

# NUMEROUS

M150 – Kartenspiel Webapplikation

## Inhalt

1	Konzept .....	4
1.1	Projektidee.....	4
1.2	Vorgaben .....	4
1.2.1	Technische Vorgaben .....	4
1.2.2	Funktionale Vorgaben.....	4
1.3	Team .....	4
2	Planung .....	5
2.1	Rahmenbedingungen.....	5
2.1.1	Sicherheit.....	5
2.1.2	Verfügbarkeit .....	5
2.1.3	Transaktionsvolumen.....	5
2.2	Datenbank Schema .....	6
2.2.1	Blauer Bereich .....	7
2.2.2	Schwarzer Bereich.....	7
2.2.3	Grüner Bereich .....	7
2.2.4	Roter Bereich.....	7
2.3	User Stories.....	8
2.4	GUI Design .....	9
2.4.1	Registrierung und Login System .....	9
2.4.2	Lobby Handling.....	12
2.4.3	Profilseite.....	16
2.4.4	Spielregeln.....	17
2.4.5	Spiellogik .....	18
2.4.6	Nachrichten und Freunde .....	19
2.4.7	Meilensteine und Arbeitsschritte .....	20
2.4.8	Zuteilung.....	20
3	Technologie .....	21
3.1	Server / Backend .....	21
3.1.1	Node.js .....	21
3.1.2	Frameworks .....	21
3.2	Client / Frontend .....	22
3.2.1	Vue.js .....	22
3.2.2	Frameworks .....	22
3.3	Datenbank .....	22

3.3.1	MS SQL .....	22
3.4	Webserver .....	22
4	Anhang .....	23
4.1	Glossar .....	23
4.1.1	Abkürzungen.....	23
4.1.2	Allgemeine und Technische Bezeichnungen.....	23
4.1.3	Projektbezogene Bezeichnungen.....	23
4.2	Verzeichnis .....	23
4.2.1	Grafiken .....	23
4.2.2	Frameworks .....	23

# 1 Konzept

## 1.1 Projektidee

In der IT-Abteilung des Lehrbetrieb Komax AG wird jeweils über den Mittag und gegebenenfalls bei Freizeitaktivitäten ein Kartenspiel gespielt. Das klassische Spielen mit dem Kartenset wird zunehmend mühsam, da die Mittagszeit begrenzt ist und immer mehr Mitspielen wollen. Die Idee ist es, das Kartenspiel als Webapplikation anzubieten, damit das beliebte Spiel auf Handy und Desktop gespielt werden kann.

Der genaue Umfang des Kartenspiels ist bei den User Stories und dem Wireframe einzusehen, da aus zeitlichen Gründen und diversen anderen Vorgaben nicht das ganze System realisiert werden kann. Das eigentliche Kartenspiel wird nur zum Teil in Form eines Prototyps erstellt, da eine grafische Realisierung nicht möglich ist.

Weitere Informationen zum Kartenspiel sind auf Wikipedia zu finden:

[https://en.wikipedia.org/wiki/Shithead\\_\(card\\_game\)](https://en.wikipedia.org/wiki/Shithead_(card_game))

## 1.2 Vorgaben

### 1.2.1 Technische Vorgaben

- Eine Webapplikation mit einer frei wählbaren Technologie umsetzen
- Ein GUI erstellen
- Verwendung von MSSQL als Datenbank
- Verwendung eines klassischen Webserver

### 1.2.2 Funktionale Vorgaben

- Lesen, speichern, ändern und löschen der Daten von der Datenbank
- Benutzerverwaltung mit Sessions oder Token
- GUI muss responsive sein

## 1.3 Team

Name	Initialien	Kontakt
Gian Ott	GO	nichtgian@gmail.com
Manuel Troxler	MT	manuel.troxler@gmx.ch

## 2 Planung

### 2.1 Rahmenbedingungen

Die Realisierungsphase ist zeitlich limitiert und einige Vorgaben sind uns gesetzt. Wir möchten uns deshalb darauf Fokussieren, ein Login System und ein Lobby Management zu erstellen. Das eigentliche Kartenspiel ist clientseitig für die Darstellung relativ aufwändig umzusetzen, und werden wir deshalb nicht umsetzen können.

#### 2.1.1 Sicherheit

Wir werden keine sensiblen Daten abspeichern, werden uns jedoch auf diverse Methoden achten, um eine möglichst sichere Anwendung gegenüber dem Benutzer gewährleisten zu können.

- Verwendung von aktuellen Versionen der Frameworks
- API Abfragen über HTTPS
- JWT Token Verfahren zur Authentifizierung

