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.

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.

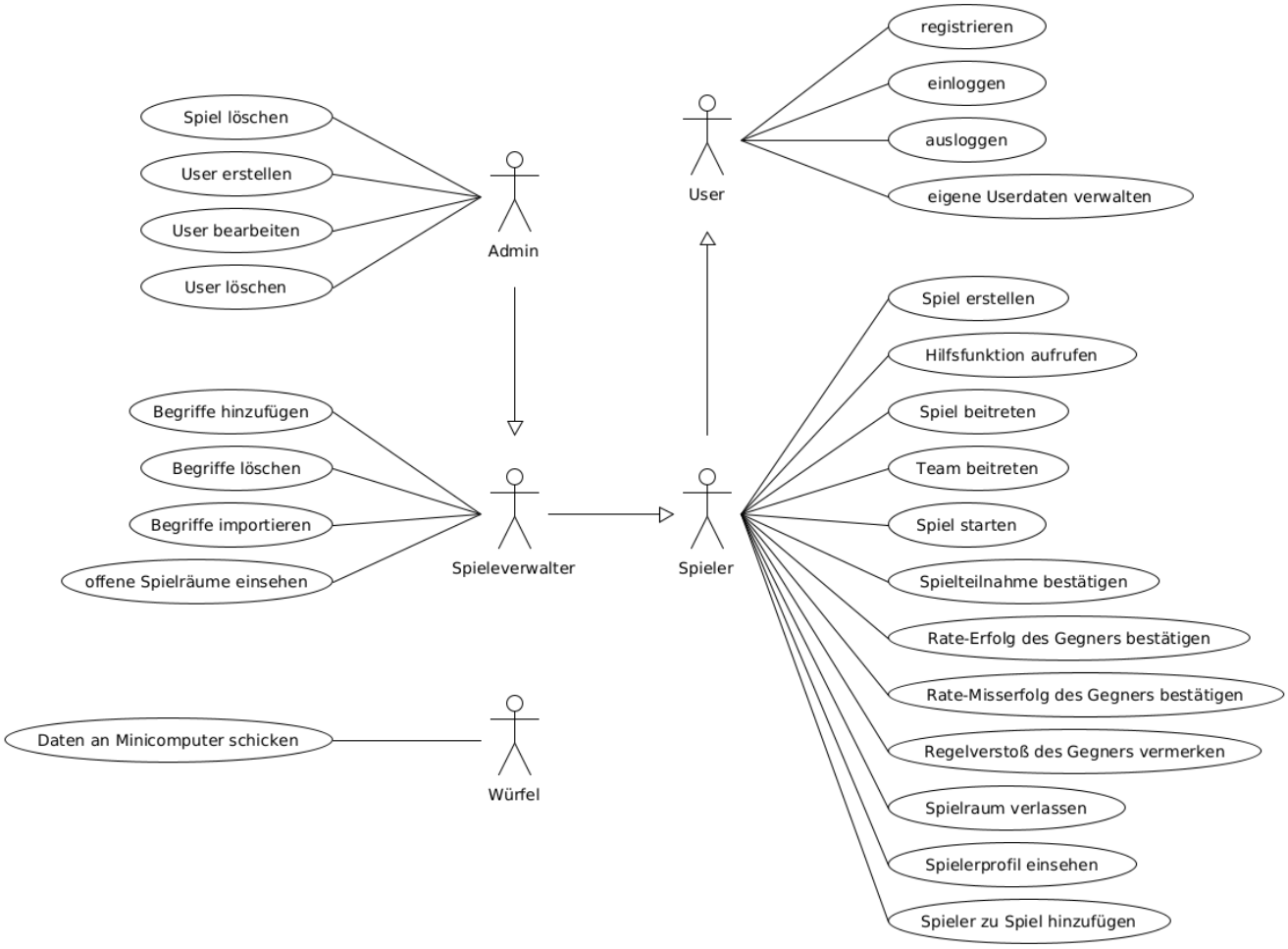
Admin

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:** Der User klickt auf “Register”. Dann gibt er in ein Registrierungsformular seinen gewünschten Usernamen und ein Passwort ein. Das Passwort muss durch eine zweite Eingabe bestätigt werden.
- **Nachbedingung:** Der User besitzt einen Account, ist angemeldet und befindet sich in der Lobby.
- **Alternativen:**
  - Die erste und zweite Eingabe des Passworts stimmen nicht überein. Eine Fehlermeldung wird angezeigt.
  - Der gewählte Username ist bereits vergeben. Eine Fehlermeldung wird angezeigt.
  - Der Vorgang wird durch den User abgebrochen.
- **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 und 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.
- **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 verwalten

- **Vorbedingung:** Der User ist eingeloggt.
- **Basisablauf:** Der User klickt auf „Settings“. Ein Einstellungsdialog wird angezeigt. Um Daten zu ändern, muss zuerst das alte Passwort eingegeben werden. Es können sowohl Username als auch Passwort geändert werden. Die Änderungen werden durch das Klicken auf “Save” übernommen.
- **Nachbedingung:** Die neuen Userdaten wurden übernommen. Der Einstellungsdialog wurde geschlossen.
- **Alternativen:**
    - Der Username ist bereits vergeben. Es wird eine Fehlermeldung angezeigt.
    - Das eingegebene alte Passwort ist falsch. 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:** Der Spieler klickt auf “Create Game”. Anschließend befindet er sich auf der Konfigurationsseite für das neue Spiel. Er legt die Anzahl der Teams und ein Themengebiet fest. Danach konfiguriert er den verwendeten Würfel. Die Verbindung zum Würfel muss hergestellt werden und das Mapping für die Würfelseiten festgelegt werden. Für die Konfiguration des Würfels kann eine Hilfsfunktion aufgerufen werden. Der Spieler bestätigt die Konfiguration.
- **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:**
    - Das Erstellen des Spiels wird abgebrochen. Der Spieler kehrt in die Lobby zurück.
    - Es kann keine Verbindung zum Würfel hergestellt werden. Es wird eine Fehlermeldung angezeigt.
- **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. Dem Spieler werden nützliche Detailinformationen für die Würfelkonfiguration angezeigt.
- **Nachbedingung:** -
- **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.
- **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:**
    - Der Spieler wartet darauf, dass noch weitere Spieler die Spielteilnahme bestätigen.
    - 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 Gesamtpunktezahl 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:** Der Spieler befindet sich in der Lobby und kann jederzeit wieder dem aktiven Spiel beitreten durch Klicken auf „Current Game“.
- **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:** Der Host klickt bei einem Team auf „Add Player“. Ein Dialogfenster öffnet sich und ein verfügbarer Spieler kann ausgewählt werden. Die Auswahl wird bestätigt und der gewählte User muss sein Passwort eingeben, um dem Spiel beitreten zu können.
- **Nachbedingung:** Der ausgewählte Spieler befindet sich im gewählten Team.
- **Alternativen:** Das eingegebene Passwort ist falsch. Eine Fehlermeldung wird angezeigt.
- **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:** Durch Klicken von “Add Expressions” wird eine Eingabemöglichkeit geöffnet. Dort kann eine Kategorie gewählt werden und es können beliebig viele neue Begriffe zu diesem Thema eingegeben werden. Die Eingabe wird danach bestätigt.
- **Nachbedingung:** Die neu eingegebenen Begriffe wurden in den Begriffskatalog aufgenommen.
- **Alternativen:** Die Eingabe wird durch den Spieleverwalter abgebrochen. Es wurden keine Begriffe gespeichert.
- **Involvierte Klassen:** Expression, Category

