Concept

Team: 1

Member 1: Fabian Amhof, 11910432

Member 2: Martin Auer, 11910432

Member 3: Simon Brandacher, 11906767

Member 4: Jonas Labermeier, 11917476

Member 5: Thomas Oberroither, 11902834

Member 6: Felix Schuler, 11909204

PS group: 3

Date: 17.06.2021

1 System Overview

This software is a web-based guessing game realized with an IoT dice, Raspberry Pi and a web application. The game “TimeGuess” is suitable for an entertaining get-together for family and friends. It is intended to replace, digitize and revolutionize old-fashioned community games.

In order to make the game more interactive and dynamic, every user has an account under which he can view past games and statistics. Every player is able to open a new game room and join open rooms. When the game starts the dice is thrown every round and the teams compete. As an Example a player from the team has to draw or mime a word depending on the dice result. Then the team members have to guess it in a given time. If the team succeeds, they get points depending on the level of difficulty.

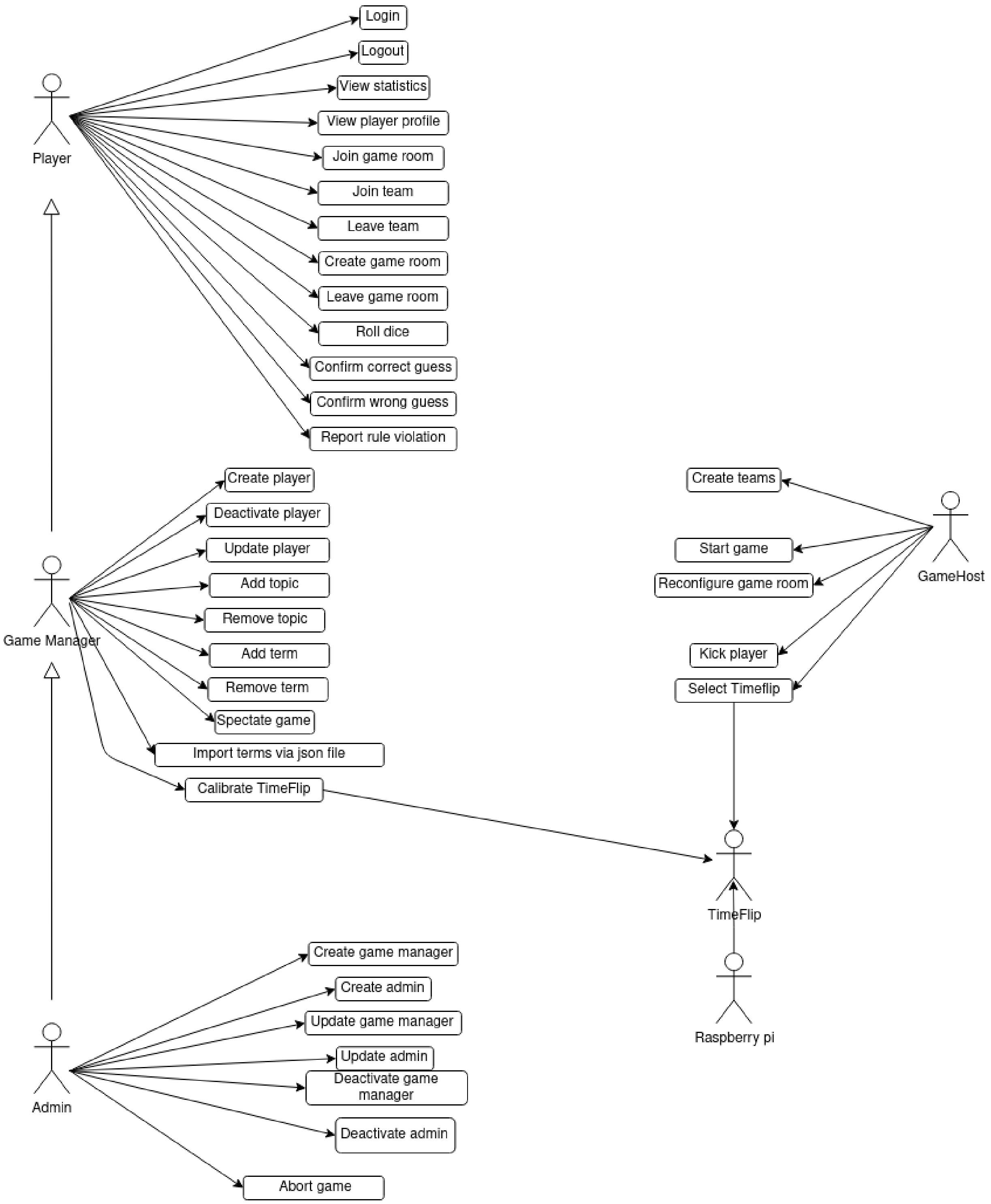
Furthermore one player gets the role of the game host assigned to do game relevant management. For example assigning players to teams or starting the game room.

Game managers ensure a varied gaming experience and take care of adding new players as well as editing and deleting them. Moreover they may add new topics to guess and watch active games.

Of course, the software also takes care of the organization and Set-up of the infrastructure. There is the administrator who can add further game managers or administrators and change the hardware configuration. He can also intervene in active games to ensure a smooth gaming experience.

2 Use Cases

2.1 Use Case Diagram



2.2 Description of the Actors

2.2.1 Actor Player

A Player is a regular user which has only the right to perform actions that do not result in any changes to the application, e.g. play a game, view statistics, etc.

2.2.2 Actor Gamemanager

A Gamemanger is a privileged user, which has the rights to perform actions that are needed to maintain the application, e.g. add new Terms and Topics or create Players. Moreover he is able to configure timeflips.

2.2.3 Admin

The Administrator is the highest role in the application, he is able to create Gamemanagers and Admins.

2.2.4 Gamehost

Gamehost is a temporary user role, which is assigned when a user creates a Gameroom. With that Role he is able to configure the Room.

A more detailed overview of the actors abilities is given in the use cases

2.3 Use Cases

2.3.1 Actor: Player

Login

- Precondition:
    - The system runs
    - The landing page is open
    - The user exists
- Procedure: The player inserts his credentials and selects “login”
- Success:
    - Game Lobby will be displayed.
    - Navigation will be displayed
- No Success:
    - An error message will be displayed
- Involved classes: User

Logout

- Precondition: The player is logged in.
- Procedure: The player selects “logout”
- Success
    - The landing page gets displayed
    - The player gets removed from every active gameroom
    - The player gets logged out
- No success: -
- Involved classes: User , GameRoom

View player profile

- Precondition: The player is logged in
- Procedure: The player selects “profile”
- Success: The players profile gets displayed
- No success: -
- Involved classes: User

View statistics

- Precondition: The player is logged in
- Procedure: The player selects “profile”
- Success: A overview of the profile and the statistics gets displayed
- No success: -
- Involved classes: User, GameRound

Join game room

- Precondition:
    - The player is logged in
    - The player has an invitation to a gameroom
- Procedure: The players clicks on the join button of the invitation
- Success:
    - The game room gets displayed
- No Success:

    - Error Message gets displayed
- Involved classes: User, GameRoom

Create game room

- Precondition: The player is logged in
- Procedure:
    - The player selects “Create game room”
- Success:
    - The game room gets created with default attributes
    - The game room gets displayed
    - The Player is Gamehost
- No success:
    - Error message gets displayed
- Involved classes: GameLobby, GameRoom, User, Team

Leave game room

- Precondition:
    - The player is in a game room
    - The game is not started
- Procedure:
    - The player selects “leave game room”
- Success:
    - Player gets removed from team
    - Player gets removed from game room
    - Game lobby gets displayed
- No success: -
- Involved classes: GameRoom, GameLobby, User, Team

Join team

- Precondition: The player is in a gameroom
- Procedure: The player selects an available team
- Success: The player is added to the team
- No success: -
- Involved classes: GameRoom, User, Team

Leave team

- Precondition:
    - The player is in a gameroom
    - The game is not started
- Procedure: The player selects “leave team”
- Success: The player is removed from the team and moved to the waiting players list.
- No success: -
- Involved classes: GameRoom, User, Team

Confirm correct guess

- Precondition:
  - Game is started
  - Opponent turns dice
- Procedure:
  - Player confirms that the term was guessed right
- Success:
  - Opponent team gets points
  - System is ready to start next round
- No success: -
- Involved classes: GameRoom, Team, User, TimeFlip

Confirm wrong guess

- Precondition:
  - Game is started
  - Opponent turns dice
- Procedure:
  - Player confirms that the term was guessed wrong
- Success:
  - Opponent team does not get any points
  - System is ready to start next round
- No success: -
- Involved classes: GameRoom, Team, User, TimeFlip

Report rule violation

- Precondition:
  - Game is started
- Procedure:
  - Player confirms that the opponent team violated the rules
- Success:
  - Opponent team loses points (score can get negative)
  - System is ready to start next round
- No success: -
- Involved classes: GameRoom, Team, User

Roll dice

- Precondition: Game is started
- Procedure:
  - Player rolls dice
- Success:
  - A new round gets started.
  - Term and remaining time get displayed
- No Success: -
- Involved classes: GameRoom, GameRound, User

2.3.2 Actor: Game Manager

Precondition for all game-manager use cases below:

The current user is logged in, and is either a game manager or an admin.