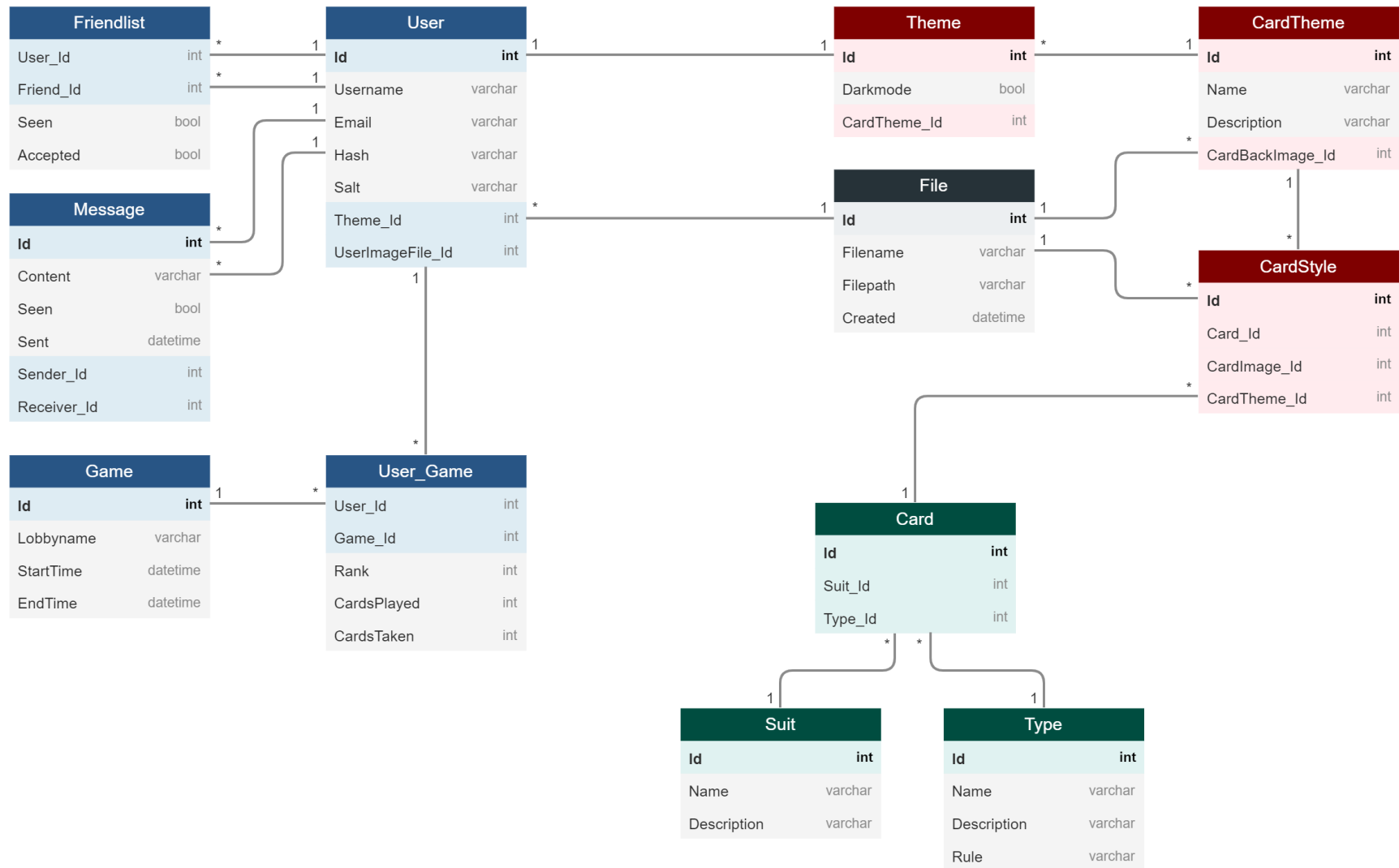
#### 2.1.2 Verfügbarkeit

Um die Verfügbarkeit zu gewährleisten werden wir Tests schreiben. So können wir das Risiko einer Unverfügbarkeit möglichst klein halten. Wir werden sicherlich auch schauen, dass wir auch bei mehr User auf einmal als wir erwarten die Verfügbarkeit sicherstellen können.

#### 2.1.3 Transaktionsvolumen

Wir rechnen mit einer kleinen Menge an User, die unsere Seite benutzen. Dies sollten wir ohne Probleme bewältigen können. Mit der Datenbank machen wir eine Verbindung sobald man sich einloggt und immer, wenn ein Spiel fertig ist. Sonst haben wir keine weitere Datenbank Transaktionen.

## 2.2 Datenbank Schema



Wir haben das Datenbank Schema farblich in verschiedene Bereiche unterteilt. Wir werden jeweils einen Abschnitt miteinander umsetzen, beginnend bei Blau und abschliessend mit Rot. Blau werden wir umsetzen. Die drei restlichen Bereiche sollen Ideen für Erweiterungen an nicht Spielrelevanten Features aufzeigen. Alle drei können aus zeitlichen Gründen nicht umgesetzt werden.

### 2.2.1 Blauer Bereich

Die registrierten Benutzer werden in der Datenbank hinterlegt und erhalten Zugriff auf ein Freunde System und Nachrichten. Darüber sollen sie sozial interagieren können und Freunde einfach zu einem Spiel einladen. Zudem kann eine Statistik aus vergangenen Spielen ausgelesen werden.

### 2.2.2 Schwarzer Bereich

Wird nicht realisiert. Es soll das hinzufügen von Dateien, insbesondere Bilddateien für die Profil Gestaltung der Bilder erlauben.

### 2.2.3 Grüner Bereich

Wird nicht realisiert. Es soll die Regeln für den Spielablauf in die Datenbank auslagern.

### 2.2.4 Roter Bereich

Wird nicht realisiert. Es soll dem Benutzer weitere Anpassungsmöglichkeiten bezüglich der Gestaltung geben.

## 2.3 User Stories

Story	Als <b>Benutzer</b> möchte ich <b>mich registrieren können</b> , um <b>Zugriff auf zusätzliche Features wie eine Freundesliste, Nachrichten und Statistiken zu erhalten.</b>
Priorität	1
Schätzung	6

Story	Als <b>Benutzer</b> möchte ich <b>mich als Gast anmelden können</b> , um <b>ohne Registrierung zu spielen zu können.</b>
Priorität	1
Schätzung	1

Story	Als <b>eingeloggter Benutzer oder Gast</b> möchte ich <b>eine neue Lobby erstellen</b> , um <b>mit anderen Leuten das Kartenspiel zu spielen.</b>
Priorität	2
Schätzung	3

Story	Als <b>eingeloggter Benutzer oder Gast</b> möchte ich <b>eine von mir erstellte Lobby auf Privat stellen</b> , um <b>nur noch Personen in meiner Lobby zu haben die das Passwort wissen und so nicht fremde Personen beitreten können.</b>
Priorität	3
Schätzung	3

Story	Als <b>eingeloggter Benutzer</b> möchte ich <b>Freunde hinzuzufügen</b> , um <b>es zu erleichtern mit der Person zusammen zu spielen.</b>
Priorität	4
Schätzung	4

Story	Als <b>eingeloggter Benutzer</b> möchte ich <b>Nachrichten schreiben</b> , um <b>meinen Mitspielern und Freunden etwas mitzuteilen.</b>
Priorität	4
Schätzung	5