Begriffe löschen

- **Vorbedingung:** Der Spieleverwalter ist eingeloggt und befindet sich in der Begriffskatalog-Ansicht.
- **Basisablauf:** Durch Betätigen eines Buttons wird ein Begriff gelöscht.
- **Nachbedingung:** Der Begriff ist nicht mehr im Begriffskatalog vorhanden.
- **Alternativen:** -
- **Involvierte Klassen:** Expression

Begriffe importieren

- **Vorbedingung:** Der Spieleverwalter ist eingeloggt und befindet sich in der Begriffskatalog-Ansicht.
- **Basisablauf:** Durch Klicken von “Import Expressions” wird ein Dialog geöffnet, wo eine Kategorie gewählt werden muss und über Drag&Drop einer JSON Datei Begriffe hinzugefügt werden können.
- **Nachbedingung:** Die importierten Begriffe wurden in den Begriffskatalog aufgenommen.
- **Alternativen:** Der Import schlägt fehl. Eine Fehlermeldung wird angezeigt.
- **Involvierte Klassen:** Expression, Category

Offene Spielräume einsehen

- **Vorbedingung:** Der Spieleverwalter 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 angezeigt werden. Durch Anklicken des gewünschten Spiels werden Informationen über das Spiel angezeigt (Zwischenergebnisse etc.)
- **Nachbedingung:** Dem Spieleverwalter werden alle aktuellen Informationen über das laufende Spiel angezeigt.
- **Alternativen:** -
- **Involvierte Klassen:** Game, Team

2.3.4 Akteur: Admin

Spiel löschen

- **Vorbedingung:** Der Admin ist eingeloggt und befindet sich in der Ansicht der offenen Spiele.
- **Basisablauf:** Durch Klicken eines Buttons 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:** Der Admin klickt auf „Create User“. Ein Dialog öffnet sich und man gibt relevante Daten ein. Die Eingabe wird bestätigt.
- **Nachbedingung:** Die neuen Daten des Users wurden übernommen. Der User existiert.
- **Alternativen:** Der Username existiert bereits. Es wird eine Fehlermeldung angezeigt.
- **Involvierte Klassen:** User

User bearbeiten

- **Vorbedingung:** Der Admin befindet sich auf der Userverwaltungsseite.
- **Basisablauf:** Der Admin wählt einen User aus. Ein Dialog wird geöffnet. Der Admin kann dann die Daten des Users ändern.
- **Nachbedingung:** Die neuen Daten des Users wurden übernommen.
- **Alternativen:** Der Username existiert bereits. Es wird eine Fehlermeldung angezeigt.
- **Involvierte Klassen:** User

User löschen

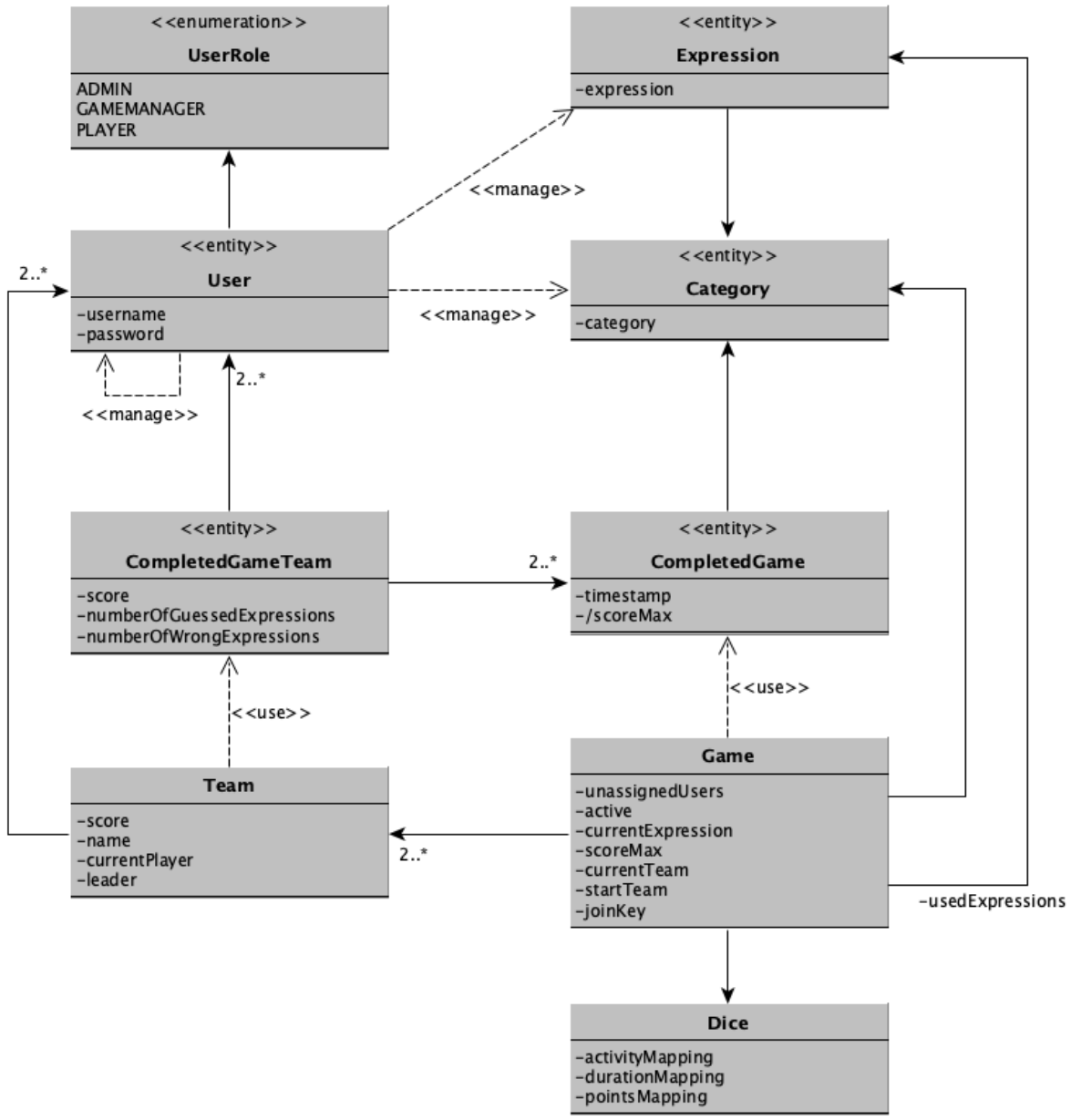
- **Vorbedingung:** Der Admin befindet sich auf der Userverwaltungsseite.
- **Basisablauf:** Der Admin wählt einen User aus und löscht ihn durch Klicken auf einen Button.
- **Nachbedingung:** Der User existiert nicht mehr.
- **Alternativen:** -
- **Involvierte Klassen:** User

2.3.5 Akteur: Würfel

Daten an Minicomputer schicken

- **Vorbedingung:** Der Würfel ist mit dem Minicomputer verbunden.
- **Basisablauf:** Der Würfel sendet seine gemessenen Positions- und Zeit-Daten an den Minicomputer.
- **Nachbedingung:** Der Minicomputer hat die Daten des Würfels empfangen.
- **Alternativen:** Die Verbindung zwischen Würfel und Minicomputer wurde unterbrochen oder die Übertragung ist fehlgeschlagen. Wiederherstellungsmaßnahmen werden ergriffen.
- **Involvierte Klassen:** -

3. Klassendiagramm



Expression
Diese Klasse stellt einen zu erratenen 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.

