Inhalt

[1 Einleitung 1](#_Toc29478884)

[2 Projektplan 2](#_Toc29478885)

[2.1 Rahmenbedingungen und Abgrenzung 2](#_Toc29478886)

[2.1.1 Datenbank Schema 2](#_Toc29478887)

[2.1.2 Frontend Seiten 2](#_Toc29478888)

[2.1.3 Sicherheit 2](#_Toc29478889)

[2.1.4 User/Gäste 2](#_Toc29478890)

[2.1.5 Soziale Funktionen 2](#_Toc29478891)

[2.1.6 Spielablauf 2](#_Toc29478892)

[2.1.7 Spiellogik 2](#_Toc29478893)

[2.2 Zeitplan 3](#_Toc29478894)

[3 Realisierung 3](#_Toc29478895)

[3.1 Testkonzept 3](#_Toc29478896)

[3.2 Testprotokoll 3](#_Toc29478897)

[4 Installationsanleitung 4](#_Toc29478898)

[4.1 Voraussetzung 4](#_Toc29478899)

[4.2 Installation 4](#_Toc29478900)

[4.3 Aufsetzten 4](#_Toc29478901)

[5 Anhang 5](#_Toc29478902)

[5.1 Glossar 5](#_Toc29478903)

[5.1.1 Abkürzungen 5](#_Toc29478904)

[5.1.2 Allgemeine und Technische Bezeichnungen 5](#_Toc29478905)

[5.1.3 Projektbezogene Bezeichnungen 5](#_Toc29478906)

[5.2 Arbeitsjournal 5](#_Toc29478907)

[5.3 Verzeichnis 6](#_Toc29478908)

[5.3.1 Grafiken 6](#_Toc29478909)

[5.3.2 Frameworks 6](#_Toc29478910)

# Einleitung

Dieses Dokument umfasst die Realisierungsphase des im Modul 150 umgesetzten Projektes Numerous.

# Projektplan

## Rahmenbedingungen und Abgrenzung

Aus zeitlichen Gründen ist es nicht möglich, alles umzusetzen. Wir setzten uns deshalb als Ziel, konkret folgende Punkte umsetzten.

### Datenbank Schema

Wir werden nur den Blauenteil umsetzen. Dazu gehören die Registrierten User und die dazugehörigen Sozialen Funktionen wie Freunde hinzufügen und Nachrichten versenden. Spiele und Lobbys werden nur zur Laufzeit erstellt und nicht auf der Datenbank gespeichert. Einzig die Statistik aus vergangenen Spielen werden persistiert.

### Frontend Seiten

Alle Erfassten Wireframes möchten wir auch umsetzten. Ausnahme ist die Spiellogik Seite, wir können aus zeitlichen Gründen dies nur als Prototyp umsetzten. Der Spielstand wird über die Konsole mittgeteilt und es können Karten nur über Buttons gespielt werden.

### Sicherheit

User werden in unserer Datenbank abgespeichert. Eine Registrierung eines Benutzerprofils wird jedoch nicht vorausgesetzt. Das Passwort soll gesaltet und gehasht abgelegt werden. Die Kommunikation und der sichere Informationsaustausch mit eingeloggten Benutzern erfolgt über JSON Web Tokens.

### User/Gäste

Bei einer Registrierung wird das Benutzerprofil auf unserer Datenbank gespeichert. Es gibt aber die Möglichkeit unsere Webseite als Gast zu besuchen, von welchem keine Daten persistent gespeichert werden.

### Soziale Funktionen

Soziale Funktionen wie Nachrichten versenden, Freundschaftsanfragen verschicken und Profile registrierter Benutzer ansehen sollen möglich sein.

### Spielablauf

Um eine Runde des Kartenspiels zu spielen werden wir einen Lobbybrowser umsetzen. Dazu gehört das Erstellen, Beitreten und Verlassen von öffentlichen und passwortgeschützten Lobbies. Diese Lobbies existieren nur zur Laufzeit und werden nicht gespeichert. Einzig geplant ist, dass Statistiken aus Lobbies und Spielen für die registrierten Benutzer abrufbar sind. Der Echtzeit Datenaustausch wird mit Web Sockets ermöglicht. Nach Spielschluss wird wieder in die Lobby zurückgekehrt. Die Statistik soll anschliessend generiert und gespeichert werden.

### Spiellogik

Die Logik und allenfalls Tests für Kartenspiel sollen umgesetzt werden. Das Kartenspiel für die Benutzer soll nur als «Prototyp» verfügbar sein, da ein spielbares Styling viel zu zeitintensiv wäre.

## Zeitplan

# Realisierung

## Testkonzept

## Testprotokoll

# Benutzerdokumentation

# Installationsanleitung

Die folgende Anleitung soll den Installationsprozess unserer Anwendung aufzeigen.

## Voraussetzung

Windows Computer mit MSSQL Server Instanz.

## Installation

Zuerst muss die aktuelle Version von Node.js auf <https://nodejs.org/en/> heruntergeladen werden. Es muss die aktuelle Long Term Support (LTS) Version installiert werden. Dabei wird NodeJS, welches JavaScript ausserhalb des Browsers ausführen kann und der Node Package Manager welcher zur Installation benutzter Code Bibliotheken gebraucht wird installiert. Unser Backend ist in Node/JavaScript geschrieben und benötigt Node um gestartet werden zu können. In der Backendtechnologie ist ein eigener Webserver integriert und hostet die kompilierten Dateien des Frontends.

## Aufsetzten

Es soll im Dateisystem an einem gewünschten Ort ein Ordner erstellt werden. Der Code der von uns vorgefertigten Version ist nun darin abzulegen. Um das Backend fertig aufzusetzen, muss nun in der Eingabeaufforderung von Windows (CMD) in den entsprechenden Ordner navigiert werden und der Befehl npm install ausgeführt werden. Mit dem Befehl npm serve wird das Backend gestartet und im Browser kann das Frontend mit localhost:4040 aufgerufen werden. Dass die Datenbank korrekt erstellt wird muss die Konfiguration in der Datei backend/ormconfig.json angepasst werden.

# Anhang

## Glossar

### Abkürzungen

**MSSQL** - Microsoft SQL Server   
**DB** - Datenbank  
**JWT** - JSON Web Tokens  
**NPM** - Node Package Manager  
**LTS** - Long Term Support

### Allgemeine und Technische Bezeichnungen

-

### Projektbezogene Bezeichnungen

**Numerous** - [Numerous](https://en.wikipedia.org/wiki/Shithead_(card_game)), oder auch ShitHead genannt, ist ein Kartenspielablauf mit besonderen regeln. Unser Projekt, welches dieses Kartenspiel mit zusätzlichen Funktionalitäten umfasst, wurde deshalb danach benannt.

## Arbeitsjournal

## Verzeichnis

### Grafiken

https://www.fiftytwo.de/products/black-rounders-playing-cards

### Frameworks

https://nodejs.org/en/   
http://expressjs.com/   
https://github.com/typeorm/typeorm   
https://socket.io/  
https://jwt.io/   
https://vuejs.org/   
https://vuetifyjs.com/en/  
https://www.npmjs.com/package/mssql