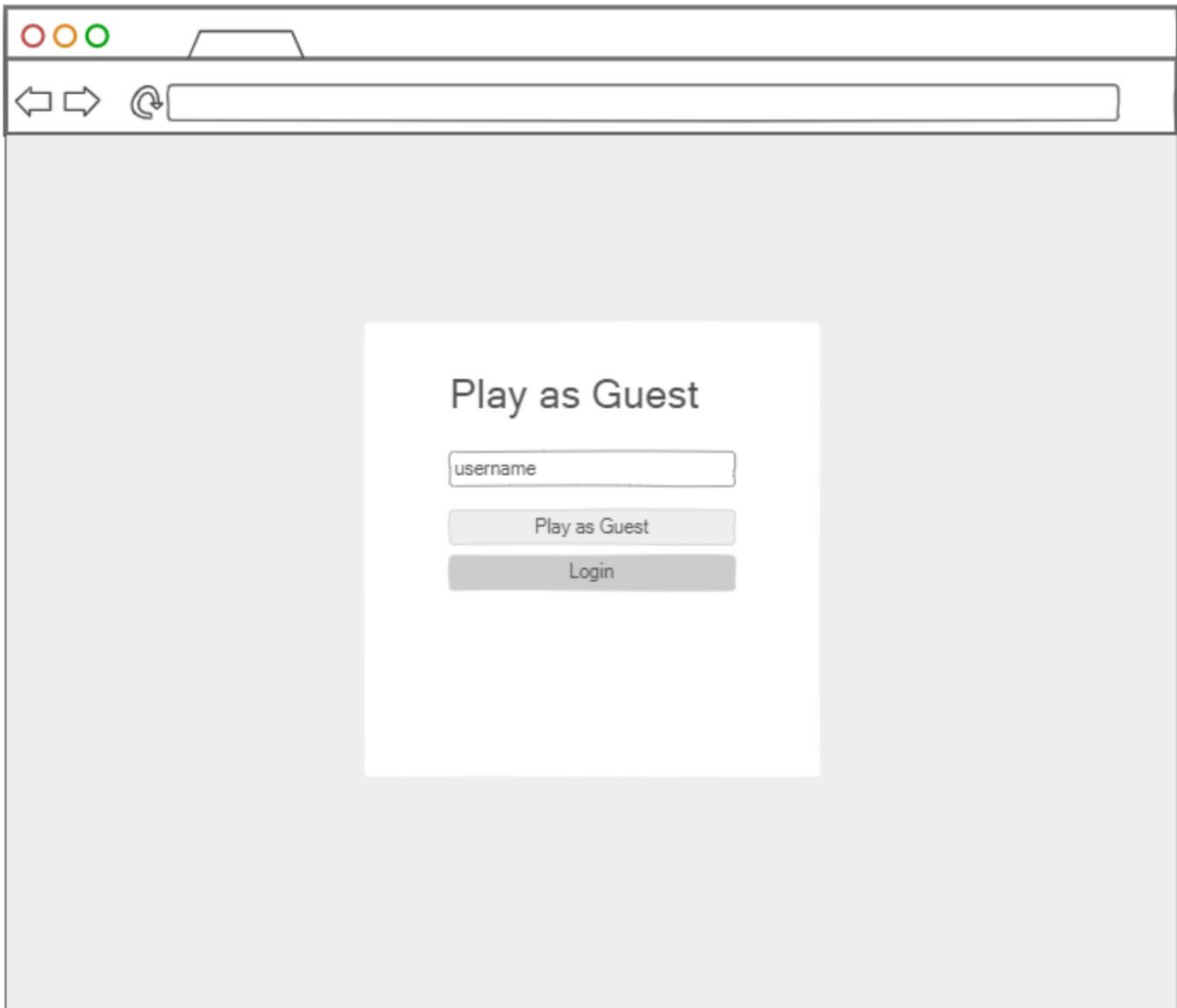


## 2.4 GUI Design

### 2.4.1 Registrierung und Login System

The image shows a wireframe of a web browser window. The browser's title bar includes three colored circles (red, yellow, green) and a tab. The address bar features navigation icons (back, forward, refresh) and a text input field. The main content area is light gray and contains a white login form. The form is titled "Login" and includes a "Play as Guest" button. Below this is a horizontal separator line, followed by an "email" input field, a password input field (represented by asterisks), a "Login" button, and a "Register" button.

Dies ist die Einstiegsseite unserer Applikation. Ein registrierter Benutzer kann sich mit Email und Passwort hier einloggen. Er landet dann auf der Lobbybrowser Startseite. Alternativ kann er zur Login als Gast oder Registrierung Seite wechseln. Ein registrierter Benutzer erhält Zugriff auf mehr Funktionen und Anpassungsmöglichkeiten. Dazu gehört auch ein Freunde und Benachrichtigung System.

