

Konzeptbeschreibung

Team: Team 1

Mitglied 1: (Michael Hauser ,01267565)

Mitglied 2: (Angela Todhri, 11815296)

Mitglied 3: (Ismail Üner, 11721981)

Mitglied 4: (Flaminia Anselmi, 11934695)

Mitglied 5: (Sebastian Hepp, 01015083)

Mitglied 6: (Maximilian Heine, 01317323)

Proseminargruppe: Gruppe 6

Datum: 18.06.2021

Inhalt

Inhalt	2
1 Systemüberblick	3
2 Use Cases	4
2.1 Actors	4
2.2 Use-Case Diagram	5
2.3 Use-Cases	6
3 Klassendiagramm	22
4 Software-Architektur	24
4.1 Bausteinsicht	24
4.2 Laufzeitsicht	26
4.3 Verteilungssicht	29
5 GUI	30
6 Projektplan	35

1 Systemüberblick

The TimeGuess software is a IoT- and web-based trivia game for two or more teams and is played with a TimeFlip. This is a 12-sided smart dice, whose sides are associated with different activities, points and durations.

The game is played in a web application, where the users have to login. Everything the players need to do or know can be found in the app. Before a game a user can form teams inside the app, select a topic and once everything is set up, start the game.

Once a game has started the web-application guides the teams through the game. A player has to roll the TimeFlip and the player in turn gets a new term shown on the device of another team. The player then has to make the teammates guess the term using only the in the current task stated means – pantomime, rhyming, drawing or speaking without using the term. A timer counts down the allowed time for this task.

An opponent teams must state in the app, whether the term was guessed correctly, was not guessed in time or rules were broken. Depending on this input the guessing team receives a certain amount of points.

In the game app, users can see statistics of past games and their account. Privileged users – game managers and admins – can create and delete terms and topics, specify Raspberries associated with the game and setup and create new games. Furthermore, admins can create, edit and delete users.

2 Use Cases

2.1 Actors

User

The user has an account to use the web application. All users can do actions such as login, logout, see the lobby with statistics, see their profile and join games. Player, game manager and administrator are all users with different roles.

Player

The player is a user with basic functionalities. A player is a regular player in a game without any further privileges.

Game Manager

In addition to the options of a regular user, the game manager can create and delete terms and topics, specify Raspberries associated with the game and setup and create new games. Basically, he should be the person that owns the game and sets it up for the others.

