

M150, BBZW Sursee

Numerous - Kartenspiel

Gian Ott & Manuel Troxler

16.01.2020

Inhalt

1	Einleitung.....	3
2	Projektinformationen	3
2.1	Teilhabende	3
3	Projektplan	4
3.1	Rahmenbedingungen und Abgrenzung	4
3.1.1	Datenbank Schema	4
3.1.2	Frontend Seiten	4
3.1.3	Sicherheit	4
3.1.4	User/Gäste	4
3.1.5	Soziale Funktionen	4
3.1.6	Spielablauf	4
3.1.7	Spiellogik	4
3.2	Zeitplan	4
4	Realisierung	5
4.1	Testkonzept	5
4.1.1	System Tests	5
4.1.2	Unit Tests	5
4.2	Testprotokoll	5
4.2.1	Testübersicht	5
4.2.2	Tests	5
4.3	Benutzerdokumentation	8
4.3.1	Login.....	8
4.3.2	Home.....	8
4.3.3	Lobby.....	10
4.4	Installationsanleitung	11
4.4.1	Voraussetzung.....	11
4.4.2	Installation	11
4.4.3	Datenbank Konfiguration	11
4.4.4	Aufsetzten	13
5	Anhang	13
5.1	Glossar	13
5.1.1	Abkürzungen	13
5.1.2	Allgemeine und Technische Bezeichnungen	13
5.1.3	Projektbezogene Bezeichnungen	13
5.2	Arbeitsjournal	14
5.3	Verzeichnis.....	16
5.3.1	Grafiken	16
5.3.2	Frameworks	16

1 Einleitung

Dieses Dokument umfasst die Realisierungsphase des im Modul 150 umgesetzten Projektes Numerous. Zudem sind eine Installationsanleitung und eine Bedienungsanleitung in diesem Dokument enthalten. In der Abgabe sind nebst der Dokumentation noch der Source enthalten.

2 Projektinformationen

2.1 Teilhabende

Name	Funktion	Kontakt
Gian Ott	Entwickler & Autor	nichtgian@gmail.com 079 199 82 03
Manuel Troxler	Entwickler & Autor	manuel.troxler@gmx.ch 078 840 84 17

Beginn: Do, 28.11.2019

Abgabetermin: Do, 16.01.2020

Arbeitsaufwand: ca. 20 Lektionen

3 Projektplan

3.1 Rahmenbedingungen und Abgrenzung

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

3.1.1 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.

3.1.2 Frontend Seiten

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

3.1.3 Sicherheit

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

3.1.4 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.

3.1.5 Soziale Funktionen

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

3.1.6 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.

3.1.7 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.

3.2 Zeitplan

Die Ziele haben wir uns ständig am Ende eines Nachmittags in der Schule im Arbeitsjournal eingetragen. Ein Eintrag im Arbeitsjournal enthält den Punkt Geplant, welcher die Aufgaben und Ziele für diesen Tag enthält.

4 Realisierung

4.1 Testkonzept

4.1.1 System Tests

Abläufe, welche mehrere Komponenten zum Einsatz kommen lassen, möchten wir mit manuellen System Tests erfassen.

4.1.2 Unit Tests

Für die Kartenspiellogeik möchten wir automatisierte Unit Tests erstellen. Sie ist eine in sich abgeschlossene Komponente und soll immer dieselben Werte zurückgeben.

4.2 Testprotokoll

Tests, welche nicht als Unit Tests markiert sind, sind manuelle System Tests.

4.2.1 Testübersicht

Test Nr.	Titel	Status
T01	Unit Test: Normale Karte spielbar	
T02	Unit Test: Immer spielbare Karten	
T03	Unit Test: Spezialregel Karte 7	
T04	Registrieren	
T05	Login	
T06	Gast Login	
T07	Lobby erstellen	
T08	Lobby beitreten	

Status	Beschreibung
	Erfolgreich
	Teilweise erfolgreich, Problembeschreibung nötig
	Nicht erfolgreich oder nicht implementiert, Problembeschreibung nötig

4.2.2 Tests

Test Nr.	T01
Titel	Normale Karte spielbar
Bedingung	-
Ablauf	Unit Test: Eine normale Karte ohne spezielle Regel soll nur spielbar sein, wenn die oberste Karte des Stapels kleiner oder gleich ist.
Test Daten	(oberste Stapel Karte / gespielte Karte) ✓ (4/4), (4/5), (6/6), (8/8), (8/9), (Q/K), (J/A) ✗ (5/4), (9/6), (A, K)
Erwartetes Ergebnis	Wenn die Karte spielbar ist, soll True, und wenn nicht False zurückgegeben werden.
Testergebnis	Die normalen Karten sind nur dann spielbar, wenn dies zulässig ist.