Create Player

- Precondition: inside the user-management view
- Procedure:
  - click onto “create Player”
  - enter the required data in the appearing form
  - click onto “accept”
- Success: insert user into the database
- No Success:
  - display error message
  - change the missing / invalid fields
  - retry
- Involved classes: GameManager, User

Deactivate Player

- Precondition: inside the user-management view
- Procedure:
  - search the player
  - click the toggle button to deactivate
- Success: archive user (archive: set flag to invalidate credentials)
- No Success: display error message
- Involved classes: GameManager, User

Update Player

- Precondition: inside the user-management view
- Procedure:
  - search the player
  - click onto “update”
  - the same form as if a new player is created appears, with prefilled values
  - change the data
  - save the form
- Success: update changed fields, and store them into the database
- No Success: display Error Message
- Involved classes: GameManager, User

Add Topic

- Precondition: inside the topic management view
- Procedure:
  - click onto “new topic”
  - fill out the form
  - save the form
- Success: insert topic into the database
- No Success:
  - if the topic already exists display an error message
  - if the entered values are incorrect display an error message

- Involved classes: Topic

Remove Topic

- Precondition: inside the topic-management view
- Procedure:
  - select the topic
  - click onto delete icon
- Success: archive topic (archive: set flag to invalidate topic, and remove it from all views).
- No Success: -
- Involved classes: Topic, Term

Add Term

- Precondition: inside the topic-management view
- Procedure:
  - select a topic to add the term
  - click onto “new term”
  - fill out the form
  - save the term
- Success: insert term into the database.
- No Success:
  - display error message
  - change the missing / invalid fields
  - retry
- Involved classes: Topic, Term

Remove Term

- Precondition: inside the topic-management view
- Procedure:
  - select the topic which contains the term
  - search the term
  - click onto “delete”
- Success: archive term (archive: set flag to invalidate term, and remove it from all views).
- No Success: display error message
- Involved classes: Term

Import terms from json

- Precondition: inside the topic-management view
- Procedure:
  - click on edit Topic
  - click onto upload icon
  - the file explorer opens, and accepts any json file for importing
- Success:
  - display success message
  - insert the new terms
- No Success: Specific failure messages for different cases are displayed like:
  - invalid format
  - file not found
  - already exists

  - import aborted

Spectate Game

- Precondition:
  - inside the dashboard view
  - active gameroom or game
- Procedure:
  - navigate to to the active games table
  - click “spectate game”
  - select desired game from the list of all games
- Success:
  - the user is now in ”spectate mode” ;He sees the gameroom as if he is a normal player in the room, with an additional message telling him he is spectating. The difference is that he doesn`t appear in the list of players of any team, and is not visible to any other player in that room. Further he cannot take any action in the room.
- No Success: -

Configure Timeflip:

- Precondition:
  - a Timeflip is or was connected
  - inside the timeflip settings view
- Procedure:
  - click on the edit symbol
  - fill in the form or use the fill randomly option
- Success:
  - TimeFlip is configured and shown as green in the column “configured”
- No Success: -

2.3.3 Actor: Game Host

Precondition for all game-host use cases below:

The current user is logged in, and is gamehost which means he created the game room.

Create teams

- Precondition: inside the game-room view
- Procedure:
  - click on the “create team” button
- Success: A new team list is shown
- No Success: -

Start game

- Precondition:
  - there are at least two teams with at least two players per team
  - inside the game-room view
  - Timeflip is selected

- Procedure:
    - all players click onto the “ready” button
    - once all players are marked ready the game is started
- Success: the game is started, and all users are redirected to the game-view
- No Success: - (grey out the start button for invalid teams, team sizes or missing timeflip)

Reconfigure game room

- Precondition: inside the game-room view
- Procedure:
    - click onto “Topic” drop-down to change the topic
    - click onto “Timeflip” drop-down to change or select a timeflip
    - click onto “Points” drop-down to change the topic
- Success: changed game settings
- No Success: display error message

Kick player

- Procedure:
    - click on the remove symbol next to a player
- Success: The player is removed from the room and ends up in his lobby or spectator view if administrator
- No-Success: -

Select Timeflip

- Precondition:
    - Atleast one timeflip is online and configured
- Procedure:
    - click on the Timeflip drop-down menu
    - select a Timeflip
- Success:
    - TimeFlip is assigned to gameroom
- No Success: -

2.3.4 Actor: Admin

Precondition for all game-manager use cases below:

The current user is logged in, and is an admin.

Abort Game

- Precondition:
    - Active game
- Procedure:
    - click on the Timeflip drop-down menu
    - select a Timeflip
- Success:
    - TimeFlip is assigned to gameroom
- No Success: -

Create Gamemanger

- Precondition: inside the user-management view
- Procedure:
    - click onto "+" icon
    - enter the required data in the appearing form
    - select role Gamemanager
    - click onto "accept"
- Success: insert user into the database
- No Success:
    - display error message
    - change the missing / invalid fields
    - retry
- Involved classes: User

Create Admin

- Precondition: inside the user-management view
- Procedure:
    - click onto "+" icon
    - enter the required data in the appearing form
    - select role Admin
    - click onto "accept"
- Success: insert user into the database
- No Success:
    - display error message
    - change the missing / invalid fields
    - retry
- Involved classes: User

Deactivate Gamemanager

- Precondition: inside the user-management view
- Procedure:
    - search a user with role Gamemanager
    - click the toggle button to deactivate

- Success: archive user (archive: set flag to invalidate credentials)
- No Success: display error message
- Involved classes: User

Deactivate Admin

- Precondition: inside the user-management view
- Procedure:
  - search a user with role Admin
  - click the toggle button to deactivate
- Success: archive user (archive: set flag to invalidate credentials)
- No Success: display error message
- Involved classes: User

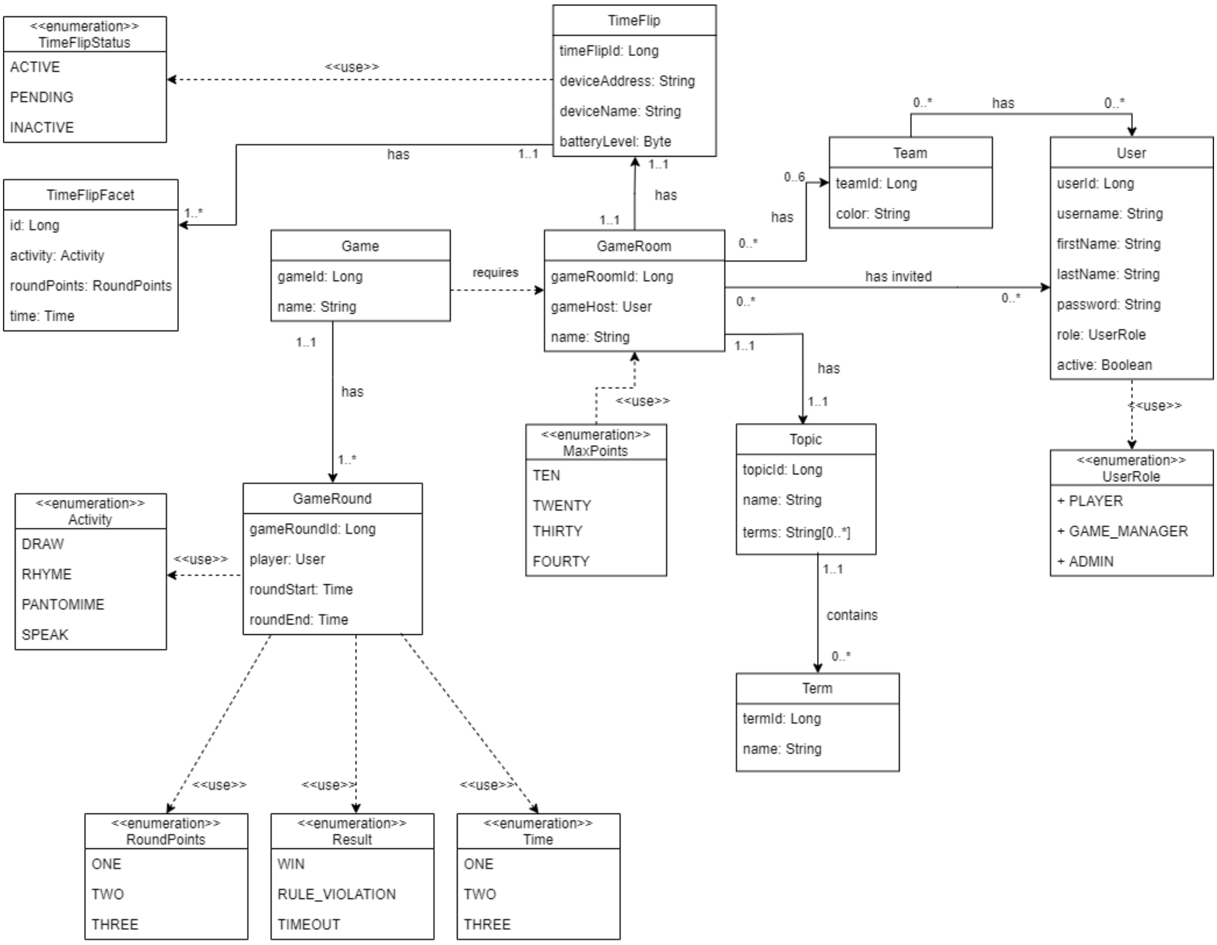
Update Admin

- Precondition: inside the user-management view
- Procedure:
  - search a user with role Admin
  - click onto edit icon
  - the same form as if a new player is created appears, with prefilled values
  - change the data e.g. name
  - save the form
- Success: update changed fields, and store them into the database
- No Success: display Error Message
- Involved classes: User

Update Gamemanager

- Precondition: inside the user-management view
- Procedure:
  - search a user with role Gamemanager
  - click onto edit icon
  - the same form as if a new player is created appears, with prefilled values
  - change the data e.g. role upgrade
  - save the form
- Success: update changed fields, and store them into the database
- No Success: display Error Message
- Involved classes: User

3 Domain Diagram



The game room is at the centre of the domain diagram. A game room consists of 0-6 teams, with each team itself having zero or more users.

Teams are used to build groups of users that play together.
Each team has a colour for identification.

