

# Konzeptbeschreibung

**Team: G4T3**

Mitglied 1: Diana Gründlinger, 01480296

Mitglied 2: Marcel Alexander Huber, 11909190

Mitglied 3: Thomas Klotz, 11909202

Mitglied 4: Aaron Targa, 11914514

Mitglied 5: Matthias Thalmann, 11914515

**Proseminargruppe: 4**

**Datum: 18.03.2021**

## 1. Systemüberblick

Bei dieser Anwendung handelt es sich um ein Gesellschaftsspiel. Dabei müssen Teams durch verschiedene Aktivitäten Begriffe erklären beziehungsweise erraten, um zu gewinnen.

Gespielt wird mit Hilfe eines IoT-Würfels und einer Web-Anwendung. Von jedem Team muss mindestens ein Mitglied in der Web-Anwendung mit einem Account eingeloggt sein und sich im Spiel befinden.

Das Spiel wird in Runden gespielt. In jeder Runde muss ein Spieler seinem Team einen Begriff innerhalb einer gegebenen Zeit erklären. Das Zeitlimit, die erreichbaren Punkte und die Aktivität werden durch einen Würfelwurf ermittelt. Wird der Begriff innerhalb der gegebenen Zeit erraten, bekommt das Team Punkte. Tritt ein Regelverstoß auf, gibt es für das Team Punkteabzug. Danach ist das nächste Team an der Reihe.

Die zu erratenden Begriffe werden zufällig einem Begriffskatalog entnommen. Die Begriffe sind jeweils einem Thema zugeordnet, welches für jedes Spiel vor Spielbeginn festgelegt wird.

Beendete Spiele werden gespeichert und in Statistiken visualisiert.

Begriffe, Themen und Spiele können von Spielerverwaltern und Administratoren verwaltet werden.

## 2. Use Cases

### 2.1 Akteure

#### **User**

Ein User besitzt einen Account im System, kann sich ein- und ausloggen, und seine eigenen Userdaten verwalten.

#### **Spieler**

Ein Spieler kann an Spielen teilnehmen und Statistiken zu vergangenen Spielen einsehen.

#### **Spieleverwalter**

Ein Spieleverwalter pflegt den Katalog an Rate-Begriffen und kann laufende Spiele einsehen.

#### **Admin**

Ein Admin verwaltet Spiele und User.

#### **Würfel**

Ein Würfel sendet Daten an das System.

## 2.2 Use-Case-Diagramm



## 2.3 Use-Cases

### 2.3.1 Akteur: User

#### Registrieren

- **Vorbedingung:** Der User hat noch keinen Account im System und befindet sich auf dem Anmeldebildschirm.
- **Basisablauf:**
  1. Der User klickt auf "Register".
  2. Dann gibt er in ein Registrierungsformular seinen gewünschten Usernamen und ein Passwort ein.
  3. Das Passwort muss durch eine zweite Eingabe bestätigt werden.
  4. Die Eingabe wird durch einen Buttonklick bestätigt.
- **Nachbedingung:** Der User besitzt einen Account, ist angemeldet und befindet sich in der Lobby.
- **Alternativen:**
  0. Der Registrierungsvorgang wird, z.B. durch einen Klick auf „Login“, durch den User abgebrochen.
  2. Der gewählte Username ist bereits vergeben. Eine Fehlermeldung wird angezeigt.
  3. Die erste und zweite Eingabe des Passworts stimmen nicht überein. Eine Fehlermeldung wird angezeigt.
- **Involvierte Klassen:** User

### Einloggen

- **Vorbedingung:** Der User hat einen Account im System, ist nicht angemeldet und befindet sich somit auf dem Anmeldebildschirm.
- **Basisablauf:** Der User gibt seine Anmeldedaten ein, klickt auf "Login" und wird zur Lobby weitergeleitet.
- **Nachbedingung:** Der User ist angemeldet und befindet sich in der Spielelobby.
- **Alternativen:** Die eingegebenen Userdaten sind falsch. Der Login schlägt fehl. Es wird eine Fehlermeldung angezeigt, in der der User darauf hingewiesen wird, dass Username oder Passwort falsch sind.
- **Involvierte Klassen:** User

### Ausloggen

- **Vorbedingung:** Der User ist angemeldet.
- **Basisablauf:** Der User klickt auf "Logout".
- **Nachbedingung:** Der User kommt auf den Anmeldebildschirm und ist nicht mehr angemeldet.
- **Alternativen:** -
- **Involvierte Klassen:** User

### Eigene Userdaten ändern

- **Vorbedingung:** Der User ist eingeloggt.
- **Basisablauf:**
  1. Der User klickt auf „Settings“.
  2. Einstellungsdialog wird angezeigt. Es können sowohl Username als auch Passwort geändert werden.
  3. Um die geänderten Daten zu bestätigen, muss das alte Passwort eingegeben werden.
  4. Die Änderungen werden durch das Klicken auf "Save changes" bestätigt.
- **Nachbedingung:** Die neuen Userdaten wurden übernommen. Der Einstellungsdialog wurde geschlossen. Der User wird ausgeloggt und befindet sich auf dem Anmeldebildschirm.
- **Alternativen:**
  3. Das eingegebene alte Passwort ist falsch. Es wird eine Fehlermeldung angezeigt.
  4. Der Username ist bereits vergeben. Es wird eine Fehlermeldung angezeigt.
- **Involvierte Klassen:** User

### 2.3.2 Akteur: Spieler

#### Spiel erstellen

- **Vorbedingung:** Der Spieler ist eingeloggt und befindet sich in der Lobby.
- **Basisablauf:**
  1. Der Spieler klickt auf "Setup a new game". Anschließend befindet er sich auf der Konfigurationsseite für das neue Spiel.
  2. Er legt die Anzahl der Teams, die maximal erreichbare Punkteanzahl und ein Themengebiet fest. Die ID des Würfels muss in das Feld "TimeFlip code" eingegeben werden. Das Mapping für die Würfelseiten muss festgelegt werden.
  3. Für die Konfiguration des Würfels kann eine Hilfsfunktion aufgerufen werden.
  4. Der Spieler bestätigt die Konfiguration durch Klicken von "Create".
- **Nachbedingung:** Der Spieler befindet sich als Host des soeben erstellten Spiels im Warteraum dieses Spiels. Ein Zahlencode wird angezeigt, der von anderen Spielern zum Beitreten des Spiels benötigt wird.
- **Alternativen:**
  0. Das Erstellen des Spiels wird abgebrochen. Der Spieler kehrt in die Lobby zurück.

4. Der Würfel ist nicht registriert. Es wird eine Fehlermeldung angezeigt. Der Spieler befindet sich noch immer in der Spielerstellung.

- **Involvierte Klassen:** Game, Category, Dice, User, Team

#### Hilfsfunktion aufrufen

- **Vorbedingung:** Der Spieler befindet sich auf der Konfigurationsseite eines neuen Spiels.
- **Basisablauf:** Die Hilfsfunktion für die Würfelkonfiguration wird durch Klicken von einem Fragezeichen-Symbol aufgerufen.
- **Nachbedingung:** Dem Spieler werden nützliche Detailinformationen für die Würfelkonfiguration angezeigt.
- **Alternativen:** -
- **Involvierte Klassen:** -

#### Spiel beitreten

- **Vorbedingung:** Der Spieler befindet sich in der Lobby und ist in keinem anderen aktiven Spiel.
- **Basisablauf:** Der Spieler gibt einen Zahlencode, den er vom Host mitgeteilt bekommen hat, ein und bestätigt dann die Eingabe.
- **Nachbedingung:** Der Spieler befindet sich im Warteraum des Spiels, dem er gerade beigetreten ist.
- **Alternativen:** Der eingegebene Zahlencode ist unbekannt. Es wird eine Fehlermeldung angezeigt. Der Spieler befindet sich immer noch in der Lobby.
- **Involvierte Klassen:** Game, User

#### Team beitreten

- **Vorbedingung:** Der Spieler befindet sich im Warteraum eines Spiels. Es wird eine Übersicht über alle verfügbaren Teams angezeigt.
- **Basisablauf:** Der Spieler kann mit einem einfachen Klick auf das gewünschte Team dem Team beitreten.
- **Nachbedingung:** Der Spieler ist in einem Team und befindet sich immer noch im Warteraum.
- **Alternativen:** -
- **Involvierte Klassen:** Game, Team, User

#### Spiel starten

- **Vorbedingung:** Der Spieler ist Host eines neu erstellten Spiels und befindet sich im Warteraum dieses Spiels.
- **Basisablauf:** Der Host klickt auf "Start Game" und wartet, bis alle Spieler im Warteraum ihre Spielteilnahme bestätigt haben. Danach wird das Spiel automatisch gestartet.
- **Nachbedingung:** Der Spieler befindet sich zusammen mit den anderen Spielern im Spielraum. Das System wählt automatisch, welches Team an der Reihe ist. Durch Werfen des Würfels kann der erste Spielzug eingeleitet werden.
- **Alternativen:** -
- **Involvierte Klassen:** Game

#### Spielteilnahme bestätigen

- **Vorbedingung:** Der Spieler befindet sich im Warteraum und der Host will das Spiel starten.
- **Basisablauf:** Der Spieler bestätigt die Spielteilnahme mit einem Button-Klick.
- **Nachbedingung:**
  - a) Der Spieler wartet darauf, dass noch weitere Spieler die Spielteilnahme bestätigen und befindet sich weiterhin im Warteraum.
  - b) Der Spieler befindet sich mit den anderen Spielern im Spielraum.

- **Alternativen:** -
- **Involvierte Klassen:** Game, User

**Rate-Erfolg des Gegners bestätigen**

- **Vorbedingung:** Der Spieler befindet sich in einem laufenden Spiel und ist Teil eines Teams, das gerade nicht am Zug ist. Das Team, das am Zug ist, konnte den Begriff vor Zeitablauf erraten. Der Würfel wurde gedreht, um den Timer zu stoppen.
- **Basisablauf:** Der Spieler bestätigt den Erfolg mit einem Klick auf einen Button.
- **Nachbedingung:** Das gegnerische Team bekommt die berechneten Punkte für den erratenen Begriff auf Ihr Punktekonto gutgeschrieben, danach ist das nächste Team an der Reihe.
- **Alternativen:** -
- **Involvierte Klassen:** Game, Team

**Rate-Misserfolg des Gegners bestätigen**

- **Vorbedingung:** Der Spieler befindet sich in einem laufenden Spiel. Er ist Teil eines Teams, das gerade nicht am Zug ist. Der Timer ist abgelaufen und der gesuchte Begriff wurde nicht vor Zeitablauf erraten.
- **Basisablauf:** Der Spieler bestätigt den Misserfolg mit einem Klick auf einen Button.
- **Nachbedingung:** Es werden keine Punkte gutgeschrieben. Das nächste Team ist an der Reihe.
- **Alternativen:** -
- **Involvierte Klassen:** Game, Team

**Regelverstoß des Gegners vermerken**

- **Vorbedingung:** Der Spieler befindet sich in einem laufenden Spiel. Er ist Teil eines Teams, das gerade nicht am Zug ist. Das gegnerische Team hat beim Erklären eines Begriffs gegen eine Regel verstoßen.
- **Basisablauf:** Der Regelverstoß wird durch einen Button-Klick bestätigt. Der Zug ist damit beendet und es wird dem Team, das gegen eine Regel verstoßen hat, ein Punkt abgezogen.
- **Nachbedingung:** Der Regelverstoß ist im System vermerkt und die Gesamtpunktezahle wurde angepasst.
- **Alternativen:** -
- **Involvierte Klassen:** Game, Team

**Spielraum verlassen**

- **Vorbedingung:** Der Spieler befindet sich in einem Warteraum, einem Spielraum oder sieht gerade die Rangliste ein, die nach Spielablauf angezeigt wird.
- **Basisablauf:** Um das Spiel zu verlassen, wird ein Button geklickt.
- **Nachbedingung:**
  - a. Der Spieler hat einen Warteraum verlassen. Er befindet sich in der Lobby.
  - b. Der Spieler hat ein laufendes Spiel verlassen. Er befindet sich in der Lobby und kann jederzeit wieder dem aktiven Spiel beitreten durch Klicken auf „Current Game“.
  - c. Der Spieler hat ein beendetes Spiel verlassen. Er befindet sich in der Lobby.
- **Alternativen:** -
- **Involvierte Klassen:** Game

**Spielerprofil einsehen**

- **Vorbedingung:** Der Spieler ist eingeloggt.
- **Basisablauf:** Der Spieler klickt auf einen beliebigen angezeigten Usernamen. Es kann sich dabei auch um den eigenen Usernamen handeln.

- **Nachbedingung:** Das Profil des gewählten Users wird angezeigt, inklusive interessanter Statistiken über vergangene Spiele.
- **Alternativen:** -
- **Involvierte Klassen:** User

#### Spieler zu Spiel hinzufügen

- **Vorbedingung:** Der Spieler befindet sich als Host im Warteraum eines Spiels.
- **Basisablauf:**
  1. Der Host klickt bei einem Team auf „Add Player“. Ein Eingabedialog öffnet sich.
  2. Durch Eingeben eines Usernamens wird ein Spieler ausgewählt.
  3. Der gewählte User muss sein Passwort eingeben, um dem Spiel beitreten zu können.
  4. Die Auswahl wird bestätigt durch Klicken von „Add“.
- **Nachbedingung:** Der Eingabedialog wurde geschlossen. Der Spieler befindet sich wieder im Warteraum. Der neu hinzugefügte Spieler befindet sich im gewählten Team.
- **Alternativen:**
  3. Das eingegebene Passwort ist falsch. Eine Fehlermeldung wird angezeigt. Der User wird nicht hinzugefügt, der Host befindet sich immer noch im Dialogfenster.
- **Involvierte Klassen:** Game, Team, User

#### 2.3.3 Akteur: Spieleverwalter

##### Begriffe hinzufügen

- **Vorbedingung:** Der Spieleverwalter ist eingeloggt und befindet sich in der Begriffskatalog-Ansicht.
- **Basisablauf:**
  1. Der Spieleverwalter klickt auf das 3-Punkte-Menü einer Kategorie rechts in der Tabellenansicht. Es öffnet sich ein Auswahlmenü.
  2. Der Spieleverwalter klickt auf „Add expression(s)“. Es öffnet sich ein Eingabedialog.
  3. Es kann ein Begriff eingegeben und durch Betätigen der Enter-Taste bestätigt werden. Weitere Begriffe können im gleichen Textfeld auf gleiche Weise hinzugefügt werden.
  4. Die Eingabe(n) werden durch Klicken von „Save“ gespeichert.
- **Nachbedingung:** Der Spieleverwalter befindet sich in der Begriffskatalog-Ansicht. Die neu eingegebenen Begriffe wurden in den Begriffskatalog in der gewählten Kategorie aufgenommen. Durch Klicken von „Show Expressions“ im 3-Punkte-Menü kann die Änderung überprüft werden.
- **Alternativen:**
  3. Die Eingabe wird durch den Spieleverwalter abgebrochen. Es wurden keine Begriffe gespeichert. Der Spieleverwalter befindet sich in der Begriffskatalog-Ansicht.
- **Involvierte Klassen:** Expression, Category

##### Begriffe löschen

- **Vorbedingung:** Der Spieleverwalter ist eingeloggt und befindet sich in der Begriffskatalog-Ansicht.
- **Basisablauf:**
  1. Der Spieleverwalter klickt auf das 3-Punkte-Menü einer Kategorie rechts in der Tabellenansicht. Es öffnet sich ein Auswahlmenü.
  2. Der Spieleverwalter klickt auf „Show expressions“. Es öffnet sich die Begriffsübersicht.
  3. Durch Klicken des Mülleimer-Symbols kann ein Begriff gelöscht werden.
- **Nachbedingung:** Der Begriff ist nicht mehr im Begriffskatalog vorhanden.
- **Alternativen:** -

- **Involvierte Klassen:** Expression

#### Begriffe importieren

- **Vorbedingung:** Der Spielverwalter ist eingeloggt und befindet sich in der Begriffskatalog-Ansicht.
- **Basisablauf:**
  1. Durch Klicken des Buttons "Create" oder eines 3-Punkte-Menüs in der Tabelle werden Optionen für den Begriffsimport geöffnet.
  2. Je nach Auswahl können Begriffe zu einer oder mehreren Kategorien hinzugefügt werden, durch Hochladen einer JSON-Datei oder CSV-Datei. Es taucht ein Datei-Explorer-Fenster auf in welchem eine Datei ausgewählt werden kann.
  3. Durch einen Button-Klick wird die Datei hochgeladen und die Begriffe importiert.
- **Nachbedingung:** Die importierten Begriffe wurden in den Begriffskatalog aufgenommen. Der Spielverwalter befindet sich in der Begriffskatalog-Ansicht.
- **Alternativen:** Der Import schlägt fehl. Eine Fehlermeldung wird angezeigt. Der Spielverwalter befindet sich in der Begriffskatalog-Ansicht.
- **Involvierte Klassen:** Expression, Category

#### Offene Spielräume einsehen

- **Vorbedingung:** Der Spielverwalter ist eingeloggt und befindet sich in der Spielelobby. Es gibt offene Spiele.
- **Basisablauf:** Durch Anklicken von „Open Games“ wird eine Ansicht geöffnet, in der alle offenen Spiele inklusive interessanter Informationen angezeigt werden.
- **Nachbedingung:** Dem Spielverwalter werden alle aktuellen Informationen über das laufende Spiel angezeigt.
- **Alternativen:** -
- **Involvierte Klassen:** Game, Team

#### Kategorie hinzufügen

- **Vorbedingung:** Der Spielverwalter ist eingeloggt und befindet sich in der Begriffskatalog-Ansicht.
- **Basisablauf:**
  1. Durch Klicken des Buttons "Create" und im weiteren Verlauf „Create Category“ wird ein Eingabedialog geöffnet.
  2. Es wird eine neue Kategorie eingegeben.
  3. Die Eingabe wird danach mit einem Klick auf „Save“ bestätigt.
- **Nachbedingung:** Der Spielverwalter bekommt die Möglichkeit zu der neu erstellten Kategorie Begriffe hinzuzufügen. Danach wird die Kategorie (mit Begriffen) gespeichert.
- **Alternativen:**
  1. Die Eingabe wird durch den Spielverwalter durch Klicken von „Cancel“ abgebrochen. Es wurden keine neuen Kategorien gespeichert, der Spielverwalter befindet sich in der Begriffskatalog-Ansicht.
- **Involvierte Klassen:** Category

#### Kategorie löschen

- **Vorbedingung:** Der Spielverwalter ist eingeloggt und befindet sich in der Begriffskatalog-Ansicht.
- **Basisablauf:**
  1. Der Spielverwalter klickt auf das 3-Punkte-Menü einer Kategorie rechts in der Tabellenansicht. Es öffnet sich ein Auswahlménü.