Test Nr.	T02
Titel	Immer spielbare Karten
Bedingung	-
Ablauf	Unit Test: Die Karten 2, 3 und 10 sind immer spielbar.
Test Daten	(oberste Stapel Karte / gespielte Karte) ✓ (A/2), (4/3), (K/10) ✗
Erwartetes Ergebnis	Die Karten mit der Regel sollen unabhängig vom Wert der obersten Karte immer spielbar sein.
Testergebnis	Die Karten können immer gespielt werden.

Test Nr.	T03
Titel	Spezialregel Karte 7
Bedingung	-
Ablauf	Unit Test: Die Karte 7 ist nur spielbar, wenn die Karte des Stapels grösser oder gleich 7 ist.
Test Daten	(oberste Stapel Karte / gespielte Karte) ✓ (A/7), (K/7), (7/7) ✗ (2/7), (6/7)
Erwartetes Ergebnis	Die Karte kann nur gespielt werden, wenn die oberste Karte grösser oder gleich 7 ist. Wenn sie spielbar ist, wird True, falls nicht False zurückgegeben.
Testergebnis	Die Karten sind nur spielbar, wenn dies zulässig ist.

Test Nr.	T04
Titel	Registrieren
Bedingung	Webseite geöffnet
Ablauf	In die Eingabe Felder werden folgende Daten eingegeben. Sind diese valide soll in der Datenbank ein neuer Benutzer eingetragen werden. Passwort soll dabei gesalted und gehashed werden. Es dürfen keine Einträge mit gleichen Benutzernamen abgespeichert werden.
Test Daten	(username / passwort / repeatPassword) ✓ (test/123/123) ✗ („/“/„/“), (test/123/321)
Erwartetes Ergebnis	Es soll ein User abgespeichert werden, wenn die Eingabe den oben genannten Punkten entspricht.
Testergebnis	Doppelte Benutzer oder bereits verwendete Benutzernamen erstellen keinen Eintrag. Ist das Passwort und das RepeatPasswort nicht gleich wird dies angezeigt. Wenn ein Benutzer registriert wurde wird man zur Login Seite eingeloggt.

Test Nr.	T05
Titel	Login
Bedingung	Webseite geöffnet und Benutzer registriert
Ablauf	Die Eingabe in das Login Feld sollen überprüft werden. Bei einem erfolgreichen Login soll der Token gesetzt werden und auf die Home Seite navigiert werden.
Test Daten	(username / passwort) ✓ (test/123) ✗ (test/„“)
Erwartetes Ergebnis	Wenn die Eingaben korrekt sind soll der Token gesetzt werden. Falls nicht soll eine Meldung angezeigt werden.
Testergebnis	Korrekte Daten führen zu erfolgreichem Login.

Test Nr.	T06
Titel	Gast Login
Bedingung	Webseite geöffnet
Ablauf	Nicht registrierte Benutzer können die Webseite auch als Gast öffnen.
Test Daten	
Erwartetes Ergebnis	Der Benutzer wird beim Klick auf als Gast einloggen zur Home Seite weitergeleitet. Er verfügt über die gleichen Funktionalitäten wie ein registrierter Benutzer, ausser dass er nicht Soziale Interaktionen wie das Schreiben an registrierte Benutzer vornehmen kann.
Testergebnis	Die entsprechenden Controls werden nicht dargestellt und können von Gästen nicht benutzt werden.

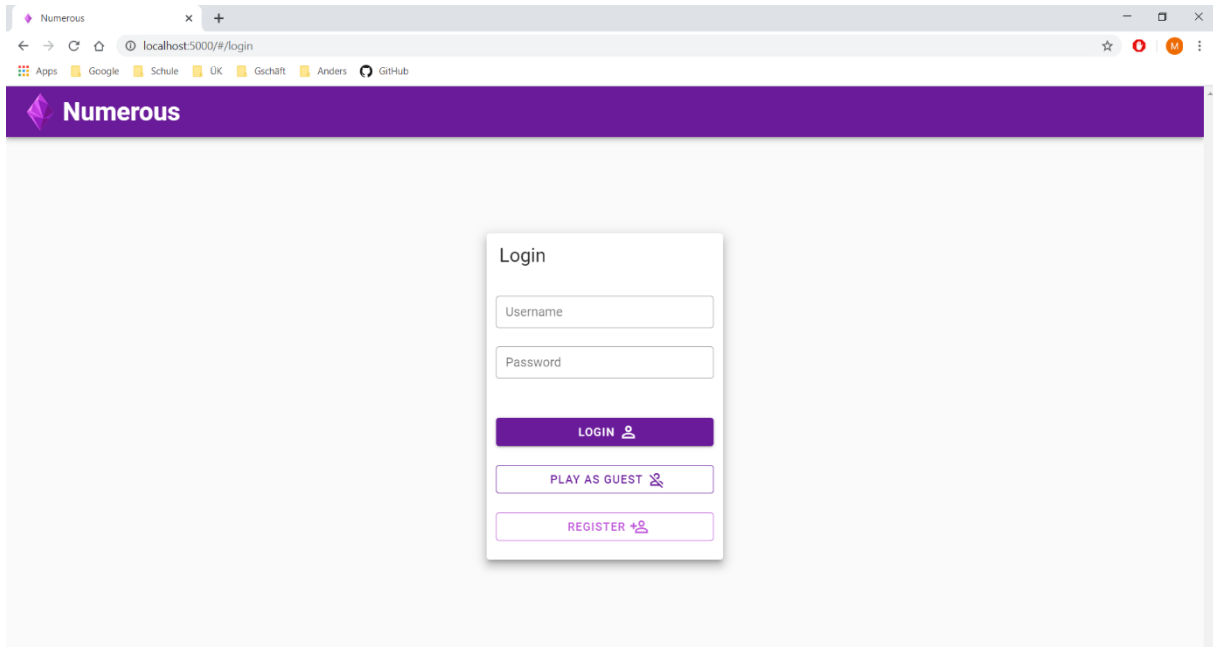
Test Nr.	T07
Titel	Lobby erstellen
Bedingung	Webseite geöffnet und in Benutzer eingeloggt / Gast
Ablauf	Es soll beim Klick auf Lobby Erstellen ein Menü zum Erstellen von Lobbies angezeigt werden.
Test Daten	(lobbyname / password / isPrivate) ✓ (test/123/true), (test/„“/false) ✗ (test/„“/true)
Erwartetes Ergebnis	Sind die Daten zum Erstellen valid (Name angegeben, wenn privat ist ein Passwort nötig) soll eine neue Lobby erstellt werden und der Lobby beigetreten werden.
Testergebnis	Die Lobby wird erstellt und es wird ihr automatisch beigetreten.