All user properties are stored in the User Entity class, and each user has a role that decides on the functionality available to the user.

Each game room saves a list of invited users, that determines which users can join the game room.

Furthermore, a game room has a topic, which in turn consists of several terms. These terms represent the objects to be guessed later in the game.

The last two components of a game room are the maximum number of points per game and the TimeFlip.

A TimeFlip is associated with a game room and has an address, name, battery level and status. This information is required in the game room and game to be able to establish a connection to the TimeFlip and check whether the TimeFlip is available or not.

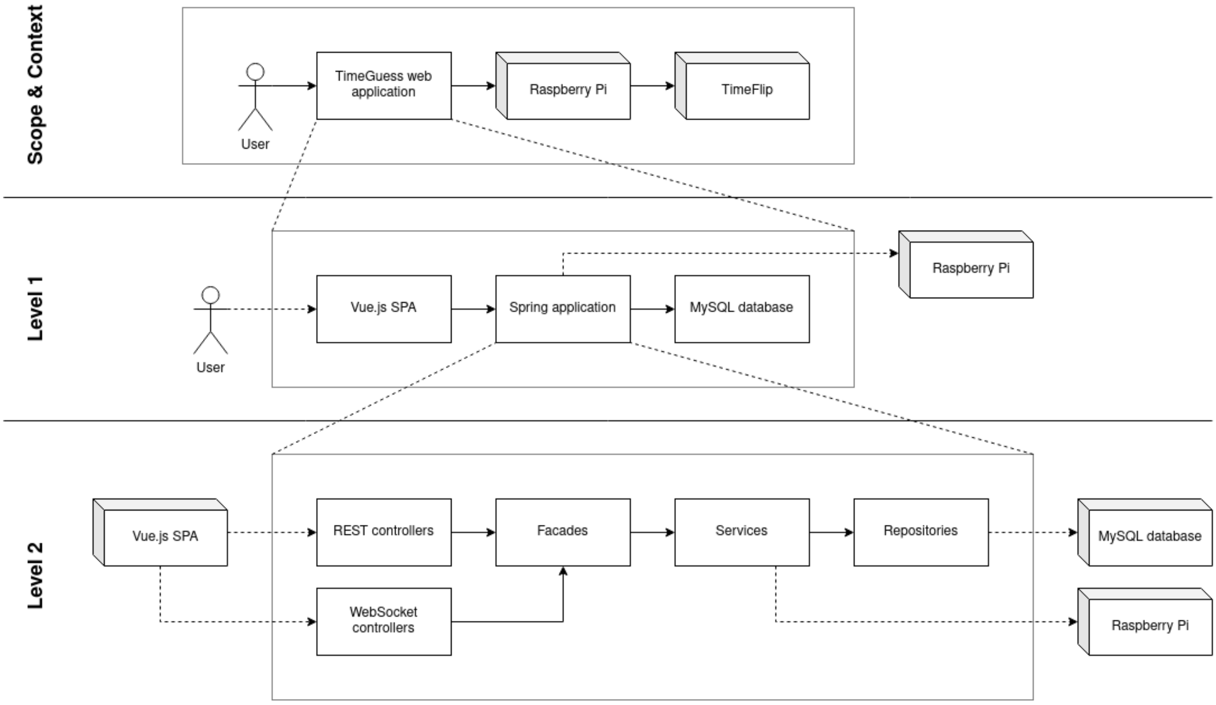
The TimeFlipFacet represents the information transmitted from the TimeFlip to the application.

Finally, a Game can be generated from a game room and has therefore all information about it. Furthermore, the game manages the game rounds played during the game.

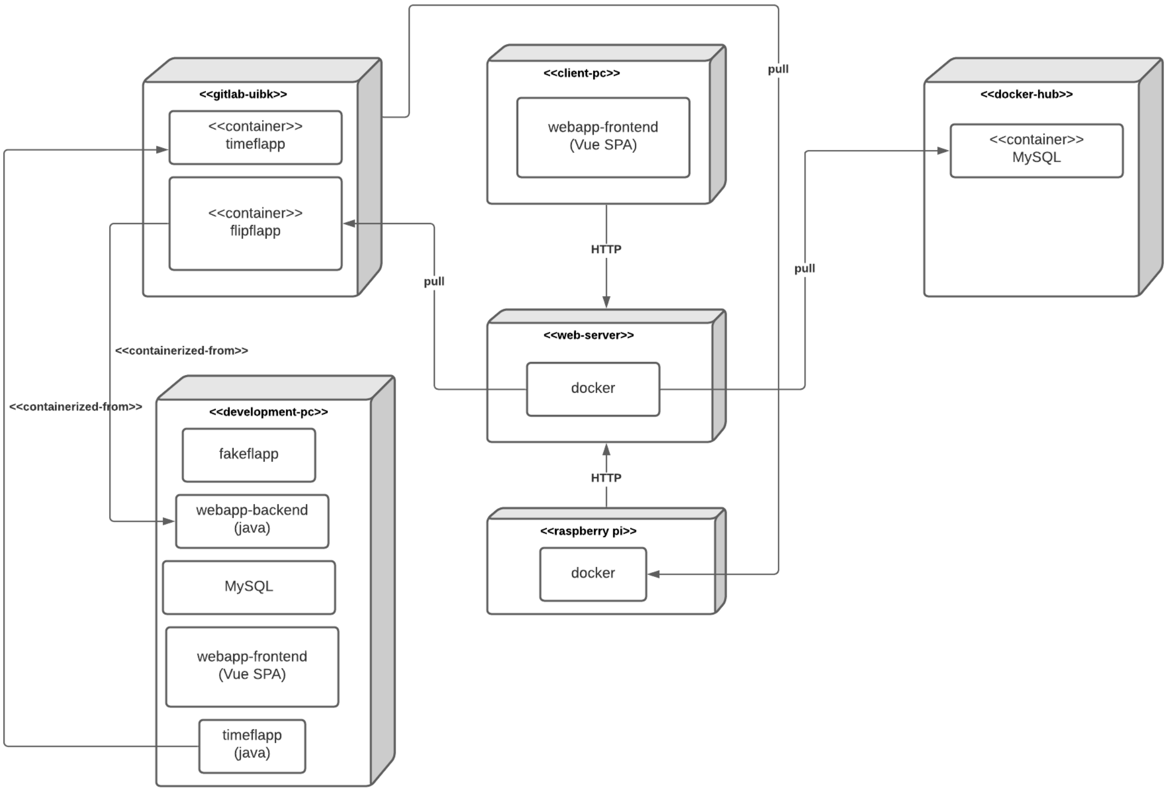
Game rounds consist of activity, round points, time for the round and the result of the round.

4 Software Architecture

4.1 Building Block View



4.2 Deployment Diagram



4.3 Technologies

4.3.1 Web Application

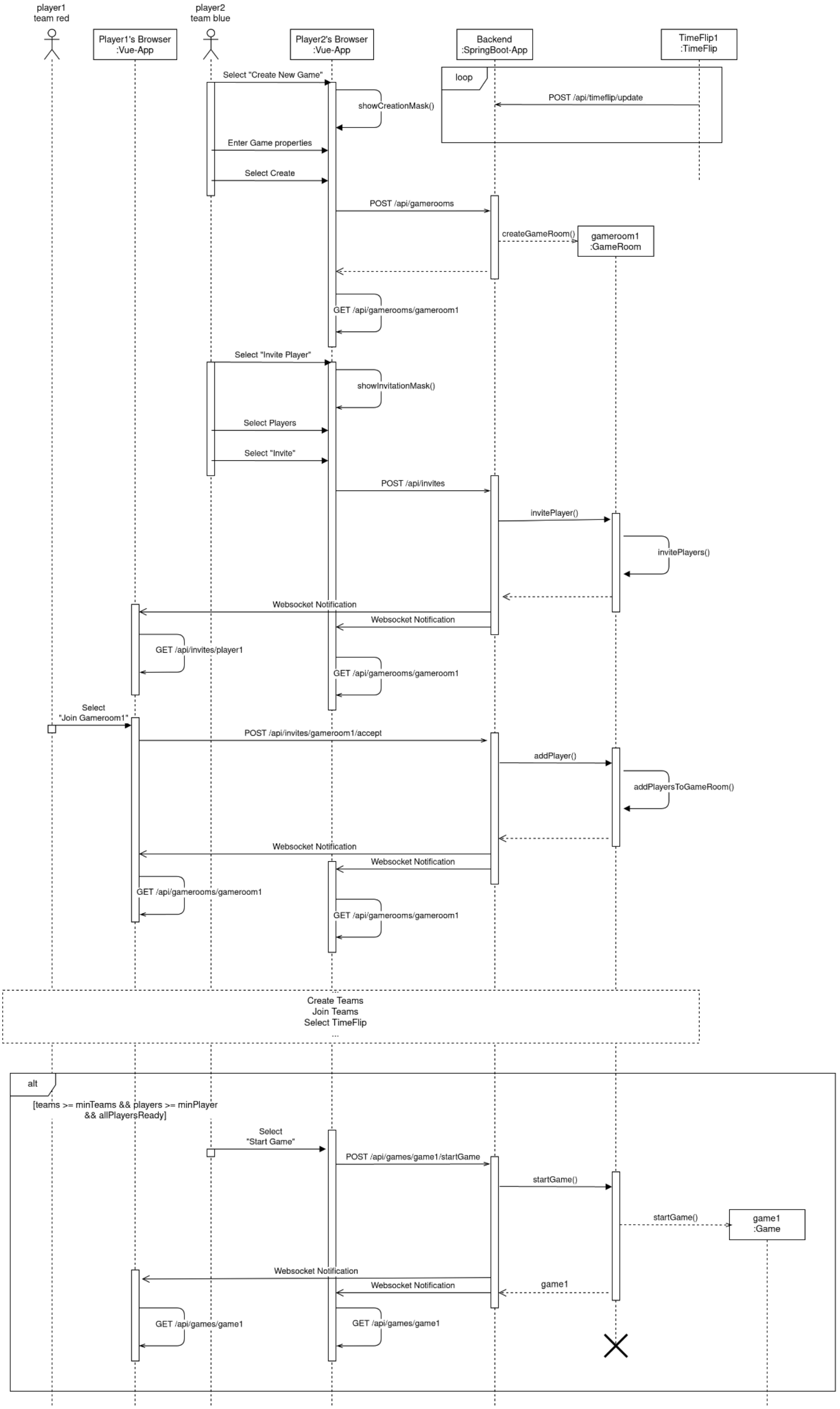
- Java
- Maven
- Vue.js: JavaScript framework for building user interfaces. We're going to build a single page application (SPA) with it. This means that the routing happens on the client. Communication with the back-end is done via REST and WebSockets.
- Vuetify: Component based UI library for Vue.js applications
- Spring (Boot): Java based framework and core of our back-end application
- JPA/Hibernate: ORM (object-relational mapping) API/library
- MySQL
- Maven: Declarative build management tool, mostly for Java projects. We are also going to integrate the front-end build process into Maven via frontend-maven-plugin.
- WebSockets: Used in the game room for the communication between clients and the server.
- JWT (JSON Web Tokens): Stored in browsers local storage and used for authentication/authorization.