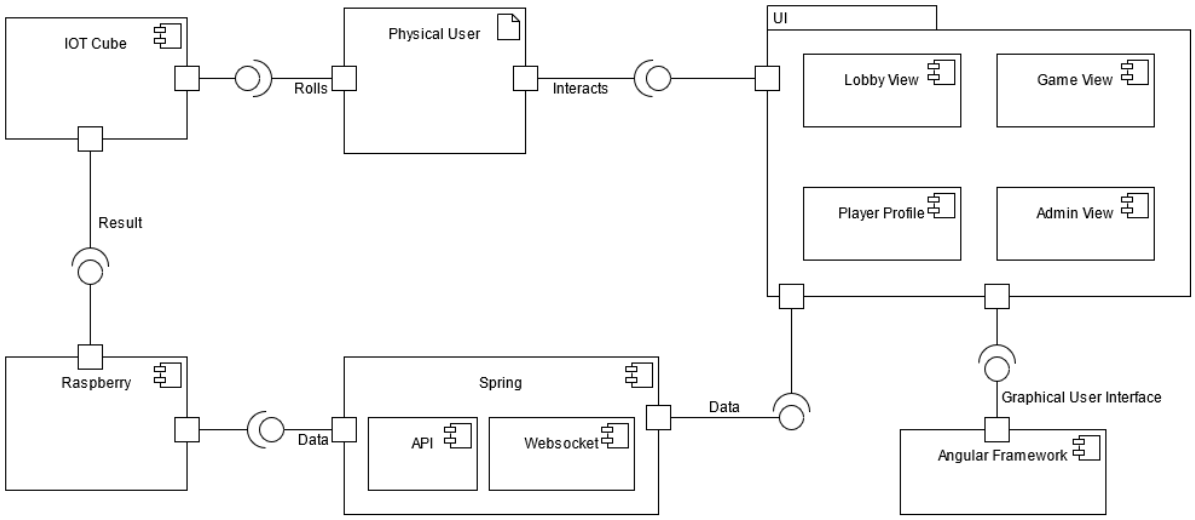
Team
Während einem laufenden Spiel werden teamspezifische Informationen in dieser Klasse verwaltet.

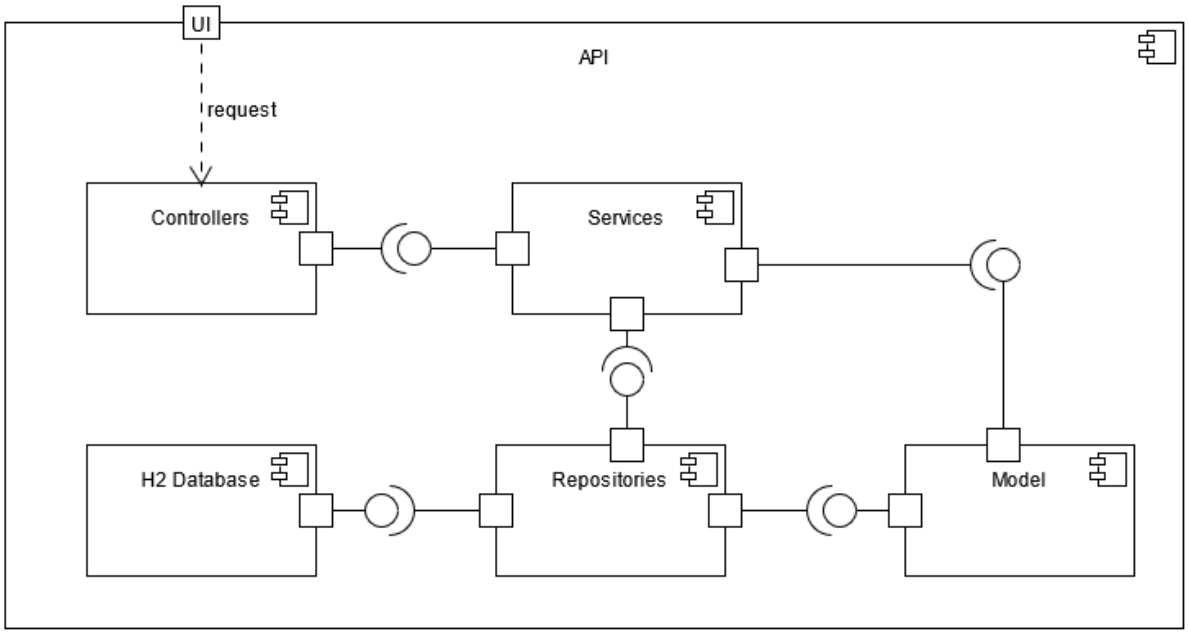
CompletedGame
Die Informationen eines abgeschlossenen Spiels werden im „CompletedGame“ gespeichert.

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

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 H2 Database Engine

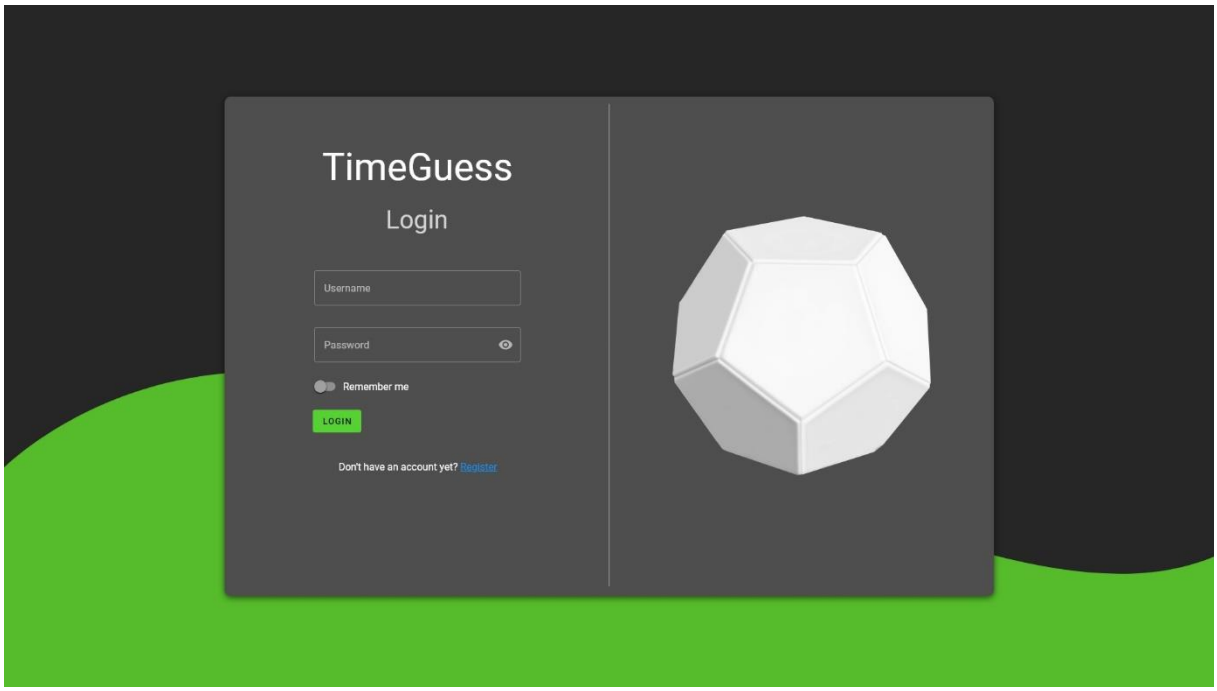
H2 Database Engine ist ein relationales in-memory Datenbankmanagementsystem, welches in Java geschrieben ist. Zum Ansprechen des Datenbanksystems können SQL, JDBC oder ORM mit Hibernate verwendet werden. H2 eignet sich gut, da es mit Spring Boot des Spring Frameworks als Standard schon implementiert ist und daher nicht noch ein zusätzlicher Datenbankserver eingerichtet werden muss.