Test Nr.	T08
Titel	Lobby beitreten
Bedingung	Webseite geöffnet und in Benutzer eingeloggt / Gast
Ablauf	Es wird eine Liste mit Lobbies angezeigt.
Test Daten	
Erwartetes Ergebnis	Wenn auf eine Lobby in der Liste geklickt wird, soll ihr Beigetreten werden können. Falls diese privat sein sollte kann das Passwort eingegeben werden. In der Lobby sollen alle beigetretenen User angezeigt werden. Verlassen alle die Lobby soll diese gelöscht werden.
Testergebnis	Das Beitreten von privaten und öffentlichen Lobbies funktioniert und das Verlassen ist auch möglich.

4.3 Benutzerdokumentation

4.3.1 Login

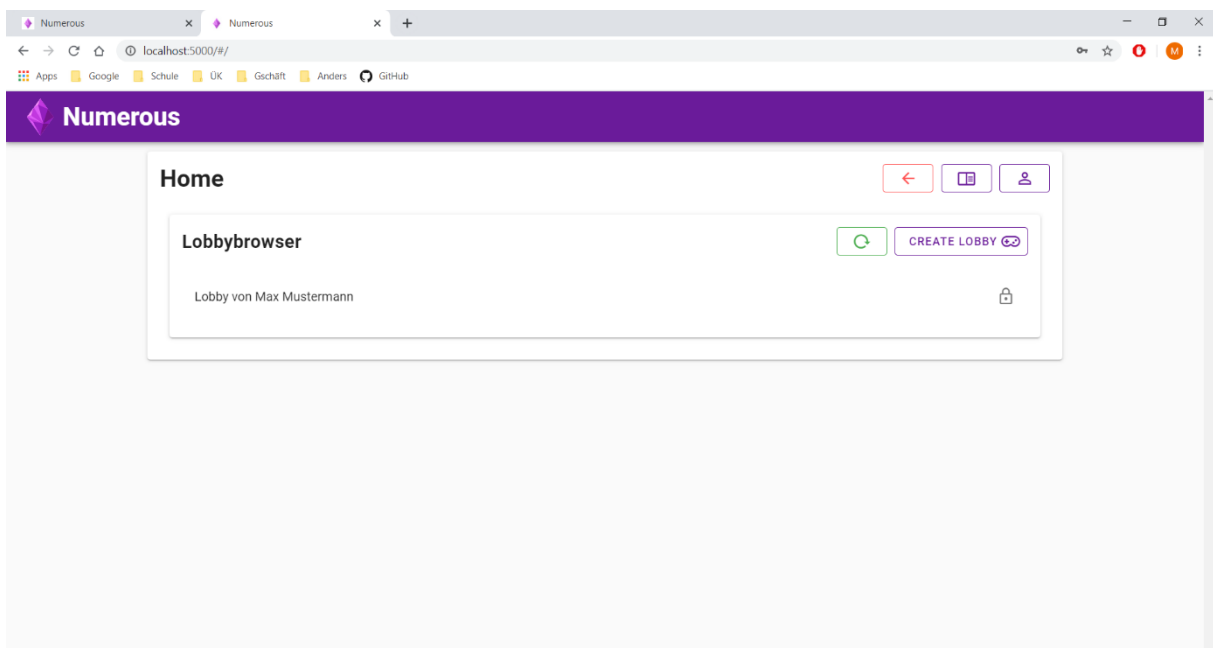
Wenn man unsere Anwendung startet, sieht man zuerst den Loginscreen. Man kann sich dort, wenn man schon einen Benutzer hat damit einloggen. Ansonsten kann man sich registrieren. Dafür muss man auf den Button «Register» klicken.