2. Durch Klicken von „Delete category“ wird die Kategorie gelöscht.
- **Nachbedingung:** Die Kategorie ist nicht mehr im Katalog vorhanden.
- **Alternativen:**
  2. Die Kategorie konnte nicht gelöscht werden, weil bereits Spiele mit dieser Kategorie gespielt wurden. Die Auswahlmöglichkeit „Delete category“ ist ausgegraut. Durch Klicken auf „Delete category“ schließt sich das Menü. Die Kategorie existiert immer noch, der Spieleverwalter befindet sich in der Begriffskatalog-Ansicht.
- **Involvierte Klassen:** Category

#### 2.3.4 Akteur: Admin

##### Spiel löschen

- **Vorbedingung:** Der Admin ist eingeloggt und befindet sich in der Ansicht der offenen Spiele.
- **Basisablauf:** Durch Klicken von „Close game“ und darauffolgende Bestätigung wird das gewählte Spiel gelöscht.
- **Nachbedingung:** Das gewählte Spiel wurde erfolgreich gelöscht und alle Spieler wurden aus der Sitzung geworfen.
- **Alternativen:** -
- **Involvierte Klassen:** Game

##### User erstellen

- **Vorbedingung:** Der Admin befindet sich auf der Userverwaltungsseite.
- **Basisablauf:**
  1. Der Admin klickt auf „Create User“.
  2. Ein Dialog öffnet sich und man gibt relevante Daten ein.
  3. Die Eingabe wird durch Klicken von „Save user“ bestätigt.
- **Nachbedingung:** Die neuen Daten des Users wurden übernommen. Der User existiert.
- **Alternativen:**
  2. Der Username existiert bereits. Es wird eine Fehlermeldung angezeigt. Es wurde kein neuer User erstellt, der Admin befindet sich noch im Dialogfenster.
- **Involvierte Klassen:** User

##### User bearbeiten

- **Vorbedingung:** Der Admin befindet sich auf der Userverwaltungsseite.
- **Basisablauf:**
  1. Der Admin klickt auf das 3-Punkte-Menü eines Users in der Tabellenansicht. Ein Auswahlménü wird geöffnet.
  2. Durch Klicken auf „Edit user“ wird ein Eingabedialog geöffnet.
  3. Der Admin kann dann die Daten des Users ändern.
  4. Durch Klicken von „Save user“ wird die Eingabe bestätigt.
- **Nachbedingung:** Die neuen Daten des Users wurden übernommen.
- **Alternativen:**
  3. Der Username existiert bereits. Es wird eine Fehlermeldung angezeigt. Die neuen Daten wurden nicht übernommen, der Admin befindet sich noch im Eingabedialog.
- **Involvierte Klassen:** User

##### User löschen

- **Vorbedingung:** Der Admin befindet sich auf der Userverwaltungsseite.
- **Basisablauf:**
  1. Der Admin klickt auf das 3-Punkte-Menü eines Users in der Tabellenansicht. Ein Auswahlménü wird geöffnet.

2. Durch Klicken auf „Delete user“ wird ein Bestätigungsfenster geöffnet.
  3. Das Löschen wird durch Klicken auf „OK“ bestätigt.
- **Nachbedingung:** Der User existiert nicht mehr. Der Admin befindet sich auf der Userverwaltungsseite.
  - **Alternativen:** -
  - **Involvierte Klassen:** User

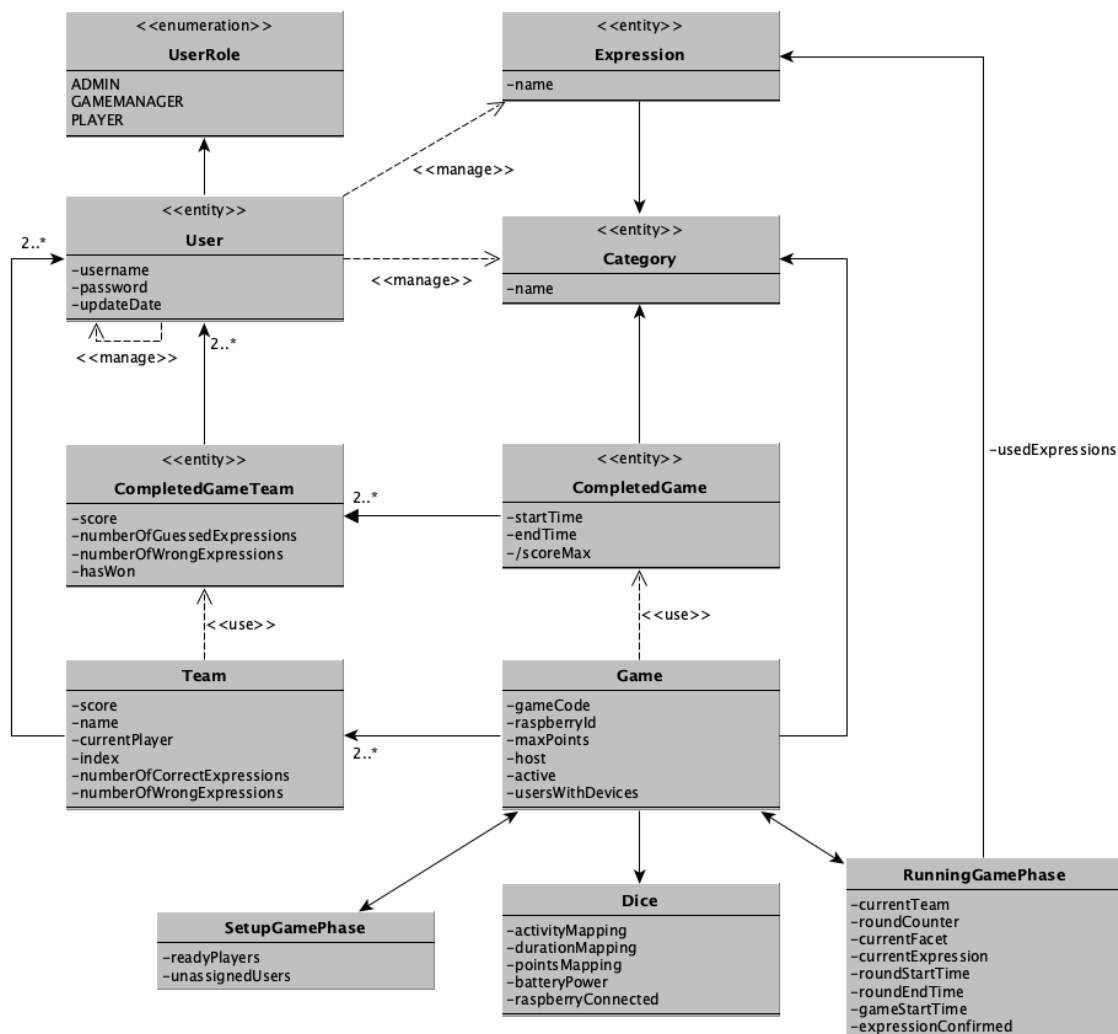
#### 2.3.5 Akteur: Würfel

##### Daten an Minicomputer schicken

- **Vorbedingung:** Der Würfel ist mit dem Minicomputer verbunden.
- **Basisablauf:** Der Würfel sendet seine gemessenen Positionsdaten an den Minicomputer bzw. es werden Daten ausgelesen.
- **Nachbedingung:** Der Minicomputer hat die Daten des Würfels empfangen. In der Konsole werden die empfangenen Daten angezeigt.
- **Alternativen:** Die Verbindung zwischen Würfel und Minicomputer wurde unterbrochen oder die Übertragung ist fehlgeschlagen. Wiederherstellungsmaßnahmen werden ergriffen.
- **Involvierte Klassen:** -

### 3. Klassendiagramm

#### 3.1 Fachliches Klassendiagramm



#### Expression

Diese Klasse stellt einen zu erratenden Ausdruck dar. Eine Expression ist immer genau einer Kategorie zugeordnet.

#### Category

Eine Kategorie dient zur Gruppierung von Begriffen. Vor einem Spiel wird eine Kategorie ausgewählt.

#### User

Benutzer können an Spielen teilnehmen. Je nach Rolle können des weiteren Ausdrücke, Kategorien oder andere Benutzer verwaltet werden.

#### UserRole

Um die Berechtigungen von Benutzern festzulegen, wird diese Klasse benötigt.

#### Game

Die „Game“-Klasse stellt die gesamte Funktionalität bereit, um ein Spiel durchzuführen. Weiters werden spielflussrelevante Informationen zwischengespeichert.

#### Dice

Die Konfiguration des Würfels wird in der „Dice“-Klasse festgehalten.

**Team**

Während einem laufenden Spiel werden teamspezifische Informationen in dieser Klasse verwaltet.

**CompletedGame**

Die Informationen eines abgeschlossenen Spiels werden im „CompletedGame“ gespeichert.

**CompletedGameTeam**

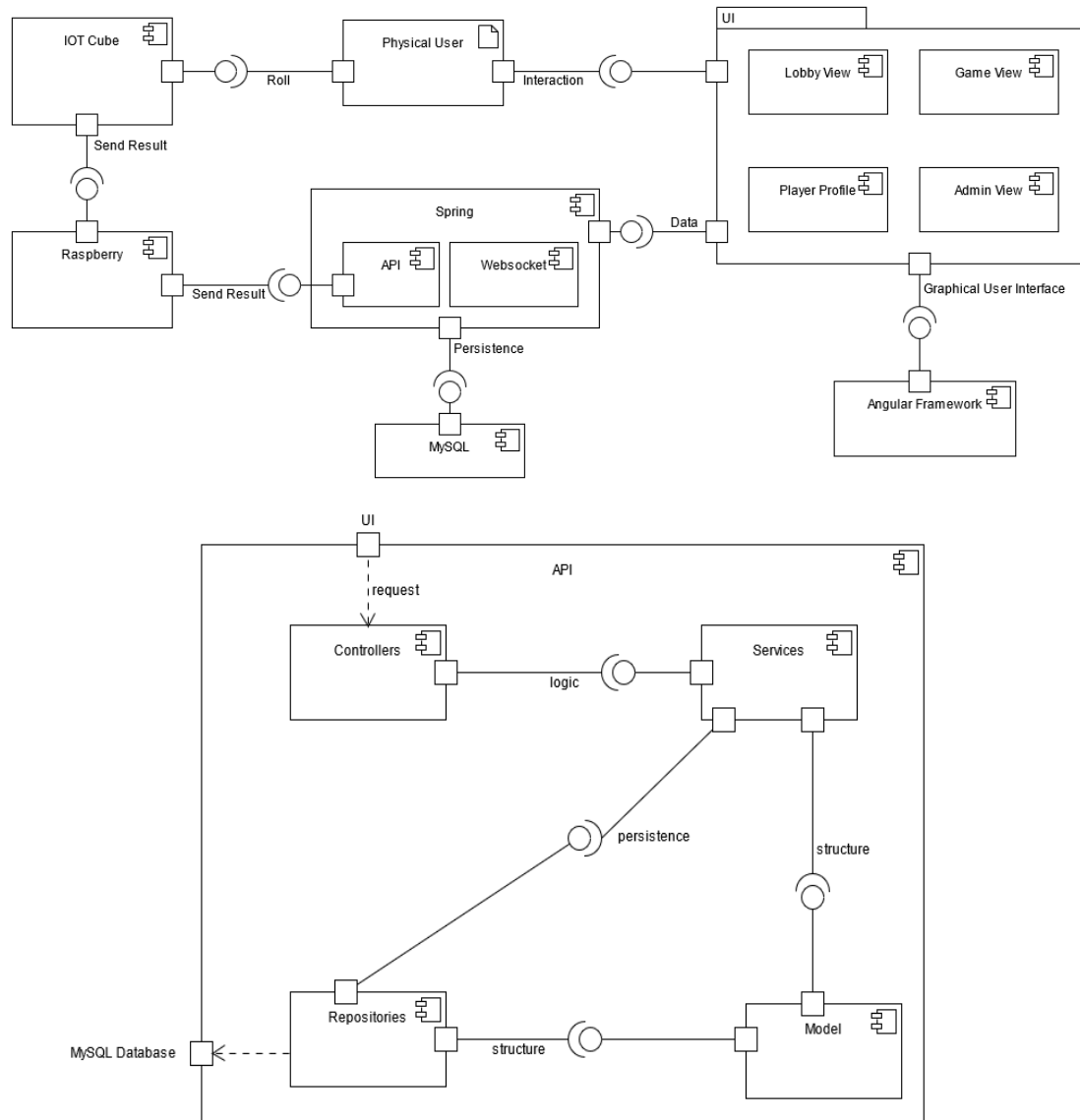
Nach Abschluss eines Spiels werden die Statistiken der Teams in der „CompletedGameTeam“-Klasse hinterlegt.

### 3.2 Technisches Klassendiagramm

Aus lesbarkeitsgründen ist das technische Klassendiagramm nur in der GIT-Repository zu finden (<https://git.uibk.ac.at/informatik/qe/sepsss21/group4/g4t3/tree/master/docs/uml/pdf>).

## 4. SW-Architektur

### 4.1 Komponentendiagramm



### 4.2 Verwendete Technologien

#### 4.2.1 Java

Java ist eine objektorientierte Programmiersprache des Unternehmens Sun Microsystems von Oracle. Der Vorteil von Java ist, dass der Code auf einer virtuellen Maschine ausgeführt wird und daher plattformunabhängig ist. Bei diesem Projekt werden der Serverteil und der Code des Minicomputers in Java geschrieben.

#### 4.2.2 Spring Framework

Spring ist ein quelloffenes Framework, das von vielen Webanwendungen in Kombination mit Java verwendet wird. Es vereinfacht die Entwicklung mit Java und fördert effizientere Entwicklungsstrategien. Spring wird in diesem Projekt verwendet, um eine REST API, welche die Schnittstelle für die Webanwendung bildet, zu realisieren.

#### 4.2.3 Apache Maven

Maven ist ein Build-Management-Tool der Apache Software Foundation zur standardisiert verwalteten Erstellung von Java-Programmen. Maven dient dazu, den Konfigurationsaufwand für die Programmierer zu minimieren.

#### 4.2.4 TypeScript

TypeScript ist eine Programmiersprache, die von Microsoft entwickelt wurde. Sie hat einen Compiler, der TypeScript in „plain“ JavaScript kompiliert, und im Vergleich zu JavaScript optionale Typen, Klassen und Module bietet. TypeScript bietet den Vorteil, dass es durch die optionalen Typen potenzielle Fehler vermeidet und, da es nach Kompilierung JavaScript Code ist, nativ auf allen Browsern ausführbar ist. In diesem Projekt wird TypeScript für den Client verwendet.