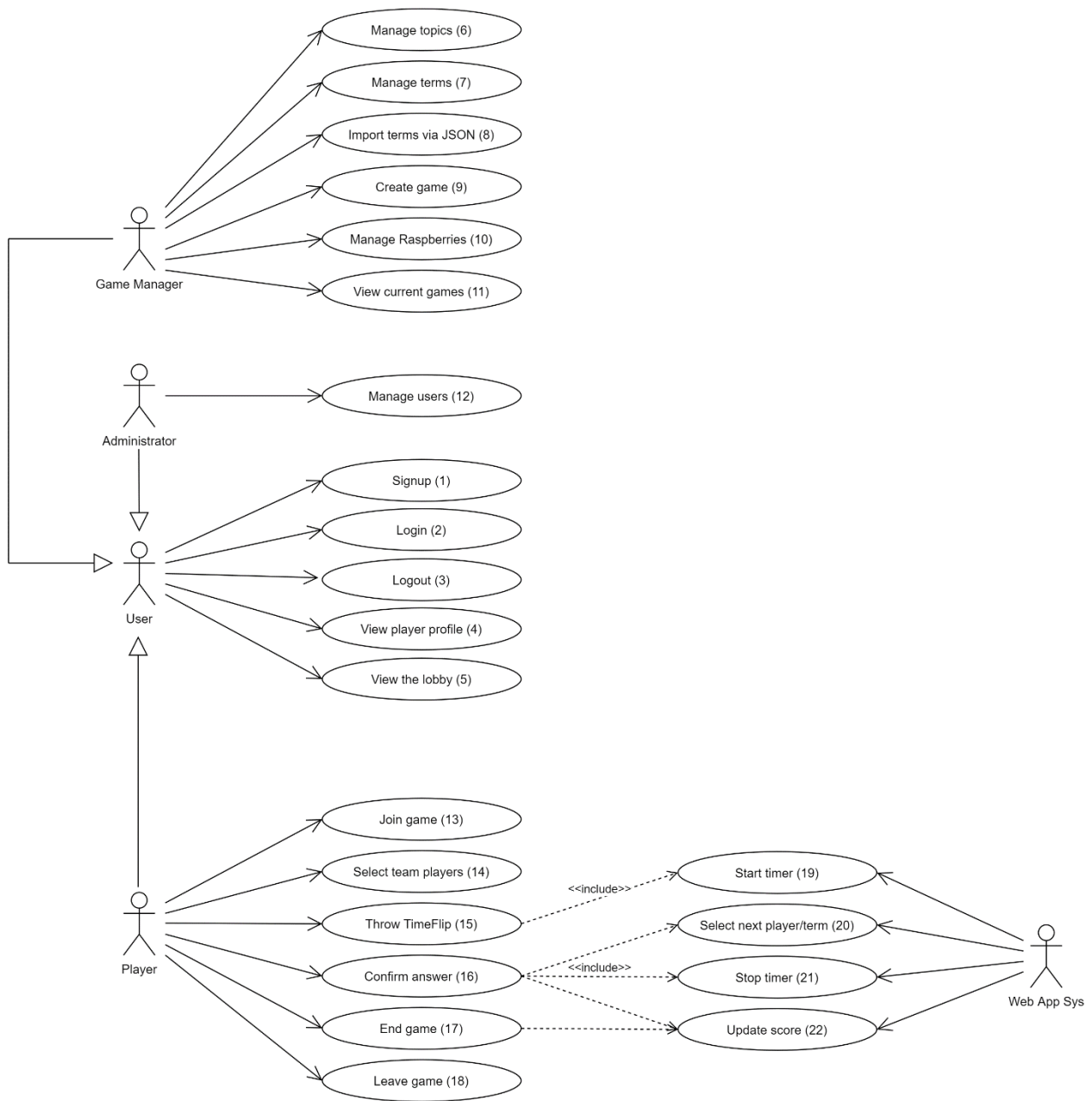
Admin

In addition to the options of a game manager, the administrator is responsible for the administration and management of players. The admin can create, delete and give rights to other users.

Web Application System

The system of the web application is responsible to randomly choose the next players and terms in the game and other tasks such as starting/ending the timer.

2.2 Use-Case Diagram



2.3 Use-Cases

2.3.1 Actor: User

Signup (1)

Precondition:

- The system is running.
- User is on the start page.

Procedure:

1. User clicks on "Signup".
2. User enters a new username, new password and confirms password.
3. Clicks "save".

Success:

- New user account is created.
- User sees the login window again and can log in now.

No success: User receives an error notification in the following cases

- User entered invalid username (less than 4 symbols or other characters than letters and numbers)
- User entered already taken username.
- User entered invalid password (less than 4 symbols).
- Confirmation password differs.

Log-in (2)

Precondition:

- The system is running.
- User is on the start page.
- User exists in the database.

Procedure:

1. User enters his/her username and password
2. Clicks on "Login".

Success: User can now view the virtual game lobby.

No success: User receives an error message and is not redirected.

Log-out (3)

Precondition:

- User is logged in.

Procedure: User clicks "Log out".

Success: User is logged out and redirected to the log-in page.

View player profile (4)

Precondition:

- User is logged in.

Procedure:

1. User clicks on "*My Profile*".

Success: The user's profile can be viewed, along with personal information:

- Username
- assigned Raspberry Pi
- Highest scored games
- Recent games
- Won games grouped by topic
- Recent teams and the teammates
- The other player with which the user is connected by having assigned same Raspberry Pi in the profile

View player profile: Change Raspberry (4.1)

Precondition:

- User is in the "*My Profile*" area.

Procedure:

1. User clicks on "*change Raspberry*" below "*Raspberry*".
2. All Raspberries are displayed in a new window. The user can select with which he wants to associate, to be able to connect to its games.
3. A new window appears, where the User has to enter the Raspberry's ip-address to confirm authorization to use it.
4. The User selects "confirm".
 - a. Alternative: User selects "Abort".
Process is aborted. Profile is shown again.

Success: The selected Raspberry is set in the user's profile.

No success:

- The User entered the wrong ip-address:
An error notification appears. The User can try again.

View player profile: Change Password (4.2)

Precondition:

- User is in the “*My Profile*” area.

Procedure:

1. User clicks on “*change password*” below the username.
2. A new window appears, where the user enters a new password twice.
3. Clicks “save”.

Success:

- New password is set.
- User sees profile again.

No success: User receives an error notification in the following cases

- User entered invalid password (less than 4 symbols).
- Confirmation password differs.

View Lobby (5)

Precondition:

- User is logged in.

Procedure:

1. User clicks on the menu item “Lobby”.

Success: User sees global statistics and the chat.

View Lobby: Search for other users (5.1)

Precondition:

- User is in the “lobby”-view

Procedure:

1. User enters an existing username in the search field.
A list of suggestions appears, as the user begins typing.
2. User clicks on search.

Success: User gets redirected to the other user’s profile.

No Success: When a non-existing name is entered, an error notification appears.

2.3.2 Actor: Game Manager

View Topics (6)

Precondition:

- User is logged in.
- User has the role "Game Manager or Admin".

Procedure:

1. User clicks on the menu item "Topic Hub".

Success: User sees a list of all topics.

Topics: Create topic (6.1)

Precondition:

- User is in the Topic view
- Topic is not yet in the list of topics.