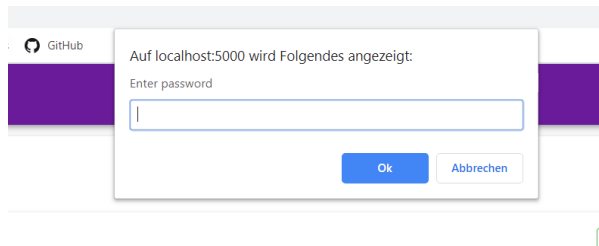
Damit öffnet sich das Register Fenster. Dies sieht fast gleich aus. Bis auf die unterschiedlichen Bezeichnungen ist der einzige Unterschied, dass es für das Passwort zwei Felder hat, um das Passwort zu bestätigen. Damit verhindern wir die Falscheingabe des Passwortes und das damit der Benutzer sich gar nicht einloggen kann. Nach dem Registrieren kann man sich mit dem Benutzer einloggen im Login Fenster. Alternativ kann man sich auch als Gast anmelden, wenn man es will.

4.3.2 Home

Wenn man sich erfolgreich als User oder Gast eingeloggt hat, wird man auf das Home Fenster weitergeleitet. Darin kann man gerade die offenen Lobbys sehen.

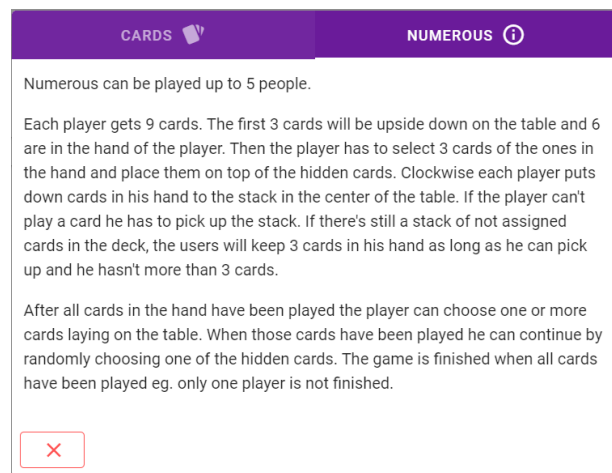
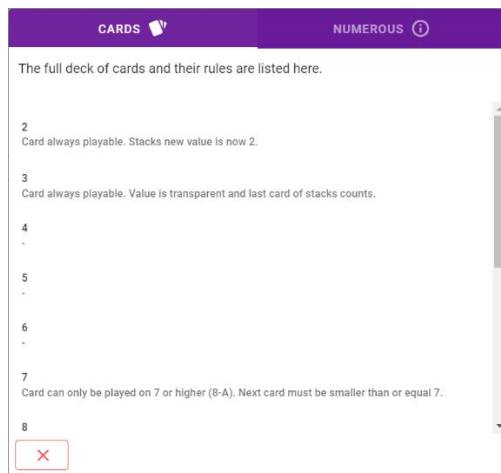


In diesem Beispiel ist eine Lobby offen. Die Lobby hat den Namen «Lobby von Max Mustermann», den kann man aber selber setzen. An dem Schloss auf der rechten Seite kann man erkennen, dass sie Privat ist und damit durch ein Passwort geschützt. Wenn sie nicht Privat wäre könnte man einfach beitreten, in diesem Fall kommt noch ein Pop-up indem man das Passwort eingeben muss.

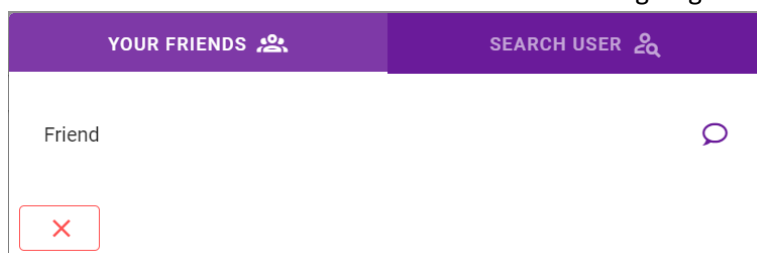


Damit kommt man dann auch in die gewünschte Lobby.

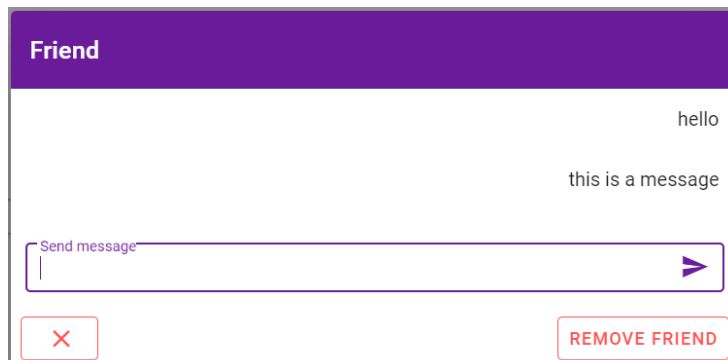
Auf dem Home findet man noch einen Button mit einem Roten Pfeil. Mit diesem kann man sich ausloggen. Der Button daneben ist die Erklärung zum Spiel. Darin sind einerseits die Karten beschrieben aber auch die Spielregeln kurz aufgeschrieben.