#### 4.2.5 Angular

Angular ist ein Frontend-Framework, das auf TypeScript und SASS basiert. Der Vorteil von Angular ist, dass der Umgang zwischen Darstellung und Programmlogik erleichtert wird, und dass es reaktiv ist, d.h. Änderungen von Daten werden direkt in der Darstellung übernommen.

#### 4.2.6 Angular Material

Angular Material ist ein Komponentenframework für Angular. Komponentenframeworks stellen Komponenten bereit, die es einfacher machen, Anwendungen zu programmieren.

#### 4.2.7 MySQL

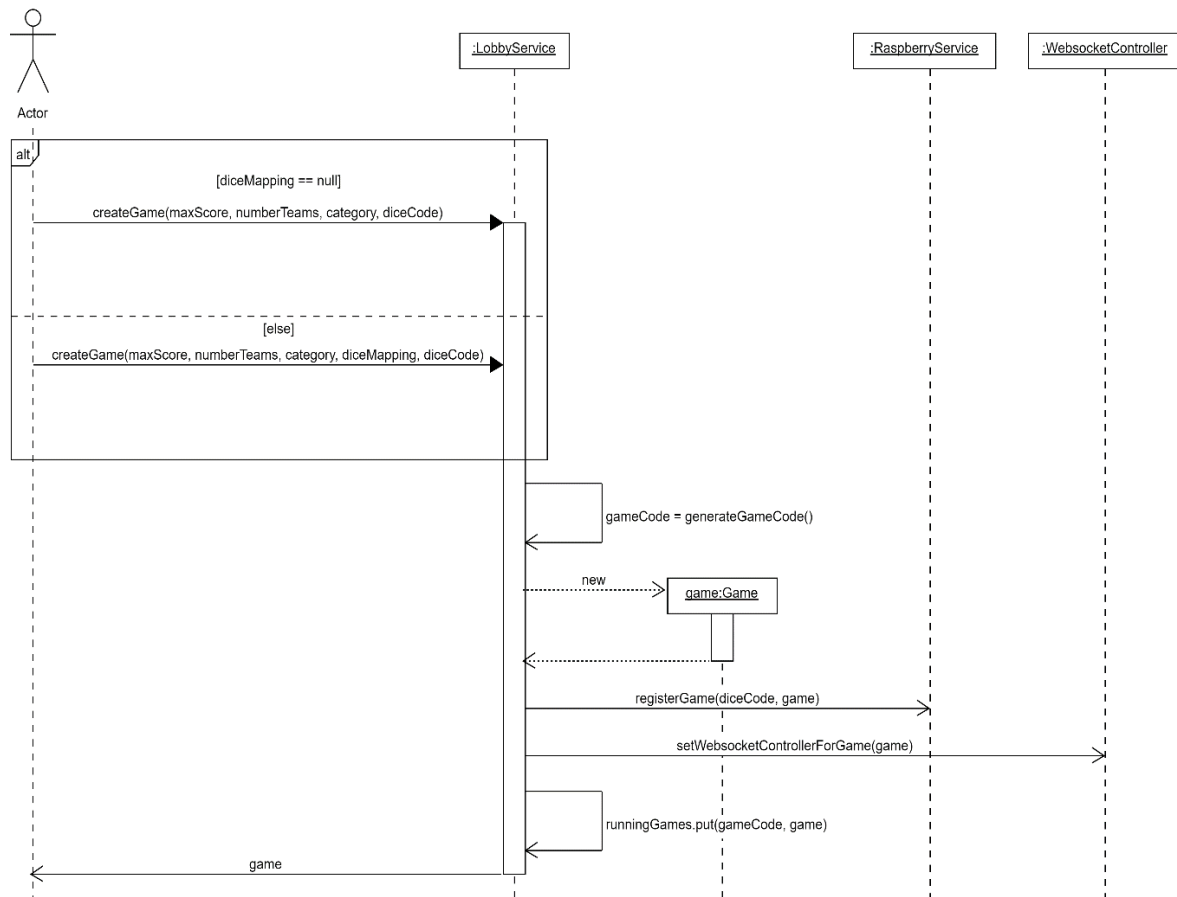
MySQL ist ein quelloffenes, relationales Datenbankmanagementsystem. Es ist ein weltweit weit verbreitetes Datenbanksystem. In diesem Projekt wird MySQL verwendet, da es gut dokumentiert ist, einfach verwendbar und es gut in Verbindung mit dem Spring Framework funktioniert.

#### 4.2.8 JUnit

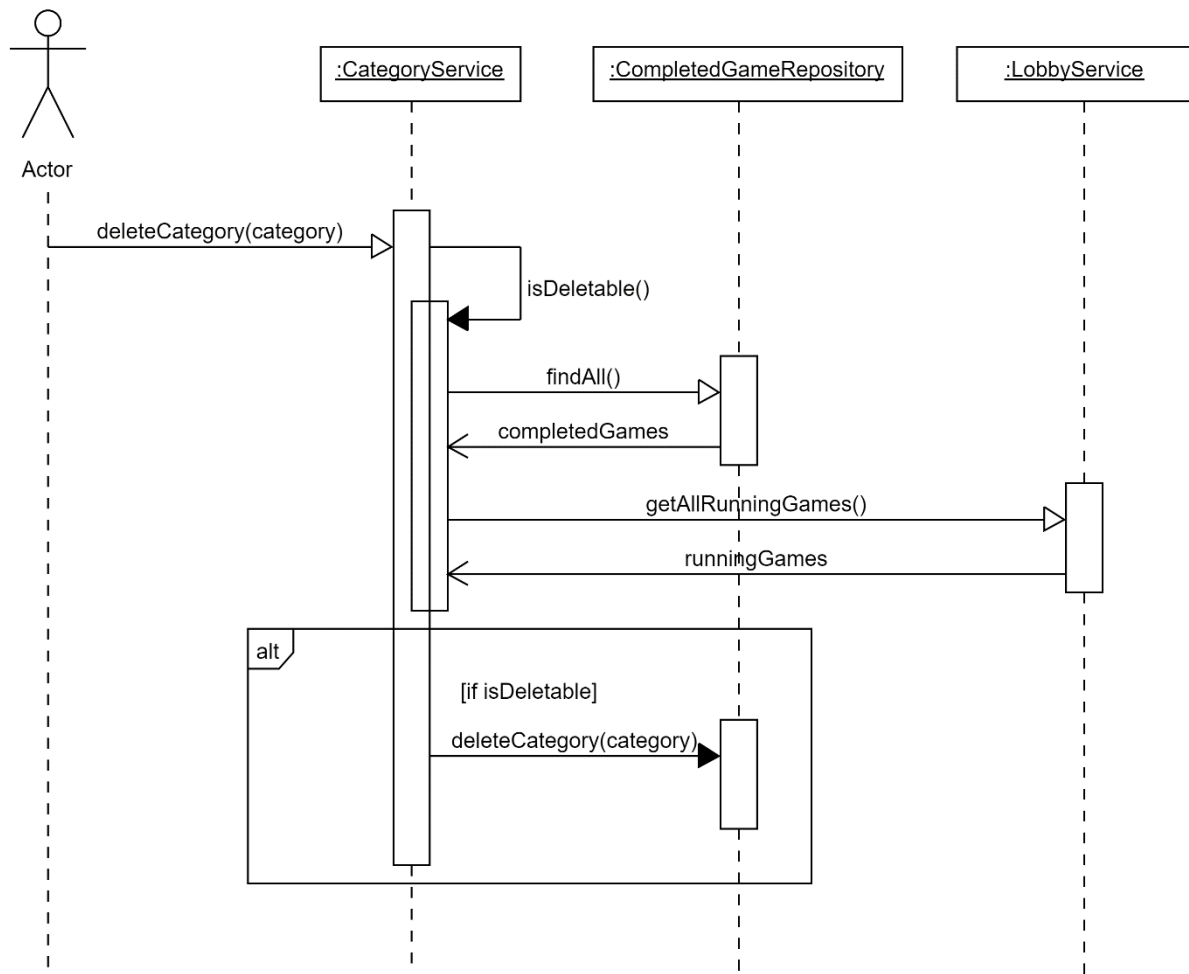
JUnit ist das Standard-Framework zum Testen von Java-Programmen und wird daher von vielen Entwicklerwerkzeugen standardmäßig unterstützt. Es wird in diesem Projekt verwendet, da man damit einfach automatisierte Unit Tests schreiben kann, die zu einer besseren Qualität der Software führen.