4.3.2 Raspberry Pi

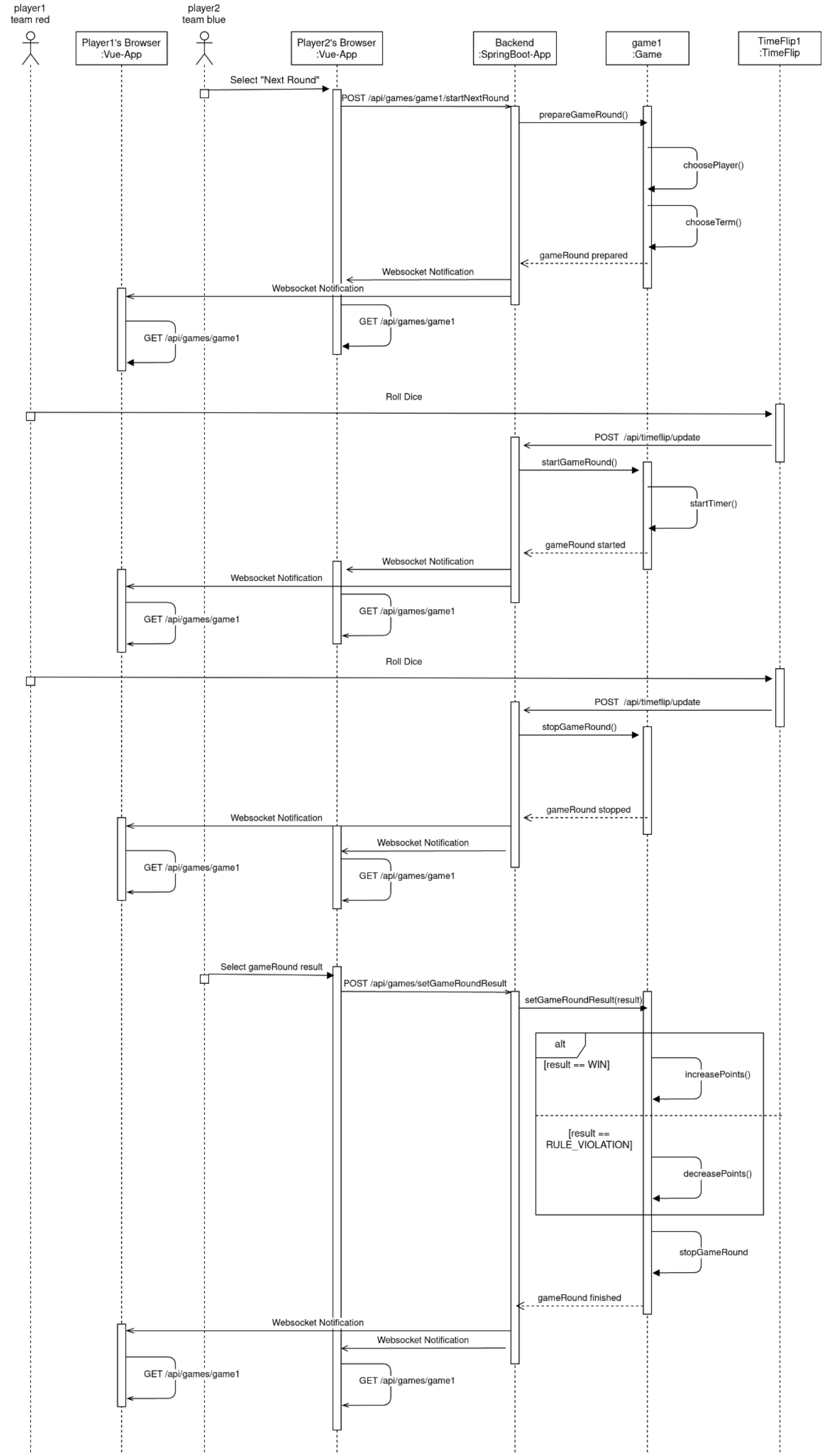
- Java
- Maven
- TinyB: Bluetooth LE library for Java and other languages
- BlueZ: Linux Bluetooth stack
- Operating System: Raspberry Pi OS
- Communication with web application via REST

