

Designbeschreibung



Version 1.0

Thomas Zimmermann

Dokumentenhistorie

Datum	Version	Beschreibung	Autor
26.03.2021	Version 1.0	Erstellung des Dokuments	Thomas Zimmermann

Inhaltsverzeichnis

1. Allgemeines.....	4
2. Produktübersicht	4
2.1. Äußere Funktionsmerkmale	4
3. Grundsätzliche Struktur und Entwurfsentscheidungen.....	5
3.1. Anwendung	5
3.2. Pakete und Komponenten	6

1. Allgemeines

Diese Designbeschreibung enthält alle wichtigen Informationen zu den Struktur- und Entwurfsprinzipien der Software.

Dem Kunden soll durch den Einsatz des Produktes die Möglichkeit gegeben werden einen Server zu hosten welcher ein Multi-User-Dungeon Spiel und Konfigurator für seine Nutzer bereitstellt.

Dieses Produkt ist ein solcher Konfigurator und bietet dem Kunden eine Möglichkeit Dungeons möglichst frei zu gestalten und seine Kreativität umzusetzen. Die konfigurierten Dungeons können von anderen Nutzern nach der Erstellung und Veröffentlichung gespielt werden.

Das Produkt basiert auf einer Web-Architektur und nutzt WebSockets sowie http-Requests für die bidirektionale Übermittlung von Daten von der Web-Client/-Server-Architektur zum Anwendungs-Server. Der Anwendungs-Server, welcher für die Steuerung von Geschäftsprozessen zuständig ist, ist in Python verfasst und bietet ebenfalls eine Anbindung an den Daten-Server in Form einer MySQL-Datenbank.

2. Produktübersicht

2.1. Äußere Funktionsmerkmale

Dieses Produkt ist eine Browser Anwendung, welche eine grafische Oberfläche bietet.

Bevor der Nutzer mit der grafischen Oberfläche interagieren kann, muss er sich registrieren. Hat der Nutzer sich bereits registriert, muss er sich nur einloggen. Der Nutzer kann jederzeit sein Passwort zurücksetzen, oder seinen Account unwiderruflich löschen.

Sobald der Nutzer eingeloggt ist, hat er die Möglichkeit sich einen Dungeon auszusuchen, indem er eine Liste durchsucht oder nach bestimmten Kriterien filtert.

Zusätzlich kann ein Nutzer neue Dungeons erstellen. Hierbei kann er die Größe des Dungeons festlegen und beliebig viele Räume hinzufügen. Neben der Kartenkonfiguration hat der Nutzer die Möglichkeit, Aspekte wie beispielsweise Klassen, Rassen, Items und NPCs zu erstellen. Ist der Dungeon erstellt, so kann der Nutzer den Dungeon speichern und anschließend veröffentlichen. Dabei kann er eine Zugangsbeschränkung festlegen, wobei er hierbei die Auswahl zwischen einem öffentlichen, beziehungsweise einem privaten Dungeon hat. Öffentlich bedeutet, dass jeder Nutzer beitreten kann, sofern die maximale Spieleranzahl nicht erreicht ist. Privat legt fest, dass der Dungeon Master bestimmen kann, ob ein Nutzer beitreten kann oder nicht. Dies geschieht über eine Beitrittsanfrage.

Jeder Nutzer kann seine selbst erstellten Dungeons kopieren und gegebenenfalls bearbeiten, oder löschen. Der Dungeon Master kann dies auch zur Laufzeit des Spiels, was zur Folge hat, dass vorhandene Charaktere und ihre Daten (Klasse, Rasse, Beschreibung des Charakters und Items) unwiderruflich gelöscht werden.

Sobald sich ein Nutzer für einen Dungeon entschieden hat, muss er sich zuerst einen Charakter konfigurieren. Dabei muss er sich eine definierte Klasse, Rasse aussuchen sowie einen Namen und eine Beschreibung zu seinem Charakter hinterlegen. Wenn der Nutzer das Spiel verlässt, kann er danach immer wieder beitreten, ohne sich einen neuen Charakter anzulegen und wird an der zuletzt hinterlegten Position in das Spiel gelassen. Fortan nimmt er aktiv als Spieler teil und wird von dem Dungeon Master durch das Geschehen geführt. Der Spieler kann durch vordefinierte Befehle, die er in die Konsole eingibt, bestimmte Aktionen ausführen. Unter diesen Befehlen gibt es bestimmte Basisaktionen, welche in jedem Spiel verfügbar sind.

Zu Basisaktionen zählen Befehle, wie beispielsweise sich im Raum umzuschauen, den Raum in die vier Himmelsrichtungen zu verlassen, sofern ein Raum in dieser Richtung vorhanden ist. Spieler können ebenfalls Gegenstände aufnehmen, welche sie in einem Inventar ihres Charakters lagern können.

