

Planung der Implementation (V1.3)



Dokumenteninformation

Erstellt von: Leon Daskov

Bearbeitet von:

Dateiname: Implementierungsplanung_V1.3.pdf

Versionierung

Version	Datum	Bearbeiter	Änderung	Stand
1.0	22.04.2021	Leon Daskov	Initiale Erstellung	Abgenommen
1.1	25.04.2021	Leon Daskov	Änderung unter der Erstellung der Datenbanken #12	Abgenommen
1.2	25.04.2021	Leon Daskov	Hinzufügen von groben Zielen für Sprint 2-4	Abgenommen
1.3	26.04.2021	Leon Daskov	Hinzufügen von Testzielen	Abgenommen

Inhalt

1. Das System zur Implementierung	1
1.1 Sprintplanung	1
1.2 Implementierung und Versionskontrolle	1
2. Die 4 Sprints	2
2.1 Sprint 1 [19.04.2021 – 26.04.2021] – UI, Authentifizierung und Mailing-Service	2
2.2 Sprint 2 [26.04.2021-02.05.2021]	4
2.3 Sprint 3 [03.05.2021-09.05.2021]	4
2.4 Sprint 4 [10.05.2021-16.05.2021]	4

1. Das System zur Implementierung

1.1 Sprintplanung

Zum Anfang jeder Woche, trifft sich die Gruppe, um ein Sprintplanning durchzuführen. In diesem Planing werden aus groben Implementationsschritten, Issues, oder auch Stories, definiert.

Diese Stories sollen in dem Zeitabschnitt einer Woche erstellt werden können und als Feature testbar sein. Unittests werden direkt in Kombination mit der Implementierung der Storys erstellt.

Die erstellten Storys werden in unserem GitHub-Project-Board gehalten. In unserem Projektmanagement innerhalb der GitHub-Seite, werden den verschiedenen Stories Prioritäten und IDs zugewiesen. Die IDs werden in diesem Dokument mit der „#Zahl“ Form angegeben.

Die Implementierung passiert in Paaren, um dem Ansatz des Pair-Programming gerecht zu werden. Jede Woche werden neue Programmierpaare zugewiesen, so dass wenn möglich, immer eine Person aus dem Paar, Wissen in den Themenbereich der zu bearbeitenden Stories besitzt.

1.2 Implementierung und Versionskontrolle

Die Implementierung geschieht nach den Codingrichtlinien von Kotlin. Abweichungen von diesen sollen durch Refactorings behoben werden.

Die genutzte Versionskontrolle ist Git, mithilfe der Plattform Github.

Zu jedem Issue oder Pairprogramming-Paar, soll ein Branch erstellt werden, damit unser Master-Branch möglich funktionsfähig bleibt.

Auf diesem Branch wird die Implementierung der Issues von dem jeweiligen Paar geschehen und nach eigenem Ermessen des Fortschritts, iterativ gepusht.

Wenn ein Paar fertig mit ihren Stories, einer Story, welches andere Gruppen benötigen oder eines komplexen und großen Story, soll ein erster Pull-Request erstellt werden. Hierbei wird nach Kontext entweder direkt in den Master-Branch requested oder auf einen Branch einer Gruppe, die den aktuellen Fortschritt der Gruppe benötigt.

Bei einer Pull-Request werden, nach unserer Testdokumentation: Testkonzept_v1.1.pdf, alle Tests ausgeführt und das Ergebnis der Tests wird angezeigt.

Um jedoch einen Pull-Request zu akzeptieren, muss ein weiterer dritter Programmierer, welcher nicht an der Story beteiligt war, den Code reviewen und die Pull-Request akzeptieren.

Insbesondere bei Pulls in den Master-Branch, sollte der Code mit hoher Sorgfalt reviewed werden.



2. Die 4 Sprints

2.1 Sprint 1 [19.04.2021 – 26.04.2021] – UI, Authentifizierung und Mailing-Service

Initialplanung:

Dieser Eröffnungssprint fokussiert sich auf die Erstellung der ersten UI-Komponenten, der User-Authentifizierung und des Mailing-Services.

Testzielsetzung:

Die Tests sollen laut unserer Testkonzept-Dokument umgesetzt werden und spätestens im nächsten Sprint hinzugefügt werden.

In diesem Sprint werden nur Unit Tests implementiert.

Authentifizierung #9 – Leon Daskov & Marius Armbruster

Diese Story umfasst die Backend-Implementation des Authentifizierungssystems.

Die Arbeitspunkte sind aufgeteilt in:

- Abfrage und Ablegen der User-Daten in der Datenbank
- Erstellung des Registrierungsprozesses
- Erstellung des Loginprozesses
- Verschlüsselung des Passworts, um dieses auf der Datenbank sicher zu speichern
- Optional: Erstellung eines Logins basierend auf einem Session-Cookie

Dieses Arbeitspaket wird Zusammenarbeit mit dem Datenbankpaar aus Story #12 erfordern, da datenbankseitige Arbeit notwendig ist.

Mailing-Service #11 – Leon Daskov & Marius Armbruster

Diese Story umfasst die Backend-Implementation des Mailing-Services.