4.4 Runtime views

4.4.1 Start game

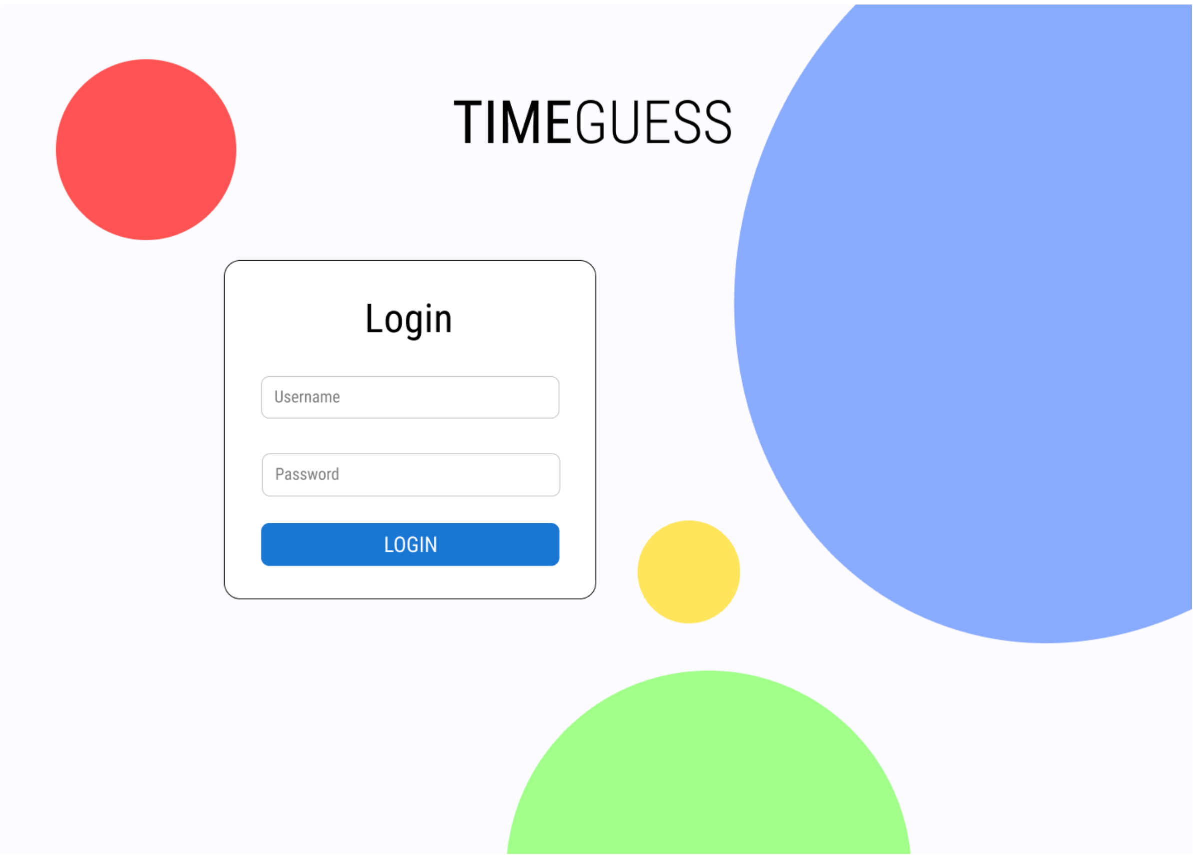


4.4.2 Game round

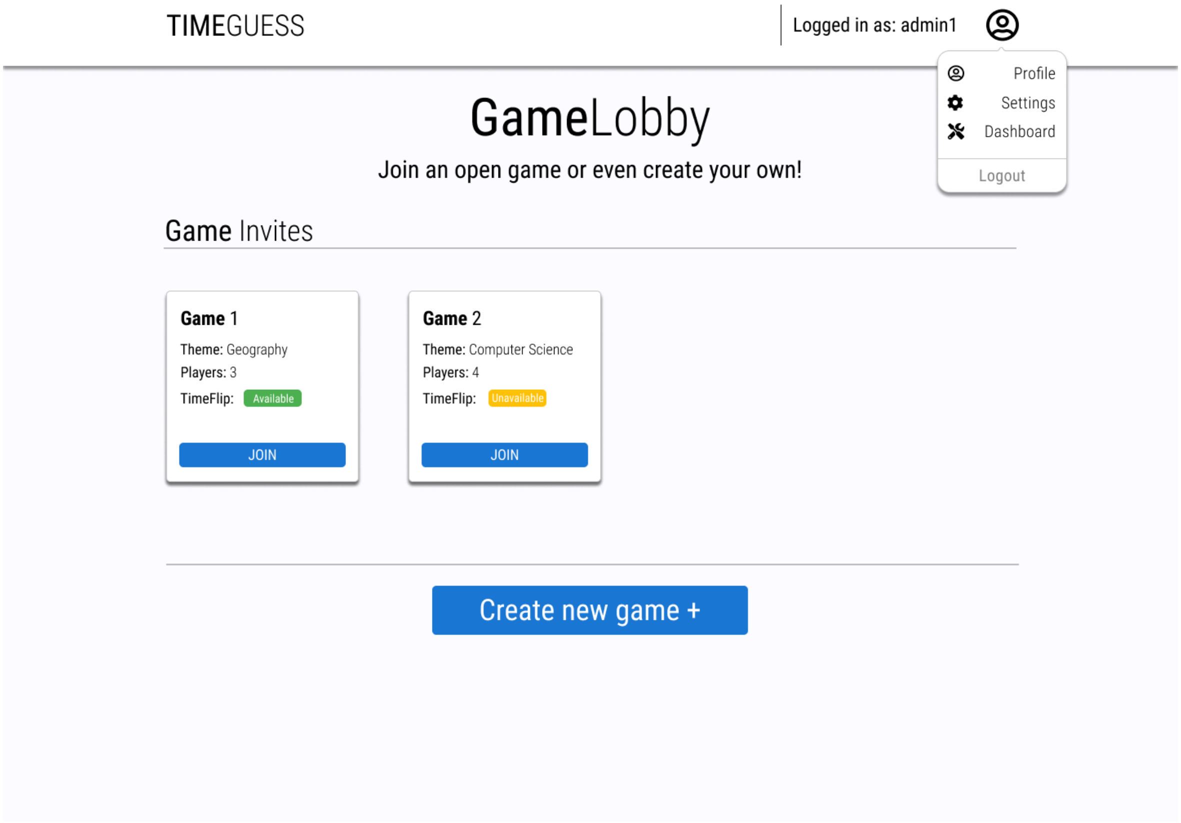


5 GUI Prototype

5.1 Login



5.2 Game Lobby

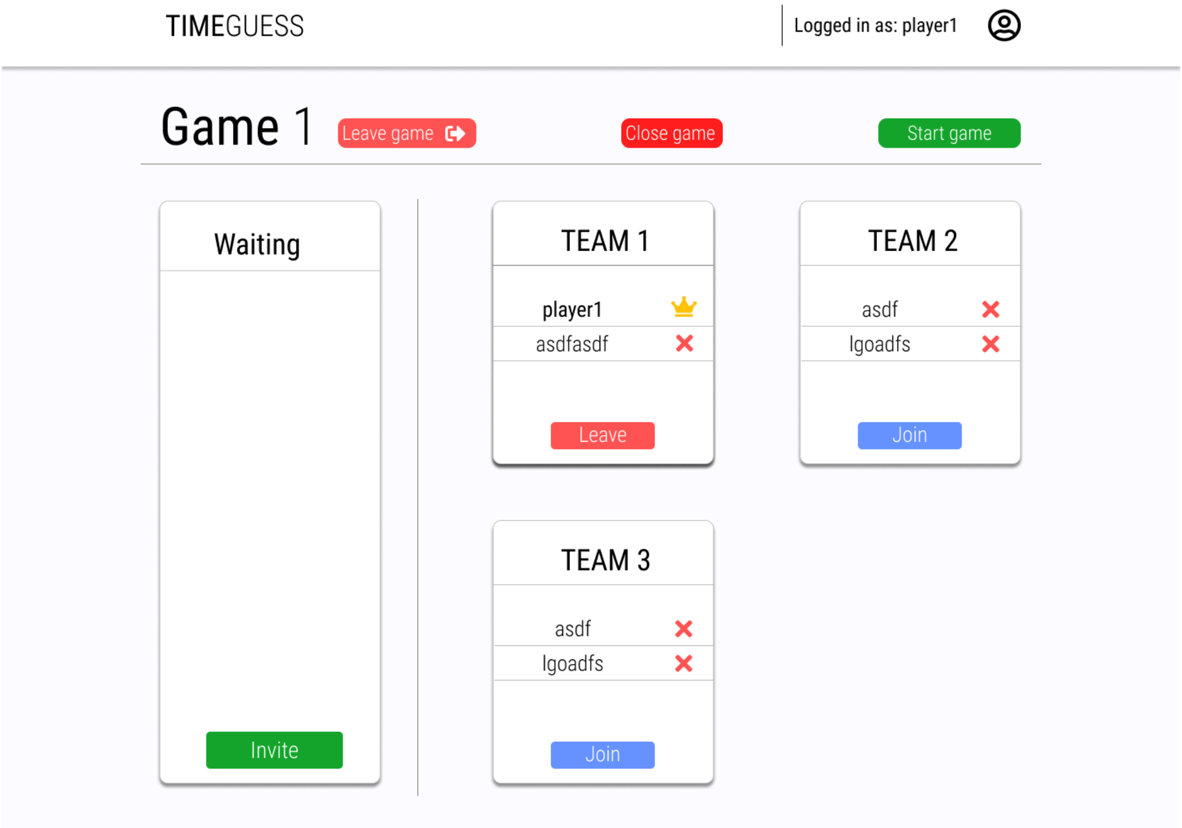


Join an open game or even create your own!

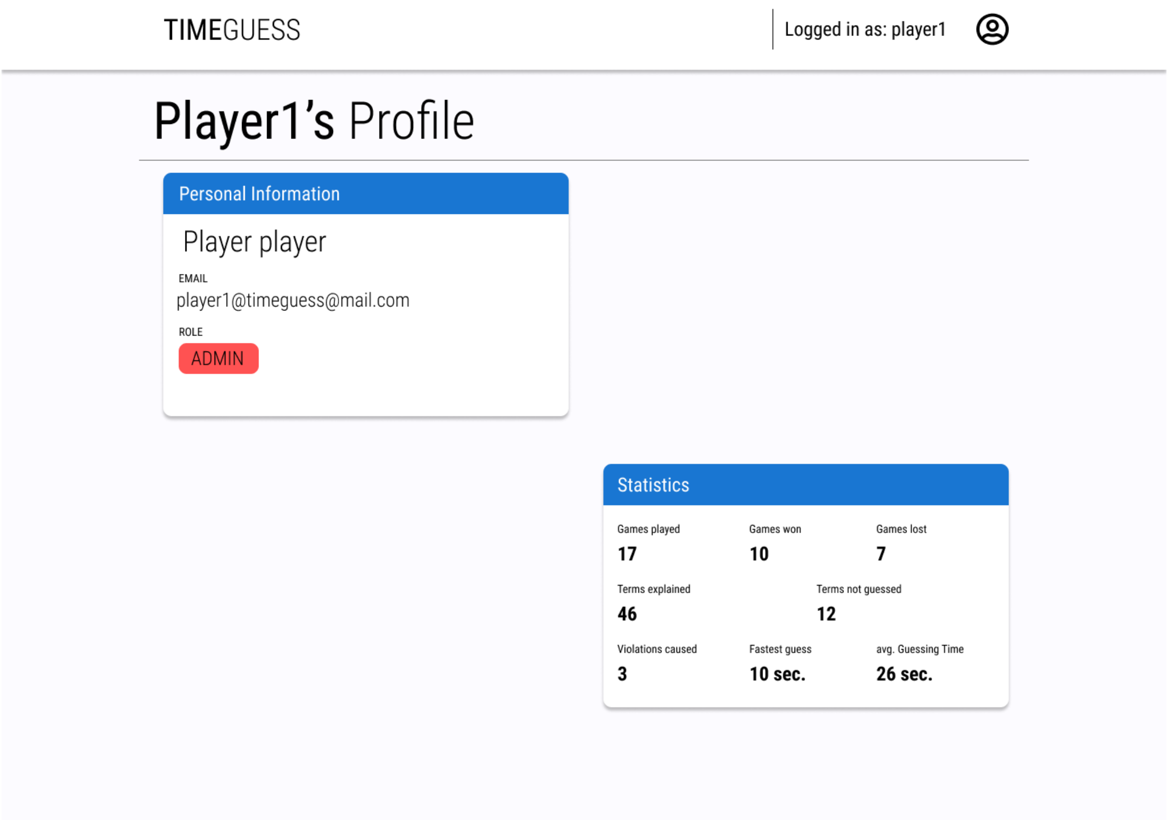
Game 1
Theme: Geography
Players: 3
TimeFlip: Available