## 4.3 Laufzeitschicht

## 4.3.1 CreateGame

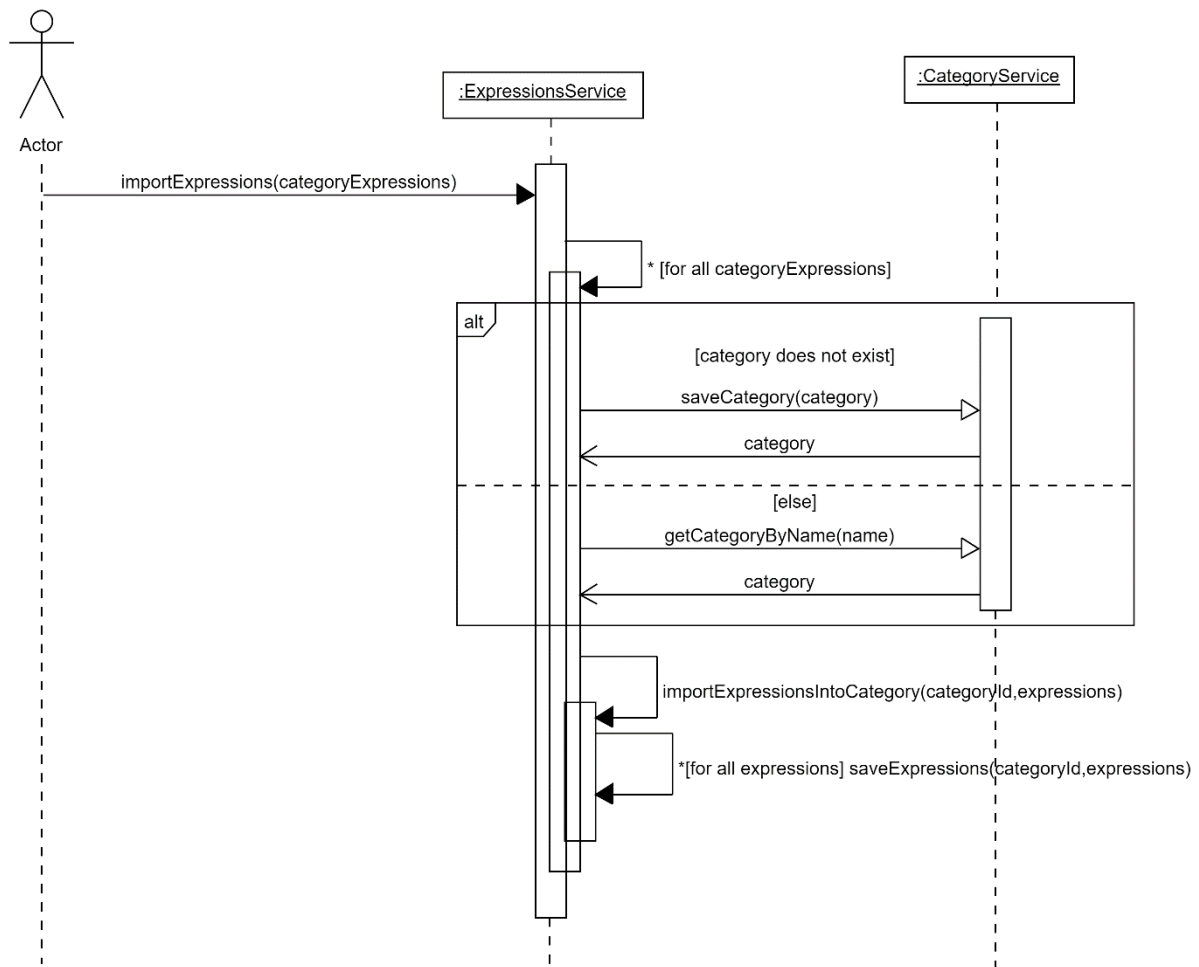


## 4.3.2 DeleteCategory

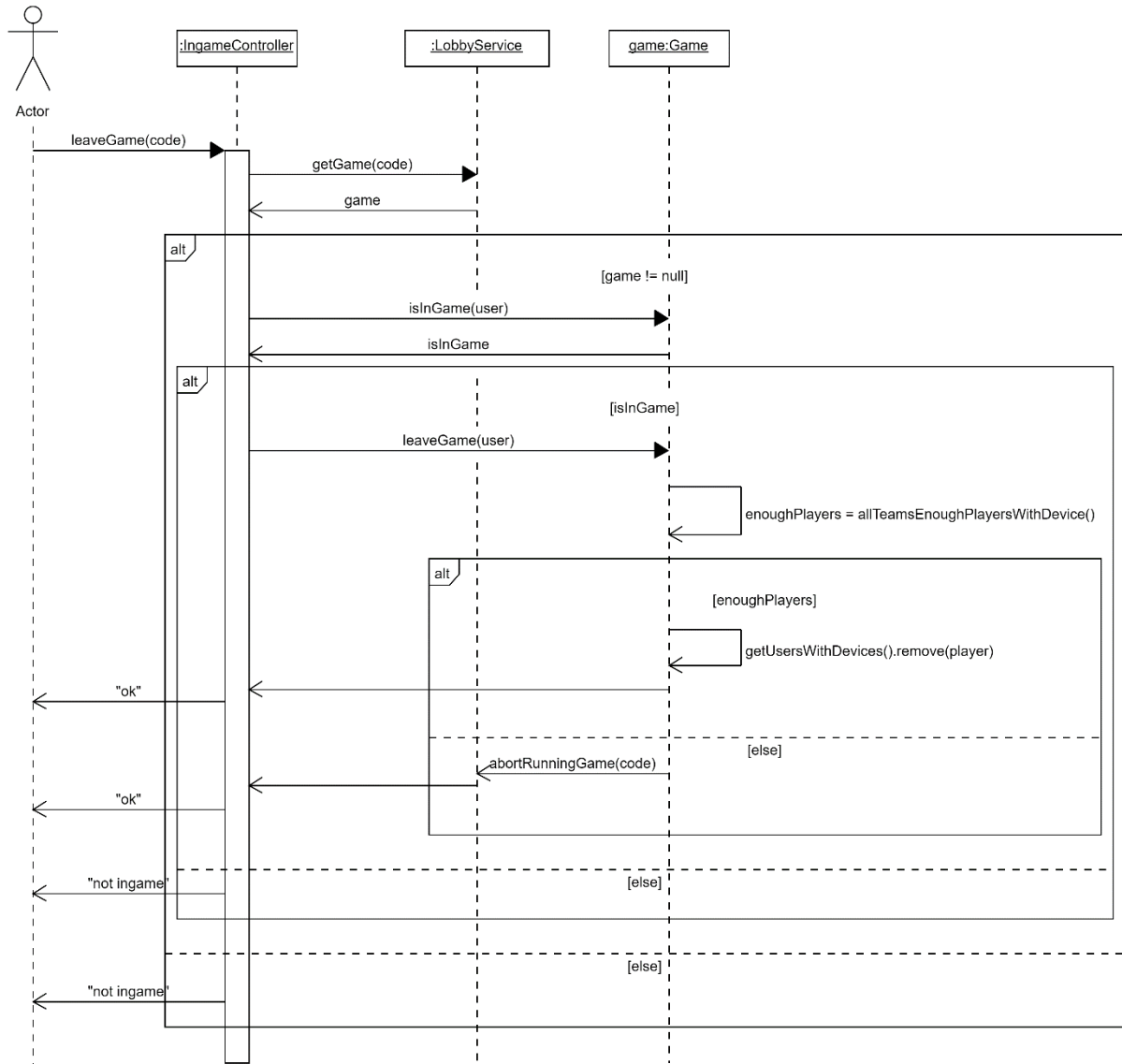




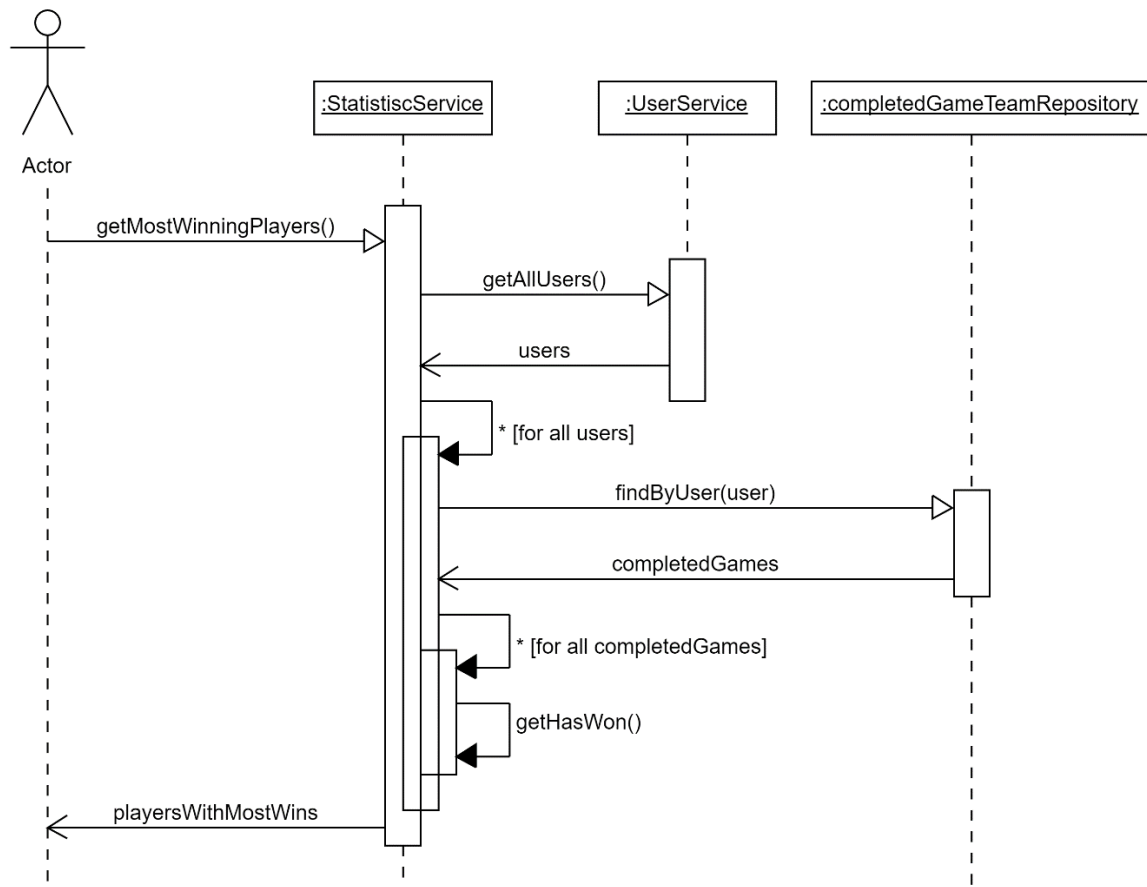
## 4.3.3 Import Expressions



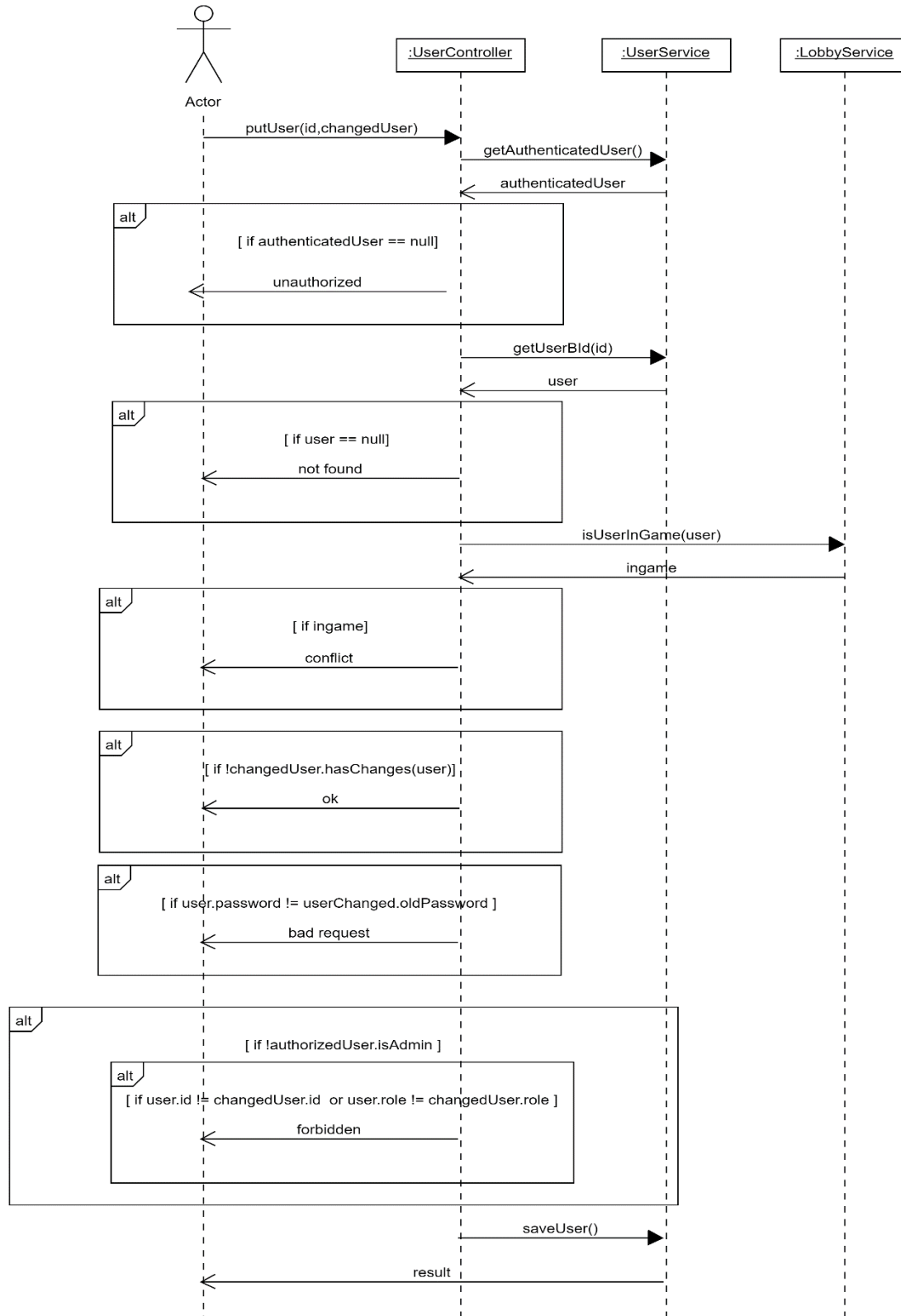
## 4.3.4 LeaveGame



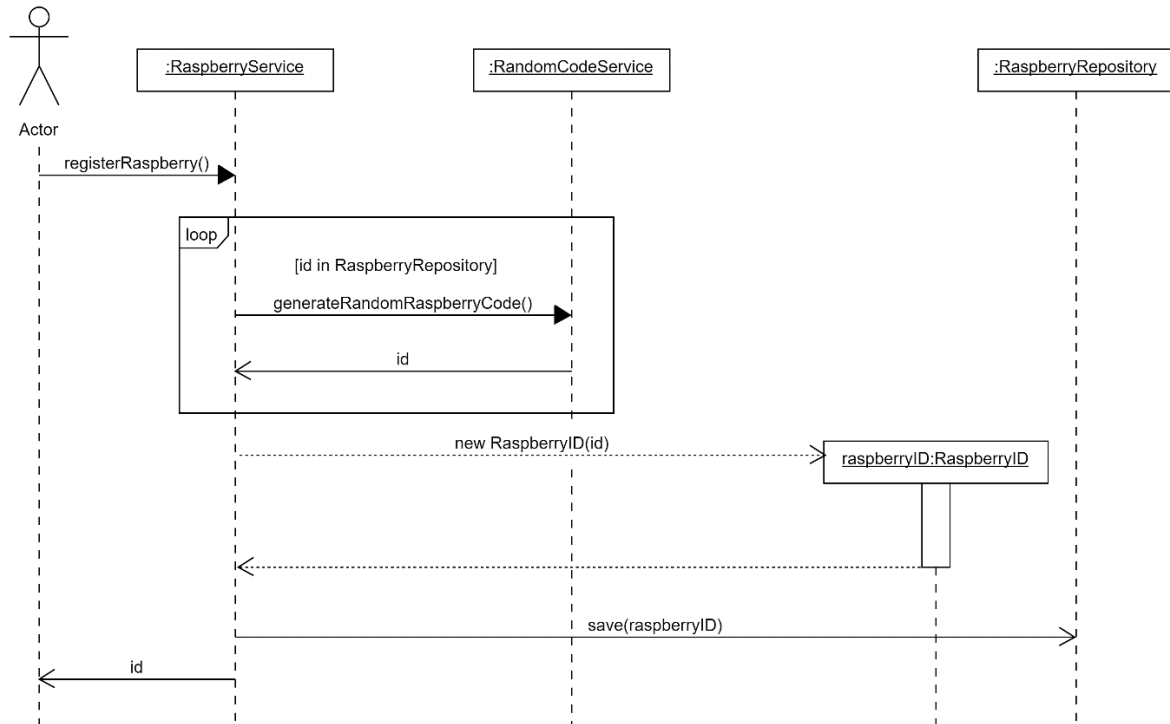
## 4.3.5 MostWinningPlayers



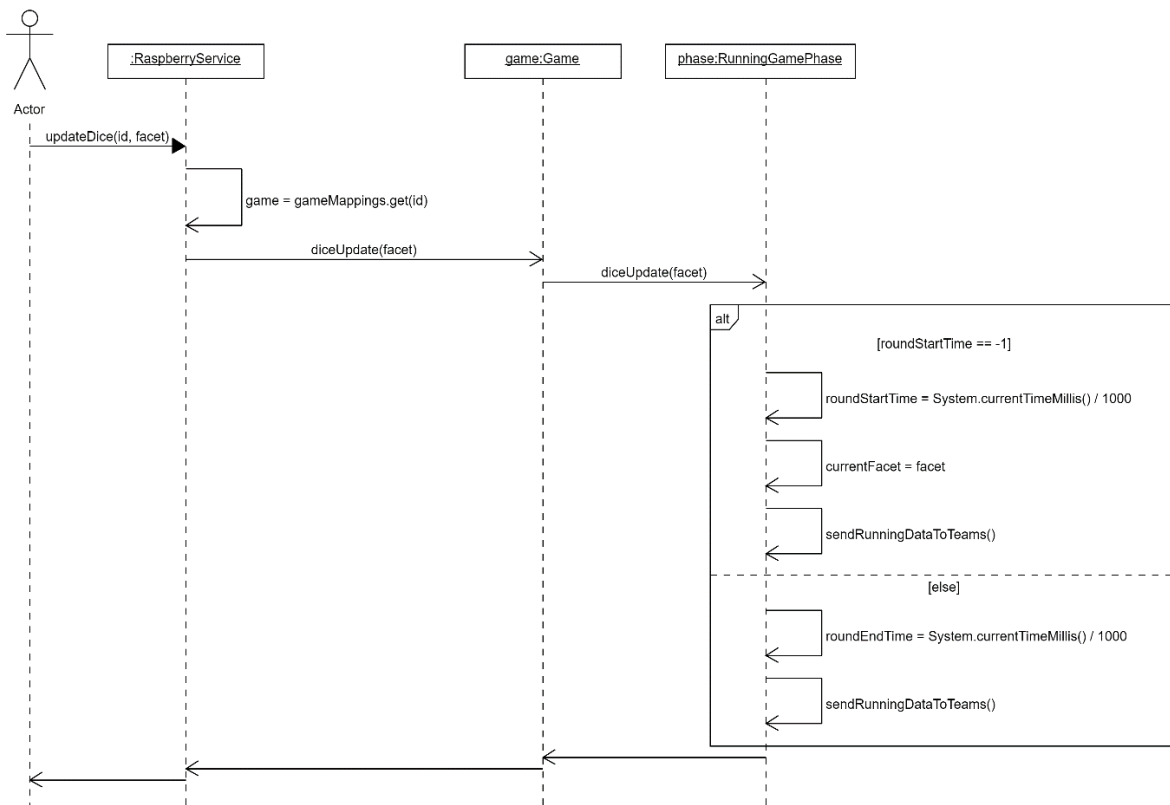
## 4.3.6 PutUser



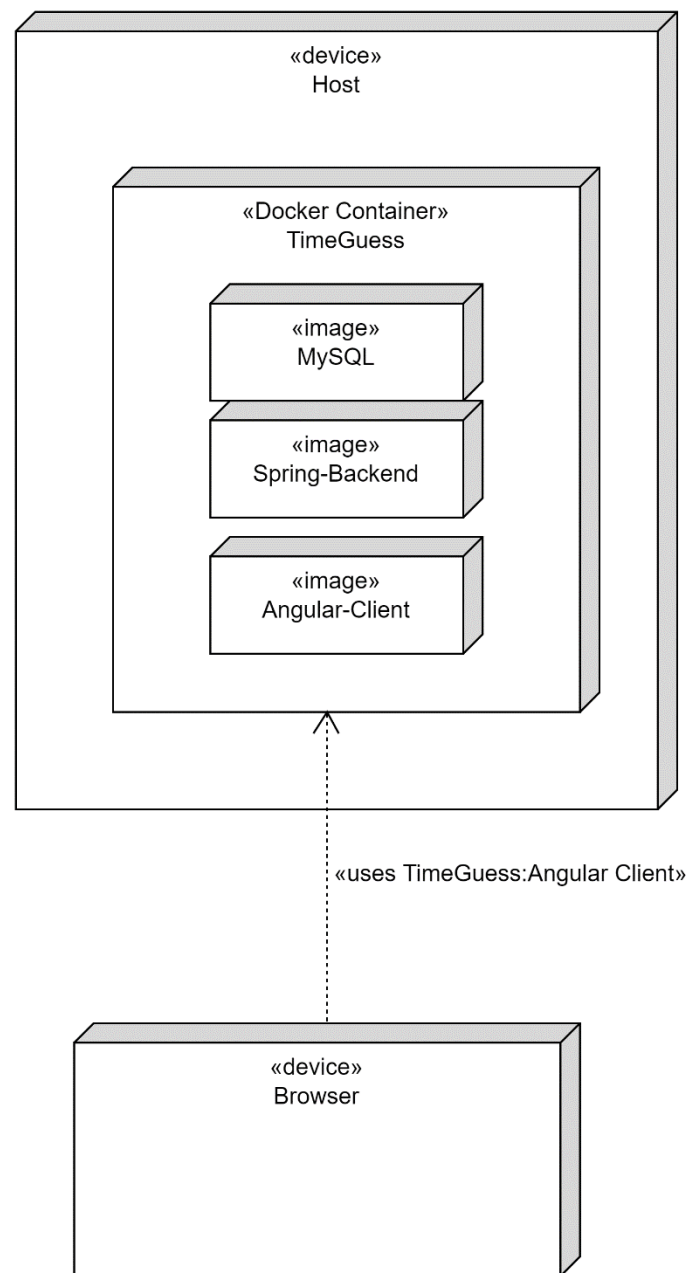
## 4.3.7 RegisterRaspberry



## 4.3.8 RollDice



## 4.4 Verteilungsschicht

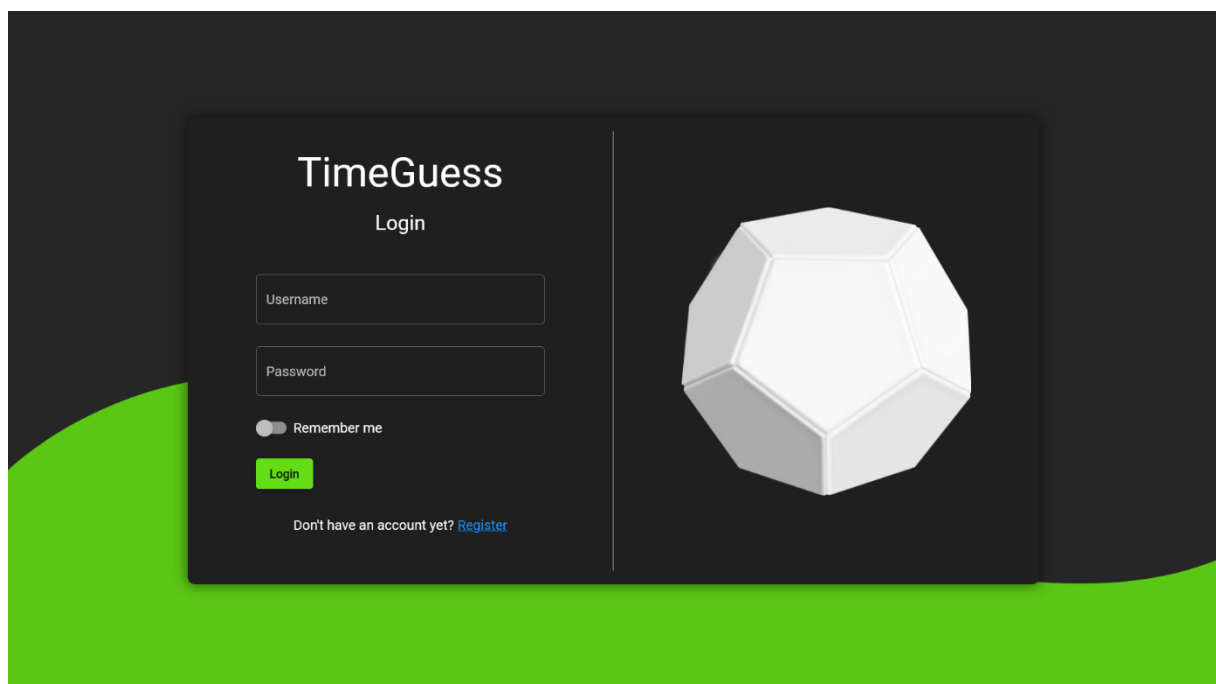
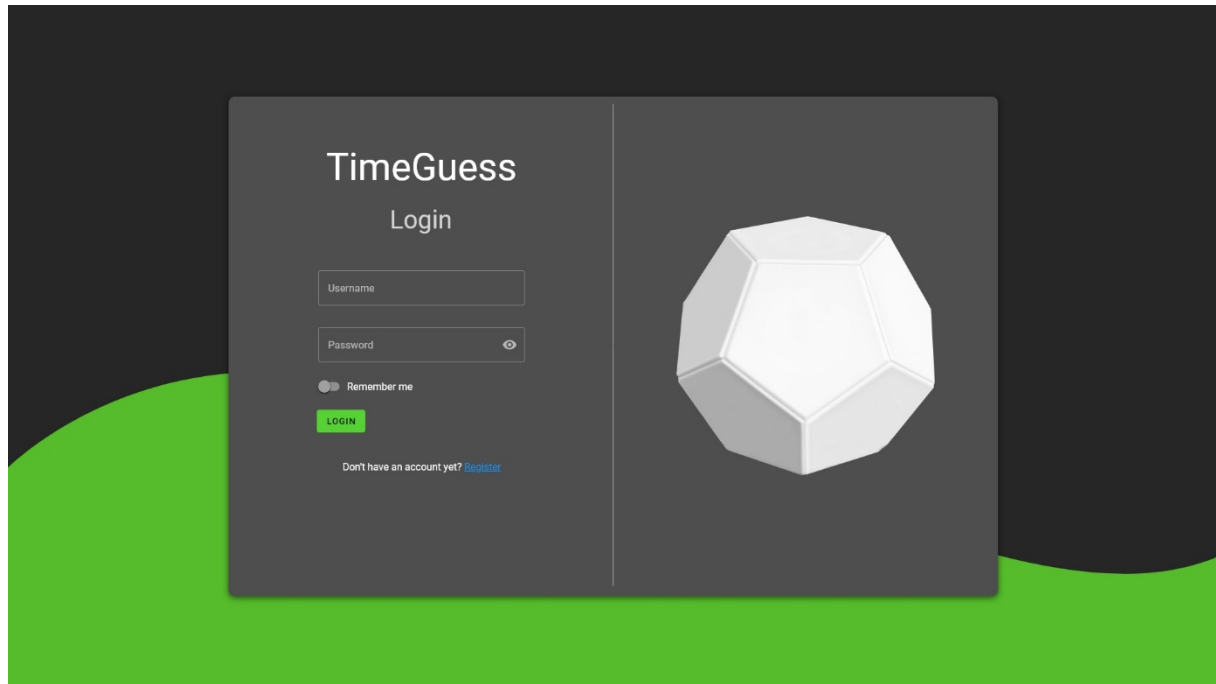


Knoten	Beschreibung
MySQL	Offizielles MySQL Docker Image, verwendet als Datenbankserver.