Oben rechts kann man auch noch einen Button sehen mit einer Figur drauf. Dies ist das Soziale Menü. Man kann dort freunde suchen und die bereits hinzugefügten Freunde sehen.



Wenn man dann auf einen Freund klickt kann man ihm auch eine Nachricht schreiben.

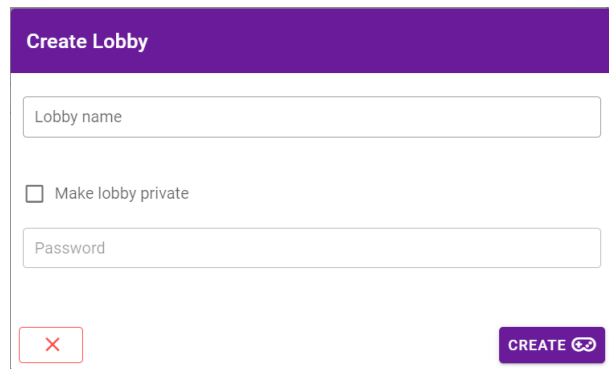


Diese Sozialen Features gibt es aber nur für eingeloggte Benutzer. Wenn man als Gast eingeloggt ist sieht man den Button gar nicht.

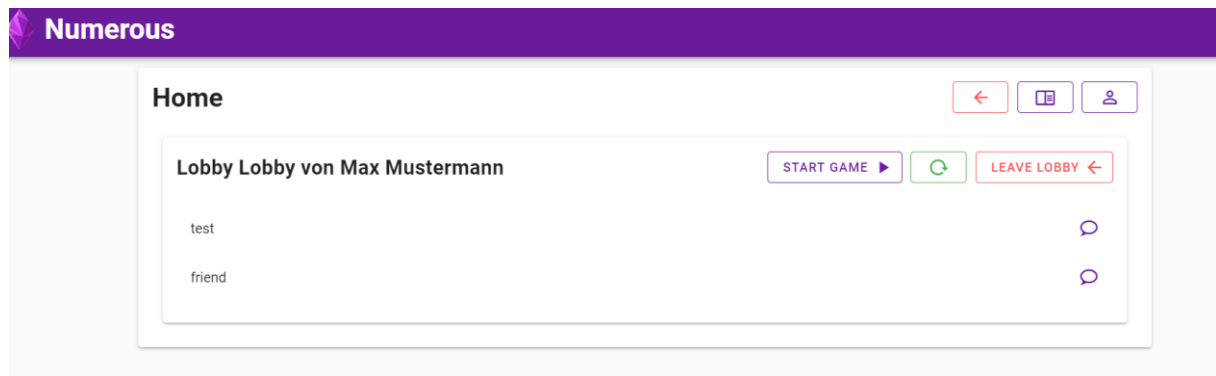
Weiter kann man eine neue Lobby erstellen. Dabei kann man Namen und falls man will ein Passwort setzen.

4.3.3 Lobby

Wenn man in einer Lobby ist, sieht man aufgelistet, wer sich sonst noch in der Lobby befindet.



The 'Create Lobby' form has a purple header. It contains a text input for 'Lobby name', a checkbox for 'Make lobby private', and a text input for 'Password'. At the bottom left is a red square button with a white 'X', and at the bottom right is a purple 'CREATE' button with a lock icon.



The lobby interface has a purple header with the 'Numerous' logo. Below it is a 'Home' section with a title 'Lobby Lobby von Max Mustermann'. To the right of the title are three buttons: a red back arrow, a purple icon of a document with a checkmark, and a purple user icon. Below the title are two buttons: a purple 'START GAME' button with a right arrow and a green circular refresh icon. To the right of these buttons is a red 'LEAVE LOBBY' button with a left arrow. Below the buttons is a list of players: 'test' and 'friend', each with a purple speech bubble icon to its right.

Die Buttons oben haben immer noch dieselbe Bedeutung wie im Home. Neu ist der Button, um die Lobby zu verlassen und der Start Game Button. Letzterer funktioniert aber noch nicht, da wir das Spiel ja noch nicht implementiert haben. Wenn man auf die Spieler klickt kann man ihm eine Nachricht schreiben und ihn als Freund hinzufügen. Die Lobby ist solange offen, wie ein Spieler in der Lobby ist.

4.4 Installationsanleitung

Die folgende Anleitung soll den Installationsprozess unserer Anwendung aufzeigen.