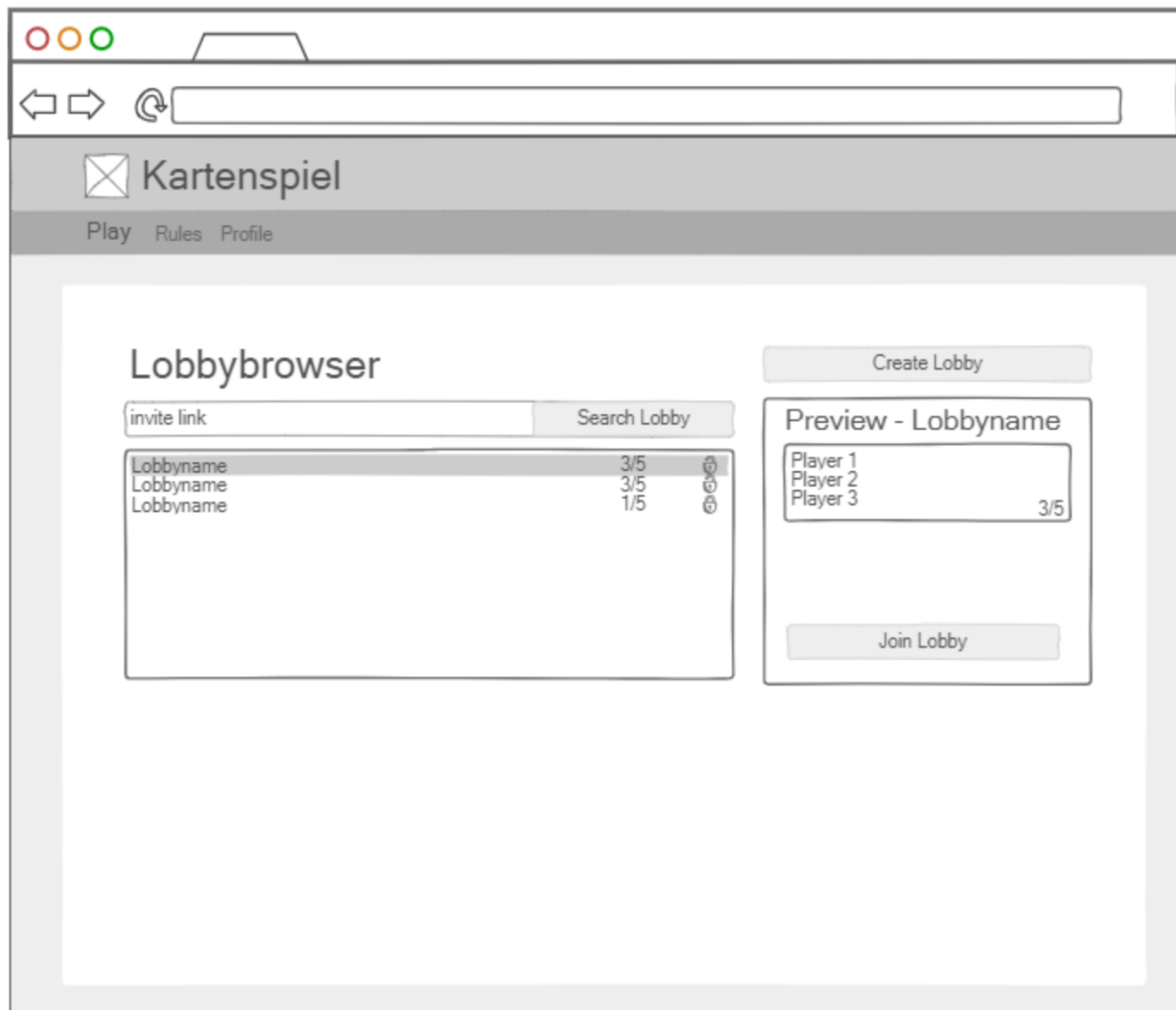


Benutzer müssen sich nicht registrieren, um das Spiel zu spielen. Hier kann ein temporäres Gästeprofil mit Name gesetzt werden, welches nach verlassen der Seite nicht mehr existiert.

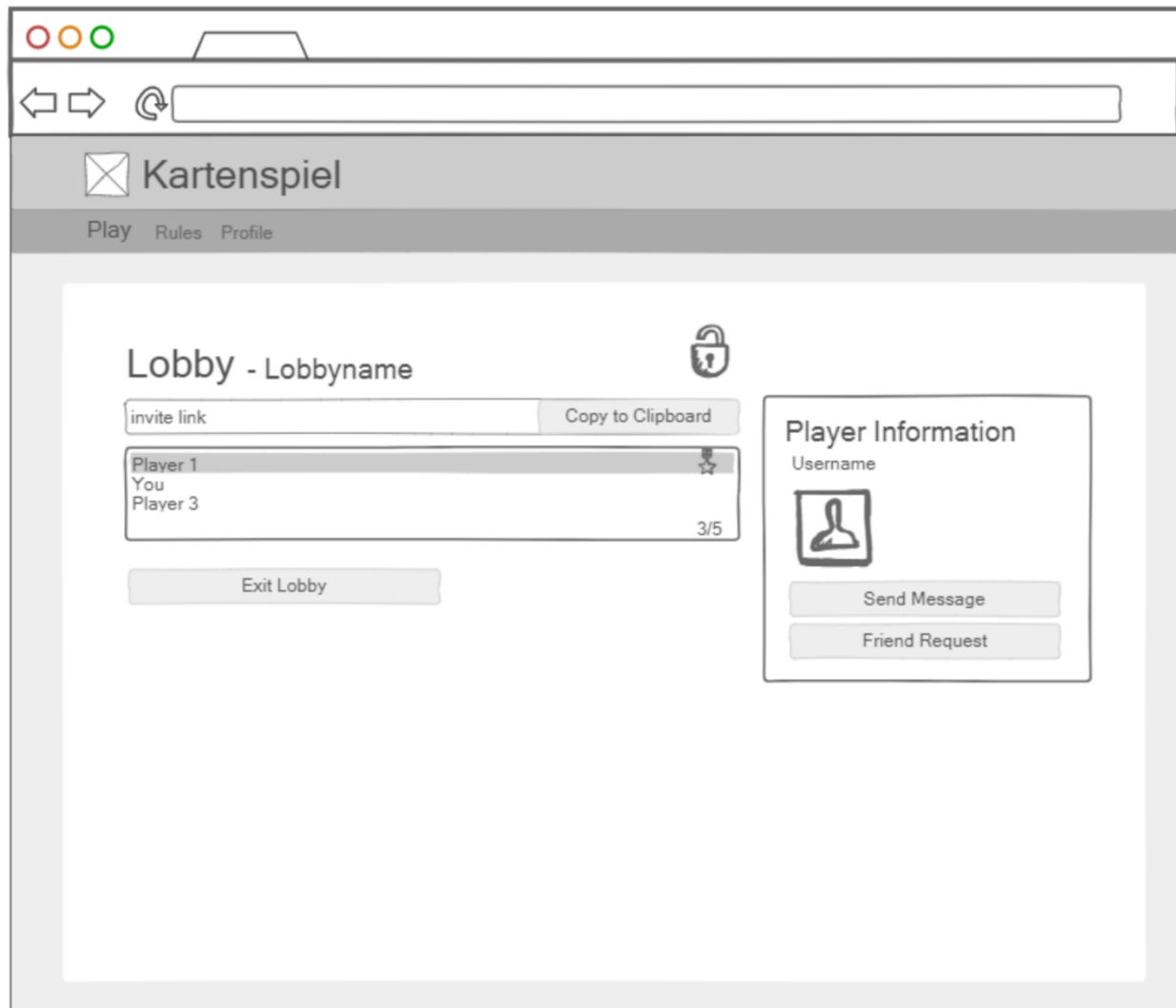
The image shows a web browser window with a single tab. The address bar is empty. The main content area has a light gray background. In the center, there is a white rectangular box containing a registration form. The form is titled "Register" in a bold, sans-serif font. Below the title, there are four input fields: "email", "username", and two password fields (each with a placeholder of "\*\*\*\*\*"). At the bottom of the form are two buttons: "Register" and "Login". The "Register" button is light gray, and the "Login" button is a darker gray.

Hier kann sich ein Benutzer registrieren. Nach erfolgreicher Registration wird zur Loginseite weitergeleitet, wo der Benutzer sich zum ersten Mal einloggen kann.

## 2.4.2 Lobby Handling



Dies ist die Startseite unserer Applikation. Um eine Runde des Kartenspiels spielen zu können, muss man in einer Lobby sein. Es kann entweder einer bestehenden Lobby, welche nicht am Spielen ist, beigetreten werden, oder eine eigene Lobby erstellt werden. Wenn ein man einen invite Link erhält, wird automatisch die Lobby selektiert oder man kann eine Lobby aus der auflistung aussuchen. Die Angewählte Lobby wird rechts angezeigt und es kann beigetreten werden. Bei privaten lobbys muss zudem das vom Lobby Leader gesetzte Passwort beim Beitreten eingegeben werden.