4.2.8 JUnit

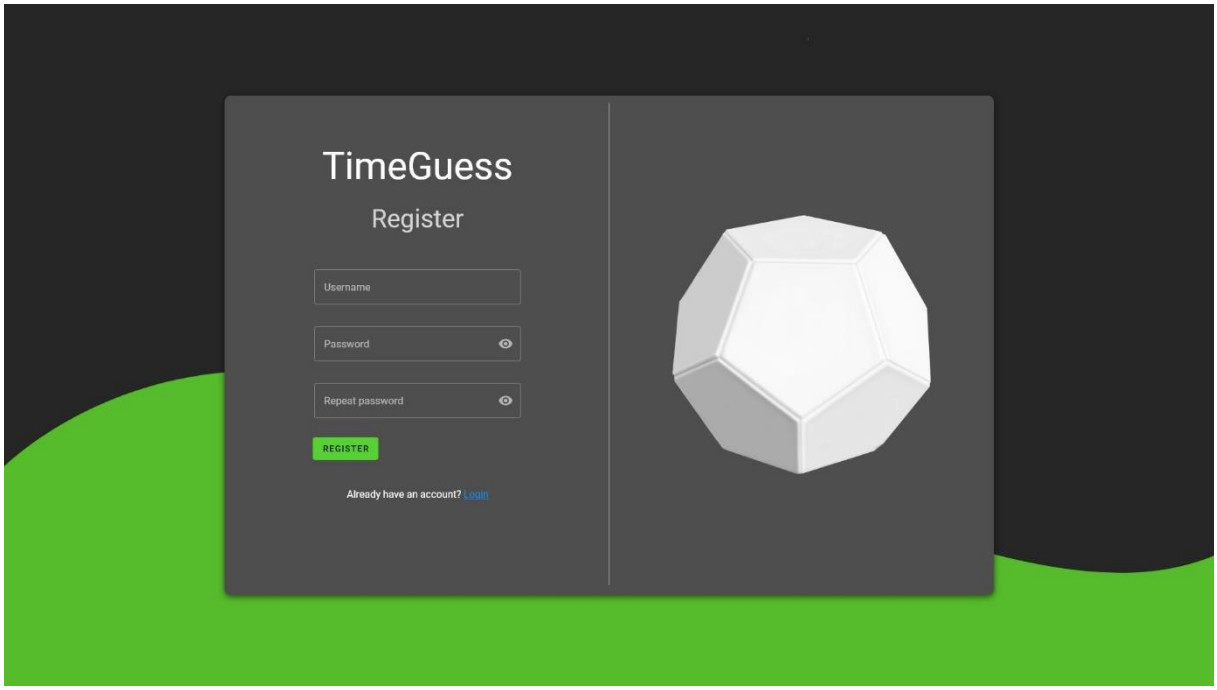
JUnit ist ein Framework zum Testen von Java Programmen. Es wird in diesem Projekt verwendet, um automatisierte Unit Tests zu schreiben, die zu einer besseren Qualität der Software führen.