4.4.1 Voraussetzung

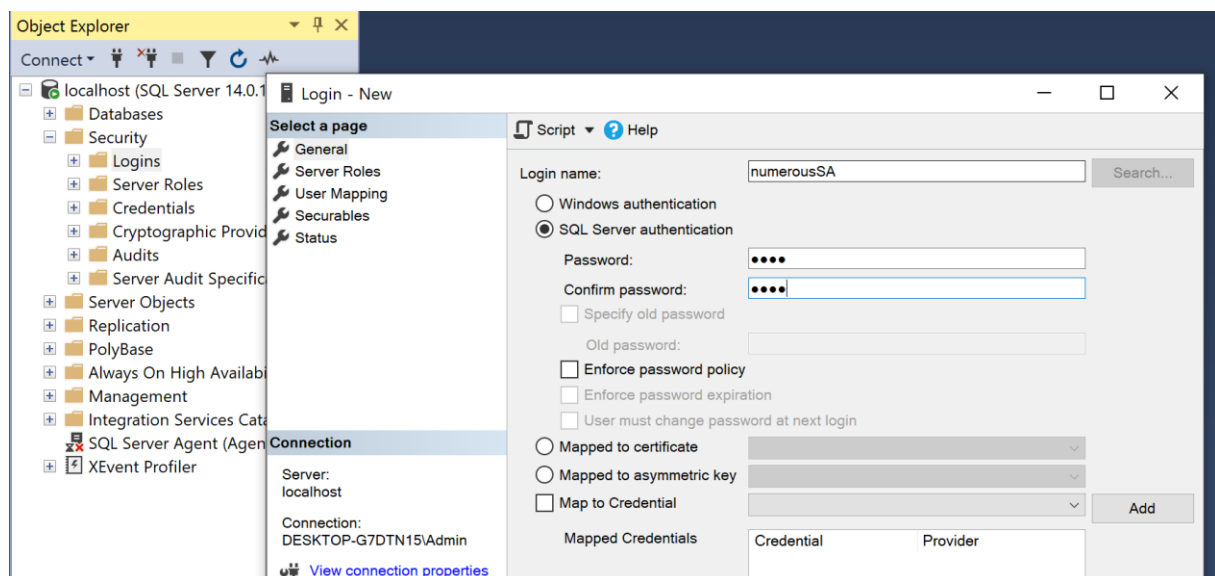
Windows Computer mit MSSQL Server Instanz. Dazu gehört das Microsoft SQL Server Management Studio und der SQL Server Configuration Manager.

4.4.2 Installation

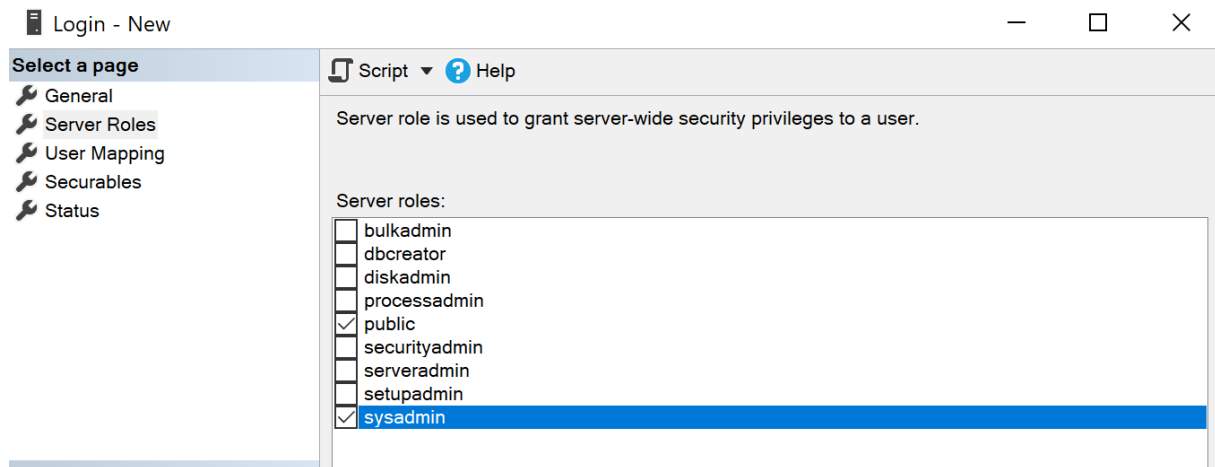
Zuerst muss die aktuelle Version von Node.js auf <https://nodejs.org/en/> heruntergeladen werden. Es muss die aktuelle Long Term Support (LTS) Version installiert werden. Dabei wird NodeJS, welches JavaScript ausserhalb des Browsers ausführen kann und der Node Package Manager welcher zur Installation 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.