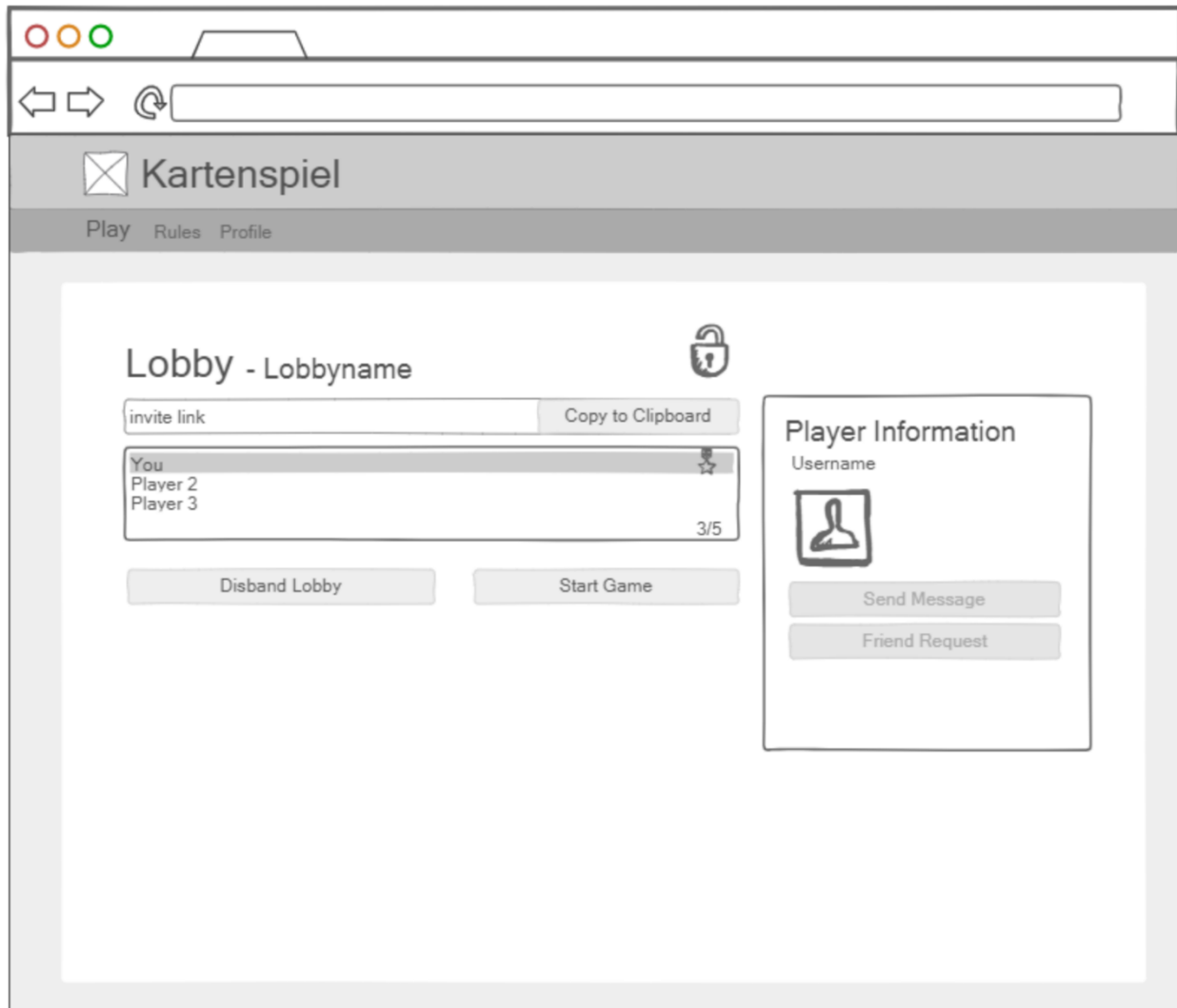


Dies ist die Ansicht, wenn man einer Lobby beigetreten ist. Die beigetretenen Spieler werden angezeigt und es können einige Profilinformationen abgerufen werden. Sobald der Lobby Leader ein Spiel startet, wird in die Spielen Ansicht gewechselt.

The screenshot shows a web browser window with a single tab. The address bar is empty. The page title is 'Kartenspiel' with a card icon. Below the title is a navigation bar with links: 'Play', 'Rules', and 'Profile'. The main content area is titled 'Create Lobby' and contains the following elements:

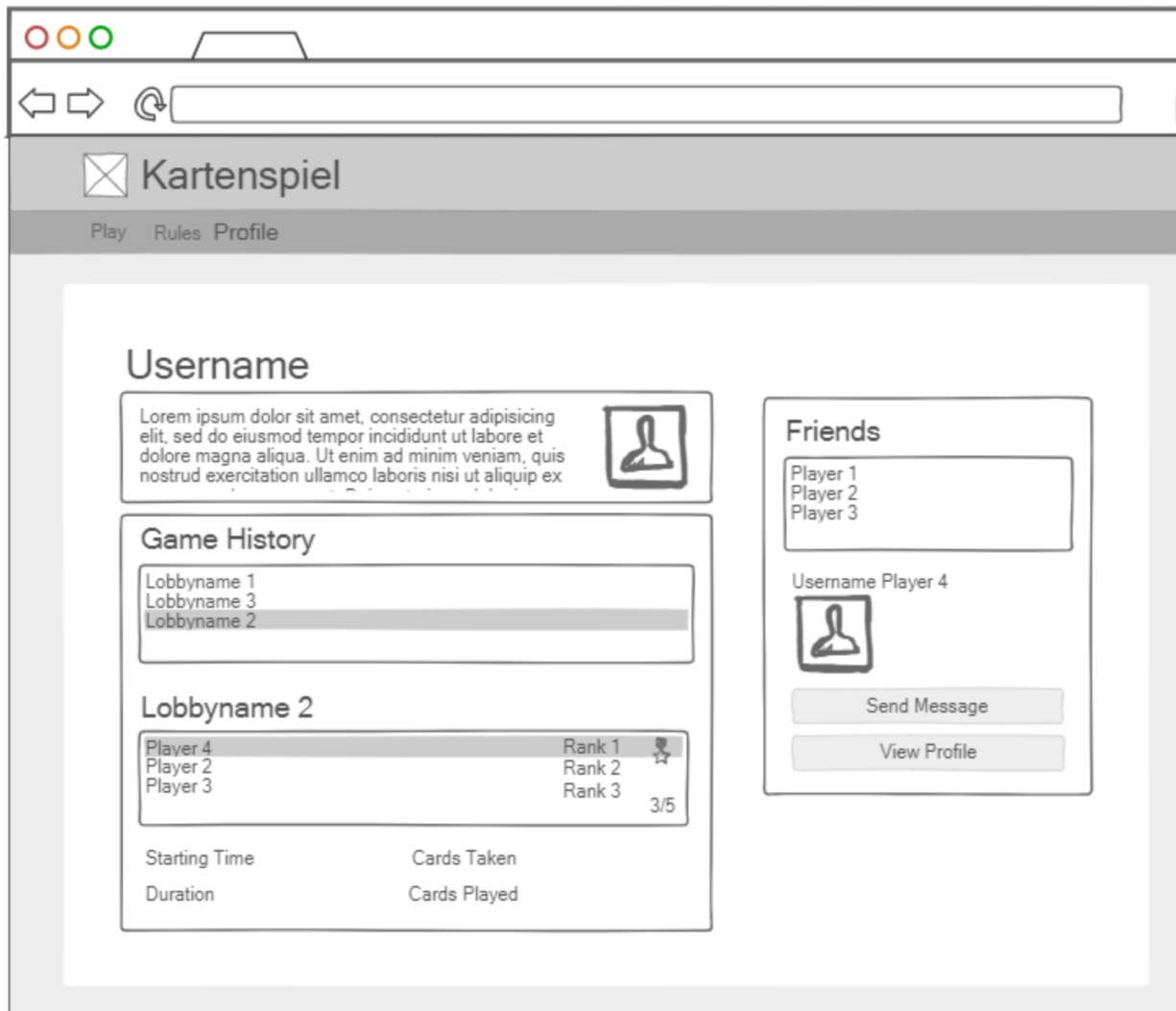
- A text input field labeled 'Lobbyname'.
- Two radio buttons for lobby type: 'Public Lobby' (selected) and 'Private Lobby'.
- Each radio button is accompanied by a lock icon.
- A password input field with six asterisks (\*\*\*\*\*).
- Two buttons at the bottom: 'Disband Lobby' and 'Create Lobby'.