5. GUI Prototyp

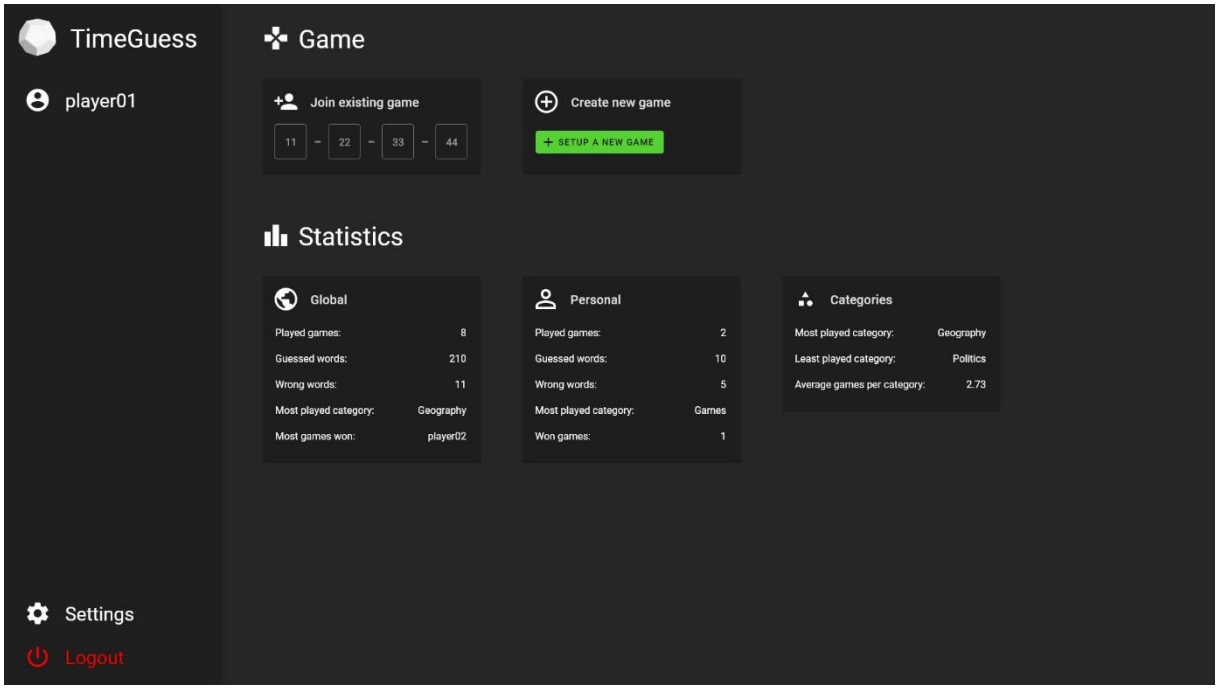
5.1 Anmelden Seite



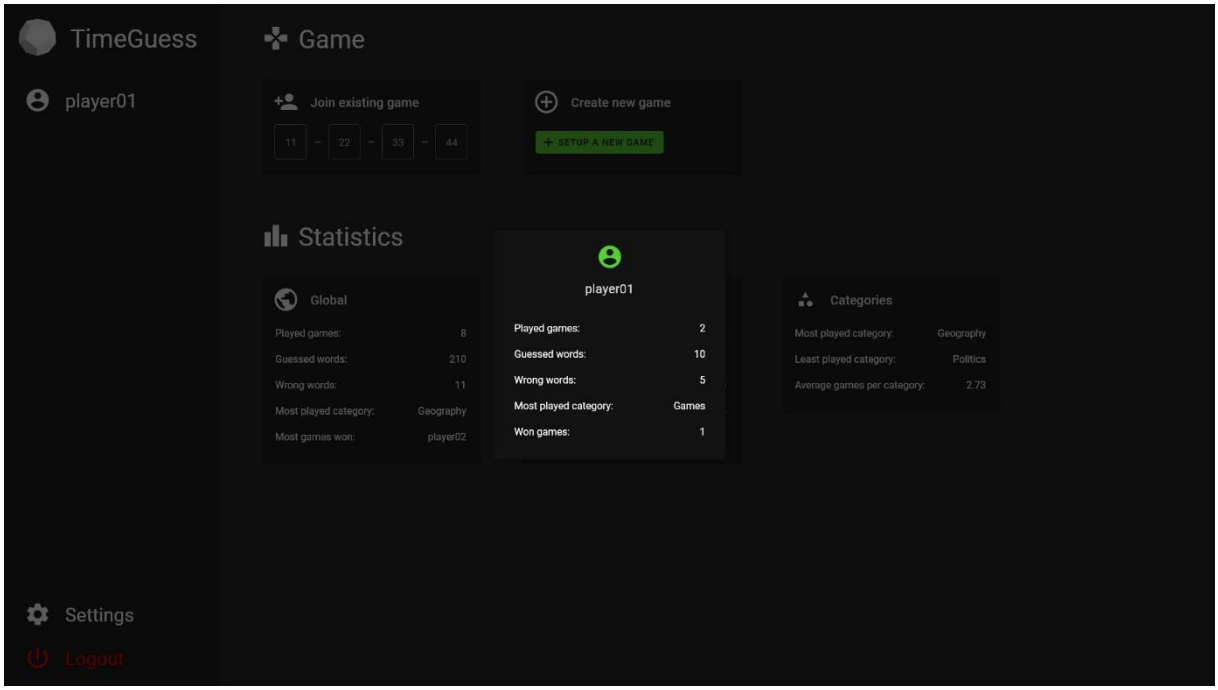
5.2 Registrieren Seite



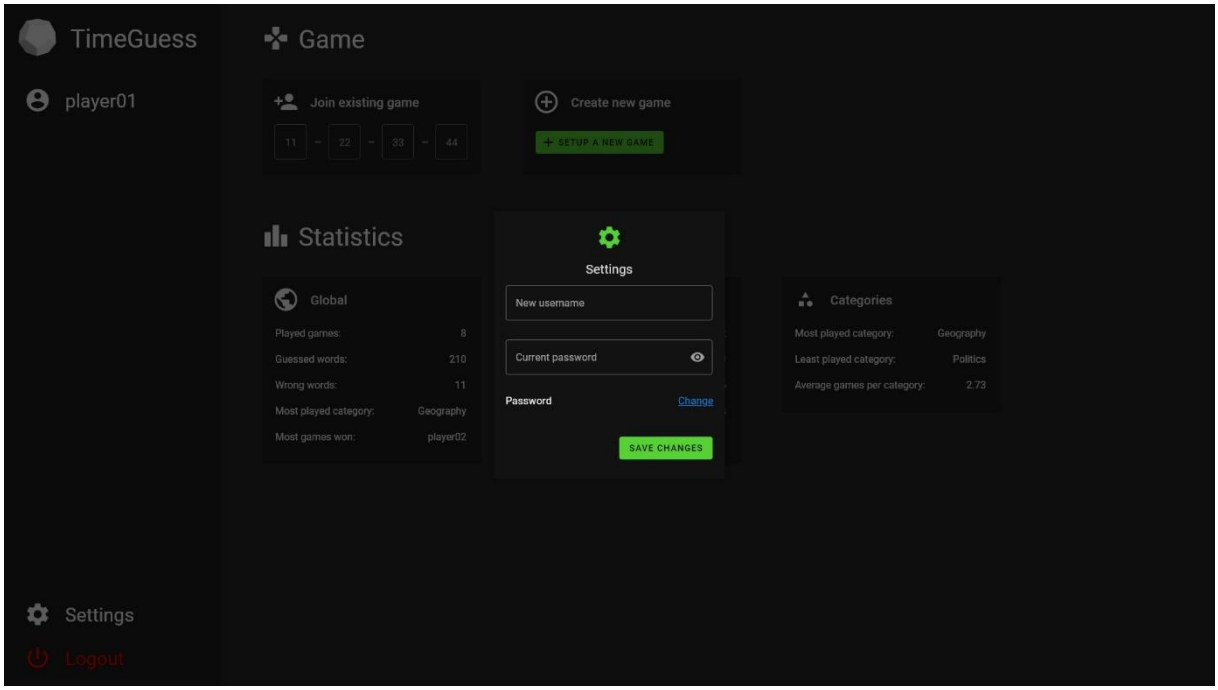