4.4.3 Datenbank Konfiguration

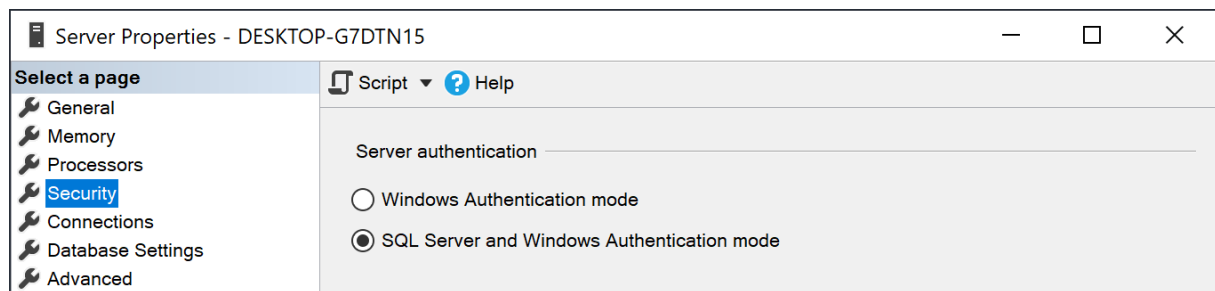
Damit das Backend funktioniert, muss eine Datenbankverbindung aufgebaut werden können. Zuerst muss, falls noch keines existiert, ein Datenbank Login erstellt werden. Öffnen sie den Microsoft SQL Server Management Studio. Mit rechtsklick auf Security/Logins kann unter New Login.. ein neuer Login erstellt werden. Es muss ein Login Name und ein Passwort angegeben werden. Optional können hier auch die Password Policies entfernt werden.



Unter Server Roles müssen nun dem Login Rollen zugewiesen werden. Am einfachsten ist es, wenn dem Login die Sysadmin-Rechte zugewiesen werden. Mit OK kann der User gespeichert werden.

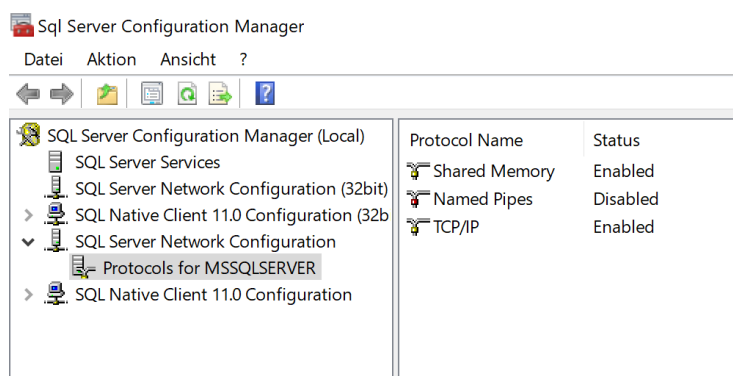


Auf der Datenbank müssen nun noch folgende Konfigurationen vorgenommen werden. Zuerst muss das Login von nicht Windows Benutzern erlaubt werden. Mit Rechtsklick auf den Server kann unter dem Menüpunkt Properties folgendes Menü geöffnet werden. Die Einstellung SQL Server and Windows Authentication mode muss ausgewählt werden.



Nun muss noch die Default DB erstellt werden. Es empfiehlt sich direkt das neu erstellte Login in einer neuen Datenbank Verbindung zu brauchen. Unter Databases New Database.. kann folgendes Menü geöffnet werden. Erstellen sie eine neue Datenbank.

Damit das Backend über die Protokolle TCP/IP mit der Datenbank kommunizieren können, muss noch folgende Einstellung überprüft werden. Öffnen sie den SQL Server Configuration Manager. Unter dem dargestellten Menüpunkt muss das Protokoll TCP/IP auf Enabled gesetzt werden. Die Datenbank ist nun fertig aufgesetzt.



4.4.4 Aufsetzen

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 start** wird das Backend gestartet und im Browser kann das Frontend mit **http://localhost:5000** aufgerufen werden. Dass die Datenbank korrekt erstellt wird muss die Konfiguration in der Datei **backend/ormconfig.json** angepasst werden.

Optional:

Falls das Frontend selbst angepasst werden möchte, muss in einer Eingabeaufforderung (CMD) ins Frontend navigiert werden. Mit **npm install** werden die Abhängigkeiten installiert. Mit **npm run serve** kann das Frontend zur Entwicklung gestartet werden. Möchte man nun Änderungen dem Backend zum hosting geben muss **npm run build** ausgeführt werden. Das Frontend wird so «kompiliert» und die Dateien im dist Ordner abgelegt, welcher vom Backend gehostet wird.

5 Anhang

5.1 Glossar

5.1.1 Abkürzungen

MSSQL - Microsoft SQL Server

DB - Datenbank

JWT - JSON Web Tokens

NPM - Node Package Manager

LTS - Long Term Support

CMD -Command-Line Interpreter

5.1.2 Allgemeine und Technische Bezeichnungen

Begriff	Beschreibung
Command-Line Interpreter	Das in Windows integrierte Command Prompt. Auch bekannt als Eingabeaufforderung. Wichtige Befehle sind das Laufwerk wechseln mit Laufwerkbuchstabe (z.B. C: oder D:) und im Dateisystem navigieren mit cd PFAD.