Procedure:

1. User selects "*Create new Topic*".
2. User enters the name of the topic.
3. User selects "*Save*".
 - a. User selects "*Abort*": Action is aborted.

Success: The topic is added to the list of topics.

No success:

- If topic already exists, an error message is shown.
- If no name was entered, an error message is shown.

Topics: Edit topic (6.3)

Precondition:

- User is in the Topic view
- Topic exists.

Procedure:

1. User selects "*Edit*"-button beside the topic.
2. User enters a new topic name, which is not already used.

Success: The topic renamed. The user sees the list of topics again.

No success: If name is empty or already taken, an error notification appears.

Topics: Delete topic (6.3)

Precondition:

- User is in the Topic view
- Topic contains no terms.

Procedure:

1. User selects "*Delete*"-button beside the topic.

Success: The topic is removed from the list of topics.

No success: If topic contains terms, an error message is shown.

Topics: Show terms (6.4)

Precondition:

- User is in the Topic view

Procedure:

1. User selects "*Terms*"-button beside the topic.

Success: The user sees a list of all terms in this topic.

Topics – Terms: Create term (7.1)

Precondition:

- User is in the Terms view
- Term is not yet in the list of terms.

Procedure:

1. User selects "*Create new Term*".
2. User enters the name of the term.
3. User selects "*Save*".
 - a. User selects "*Abort*": Action is aborted.

Success: The term is added to the list of terms.

No success:

- If term already exists, an error message is shown.
- If no name was entered, an error message is shown.

Topics – Terms: Delete term (7.2)

Precondition:

- User is in the Terms view
- Topic contains no terms.

Procedure:

1. User selects *“delete”*-button besides the term to delete.

Success: The term is removed from the list of topics.

Topics: Import terms via JSON-file (8)

Precondition:

- User is in the Topic view
- Topic is not yet in the list of topics.

Procedure:

1. User selects *“Choose”*.
2. User selects the JSON-file from the device's filesystem.
3. User selects *“Submit”*.

Success:

The topic from the file is created if not existent. The terms are added to the topic. A notification is displayed.

No success: An error message is shown.

View Games (9)

Precondition:

- User is logged in.
- User has the role *“Game Manager or Admin”*.

Procedure:

1. User selects *“Games”* from the menu.

Success:

- As Game Manager: The user sees a list of all games running on his or hers Raspberry Pi.
- As Admin: The user sees all games.

Games: Create game (9.1)

Precondition:

- User is the Games view.

Procedure:

1. User selects *“Create new Game”*.
2. User enters the name of the game.

3. User enters the points needed to win the game.
4. User selects the topic, number of teams and number of players per team.
5. User selects *"Save"*.
 - a. User selects *"Abort"*: Action is aborted.

Success: The game is added to the list of games.

No success:

- If the name already exists, an error message is shown.
- If no name was entered, an error message is shown.
- If a score for win under 12 points is entered, an error message is shown.
- If a topic is selected, that contains less than 10 terms, an error message is shown.

Games: View Teams in a game (9.2)

Precondition:

- User is the Games view.
- An editable game exists.

Procedure:

1. User selects the *"Teams"*-button next to the game to edit.

Success: A list of all Teams in a game is displayed.

No success: If the game has already been started, the button is disabled.

Games – Teams: Add team to game (9.2.1)

Precondition:

- User is the Game – Teams view.

Procedure:

1. User selects *"Create new Team"*.
2. User enters the name of the team.
3. User selects *"Save"*.
 - a. User selects *"Abort"*: Action is aborted.

Success: The team is added to the list of teams of this game.

No success:

- If the team-name already exists in this game, an error message is shown.
- If no name was entered, an error message is shown.
- If the allowed number of teams has already been reached, an error message is shown.

Games – Teams: Delete team from game (9.2.2)

Precondition:

- User is the Games – Teams view.
- A Team exists.

Procedure:

1. User selects “Delete”-button next to the team to delete.

Success: The team is deleted from the list of teams of this game.

Games - Teams: View Players in a Team (9.2.3)

Precondition:

- User is the Games – Teams view.
- A Team exists.

Procedure:

1. User selects “Player”-button next to the team.

Success: A list of all Players in a Team is displayed.

Games – Teams – Players: Add player to team (9.2.3.1)

Precondition:

- User is the Games – Teams – Players view.

Procedure:

1. User selects “Add Player”.
2. User selects the player.
3. User selects “Save”.
 - a. User selects “Abort”: Action is aborted.

Success: The player is added to the list of players of this team.

No success:

If no free spots in the team are available anymore, an error message is shown.

Games – Teams – Players: Remove player from team (9.2.3.2)

Precondition:

- User is the Games – Teams – Players view.
- A Player exists.

Procedure:

1. User selects "*Delete*"-button next to the player to remove.

Success: The player is removed from the list of players of this team.

Games: Start game (9.3)

Precondition:

- User is the Games view.
- A startable game exists.