Spring-Backend	Java Code des Spring Backends wird mit Maven kompiliert und mit einem Image im Container zur Verfügung gestellt.
Angular-Client	Angular Code des Frontends wird kompiliert und auf einen Apache Webserver kopiert. Das resultierende Image wird im Container zur Verfügung gestellt.
Browser	Ein Gerät mit einer aktuellen Version eines Browsers, um den Angular-Client zu nutzen.

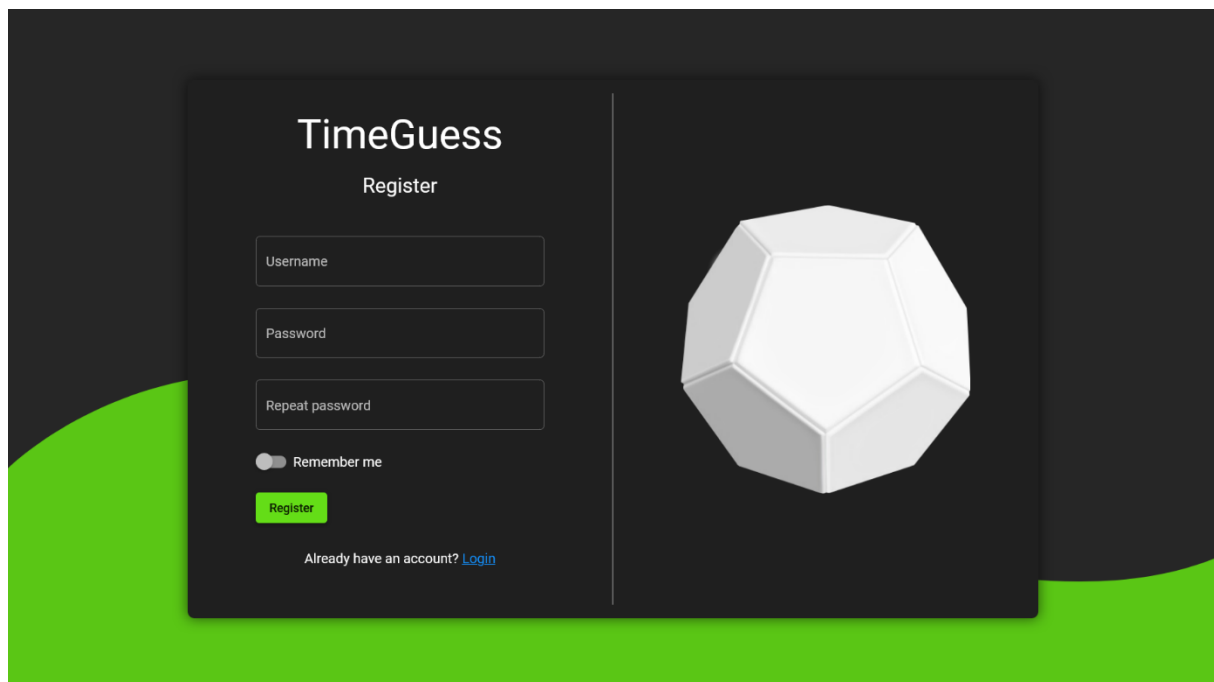
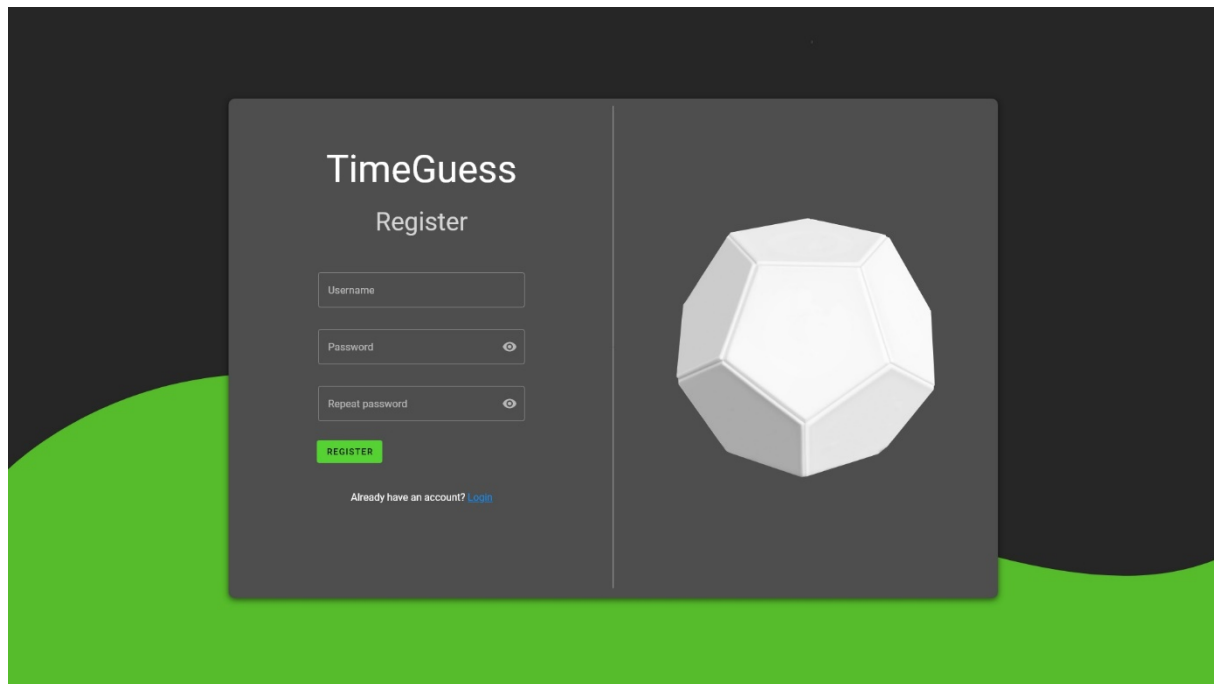
## 5. GUI Prototyp

Das obere Bild ist das Design, das untere ein Screenshot aus der laufenden Anwendung.

### 5.1 Anmelden Seite

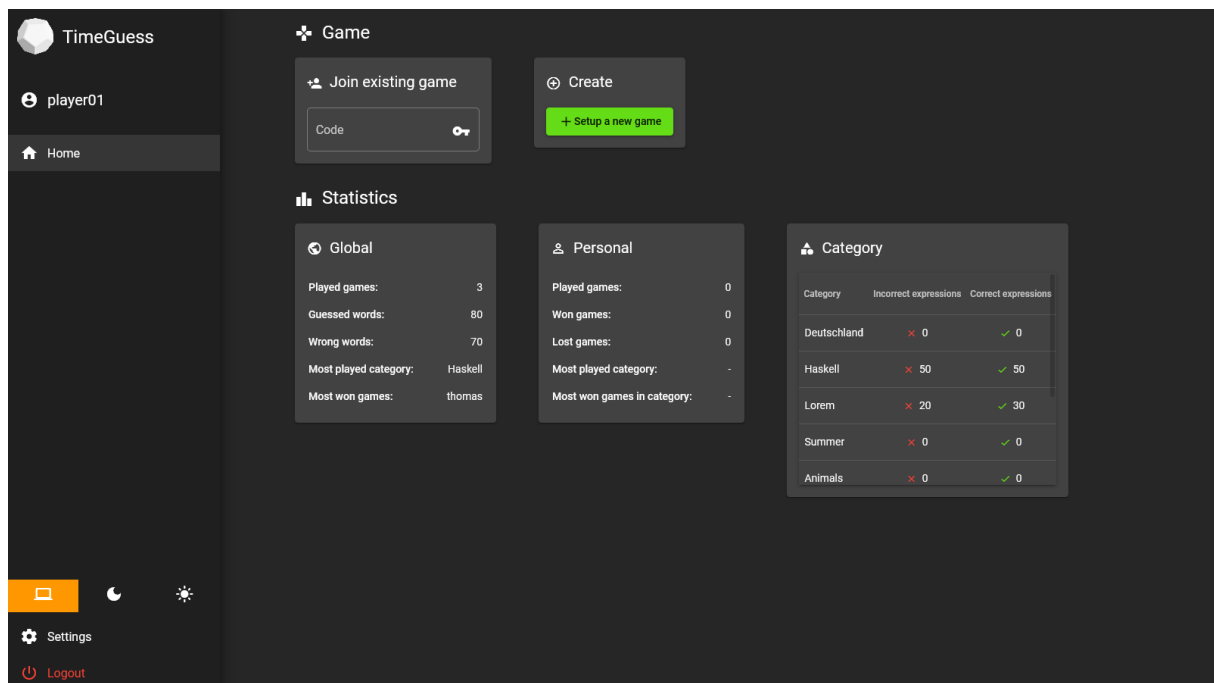
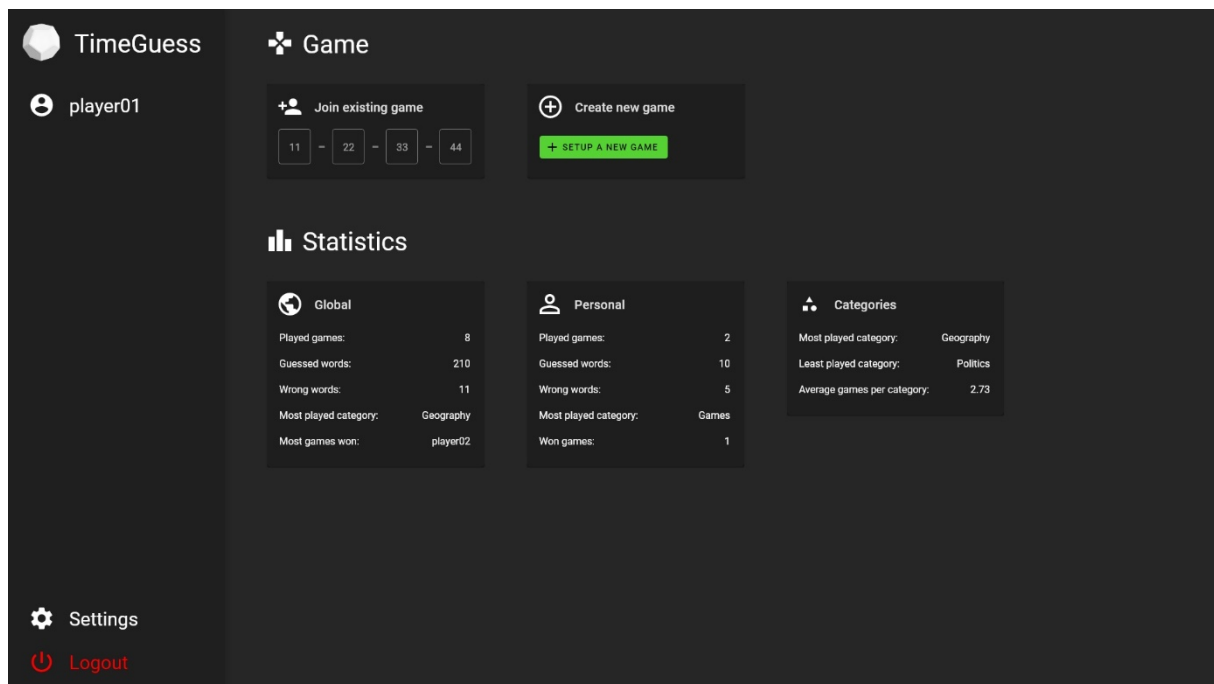


