich zu den Dungeondaten innerhalb der Datenbank festgehalten.

Siehe **UseCase\_WritenRealms\_PlayDungeon.pdf**





Um ein Dungeon (außerhalb des Tutorials) spielen zu können muss sich der Nutzer bei dem Server anmelden die geschieht über einen Authentifizierungsmechanismus, welcher die entsprechenden Daten am Server anfragt.

Nun kann der Nutzer ein ausgewähltes Dungeon spielen. Hierfür werden dann die entsprechenden Daten des Charakters abgefragt und für das entsprechende Dungeon verwendet. Innerhalb dieses Dungeons ist es dann dem Nutzer möglich durch einen speziellen Befehl in den Chat des Dungeons zu schreiben. Außerdem gibt es die Möglichkeit, jeweils mit spezifizierten Befehlen, durch Räume zu navigieren, mit Monstern oder anderen Mobs zu interagieren oder mit Items zu interagieren.

Diese jeweilige Tätigkeit muss dann im spezifischen auch in die Datenbank geschrieben werden.

Zuletzt noch ein Komponentendiagramm, welches eine ungefähre Interaktion zwischen den einzelnen Komponenten von Writen Realms darstellt.

Siehe **Komponentendiagramm\_WritenRealms.pdf**

Der Nutzer interagiert mit dem Interface eines Webclients, welches wiederum mit einem eigenen Web Server via Antwort und Anfrage kommuniziert.

Hinter diesem Webserver befindet sich eine Datenbank mit variablen Daten wie beispielsweise aktuellen Spieldaten wie der Standpunkt des Spielers.

Es wird jedoch auch mithilfe einer RESTful API auf eine statische Datenbank zugegriffen, in der Dinge wie Nutzerdaten oder ähnliches gespeichert sind.