Hier kann eine Lobby erstellt werden. Will man eine Private Lobby erstellen, muss zudem ein Passwort gesetzt werden, welches von beitretenden Benutzern eingegeben werden muss.



Als Lobby Leader kann ab 2 Spielern ein Spiel gestartet werden. Er erhält zusätzliche Funktionen wie das entfernen von Beigetretenen Spielern. Wenn er die Lobby verlässt, wird der zuletzt Beigetretene Spieler zum Lobby Leader ernannt, er kann aber auch vorher einen Spieler ernennen.

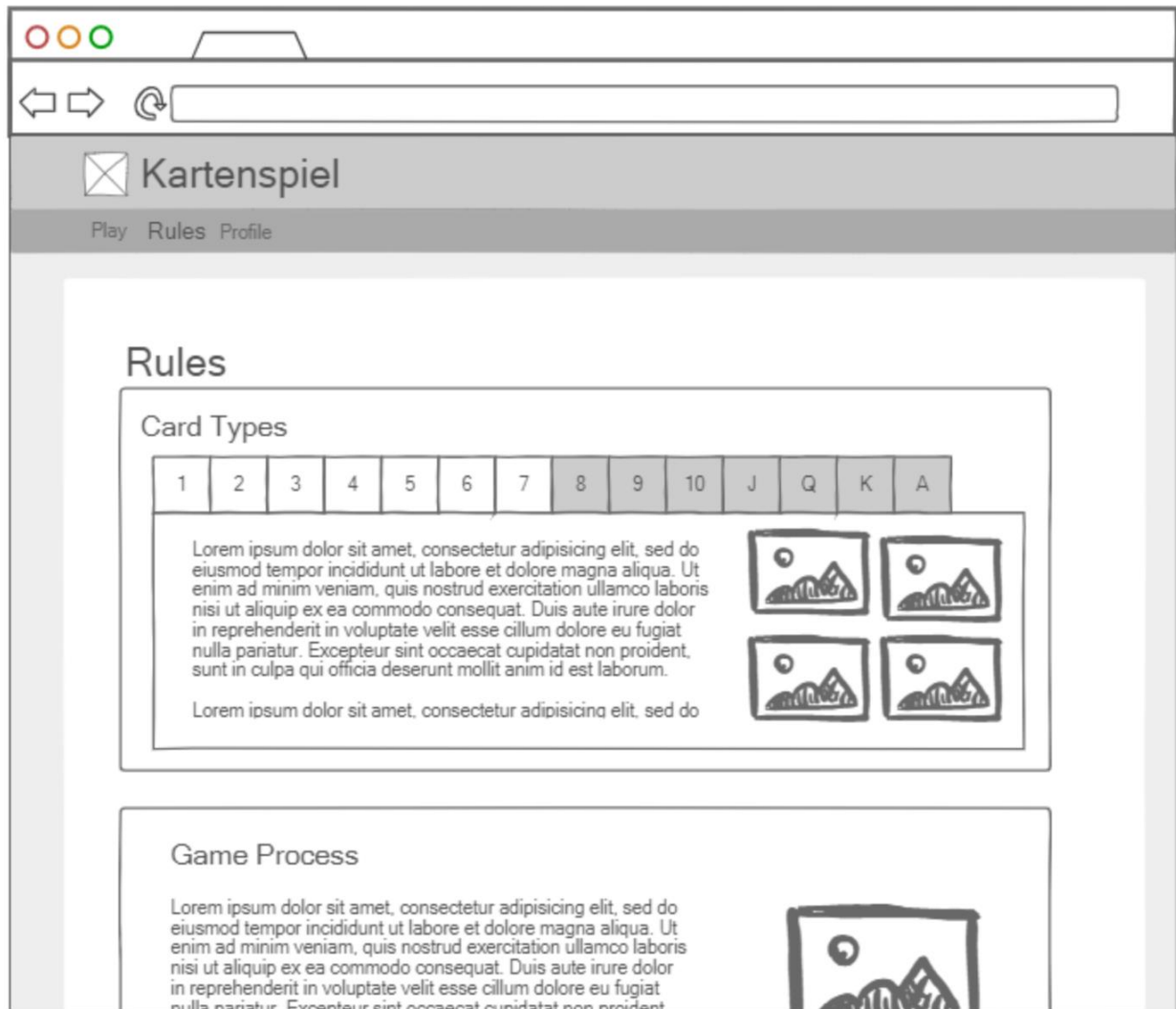
### 2.4.3 Profilseite



Hier können Informationen zum eigenen Profil oder anderen, registrierter Benutzern abgerufen werden.