Dem User sollen durch dieses Feature E-Mails gesendet werden, wenn dieser:

- Sich registriert, damit er sich verifizieren kann
- Sein Passwort vergessen hat und auf die „Reset Password“-Funktion zugreift.

Dieses Arbeitspaket erfordert Code aus der Story #9, weshalb bei beiden Stories das Programmierpaar Marius Armbruster und Leon Daskov zugeteilt worden sind.



Erstellung der Spiele- und Konfigurationsdatenbanken #12 – Jakub Cielecki und Johannes Hüttinger

Hier wird das Datenmanagement der Spiele und Konfigurationen behandelt.

Die Datenbanken für die Spieleerstellung, der Speicherdaten der verschiedenen Spieler und die Konfigurationsdateien sollen in Datenbanken gespeichert werden

Hinzugefügt am 25.04.2021:

Die Erstellung der Konfigurationsdatenbank wird in einem späteren Sprint erfolgen, da es noch nicht endgültig geklärt worden ist, wie die Konfigurationen gespeichert werden sollen.

Hinzugefügt am 26.04.2021:

Die Erstellung der Konfigurationsdatenbank wird in Sprint 2 erfolgen, da es endgültig geklärt worden ist, dass die Konfigurationen in einer relationellen Datenbank gespeichert wird.

Erstellung der Angular UI #7 – Linus Kadner & Daniel Scherer

In dieser Story wird die Erstellung der Angular UI in Form einer Nachbildung der UI-Mocks angestrebt.

Folgende UI Seiten werden in diesem Sprint mit hoher Priorität bearbeitet:

- Die SpieleUI, um das MUD spielen zu können
 - o Hierbei wird MUD-Master- und normale Spieler-UI getrennt erstellt
- Die Basis-UI für den Konfigurator
- Die Register und Login UI Komponente
- Das Dashboard, über welches man, die verschiedenen Komponenten erreichen kann

Bereitstellung von Dummydaten für die UI #8 – Jakub Cielecki und Johannes Hüttinger

Es sollen Dummydaten bereitgestellt werden, damit die erstellten UI Komponenten abgenommen werden können und bereits einen Eindruck für das spätere Produkt bieten können.

Aufstellen der Kommunikation zwischen Authentifizierungslogik und UI #10 –

Linus Kadner & Daniel Scherer

Die alleinige Erstellung von UI und Backendlogik der Authentifizierung führen zu keiner funktionierenden Software. Damit diese Software funktioniert, werden in dieser Story die Ergebnisse aus #11 und #7 zusammengeführt damit der Registrierungs- und Login-Prozess funktionieren.



2.2 Sprint 2 [26.04.2021-02.05.2021]

Grobe Zielsetzung:

In diesem Sprint sollen die ersten groben Gameplay und Konfigurator-Features implementiert werden, diese beinhalten:

- Erstellung einer simplen und abgespeckten MUD-Konfiguration
- Erstellung eines MUD-Spiels auf Basis einer Konfiguration
- Die Erstellung von einem Avatar in Koalition mit den Speicherständen
- Die ersten Moderierungsmöglichkeiten eines MUD-Masters in einem MUD
 - o Spawnen von Nicht-Spieler-Charakteren
 - o Teleportieren von Spielern in andere Räumen
- Das Bespielen eines MUDs in seinen fundamentalen Features (Laufen, interagieren, umsehen), somit soll eine ähnliche Feature Varianz, wie in dem Prototypen gegeben werden

2.3 Sprint 3 [03.05.2021-09.05.2021]

Grobe Zielsetzung:

In diesem Sprint sollen die Gameplay-Features ausgeweitet werden und dem Spiel einen ersten „Spielwert“ verleihen. Auch wird in diesem Sprint dem User und dem MUD-Master endlich möglich sein miteinander zu chatten. Die einzubauenden Features sind unten gelistet:

- Erweiterung des Spieles um Features wie ein sehr simples Kampfsystem
 - o Aktionen wie greife an und nehme Schaden sind möglich
- Erweiterung des Spielers um Stats
 - o Erweiterung des Spielers um Ausrüstung
- Einfügen von Aliases für bestehende „Kommandos/Aktionen“ der Spieler und MUD-Master
- Erweiterung der MUD-Master-Basis-Kommandos
- Erweiterung der Möglichkeiten mit Nicht-Spieler-Charakteren und Items zu interagieren
- Zuweisen von Effekten auf Gegenstände soll möglich sein

2.4 Sprint 4 [10.05.2021-16.05.2021]

Grobe Zielsetzung:

In diesem Sprint soll die Spielqualität gesichert werden und Features, welche nicht unbedingt notwendig sind, um ein simples Spielerlebnis zu bieten, implementiert.

- Erstellung von konfigurierbaren Kommandos (Es werden die vordefinierten Kommandos aneinandergereiht, um komplexere Kommandos möglich zu machen)
 - o Diese werden im Konfigurator konfiguriert und werden dann im Spiel nutzbar sein
- Finale Systemtests, um zu sichern, dass das System als Ganzes funktioniert
- Implementierung von Luxusfeatures