## 5.2 Registrieren Seite

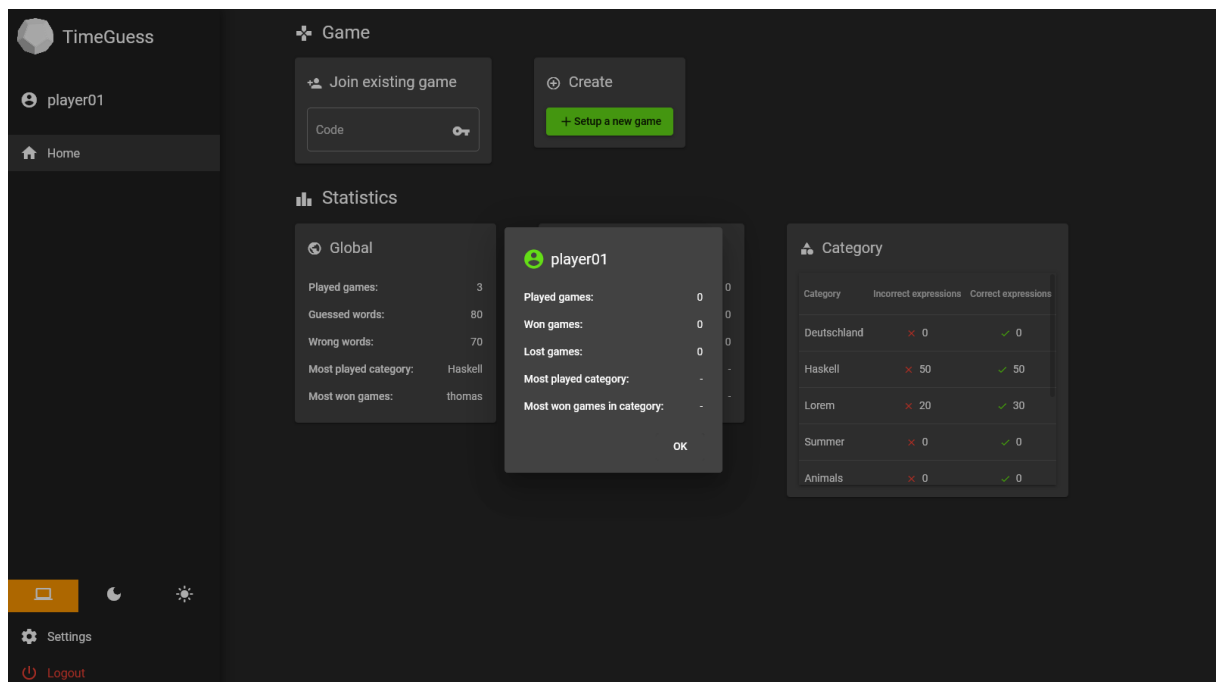
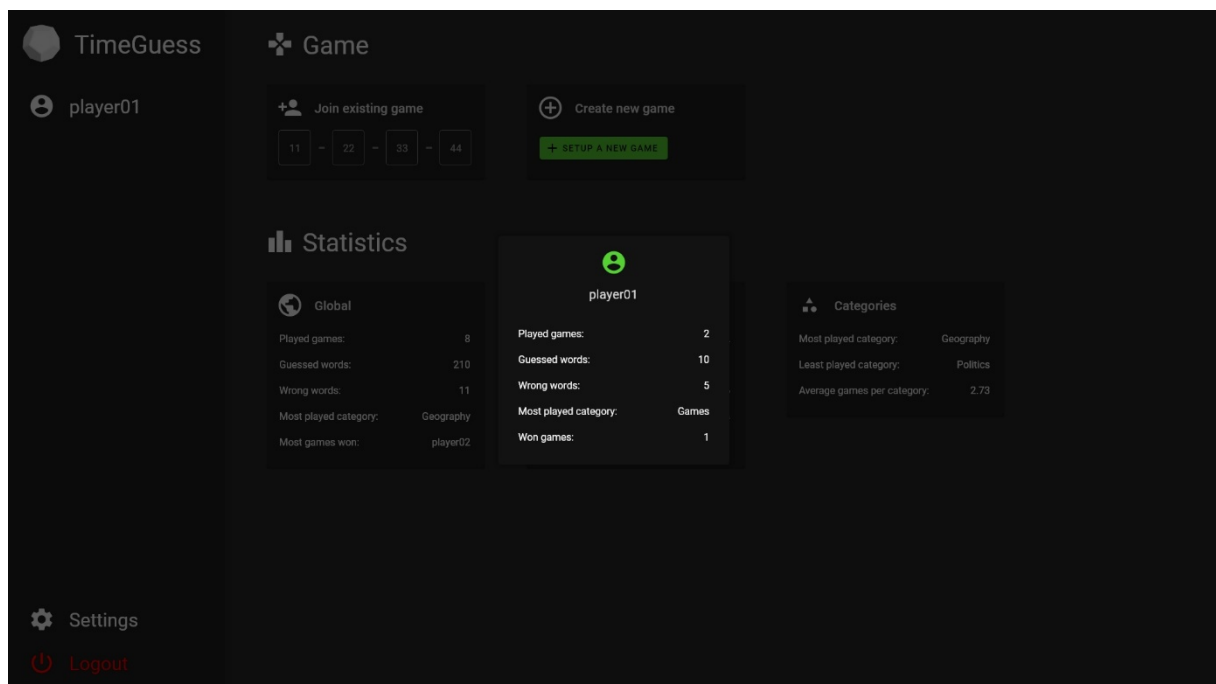




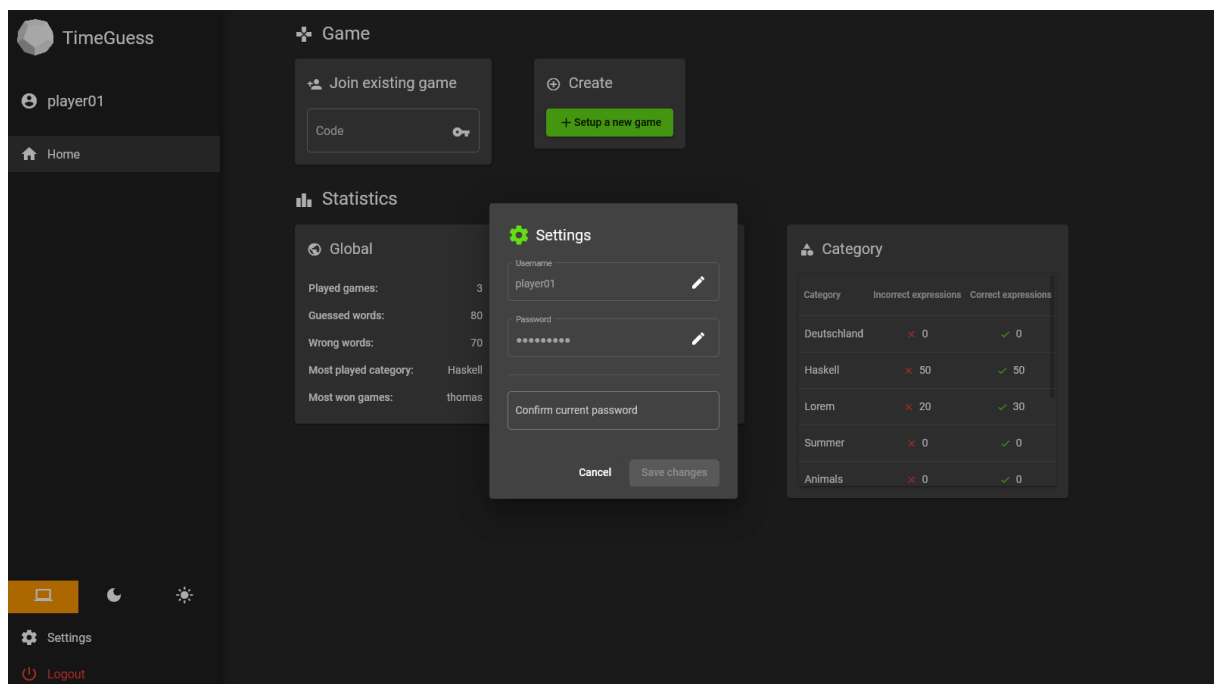
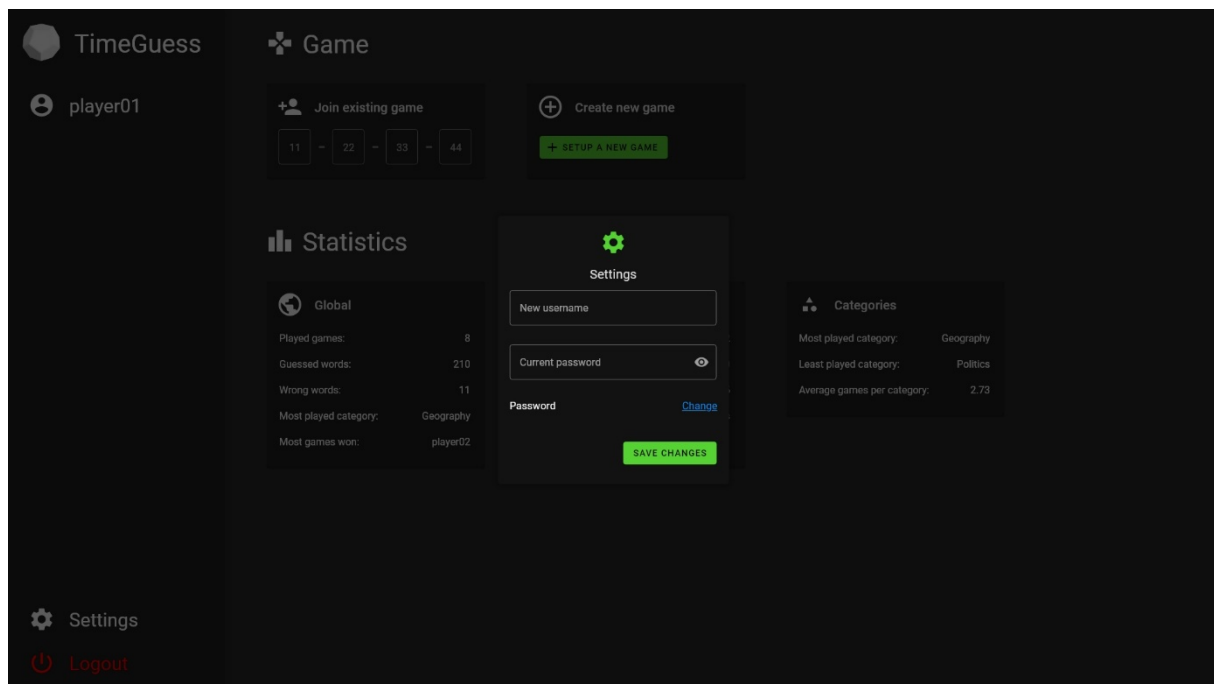
## 5.3 Virtuelle Lobby



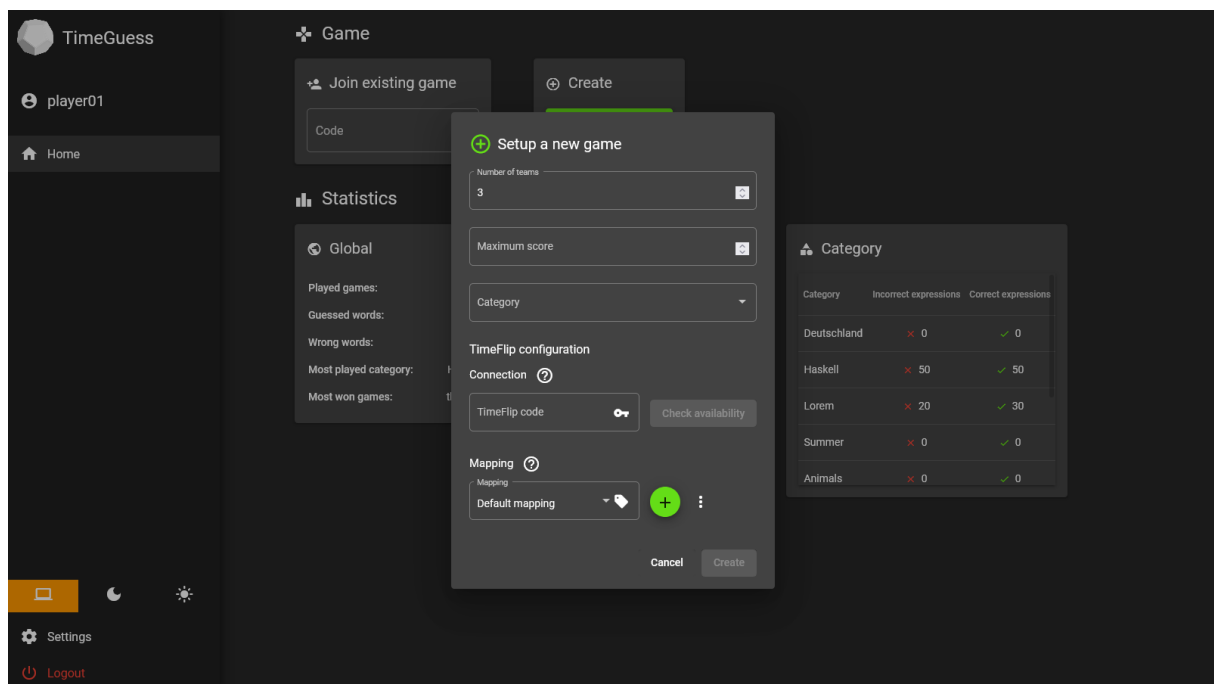
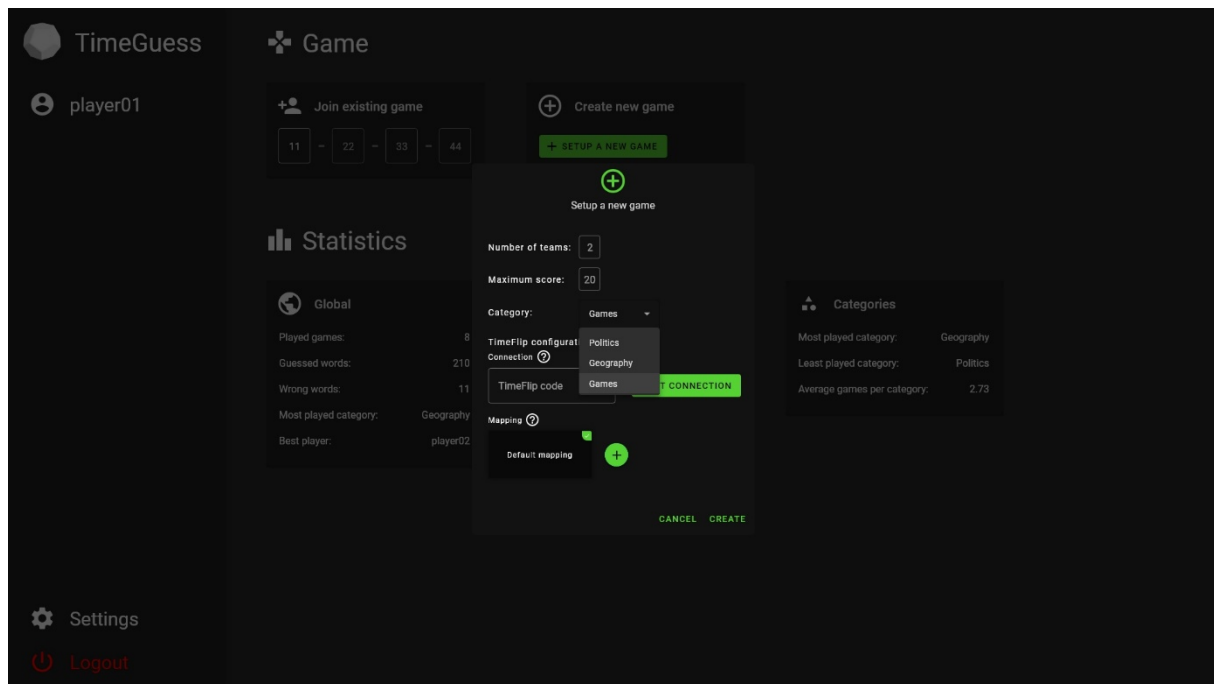
## 5.4 Spieler Profil