Eine weitere wichtige Basisaktion ist das Chatten. Der Spieler hat hierbei drei Möglichkeiten, um zu kommunizieren:

1. Innerhalb eines Raumes mit allen lokalen Spielern.
2. Mit dem Dungeon Master, unabhängig von der Position.
3. Mit einem spezifischen Spieler privat, welcher sich im selben Raum befindet.

Wenn eine Aktion ausgeführt wird, führt dies zu einer Benachrichtigung bei dem Dungeon Master, welche er beantworten muss, damit die Aktion als abgeschlossen angesehen wird. Das Chatten ist von dieser Regelung ausgeschlossen.

Die Antwort des Dungeon Masters beinhaltet eine Beschreibung der Ausgangssituation und kann die Lebenspunkte des Spielers beeinflussen. Hierbei kann der Dungeon Master Aspekte des Charakters (Klasse, Rasse, Beschreibung des Charakters und Items) in Betracht ziehen, um den Ausgang zu spezifizieren. Wenn die Lebenspunkte eines Charakters nach einer Aktion auf null fallen sollten, so wird der Charakter aus dem Dungeon gelöscht und der Spieler muss sich einen neuen Charakter anlegen.

Wenn der Dungeon Master das Spiel verlässt wird das Geschehen unterbrochen. Dies bedeutet, dass dieses pausiert wird und die Spieler innerhalb des Dungeons keine Aktionen mehr ausführen können. Ein Dungeon Master kann nach Belieben in ein Spiel wiedereintreten. Der Wiedereintritt des Dungeon Masters hebt jegliche Pausierung des Spielgeschehens auf. Damit der Dungeon Master mit den Spielern direkt kommunizieren kann, hat er die Möglichkeit Spieler privat über eine Chatfunktion anzuschreiben. Darüber hinaus kann er eine Rundnachricht an alle Spieler in einem Spiel senden.

3. Grundsätzliche Struktur und Entwurfsentscheidungen

3.1. Anwendung

Die Anwendung teilt sich in Frontend, Backend und eine Datenbankanbindung auf. Für das Frontend wird Angular benutzt. Angular ist ein mächtiges Framework mit einem guten Fundament, welches stetig weiterentwickelt wird. Daher ist es leicht einzelne Komponenten umzusetzen. Angular folgt dem Prinzip eines MVVM Design Pattern und ist aufgrund dessen ideal, um eine Oberfläche ordentlich zu strukturieren. Komponenten, die mit Angular erstellt werden, können aufgrund eines eingebauten Unittests Modul einfach getestet werden. Angular ist ebenfalls aufgrund der Modularität hervorragend zu warten. Die Interaktivität in Angular ist durch Typescript gegeben.

Das Backend besteht aus einem Python Anwendungsserver. Das Frontend kommuniziert mit diesem Server mit Hilfe von WebSockets. WebSockets ist ein Kommunikationsprotokoll, welches ermöglicht interaktive und asynchrone Datenübertragung zu implementieren. Dies ist vor allem bei dem Durchführen eines Dungeons essenziell, da man hier den Vorteil hat, da man keine Intervalle für neue Abfragen einbauen muss im Gegensatz zu einem Standard http-Request.

Nichtsdestotrotz werden auch http-Requests bei dem Anmelde/Registrierungsprozess verwendet, oder für allgemeine Abfragen, welche das Spielgeschehen nicht beeinflussen. Das Python Backend beinhaltet die Steuerung der Geschäftsprozesse und bietet dem Nutzer alle Funktionen, die er für das Spielen benötigt. Python wird aufgrund der leichten Erlernbarkeit und der hohen Flexibilität benutzt. Darüber hinaus arbeitet Python aufgrund einer ODBC-Schnittstelle hervorragend mit der gewählten Datenbank MySQL zusammen.

Damit die Daten eines Nutzers alle gesichert werden, wird die relationale Datenbank MySQL benutzt. Relationale Datenbanken haben den Vorteil, dass sie exakt strukturiert und definiert sind. Ebenfalls ist jede Datenmenge konsistent, was heißt, dass es keine Anomalien geben kann. Falls benötigt, ist das Modifizieren der einzelnen Tabellenstrukturen einfach gestaltet. Dies ist vor allem hilfreich, wenn weitere Funktionen im Anwendungsserver implementiert oder geändert werden sollen, die eine Datenanbindung benötigen.

3.2. Pakete und Komponenten

Das Frontend besteht aus mehreren Sub Pages, welche sich in Angular in einzelne Komponenten aufteilen. Jede Komponente besitzt einen eigenen Unittest.

Im Backend gibt es grob zwei unterschiedliche Pakete. Ein Paket enthält die Logik für die Kommunikation zwischen Front- und Backend. Dieses Paket soll verschiedene Anfragen verarbeiten und intern weiterleiten, beziehungsweise gewisse Unterklassen aufrufen.

Ein weiteres Paket enthält die Anbindung an die Datenbank und verarbeitet die Datenanfragen, die von dem Web-Client getätigt werden. Die Anbindung wird durch ein Modul der MySQL Bibliothek geregelt.