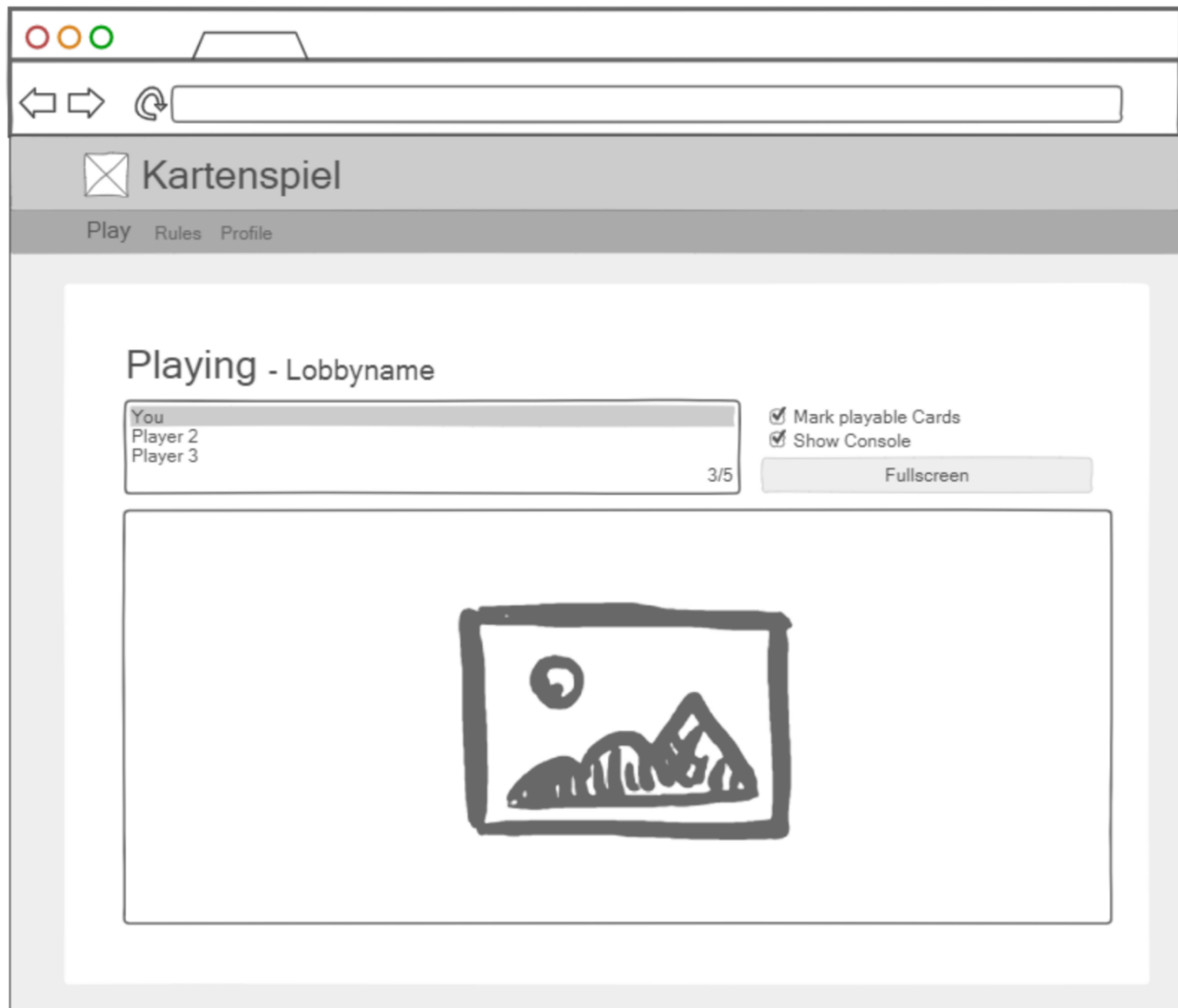


## 2.4.4 Spielregeln



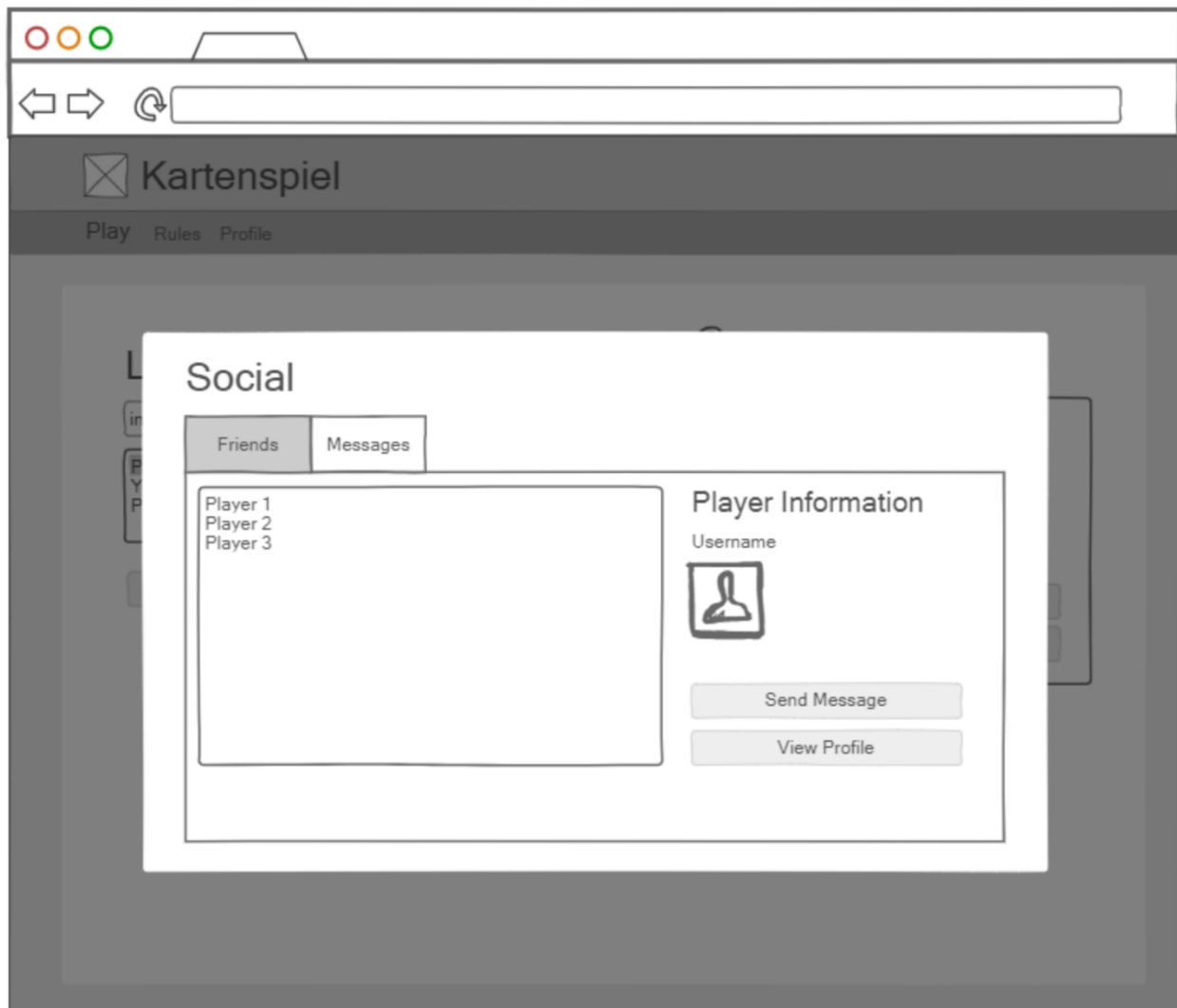
Die Bedeutung der Einzelnen Karten und alles weitere zum Spielablauf kann hier abgerufen werden.