Game 2
Theme: Computer Science
Players: 4
TimeFlip: Unavailable

5.3 Game Room



5.4 Player Profile



TIMEGUESS

Logged in as: player1

Player1's Profile

Personal Information

Player player

EMAIL
player1@timeguess@mail.com

ROLE
ADMIN

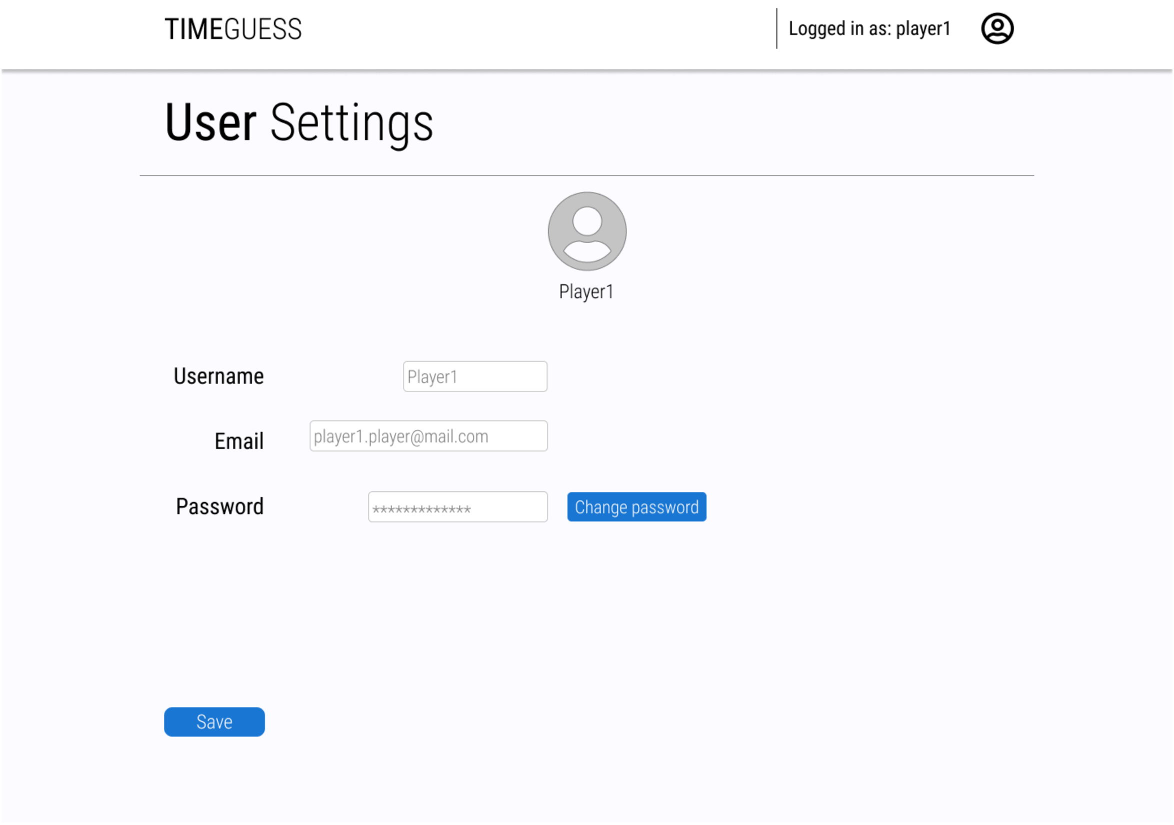
Statistics

Games played	Games won	Games lost
17	10	7

Terms explained	Terms not guessed
46	12

Violations caused	Fastest guess	avg. Guessing Time
3	10 sec.	26 sec.