Procedure:

1. User selects the "*Start*"-button next to the game to start.

Success: The user is redirected to the select-team-players-view.

No success:

- If the game has already been started, the button is disabled.
- If all teams in the game are fully set with players, but the game creator is not among those players, an error message is shown.

View Raspberry Pi List (10)

Precondition:

- User is logged in.
- User has the role "Game Manager or Admin".

Procedure:

1. User selects "*Raspberries*".

Success: A list of all Raspberry Pi is displayed.

Raspberries: Edit Raspberry (10.1)

Precondition:

- User is in the Raspberries view.

Procedure:

1. User selects "*Raspberries*".
2. User selects the "*Edit*"-action next to the Raspberry to edit.
3. The user enters the IP-address of the device.
4. Clicks "*Confirm*".
5. User can change the device name, API-key, IP-address and enable/disable it.
6. User selects "*Save*".

- a. User selects "*Abort*": Action is aborted.

Success:

- The Raspberry is edited.
- The Raspberry-List is displayed again.

No success: If the IP-address is wrong, an error notification is shown.

Raspberries: Deactivate Raspberry (10.2)

Precondition:

- User is in the Raspberries view.

Procedure:

1. User selects "*Raspberries*".
2. User selects the "*Invalidate API-key*"-action next to the Raspberry to disable.
3. The user enters the IP-address of the device.
4. Clicks "*Confirm*".

Success:

- The API-key is removed from the Raspberry entry.
- The Raspberry-List is displayed again.

No success: If the IP-address is wrong, an error notification is shown.

Raspberries: Delete Raspberry (10.3)

Precondition:

- User is in the Raspberries view.

Procedure:

1. User selects "*Raspberries*".
2. User selects the "*Delete*"-action next to the Raspberry to delete.
3. The user enters the IP-address of the device.
4. Clicks "*Confirm*".

Success:

- The Raspberry is deleted from the list.
- The Raspberry-List is displayed again.

No success: If the IP-address is wrong, an error notification is shown.

View active games (11)

Precondition:

- User is logged in.
- User has the role "Game Manager or Admin".

Procedure:

1. User selects "*Manager Hub*".

Success: All games are listed.

ActiveGames: View teams of active games (11.1)

Precondition:

- User is in the Manager view
- A game exists.

Procedure:

1. User selects "*Teams*"-button next to the game.

Success: All teams in the game and their scores are listed.

2.3.3 Actor: Administrator

Create new user (12)

Precondition:

- User is logged in.
- User has the role "Admin".

Procedure:

1. User selects "*User Hub*".
2. User selects "*Create new User*".
3. User enters a username and password and selects the roles and the associated Raspberry.
4. User selects "*Save*".
 - a. User selects "*Abort*": Action is aborted.

Success: User is added to the list of users.

No success:

- If a field was not set, an error message is shown.
- If the username already exists, an error message is shown.

Users: Edit user (12.1)

Precondition:

- User is in the User view.
- A user exists.

Procedure:

1. User selects "*Edit*"-button next to the user to edit.
2. User can change the password, the roles, the associated Raspberry Pi and disable the selected user.
3. User selects "*Save*".
 - a. User selects "*Abort*": Action is aborted.

Success: User edited.

No success:

- If no role is set, an error message is shown.
- If the password has less than 4 characters, an error message is shown.

Users: Delete user (12.2)

Precondition:

- User is in the User view.
- A user exists.

Procedure:

1. User selects "*Delete*"-button next to the user to edit.

Success: User is deleted.

No success:

- If user is enabled, an error message is shown.
- If user is currently logged in, an error message is shown.

2.3.4 Actor: Player

Join game (13)

Precondition:

- User is logged in.
- A game has been started by the game creator.
- The game creator is associated to the same Raspberry as the user.
- User is either already assigned to a team or there is at least one free place.

Procedure:

1. User selects "*Join Game*".

Success: User is redirected to the select-team-player-view.

No success: If a precondition is not met, an error message is shown.

Select team players (14)

Precondition:

- User is logged in.
- User has joined the game.

Procedure:

1. User selects players until all open spots in the team are taken.
2. User sets the team name or, if already set, can change it.
3. User selects *“Join Game”*.

Success:

The user is displayed a message and waits until all other teams have done the same. The user is then redirected to the game room.

No success:

- If no team name is entered, an error message is shown.
- If not all seats in the team are taken, an error message is shown.

Throw TimeFlip (15)

Precondition:

- User is logged in.
- For each team a user has joined the game and entered the game room.
- User has been chosen by the system to throw the TimeFlip.

Procedure:

1. After throwing the TimeFlip, the face at the top of the die is the one determining how and in what time the explanation has to be done.
 - There is a letter on each side:
 - P = Pantomime
 - R = Rhyme
 - S = Speaking
 - Z = Drawing
 - A number near the letter:
 - 1 = 1 point
 - 2 = points
 - 3 = 3 points
 - And another number indicating the time:
 - 1 minute

- 2 minutes
- 3 minutes

2. The opponent's teams devices show the task for the user.

Success: The opponents teams devices show the task for the user; the timer begins.