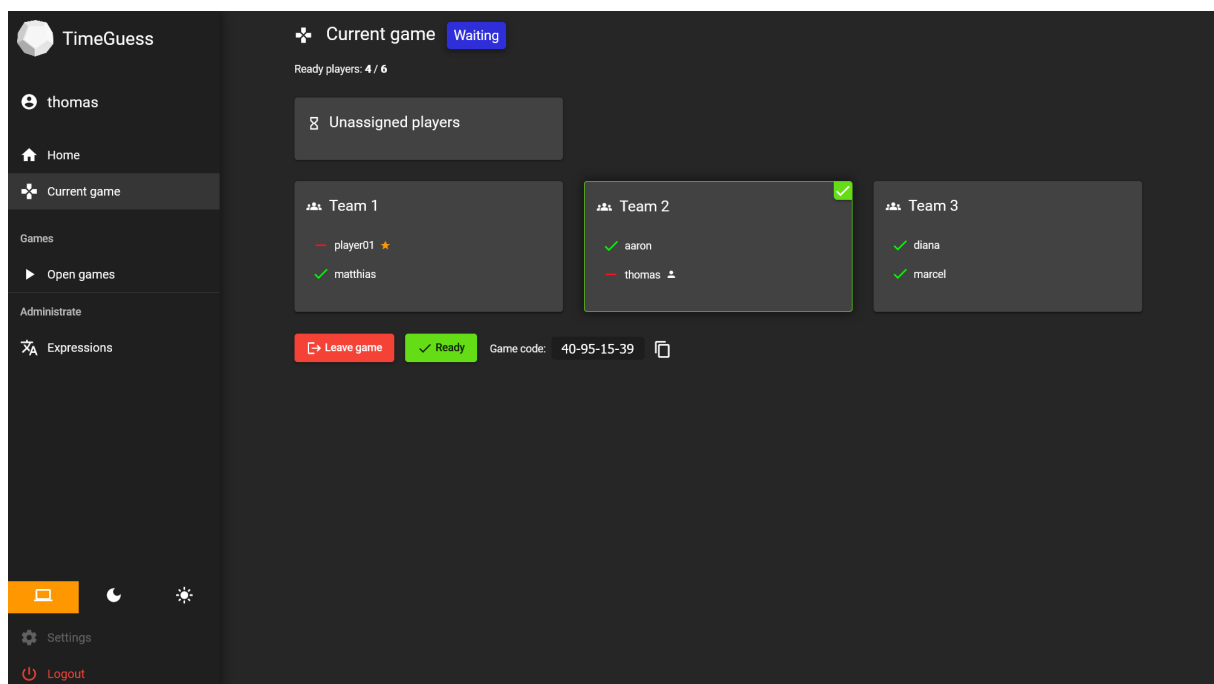
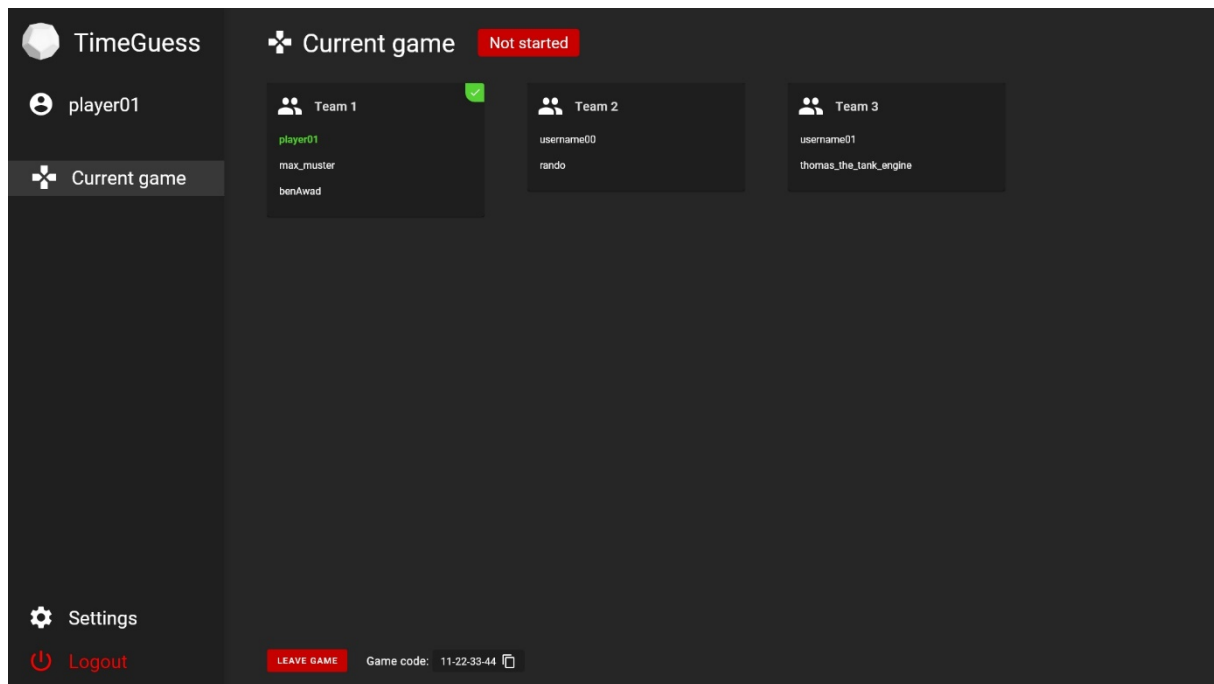
## 5.5 Benutzer Einstellungen



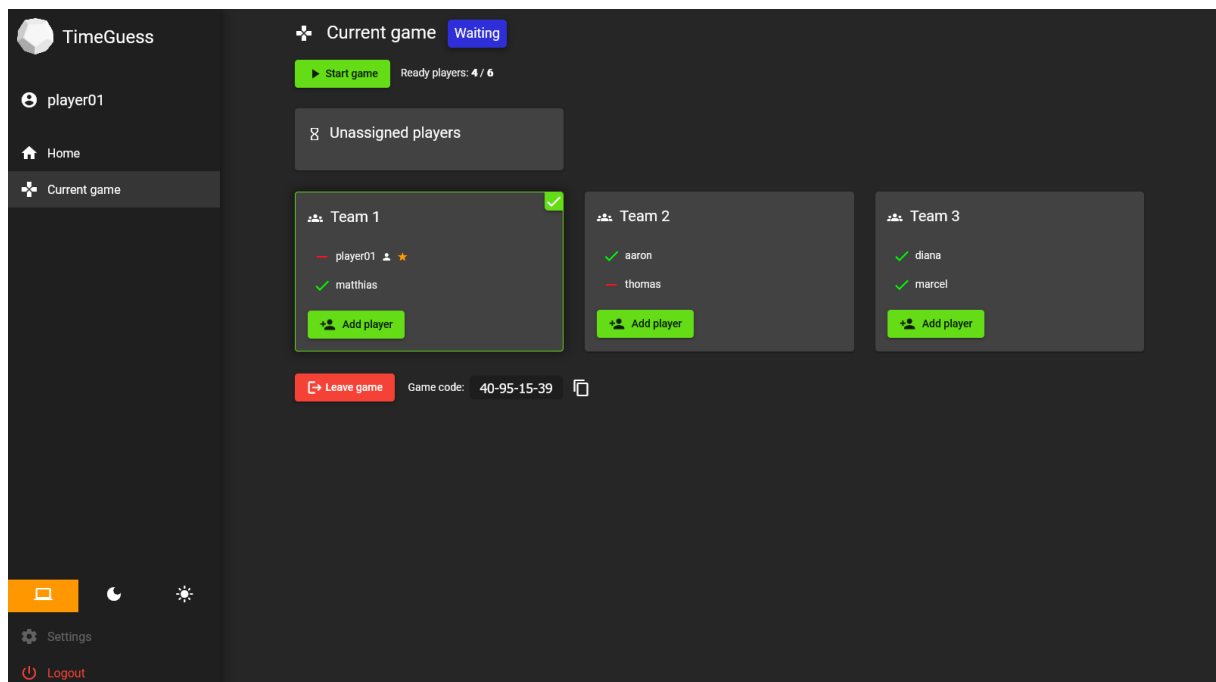
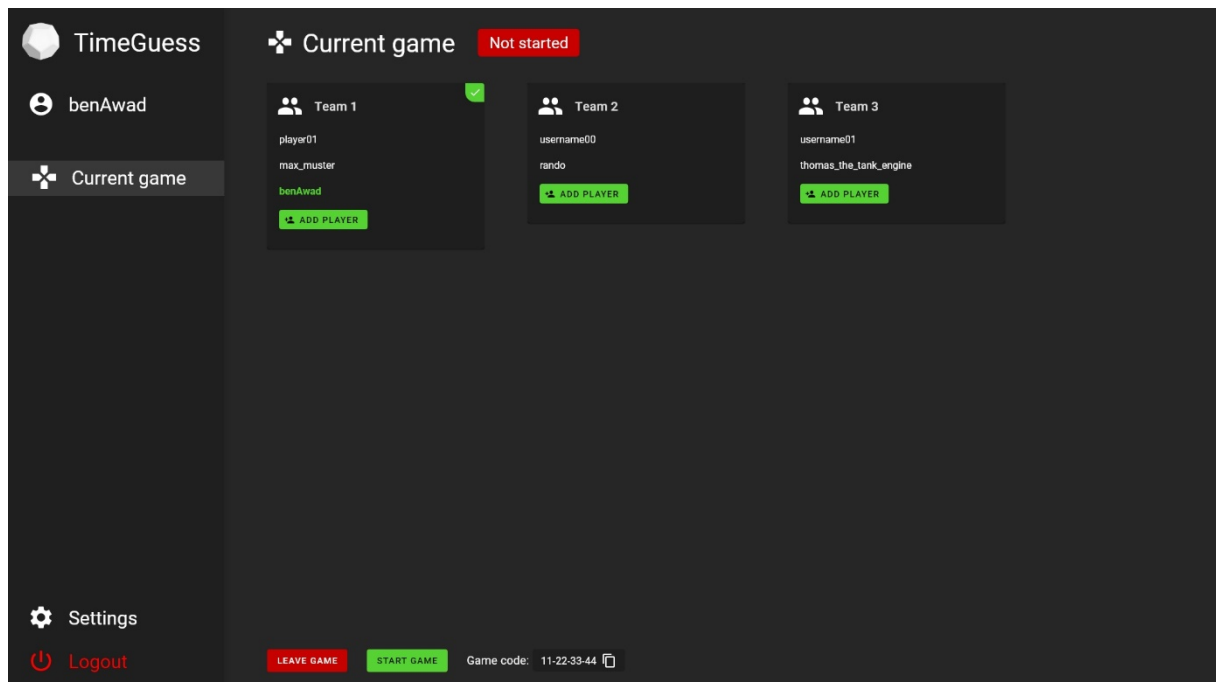
## 5.6 Spiel erstellen



## 5.7 Virtueller Spiel-Wartebereich



## 5.8 Virtueller Spiel-Wartebereich (Host)



## 5.9 Spiel – Gegnerischer Zug

**TimeGuess**

player01

Current game

Settings Logout

**Current game** **LIVE**

**Information**

Round	1
Current team	2
Category	Geography
Maximum points	30

**Current round**

Player	rando
Points	2
Available time	1:00

**Timer**

0:31

**Searched expression**

Expression	Brasil
Action	Pantomime

**Teams**

**Team 1** **5 Points**

- player01
- max\_muster
- benAwad

**Team 2** **3 Points**

- username00
- rando

**Team 3** **0 Points**

- username01
- thomas\_the\_tank\_engine

**LEAVE GAME**

**TimeGuess**

player01

Home

Current game

Settings Logout

**Current game** **LIVE**

**Information**

Round	0
Current team	Team 2
Category	Deutschland
Maximum points	30
Dice battery	65%

**Current round**

Player	aaron
Points	3
Available time	03:00

**Timer**

01:54

**Searched expression**

Expression	Zugspitze
Action	Speak

**Teams**

**Team 1** **3 Points**

- player01
- matthias

**Team 2** **0 Points**

- aaron
- thomas

**Team 3** **0 Points**

- diana
- marcel

**Leave game** Game code: 40-95-15-39

## 5.10 Spiel – Eigener Zug

**TimeGuess**

player01

**Current game**

**Current game LIVE**

**Information**

Round	2
Current team	1
Category	Geography
Maximum points	30

**Current round**

Player	max_muster
Points	4
Available time	0:30

**Timer**

0:08

**Searched expression**

Action: **Draw**

**Teams**

<b>Team 1</b> 5 Points	<b>Team 2</b> 4 Points	<b>Team 3</b> 0 Points
player01 max_muster benAwad	username00 rando	username01 thomas_the_tank_engine

**Settings**

**Logout**

**LEAVE GAME**

**TimeGuess**

player01

**Home**

**Current game**

**Current game LIVE**

**Information**

Round:	0
Current team:	Team 1
Category:	Deutschland
Maximum points:	30
Dice battery:	65%

**Current round**

Player:	player01
Points:	3
Available time:	03:00

**Timer**

02:42

**Searched expression**

Action: **Rhyme**

**Teams**

<b>Team 1</b> 0 Points	<b>Team 2</b> 0 Points	<b>Team 3</b> 0 Points
player01 ★ matthias	aaron thomas	diana marcel

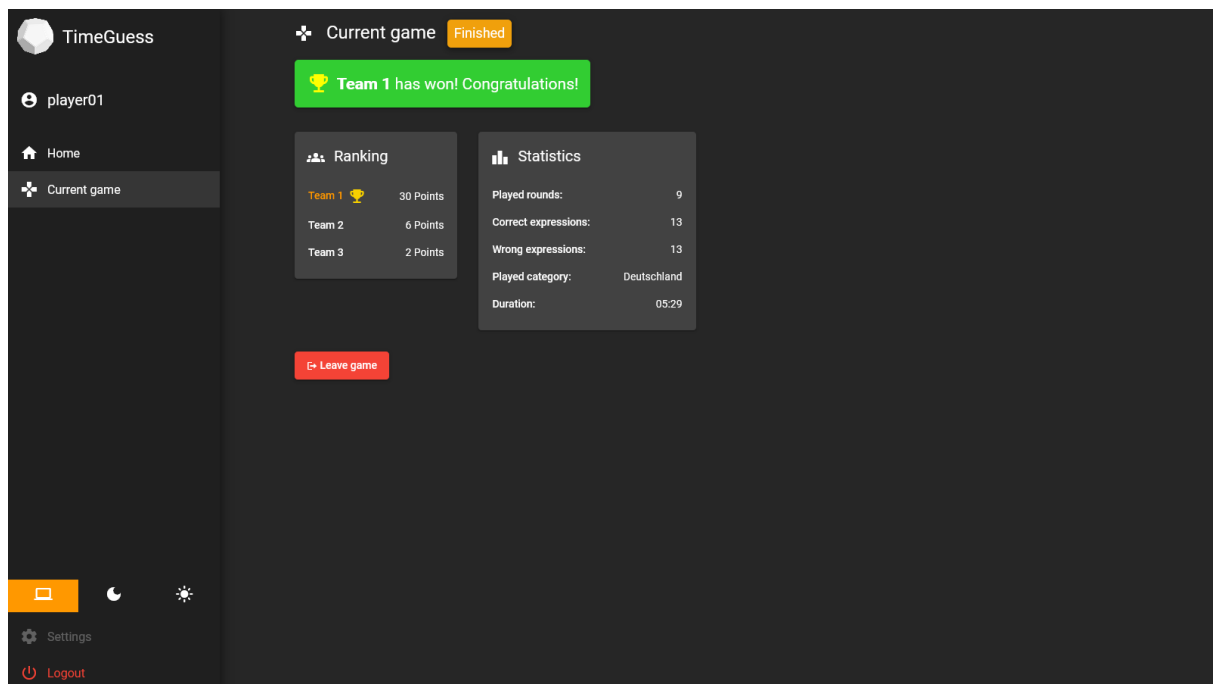
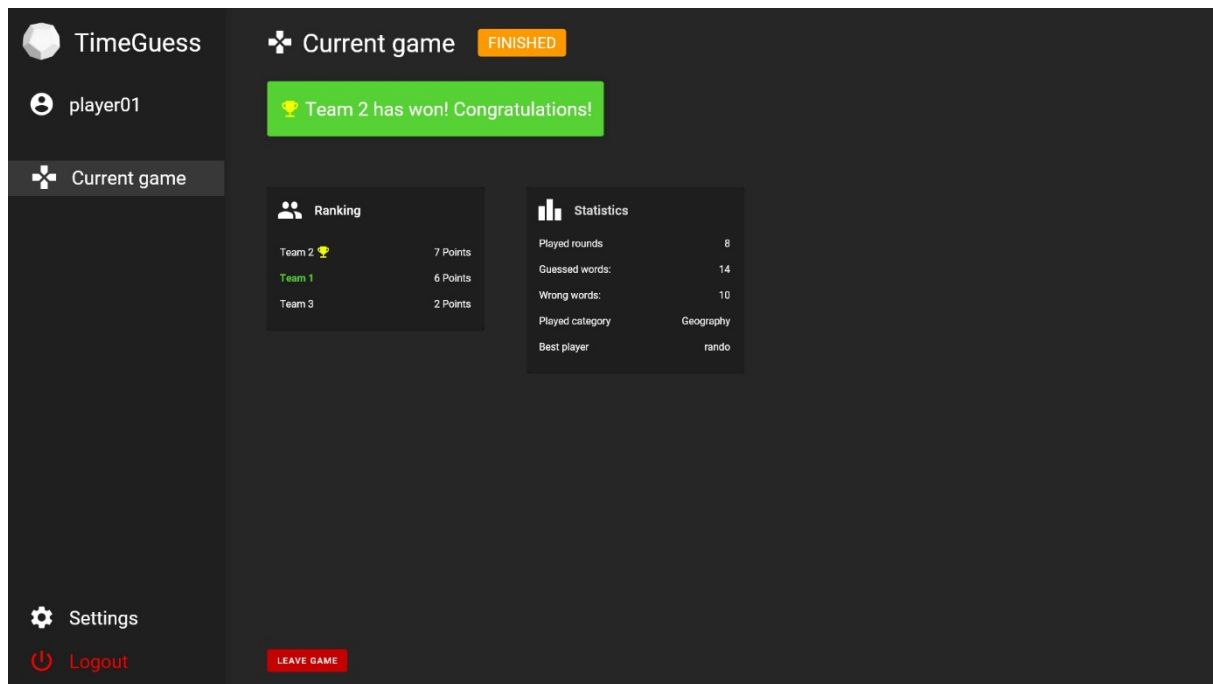
**Leave game** Game code: 40-95-15-39