5.3 Virtuelle Lobby



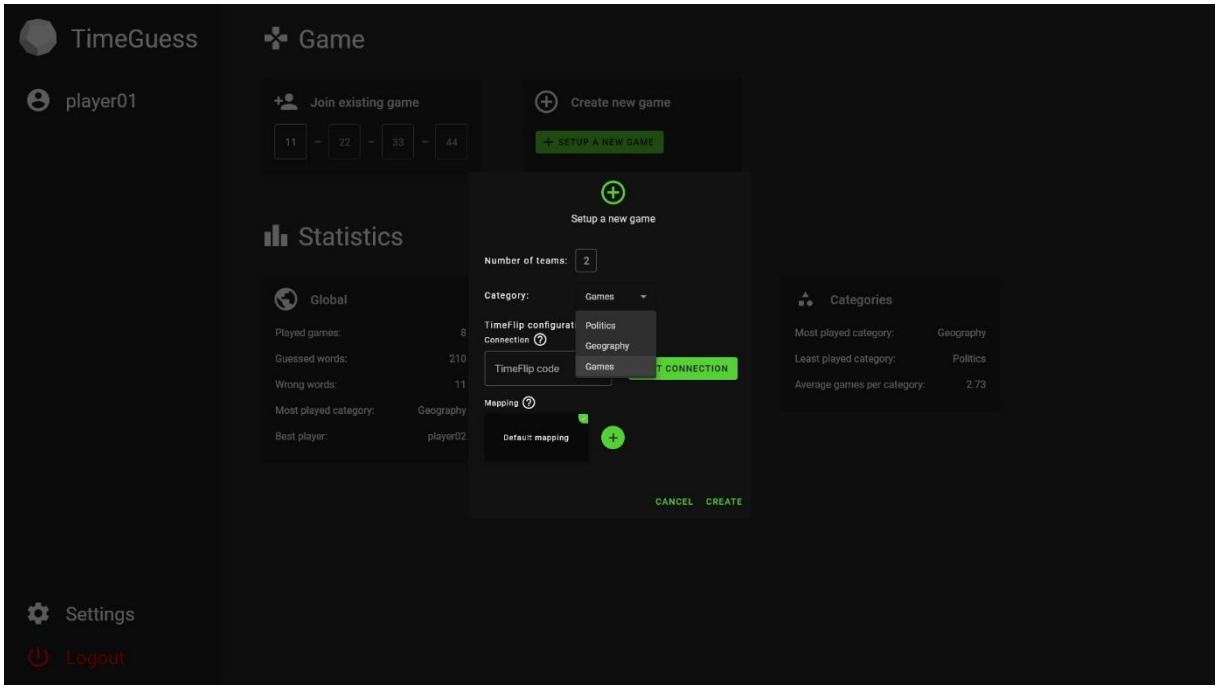
5.4 Spieler Profil



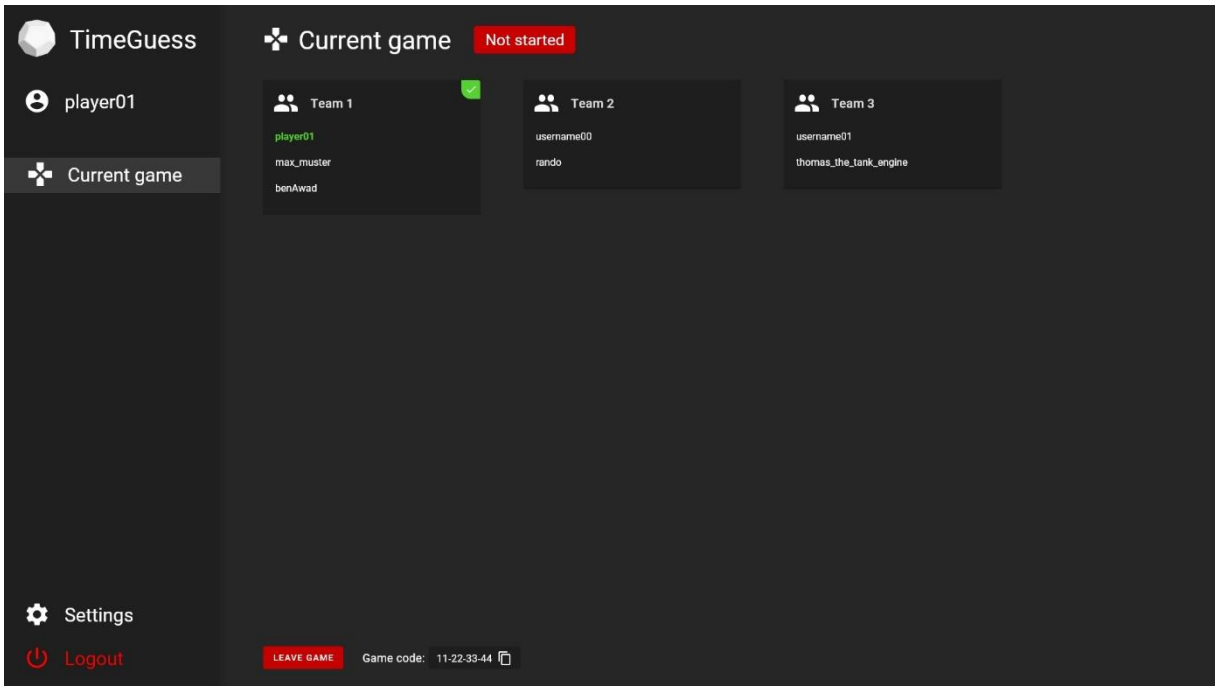
5.5 Benutzer Einstellungen



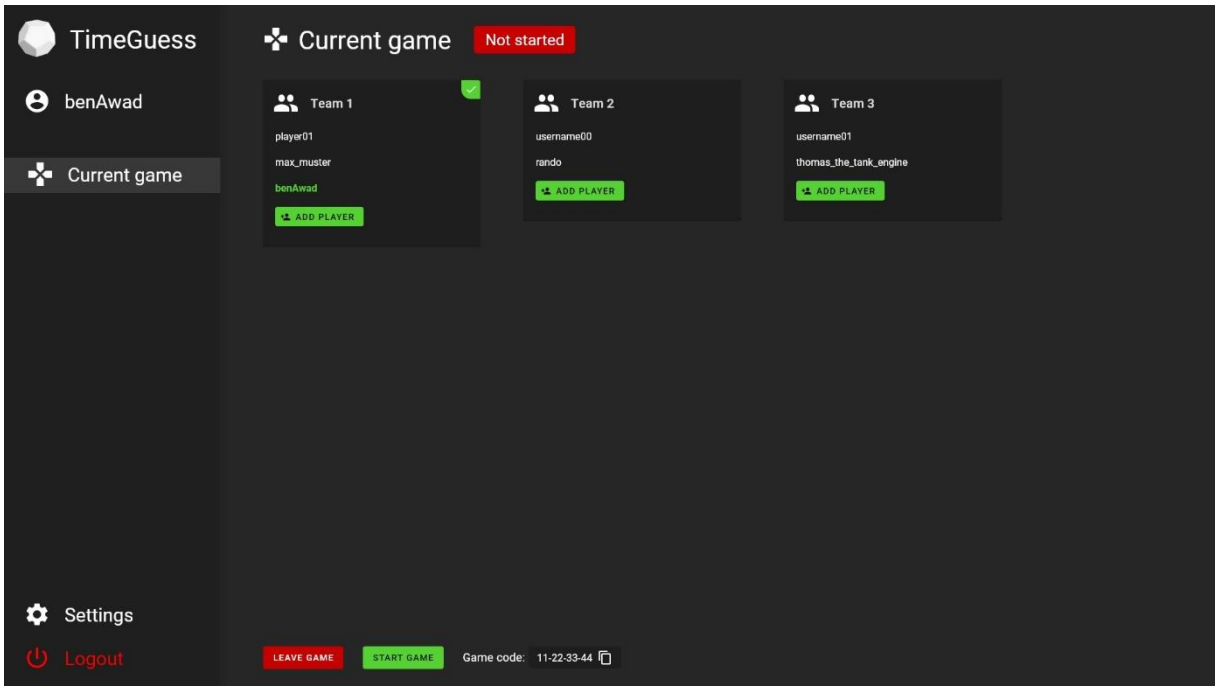