Confirm answer (16)

Precondition:

- TimeFlip has been thrown
- Task has been displayed.
- Timer has started.
- *optional*: Timer has ended.

Procedure:

1. On opponent team selects
 - a. *"Guessed Correctly"* if the term has been guessed in time.
 - b. *"Not guessed"* if the term wasn't guessed in time.
 - c. *"Rulebreak"* if the explaining user has broken a rule.

Success: The guessing team receives:

- a. The points associated with the current side of the TimeFlip if term was guessed correctly.
- b. 0 points, if term was not guessed.
- c. -1 point if there was a break of rules.

End game (17)

Precondition:

- A team has earned at least as many points as stated for "score to win" at game creation

Success: All teams are redirected to the end view and game statistics are shown.

Leave game (18)

Precondition:

- User is in the game's end view.

Procedure:

1. User selects *"Back to Lobby"*.

Success: User is redirected to the game lobby.

2.3.5 Actor: Web Application System

Start timer (19)

Precondition:

- New round has started.
- TimeFlip die has been thrown.

Procedure: System begins the countdown, which varies depending on what side of the die is at the top.

Success: Timer starts counting and players start guessing.

Select next player/term (20)

Precondition:

- Game has started.
- Round has ended.
- Score has been updated.
- At least one player per team in virtual game room.

Procedure: System determines at random the start team and the term.

Success: The term is shown on the device of the opposing team, timer is displayed.

No success: The same term is suggested more than once in a same game.

Stop timer (21)

Precondition:

- Round has ended.
- Player has stopped the die by flipping the top side.

Procedure:

System keeps track of the time that is passing:

- Stops the countdown immediately if a player flips the time.
- Checks if the countdown has arrived to 0.

Success: Timer has stopped and now players will receive a notification to confirm their answers.

Update score (22)

Precondition:

- Round has ended.
- Players confirmed the answer.

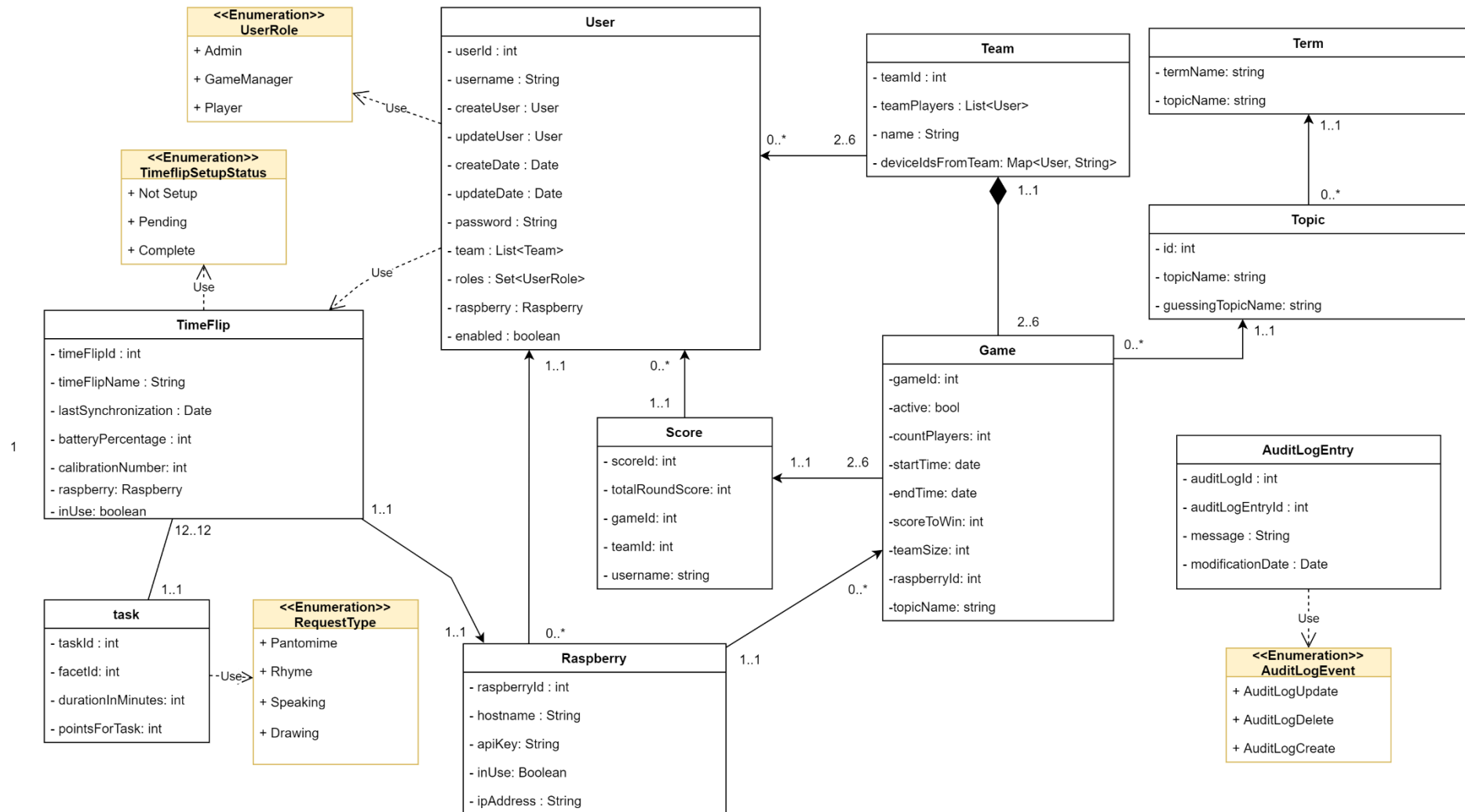
Procedure:

1. New score for the guessing team is calculated, following these options:
 - If the timer has expired completely, 0 points are awarded.
 - If the term was not guessed, 0 points are awarded.
 - If the times has not expired and the term has been guesses, the number of points written at the top side of the TimeFlip are awarded.
 - If a rule was violated, 1 point is subtracted.
2. After the score has been updated:
 - A new round is started and is time for the opposing team to play.

Success: Players start a new round.

Alternative: If the maximum of points has been reached

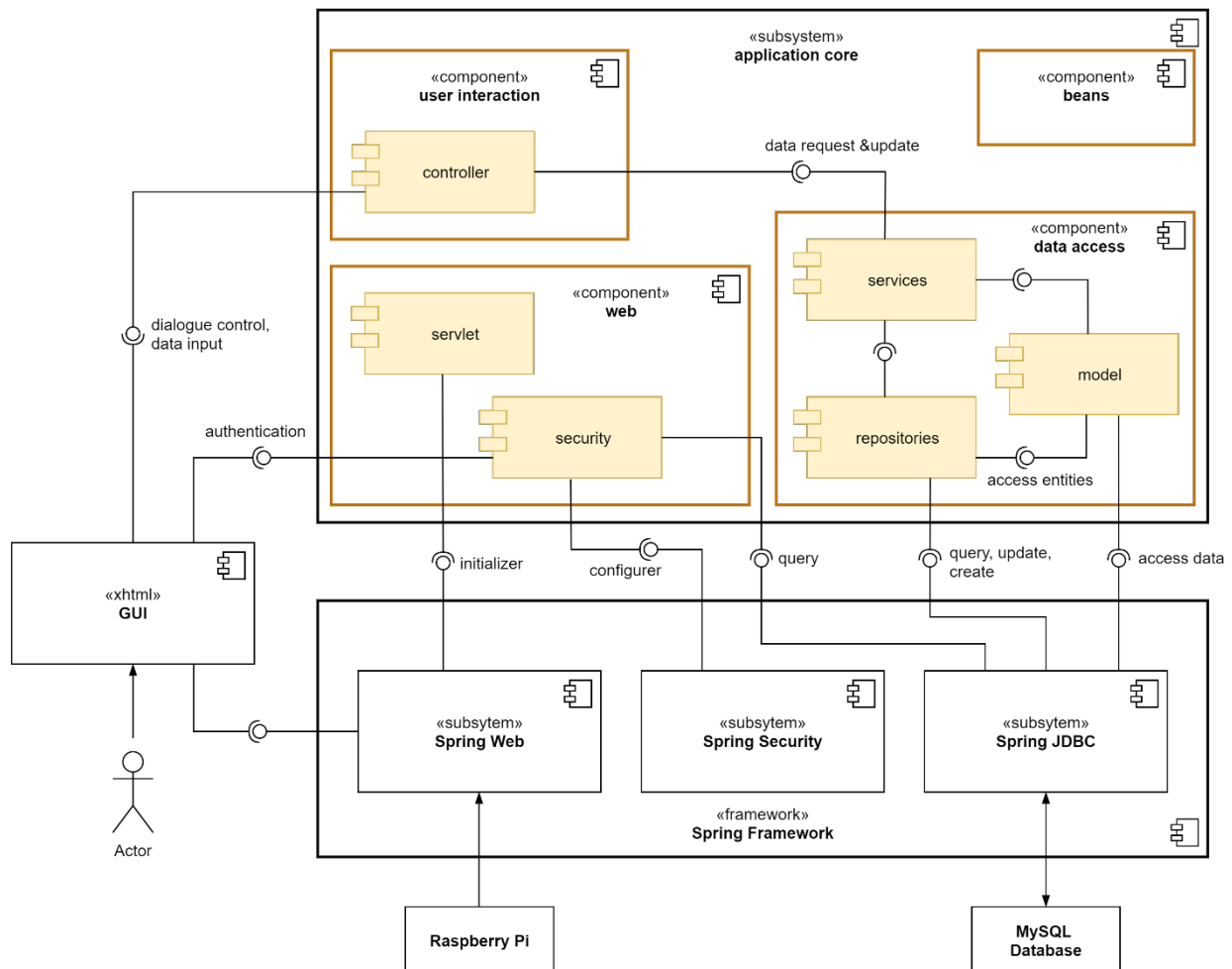
3 Klassendiagramm



class	role
Game	models a single game, with all necessary information for the gameplay; references the teams; is associated with a raspberry
Raspberry	models a single Raspberry with information necessary for the API; references a list of users, that use this device
Score	Score is instanced for each game and user and contains information for the highscore statistics; references the team the user played in
Team	models a single team in a specific game; references the users assigned to the team and their scores
Term	a single term that will be chosen from the app for players to guess
Topic	a topic will assigned to a game and references a set of terms