5.5 User Settings



TIMEGUESS

Logged in as: player1

User Settings

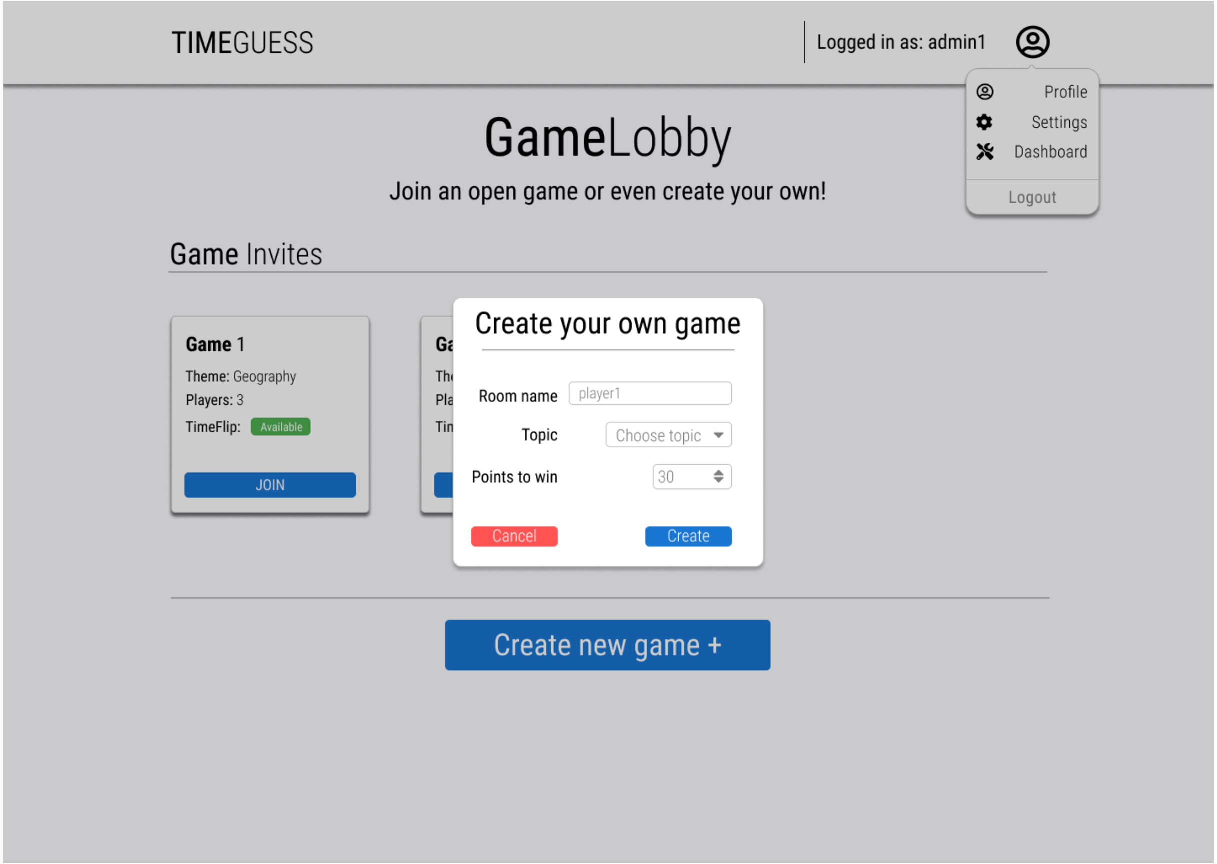
Player1

Username: Player1

Email: player1.player@mail.com

Password: *************

5.6 Create Game Room

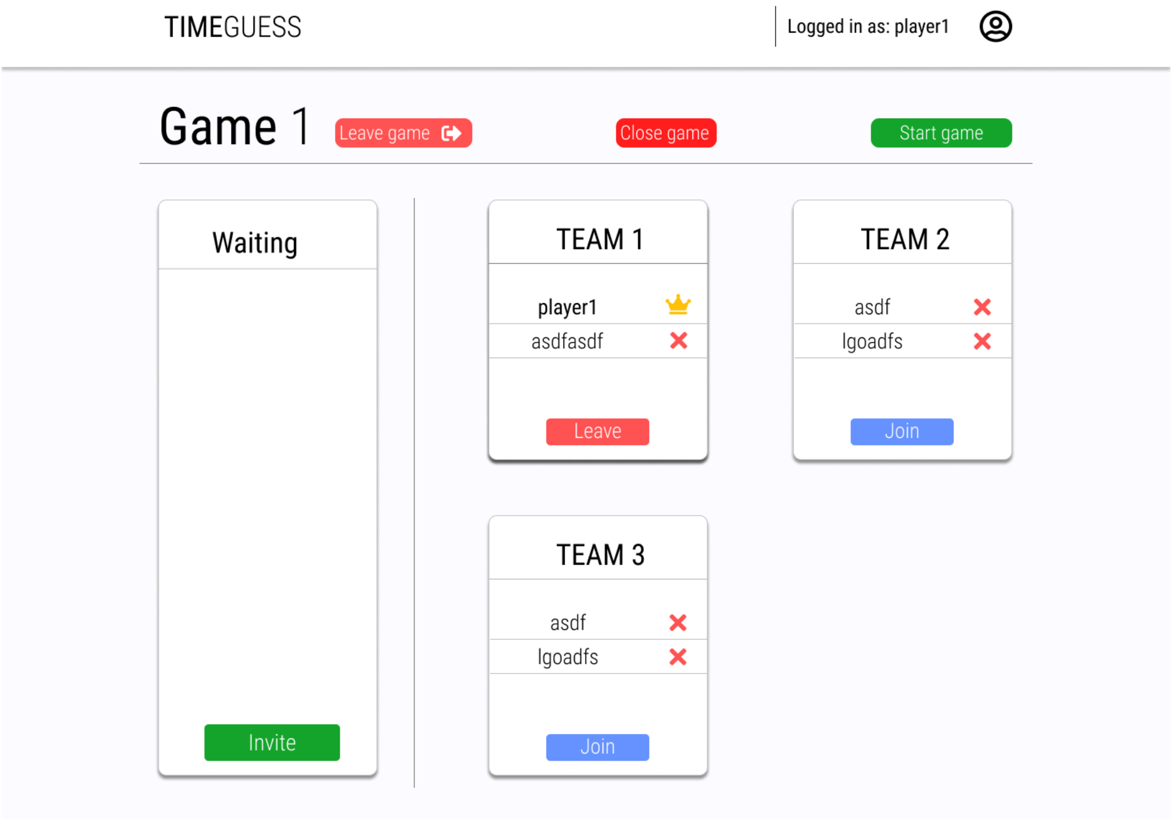


Room name: player1

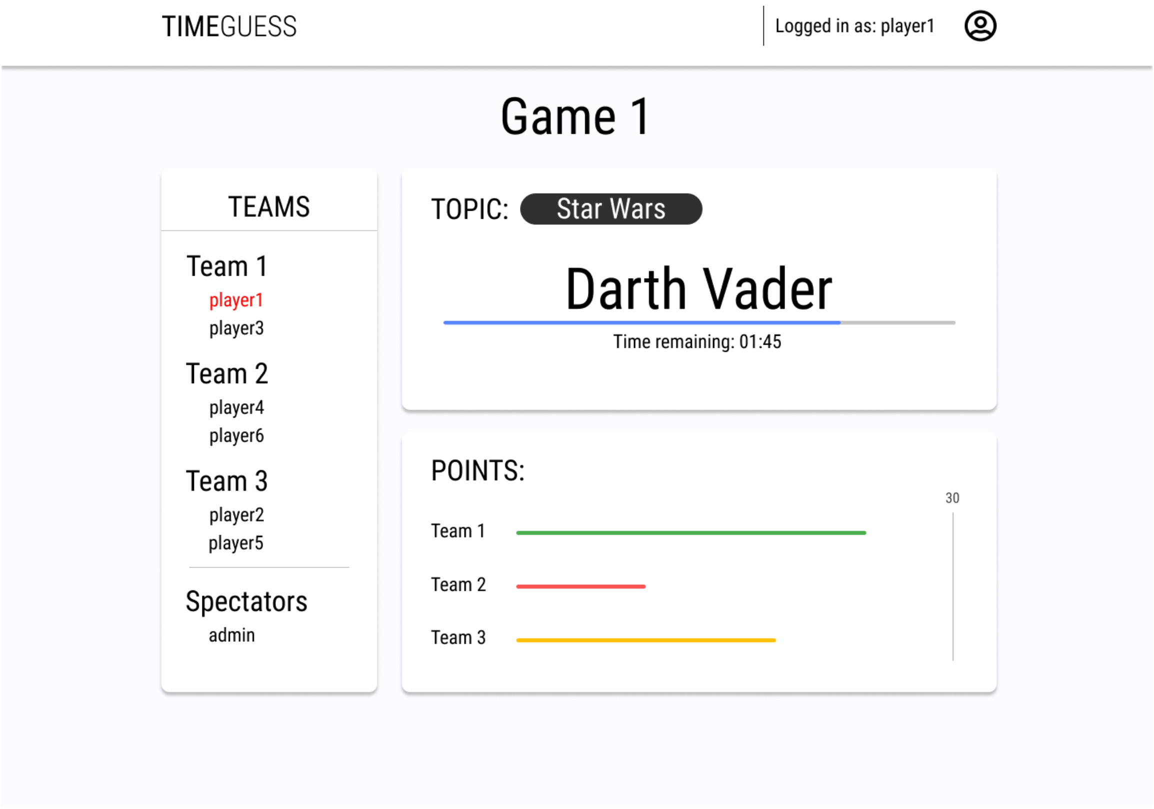
Topic: Choose topic

Points to win: 30

5.7 Game Room - open



5.8 Game Room - InGame



Game 1

TEAMS

Team 1
player1
player3

Team 2
player4
player6

Team 3
player2
player5

Spectators
admin

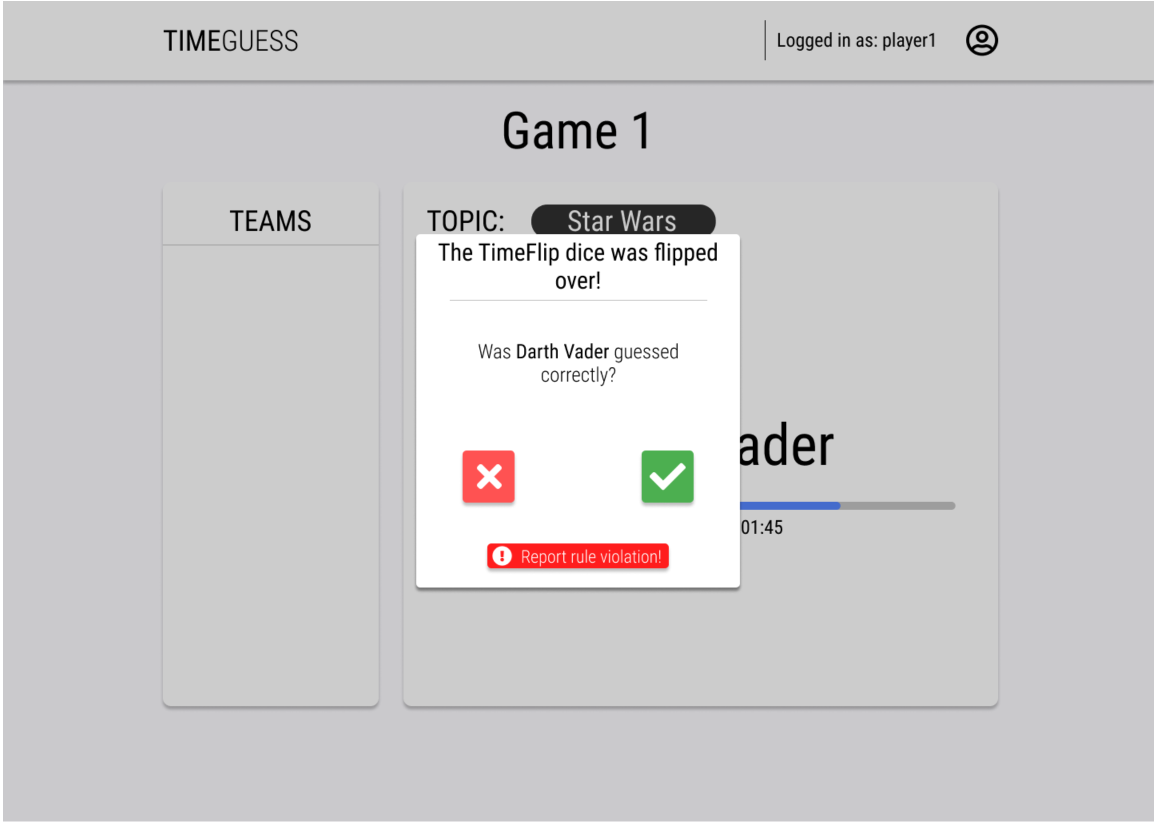
TOPIC: Star Wars

Darth Vader

Time remaining: 01:45

POINTS:

5.9 Game Room - Confirm/Deny guess

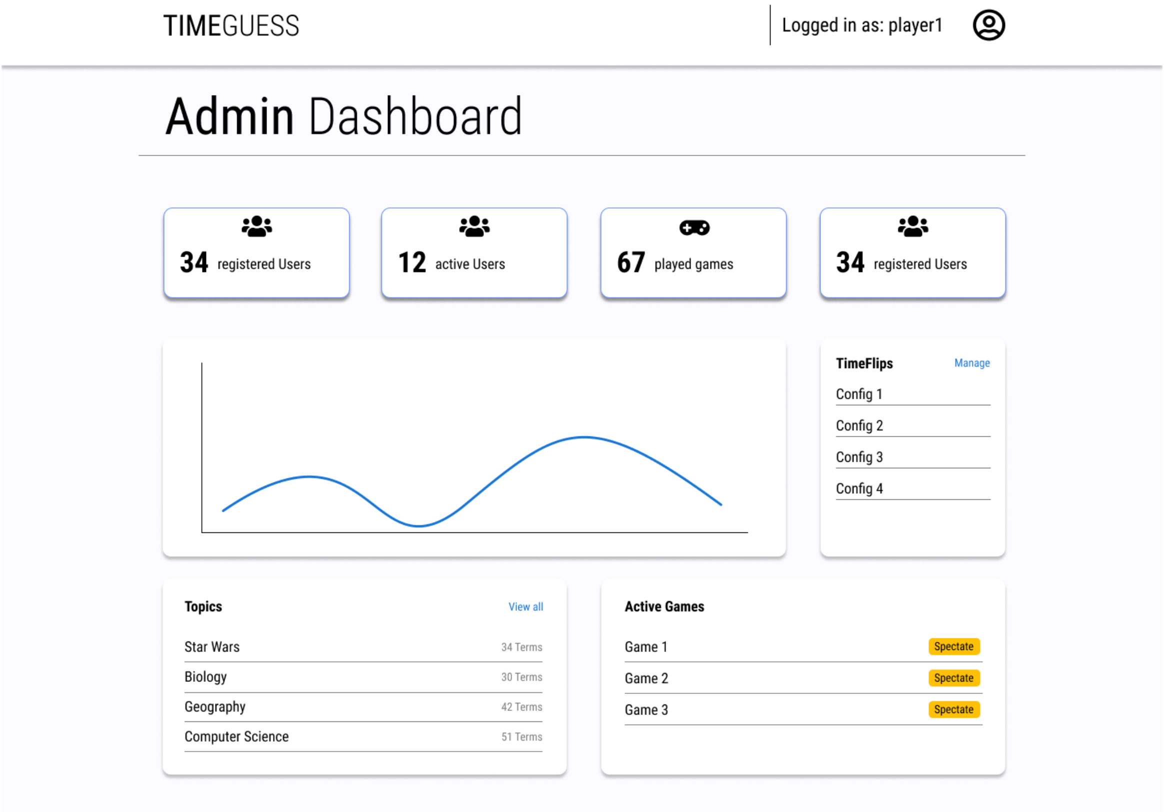


Game 1

The TimeFlip dice was flipped over!

Was **Darth Vader** guessed correctly?