5.6 Spiel erstellen



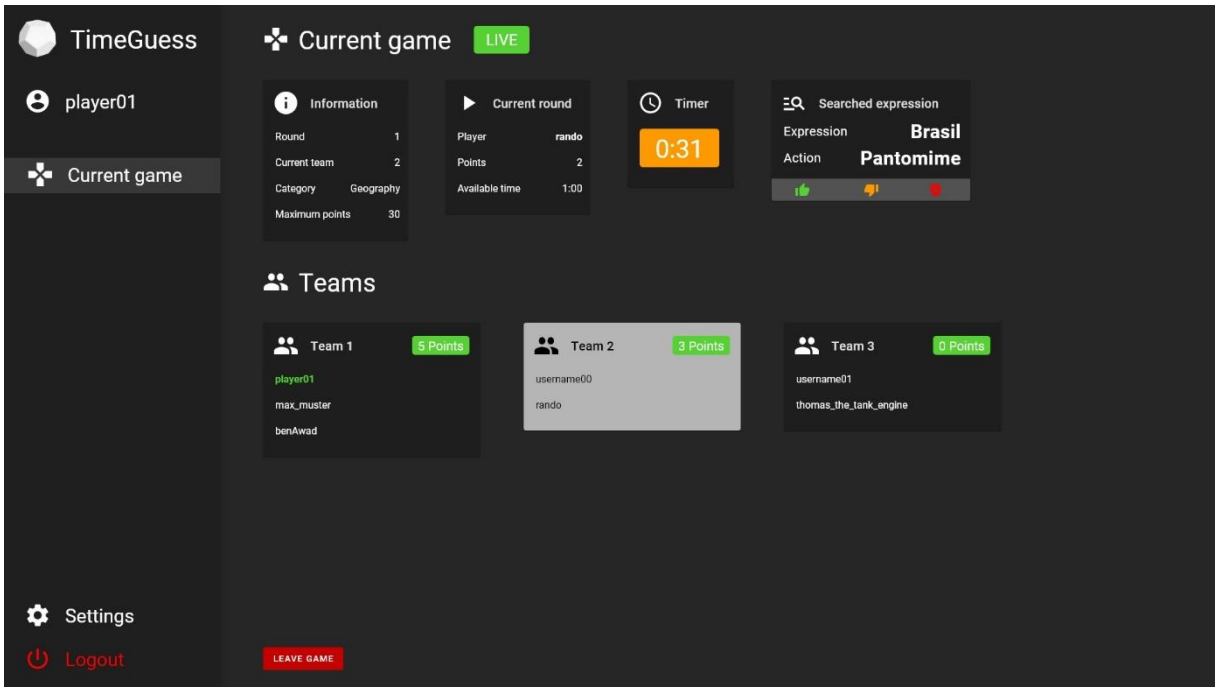
5.7 Virtueller Spiel-Wartebereich



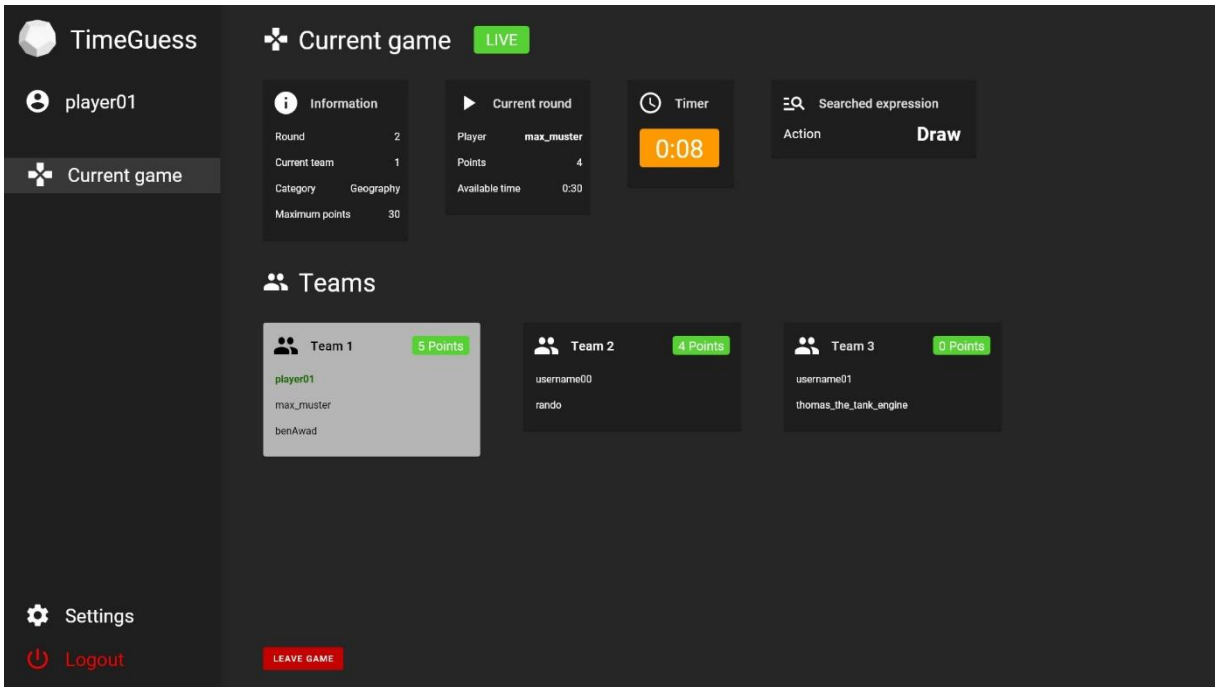
5.8 Virtueller Spiel-Wartebereich (Host)