TimeFlipConf	contains setup-information for the TimeFlip, that specify the sides of the die
User	contains the information of a single user; references all the scores of the user, the user roles, the Raspberry the user plays on and all Teams the player was assigned to

4 Software-Architektur

4.1 Bausteinsicht



The system can be divided into the spring framework it is built upon, the graphical user interface, which is realised via XHTML, and the core of the application, which holds the actual implementation of the business logic. The application core is further broken down in the following table.

component	role
data access	retrieves, creates and updates the persisted data from the MySQL database via spring's JDBC interface
► model	models the persisted data as Java objects (entities)
► repositories	used to query, create, delete and update data in the database
► services	supplies methods to other components for all operations that need to access or modify persisted data, using the repositories and model modules
web	initiates and configures the web application

► **servlet** initiates the web application

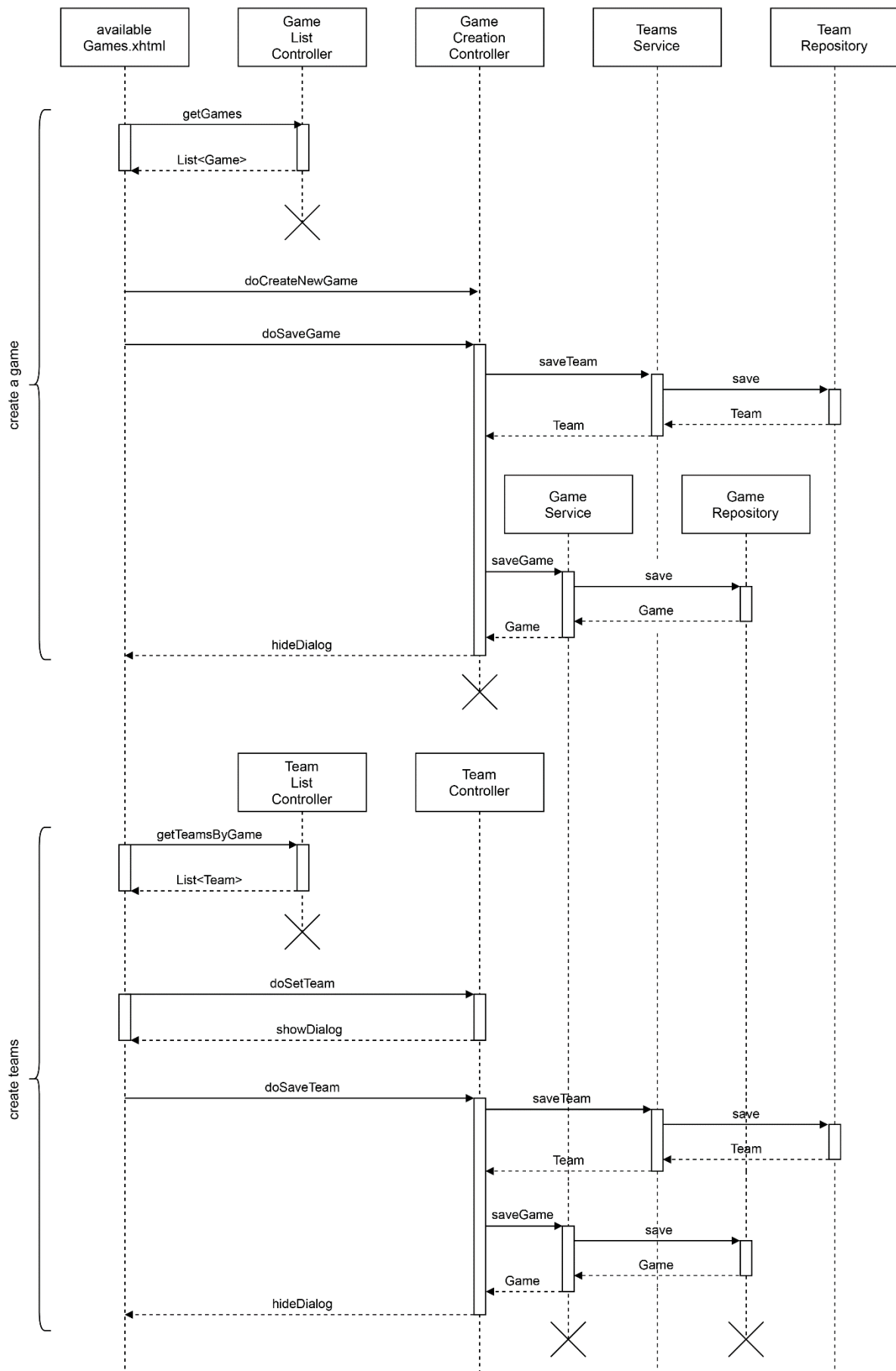
► **security** manages authentication and error handling when somebody wants to access the web application
needs to query user data from the database and receives user input from the user interface