**Settings**

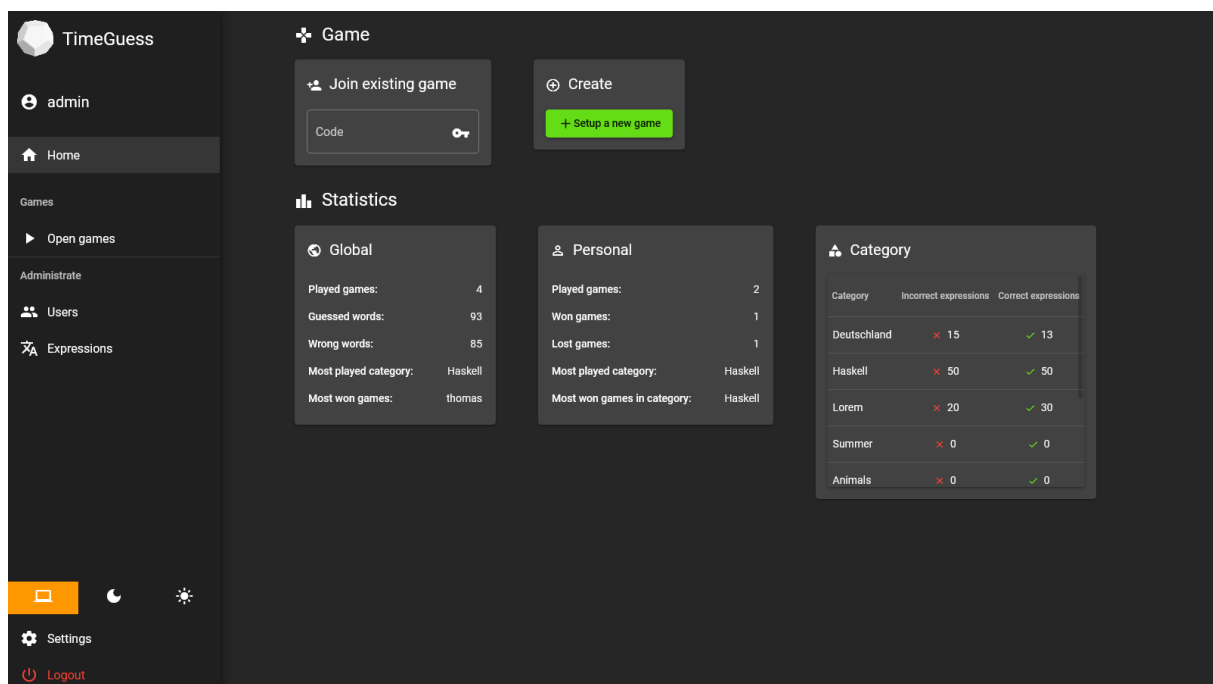
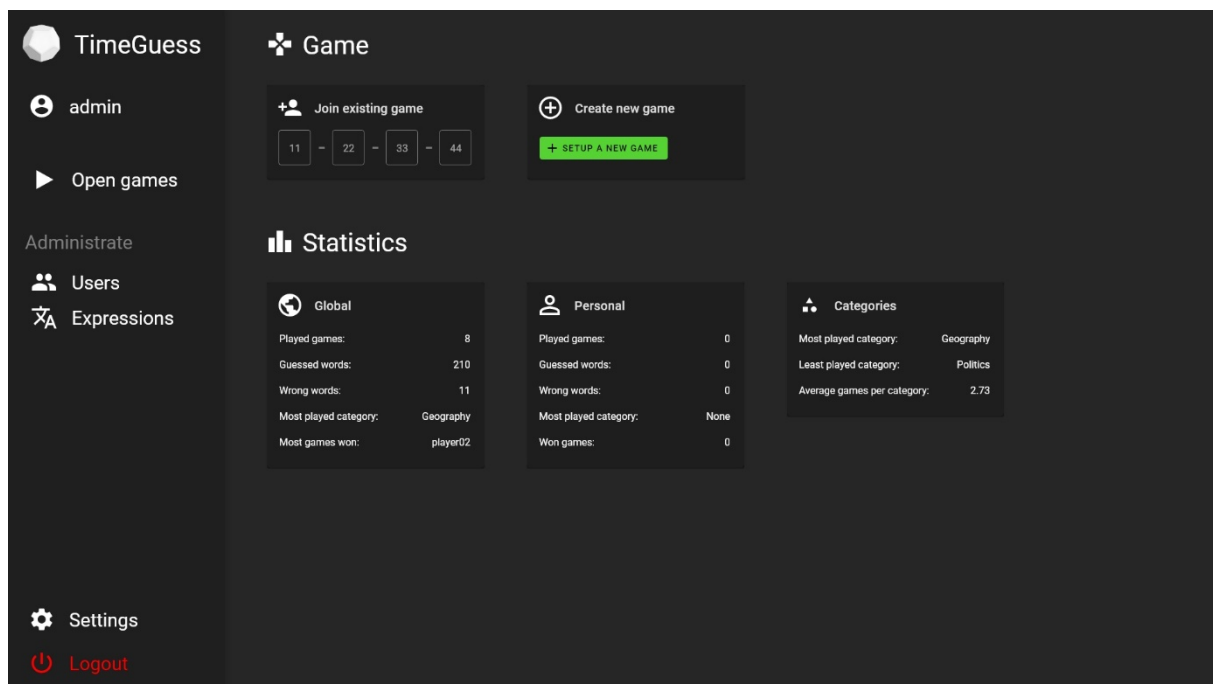
**Logout**



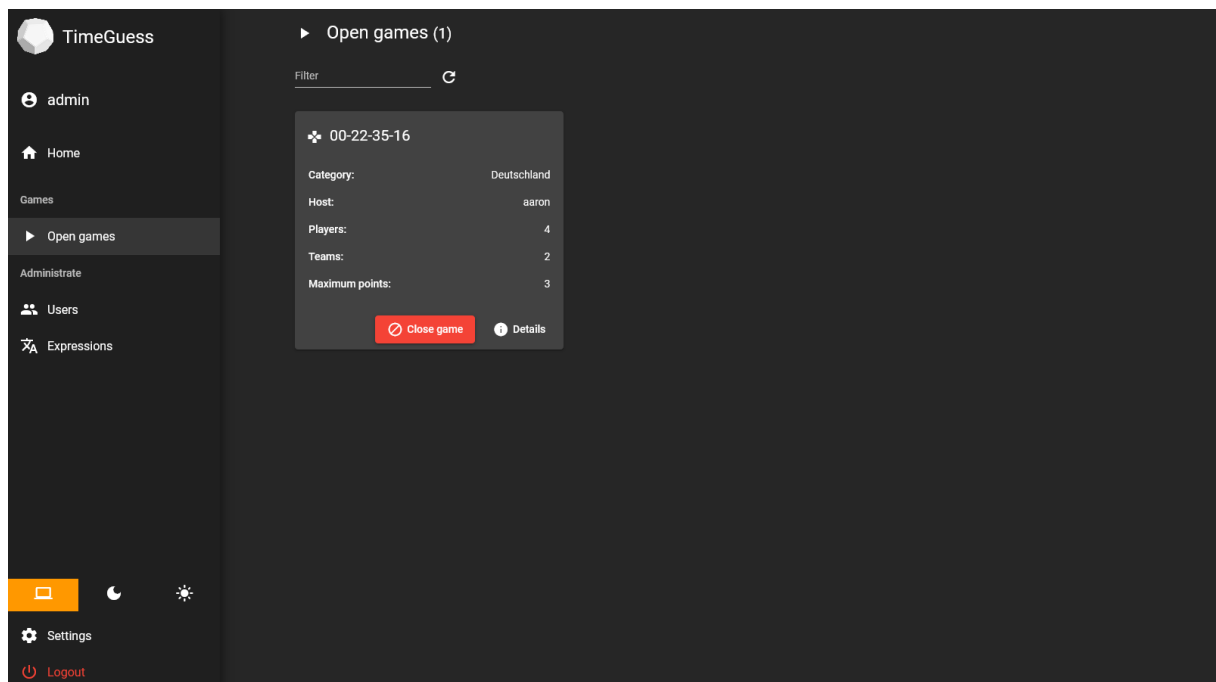
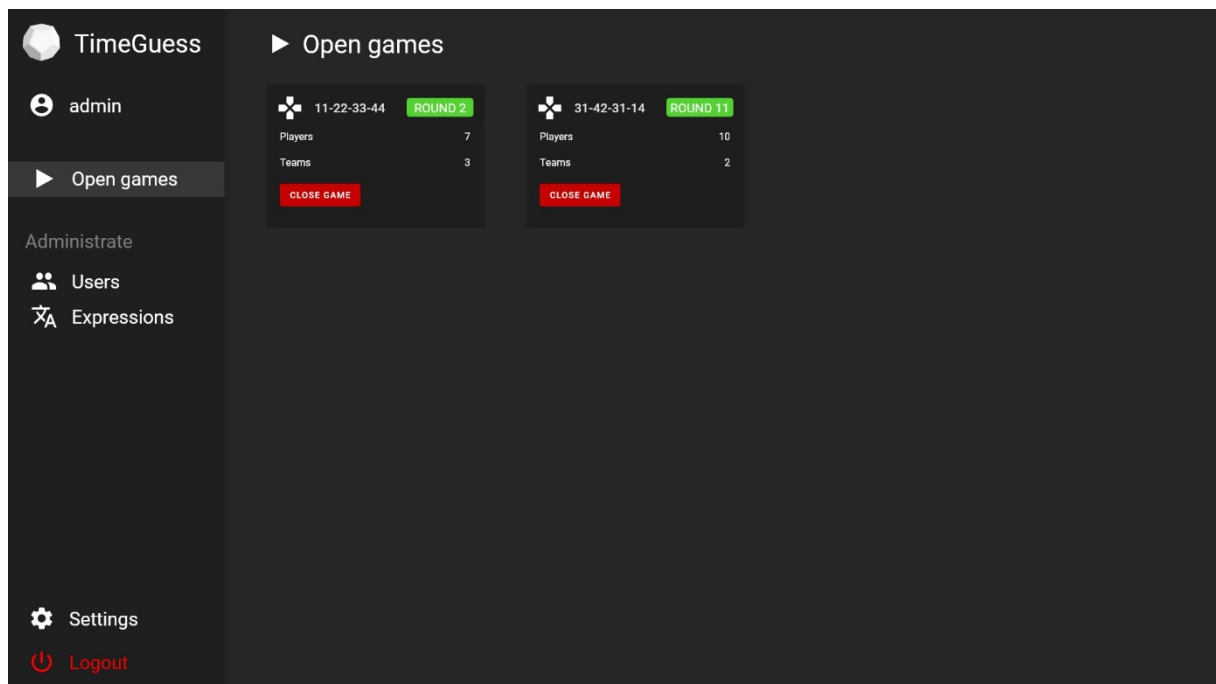
## 5.11 Spielende



## 5.12 Virtuelle Lobby (Admin)



## 5.13 Offene Spiele Seite (Admin)



## 5.14 Benutzerverwaltung (Admin)

The screenshot shows the 'Users' management page in the TimeGuess application. The left sidebar contains the following navigation items: 'admin', 'Open games', 'Administrate' (with a sub-menu containing 'Users' and 'Expressions'), 'Settings', and 'Logout'. The main content area is titled 'Users' and features a '+ CREATE USER' button. Below this is a table with the following data:

Username	Last game	Manage
admin	14.03.2021 17:23:16	
player01	14.03.2021 17:23:16	
player02	14.03.2021 17:23:16	
thomas_the_tank_engine	14.03.2021 17:23:16	
rando	14.03.2021 17:23:16	
username00	14.03.2021 17:23:16	

The screenshot shows the 'Administrare Users' page in the TimeGuess application. The left sidebar contains the following navigation items: 'admin', 'Home', 'Games' (with a sub-menu containing 'Open games'), 'Administrate' (with a sub-menu containing 'Users' and 'Expressions'), 'Settings', and 'Logout'. The main content area is titled 'Administrare Users' and features a '+ Create user' button and a refresh icon. Below this is a table with the following data:

ID	Username	Role
0	admin	Admin
1	thomas	Gamemanager
2	marcel	Gamemanager
3	matthias	Player
4	aaron	Player
5	diana	Player
6	klotz	Player
7	huber	Player
8	thalmann	Player
9	targa	Player

At the bottom right of the table, there is a context menu with the following options: 'Edit user', 'Show profile', and 'Delete user'. The bottom of the page shows 'Items per page: 10' and '1 - 10 of 12'.

## 5.14 Ausdrucksverwaltung (Admin)

The screenshot shows the 'Expressions' management page in the TimeGuess application. The left sidebar contains the following menu items: 'admin', 'Open games', 'Administrate' (with sub-items 'Users' and 'Expressions'), 'Settings', and 'Logout'. The main content area is titled 'Expressions' and features a '+ CREATE' and 'IMPORT' button. Below these buttons is a table with three columns: 'Category', 'Expressions', and 'Manage'.

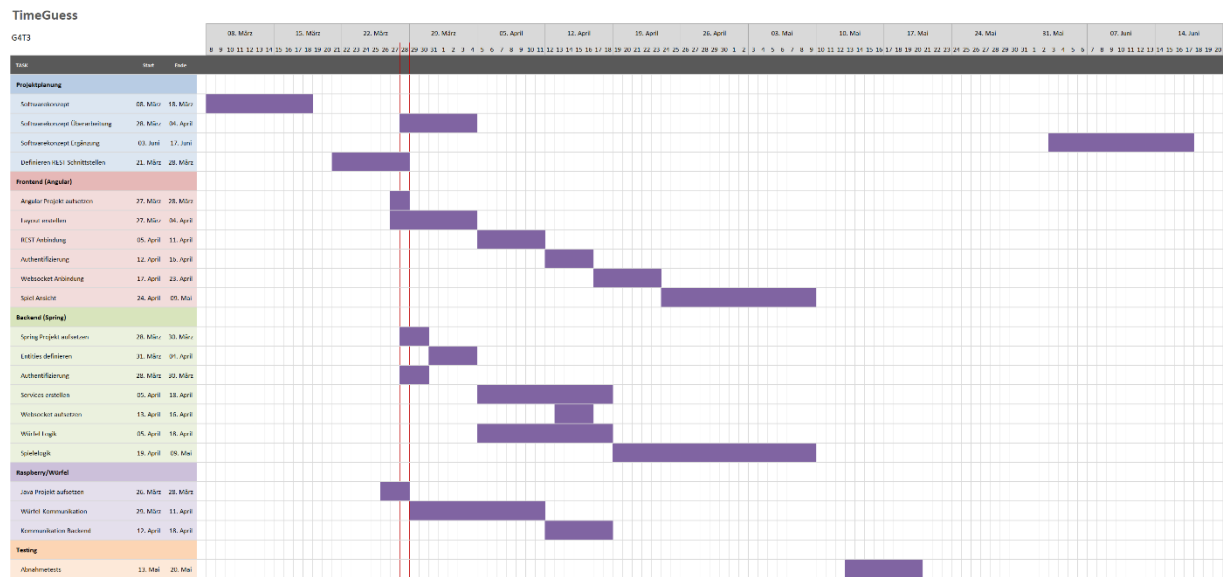
Category	Expressions	Manage
Politics	14	[Icon]
Geography	11	[Icon]
Food & Drinks	9	[Icon]
Videogames	16	[Icon]
History	7	[Icon]
Germany	0	[Icon]

The screenshot shows the 'AdministratE Expressions' page in the TimeGuess application. The left sidebar contains the following menu items: 'admin', 'Home', 'Games' (with sub-item 'Open games'), 'Administrate' (with sub-items 'Users' and 'Expressions'), 'Settings', and 'Logout'. The main content area is titled 'AdministratE Expressions' and features a '+ Create' button and a refresh icon. Below these buttons is a 'Filter' input field. The main content is a table with four columns: 'ID', 'Category', 'Expressions', and a vertical ellipsis menu.

ID	Category	Expressions	
0	Deutschland	30	⋮
1	Haskell	32	⋮
2	Lorem	30	⋮
3	Summer	38	⋮
4	Animals	35	⋮
5	Math	35	⋮
6	Actions	37	⋮
7	Winter	38	⋮

Below the table, there is a context menu with the following options: 'Show expressions', 'Add expression(s)', 'Import expressions', and 'Delete category'. At the bottom right of the table, there is a pagination control showing 'Items per page: 10' and '1 - 8 of 8'.

## 6. Projektplan



### 6.1 Zuständigkeiten

#### Hochauflösende Version

- **Frontend:** Matthias Thalmann
- **Backend:** Marcel Alexander Huber, Thomas Klotz, Aaron Targa
- **Raspberry/Würfel:** Diana Gründlinger, Thomas Klotz

### 6.2 Meilensteine und weitere Informationen

Meilensteine, Issues und Informationen darüber sind im GitLab einsehbar:

- **Issues:** <https://git.uibk.ac.at/informatik/qe/sepsss21/group4/g4t3/issues>
- **Meilensteine:** <https://git.uibk.ac.at/informatik/qe/sepsss21/group4/g4t3/-/milestones>