5.10 Dashboard Admin



TIMEGUESS

Logged in as: player1

Admin Dashboard

34 registered Users

12 active Users

67 played games

34 registered Users

TimeFlips Manage

- Config 1
- Config 2
- Config 3
- Config 4

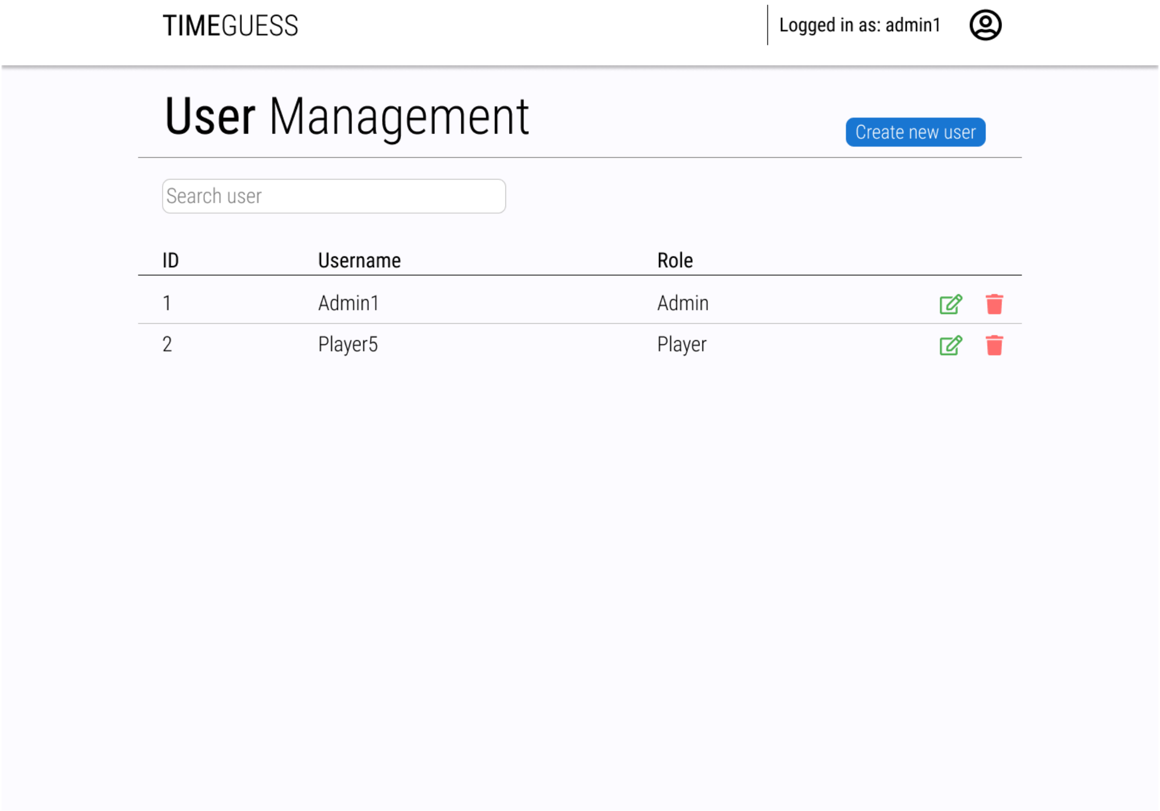
Topics View all

Star Wars	34 Terms
Biology	30 Terms
Geography	42 Terms
Computer Science	51 Terms

Active Games

Game 1	Spectate
Game 2	Spectate
Game 3	Spectate

5.11 User Management



TIMEGUESS

Logged in as: admin1

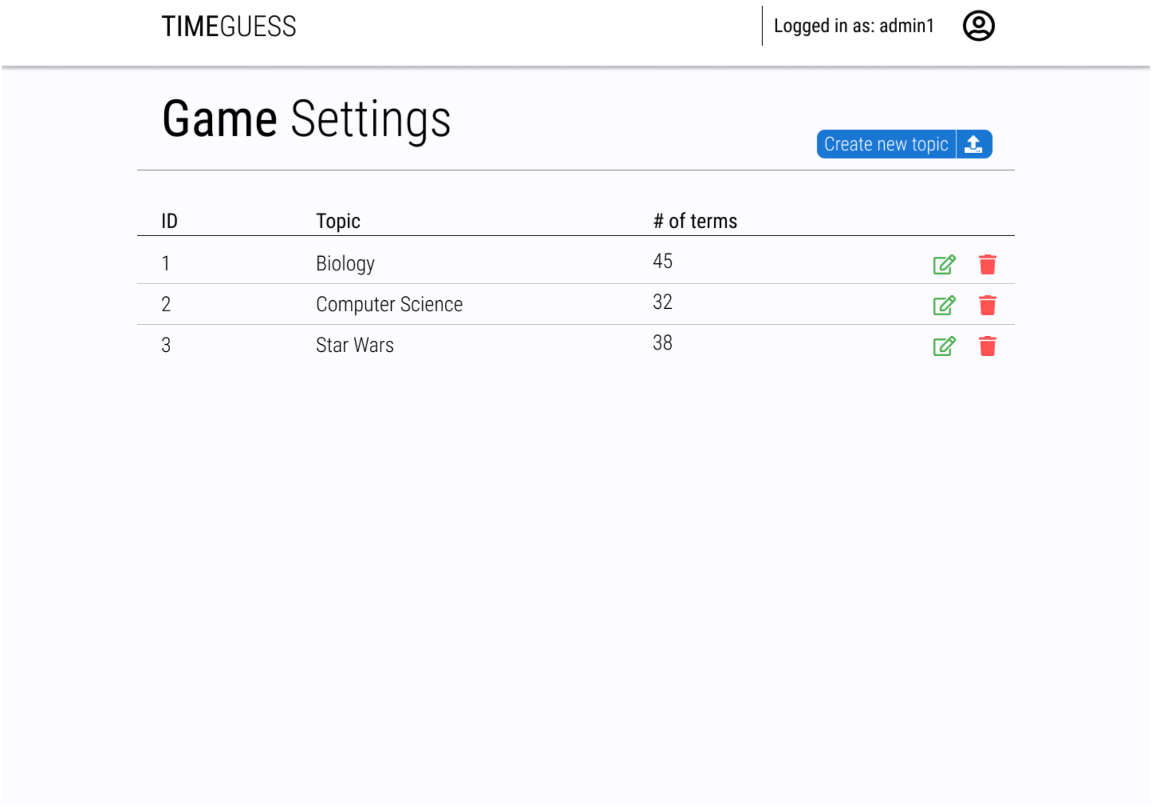
User Management

Create new user

Search user

ID	Username	Role	
1	Admin1	Admin	✎ 🗑
2	Player5	Player	✎ 🗑

5.12 Game Settings

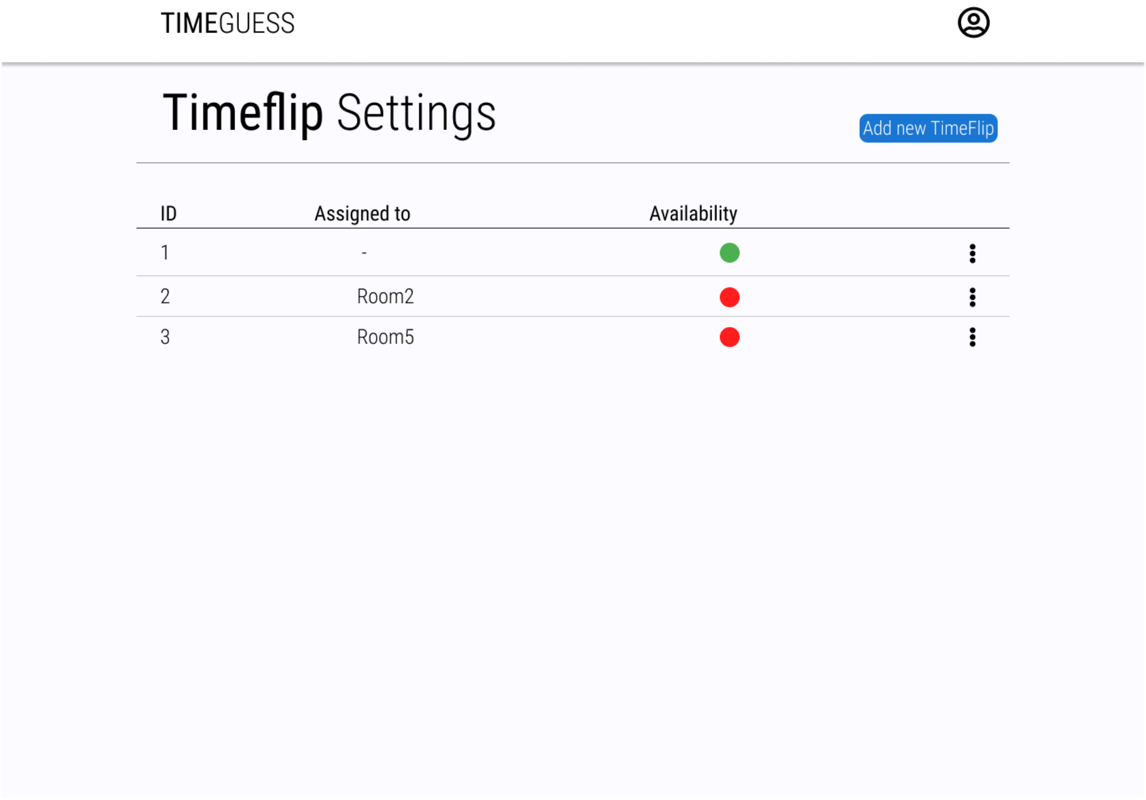


Logged in as: admin1

Game Settings

ID	Topic	# of terms	
1	Biology	45	✎ 🗑
2	Computer Science	32	✎ 🗑
3	Star Wars	38	✎ 🗑

5.13 TimeFlip Settings



Timeflip Settings

ID	Assigned to	Availability	
1	-	🟢	⋮
2	Room2	🔴	⋮
3	Room5	🔴	⋮

6 Project Plan

Milestones: https://git.uibk.ac.at/informatik/qe/sepsss21/group3/g3t1/-/milestones

Responsibilities:

- Web application: Front-end
  - Fabian Amhof
  - Felix Schuler
- Web application: Back-end
  - Martin Auer
  - Jonas Labermeier
  - Simon Brandacher
- Raspberry Pi & TimeFlip
  - Thomas Oberroither