5.1.3 Projektbezogene Bezeichnungen

Begriff	Beschreibung
Numerous	Numerous , oder auch ShitHead genannt, ist ein Kartenspielablauf mit besonderen regeln. Unser Projekt, welches dieses Kartenspiel mit zusätzlichen Funktionalitäten umfasst, wurde deshalb danach benannt.

5.2 Arbeitsjournal

Datum	Do, 28.11.2019
Pendenz	GO, MT
Geplant	- Beginn Arbeit
Tätigkeit	<p>Das Prinzip und die Regeln des Kartenspiels wurden geübt. Der Spielablauf in der Realität wurde ins Digitale «übersetzt». Allgemein wurden die Umsetzung verschiedener Funktionalitäten und Designentscheidungen getroffen.</p> <p>Anpassungen an den Zielen gemacht. Wir haben genau erfasst, was wir in der gegebenen Zeit alles machen möchten und was nicht in dieser Arbeit umgesetzt wird. Wir haben das Kartenspielprinzip besprochen und das Projekt grob mit den geplanten Technologien im Front- und Backend aufgesetzt.</p>
Reflexion	Der Spielablauf ist nun allen Teammitgliedern klar.
Ziele	- An DB arbeiten
Aufwand	4 Lektionen
Datum	Do, 05.12.2019
Pendenz	GO, MT
Geplant	- An DB arbeiten
Tätigkeit	Wir konnten weiter am Backend arbeiten und konnten die Entities fertigstellen, somit ist der DB Teil abgeschlossen. Als nächstes gilt es den Kartenspielablauf auf dem Backend umzusetzen. Dazu möchten wir zusätzlich Unit Tests einsetzen und nach TDD arbeiten, da dies dort ein sinnvoller Einsatz findet.
Reflexion	Der Einstieg in die Arbeit ging gut. Da wir zum Teil mit neuen Frameworks arbeiten, haben wir uns diese angeschaut.
Ziele	- Am Backend arbeiten
Aufwand	4 Lektionen
Datum	Do, 12.12.2019
Pendenz	GO, MT
Geplant	- An DB arbeiten
Tätigkeit	Das Prinzip und die Regeln des Kartenspiels wurden geübt. Der Spielablauf in der Realität wurde ins Digitale «übersetzt». Allgemein wurden die Umsetzung verschiedener Funktionalitäten und Designentscheidungen getroffen.
Reflexion	Heute haben wir viel Entscheidungen für die Weiterarbeit getroffen. Daher sind wir mit dem Produkt nicht viel weitergekommen aber konnten einen guten Grundstein setzen für die Zukunft. Somit hat uns der heutige Tag viel gebracht.
Ziele	Frontend beginnen
Aufwand	4 Lektionen

Datum	Do, 19.12.2019
Pendenz	GO, MT
Geplant	Erstellung Dokumentation. Frontend
Tätigkeit	Erstellung des Arbeitsjournal und Beginn der Dokumentation.
Reflexion	Da wir das Arbeitsjournal nicht von Anfang an geführt haben, da wir nicht gesehen haben, dass dies eine Anforderung war, mussten wir das Journal für die letzten beiden Donnerstage nachführen. Geplant war noch das Frontend zu beginnen was wir nicht geschafft haben. Wir können aber sicherlich in den Ferien noch damit beginnen und haben sicher noch genug Zeit für das Frontend.
Ziele	Frontend beginnen, Dokumentation machen.
Aufwand	4 Lektionen

Datum	Do, 09.01.2020
Pendenz	GO, MT
Geplant	Arbeiten an der Dokumentation und dem Frontend
Tätigkeit	Wir haben heute am Frontend und auch wieder an der Dokumentation gearbeitet. Beim Frontend müssen wir sicher noch viel machen, dies werden wir bis nächste Woche sicherlich noch machen um dann am 16. Nicht mehr programmieren müssen, sondern nur noch die Dokumentation fertigstellen können.
Reflexion	Wir konnten vom Frontend noch nicht so viel umsetzen wie es gedacht war und müssen so noch in der Freizeit daran arbeiten. Mit der Dokumentation sind wir aber schon gut weitergekommen, sodass wir das nächste Mal diese noch gut fertig machen können.
Ziele	Fertig Umsetzen
Aufwand	4 Lektionen

Datum	Do, 16.01.2020
Pendenz	GO, MT
Geplant	Abschliessende Arbeiten an der Dokumentation Abgabe des Projektes und der Dokumentation.
Tätigkeit	Wir haben die Inhalte der Dokumentation zusammengeführt und die Dateien zur Abgabe fertiggestellt.
Reflexion	Wir konnten alles abgeben und sind mit unserer Arbeit zufrieden.
Ziele	-
Aufwand	4 Lektionen

5.3 Verzeichnis

5.3.1 Grafiken

-

5.3.2 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/>