## 2.4.5 Spiellogik



Hier ist ein gestartetes Spiel zu sehen. Das darstellen der Spielelogik ist sehr aufwändig und können wir deshalb aus zeitlichen Gründen nicht realisieren. Wir werden dies aber in Form eines Prototypen erstellen. Spielzüge sollen über die Konsole ausgelesen werden können.

## 2.4.6 Nachrichten und Freunde



Soziale Funktionen wie die Freundesliste und das Nachrichtensystem sollen in einem Overlay den registrierten Benutzern zur Verfügung stehen.

### 2.4.7 Meilensteine und Arbeitsschritte

1. SQL Datenbank erstellen
2. Login- und Registrierungssystem erstellen
3. Lobby Handling
4. Profilseite machen
5. Spiellogik
6. Nachrichten und Freunde

### 2.4.8 Zuteilung

Meilenstein	Wer	Wann
1	Beide	05.12.2020
2	Gian	12.12.3020
3	Manuel	19.12.2020
4	Manuel	19.12.2020
5	Beide	09.01.2020
6	Gian	09.01.2020

## 3 Technologie

### 3.1 Server / Backend

#### 3.1.1 Node.js

JavaScript ist eine im Browser integrierte Clientseitige Programmiersprache fürs Web. Mit Node.js lässt sich jedoch JavaScript auch ausserhalb des Browsers einsetzen. Wir werden unseren Webserver mit Node realisieren, da der Datenaustausch über das JSON Format simpel und die für unser Spiel benötigte realtime mit Websockets verfügbar ist.

#### 3.1.2 Frameworks

##### 3.1.2.1 Express

Um unsere API zu gestalten, werden wir auf das Web Framework Express setzen. Es stellt uns viele Funktionen bereit.

##### 3.1.2.2 TypeORM

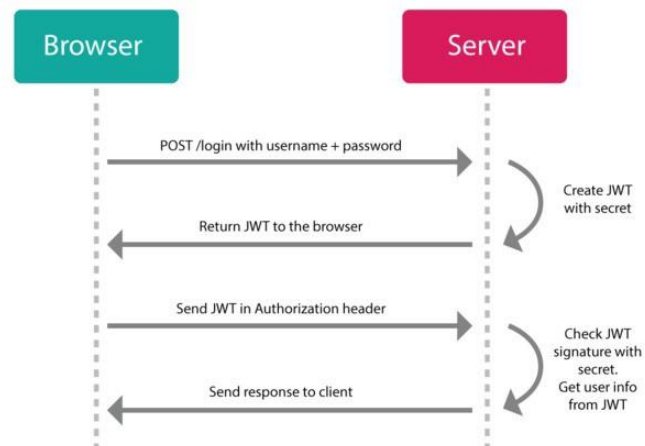
TypeORM ist ein für Node entwickelter OR Mapper und übernimmt für uns das mapping der Datensätze zu Klassen.

##### 3.1.2.3 Socket.io

Socket.io ist eine Bibliothek basierend auf Websockets und erlaubt es uns, eine realtime, beidseitige Verbindung zwischen Client und Server herzustellen. Wir benötigen diese Funktionalität ist unserem Multiplayer Spiel, da ein Spielzug direkt dem Server und den anderen Spielern mitgeteilt werden muss. Nebst dem Server muss dies auch auf dem Client implementiert werden.

##### 3.1.2.4 JWT Bearer Token

JSON Web Token ist ein Verfahren, um zwischen Client und Server eine Sichere Verbindung aufzubauen. Wir bieten zwar auch die Möglichkeit an, als Gast zu spielen, jedoch sollen Registrierte Benutzer über erweiterte Funktionalitäten Verfügen. Sie können so zum Beispiel ein Profil erstellen, eine Freundesliste Verwalten und Spielstatistiken einsehen. Das Bild zeigt den Ablauf bei einer Authentifikation Abfrage.



## 3.2 Client / Frontend

### 3.2.1 Vue.js

Vue.js ist ein clientseitiges Framework basierend auf JavaScript. Es wird zum Erstellen von Single Page Applications verwendet.

### 3.2.2 Frameworks

#### 3.2.2.1 Vuetify

Es ist das populärste Framework für Vue.js. Vuetify sehr viele Komponenten zur Verfügung und hält sich an die Material Design Spezifikationen. Es nimmt uns einen grossen Teil der Arbeit für die Anforderungen an das Design ab, wie etwa das Responsive Design für Desktop, Tablet und Mobile.

## 3.3 Datenbank

### 3.3.1 MS SQL

Die Datenbank Technologie wurde uns so vorgegeben. Um eine einfache Datenbankverbindung mit dem Backend herstellen zu können, klärten wir vorerst ab, ob unsere Backendtechnologie dies zulässt. Wir werden für unser Node.js backend das Package mssql verwenden, welches als Treiber alle gängigen moderne Versionen von SQL Servern unterstützt.

## 3.4 Webserver

Ein Webserver ist uns als Technische Vorgabe vorausgesetzt. In der Entwicklungsumgebung wird dieser nicht benötigt, da Frontend und Backend selbst gebildet werden können. In der Produktivumgebung muss das Frontend aber gebildet und die resultierenden Dateien auf einem Webserver gehostet werden. Wir werden dafür Express verwenden.

## 4 Anhang

### 4.1 Glossar

#### 4.1.1 Abkürzungen

MSSQL - Microsoft SQL Server

#### 4.1.2 Allgemeine und Technische Bezeichnungen

-

#### 4.1.3 Projektbezogene Bezeichnungen

-

### 4.2 Verzeichnis

#### 4.2.1 Grafiken

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

<http://www.jenazads.com/libraries/Implementing-JSON-Web-Token-middleware-in-Golang>

#### 4.2.2 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/>

<https://www.npmjs.com/package/mssql>