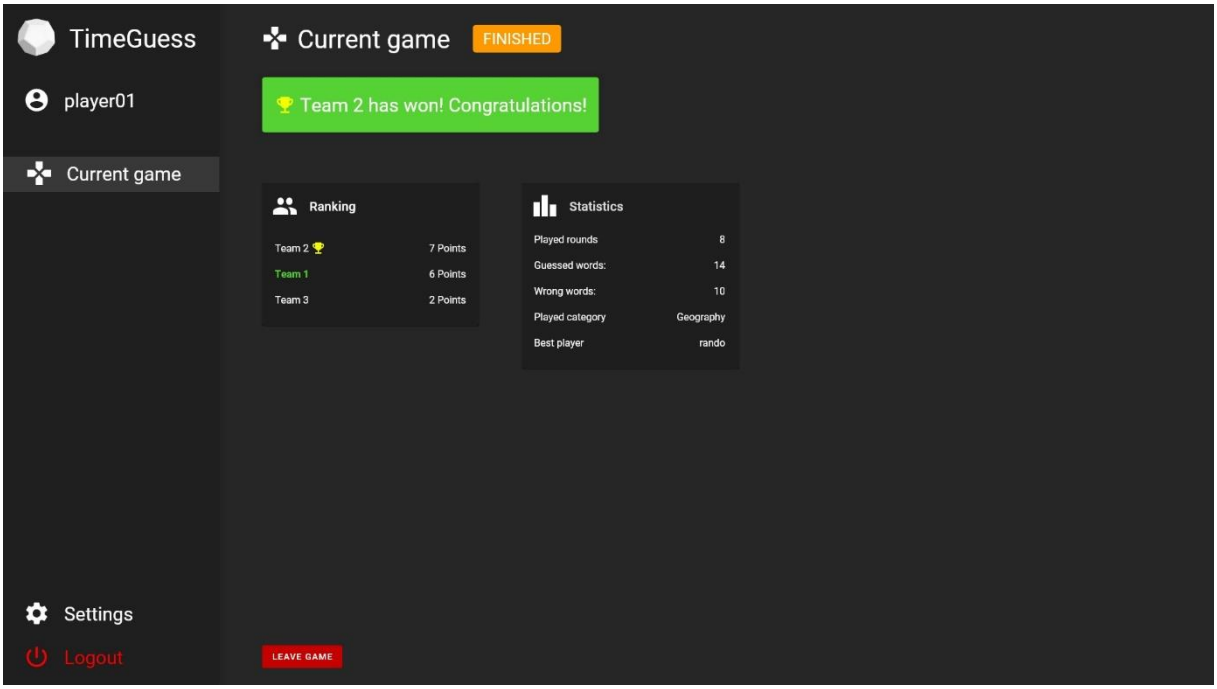
5.9 Spiel – Gegnerischer Zug



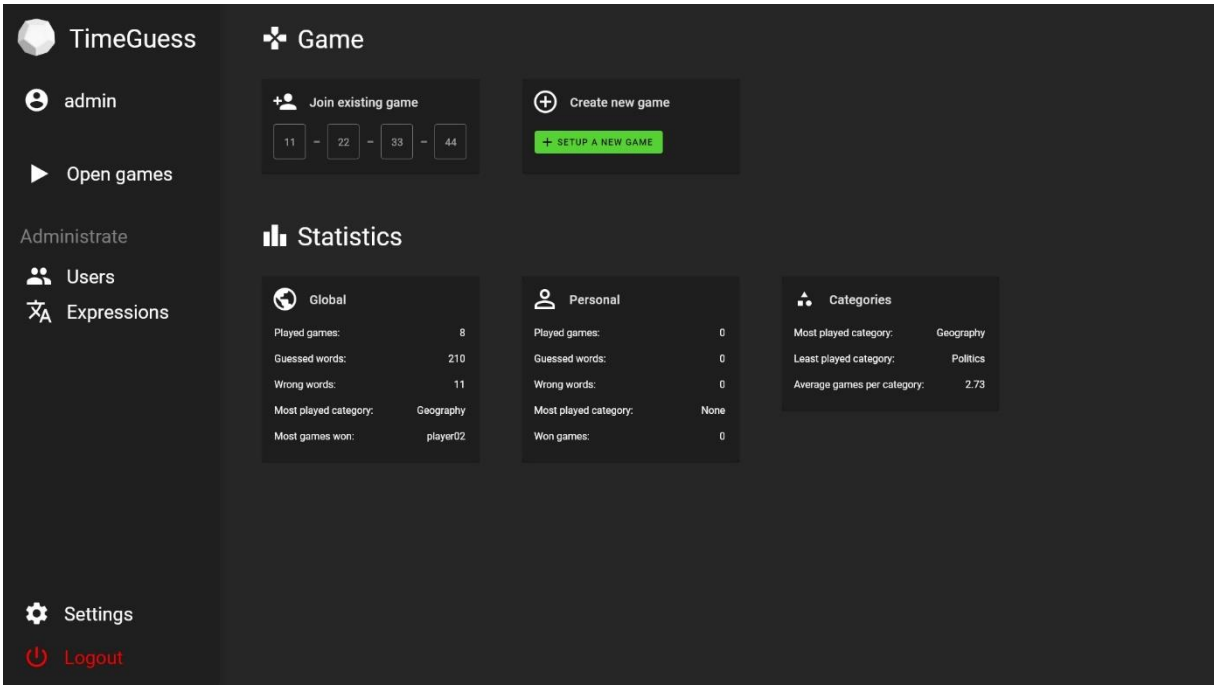
5.10 Spiel – Eigener Zug



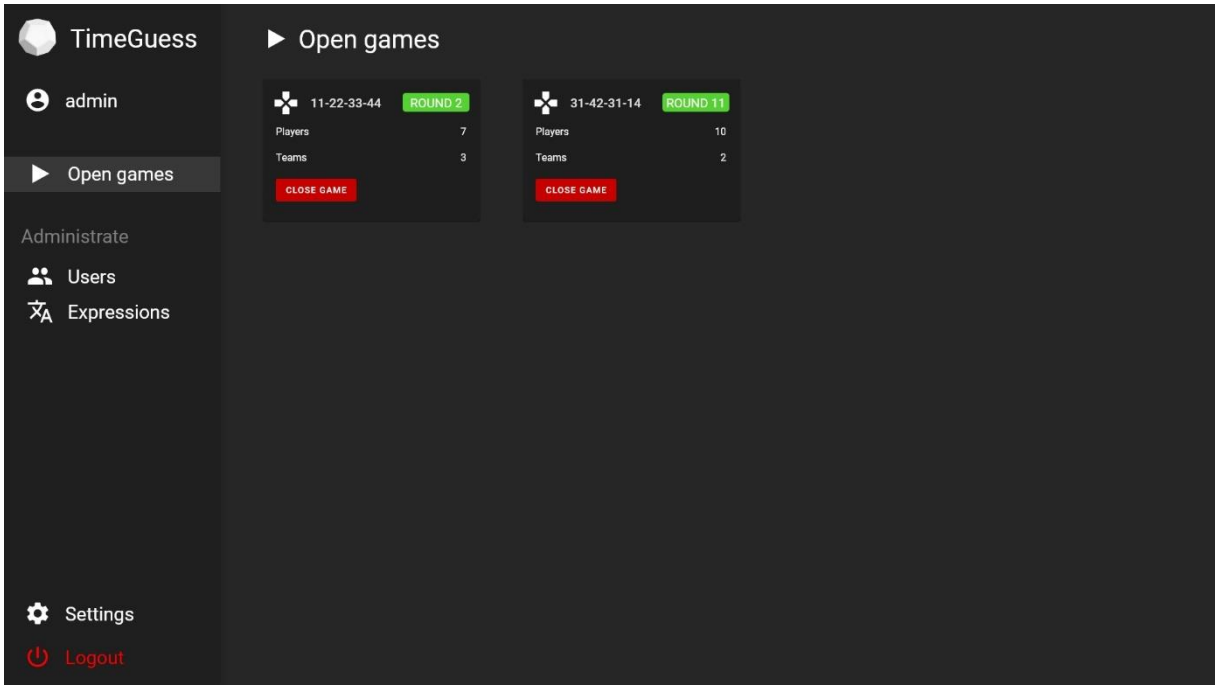
5.11 Spielende



5.12 Virtuelle Lobby (Admin)



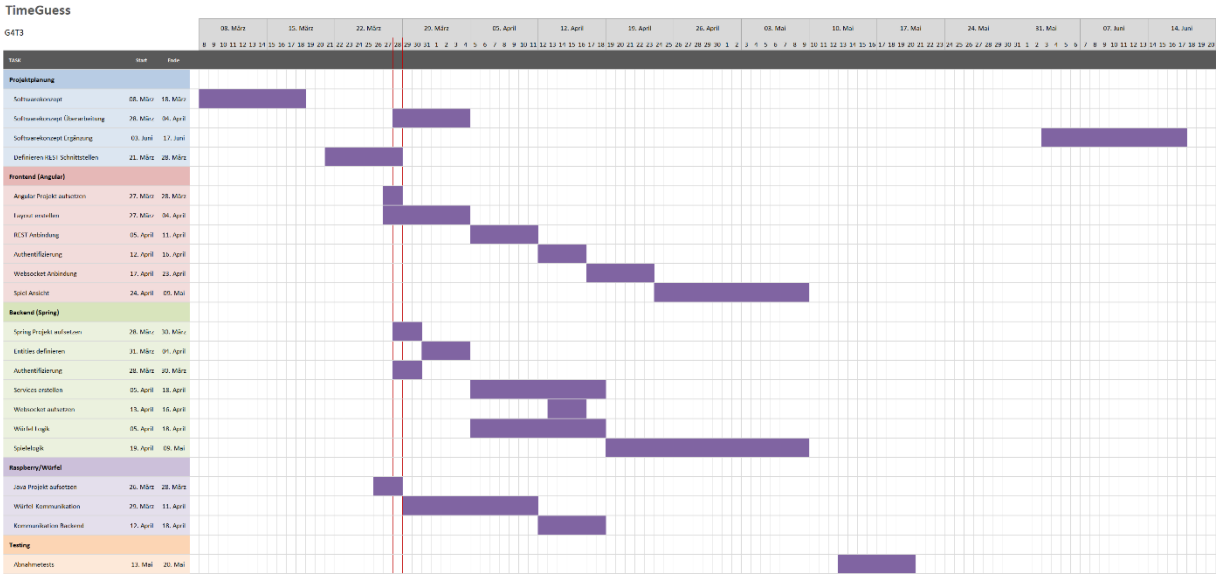
5.13 Offene Spiele Seite (Admin)



5.14 Benutzerverwaltung (Admin)

6. Projektplan

6.1 Zuständigkeiten



- **Frontend:** Matthias Thalmann
- **Backend:** Marcel Alexander Huber, 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