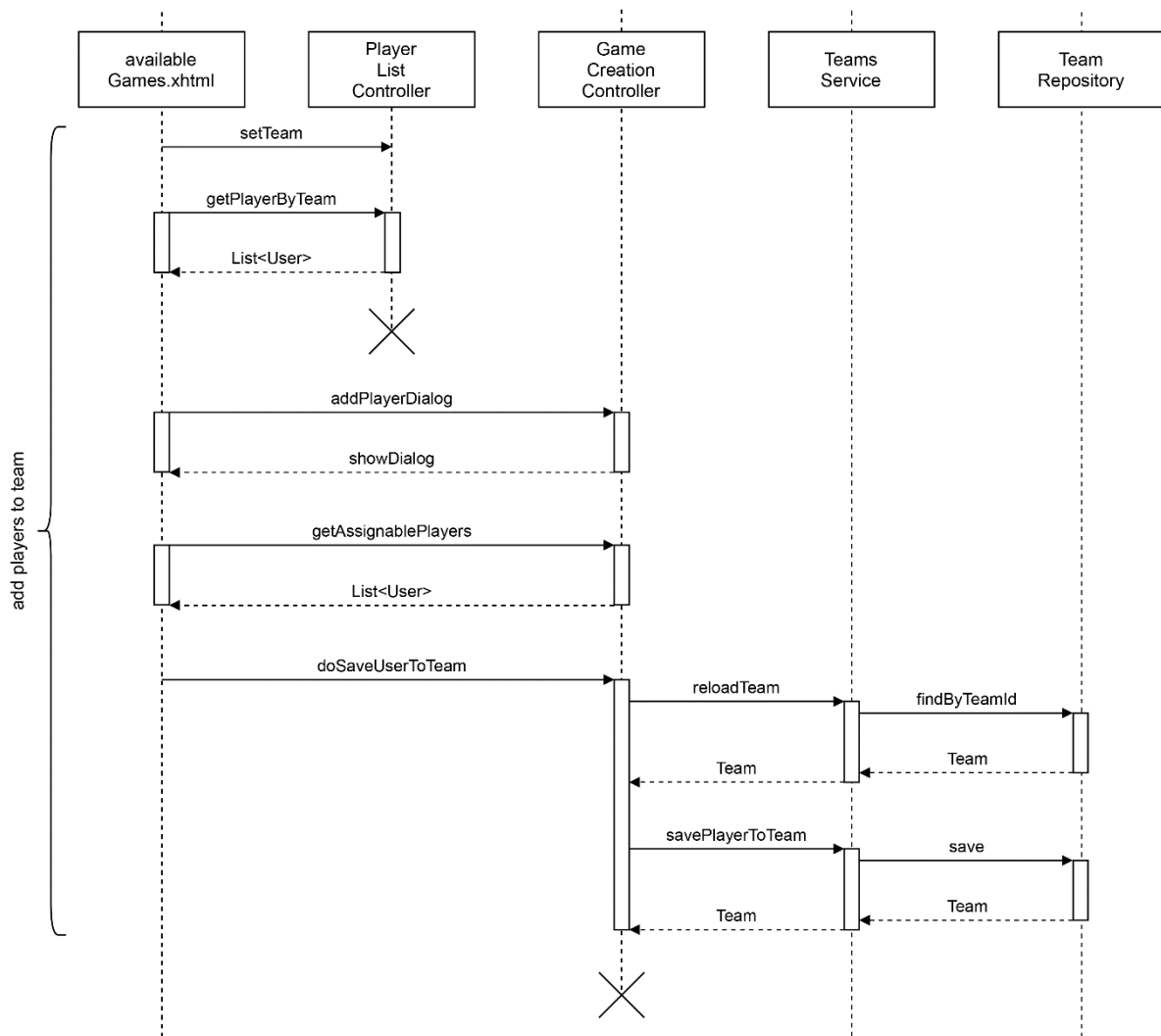
user interaction interface between GUI and the data model

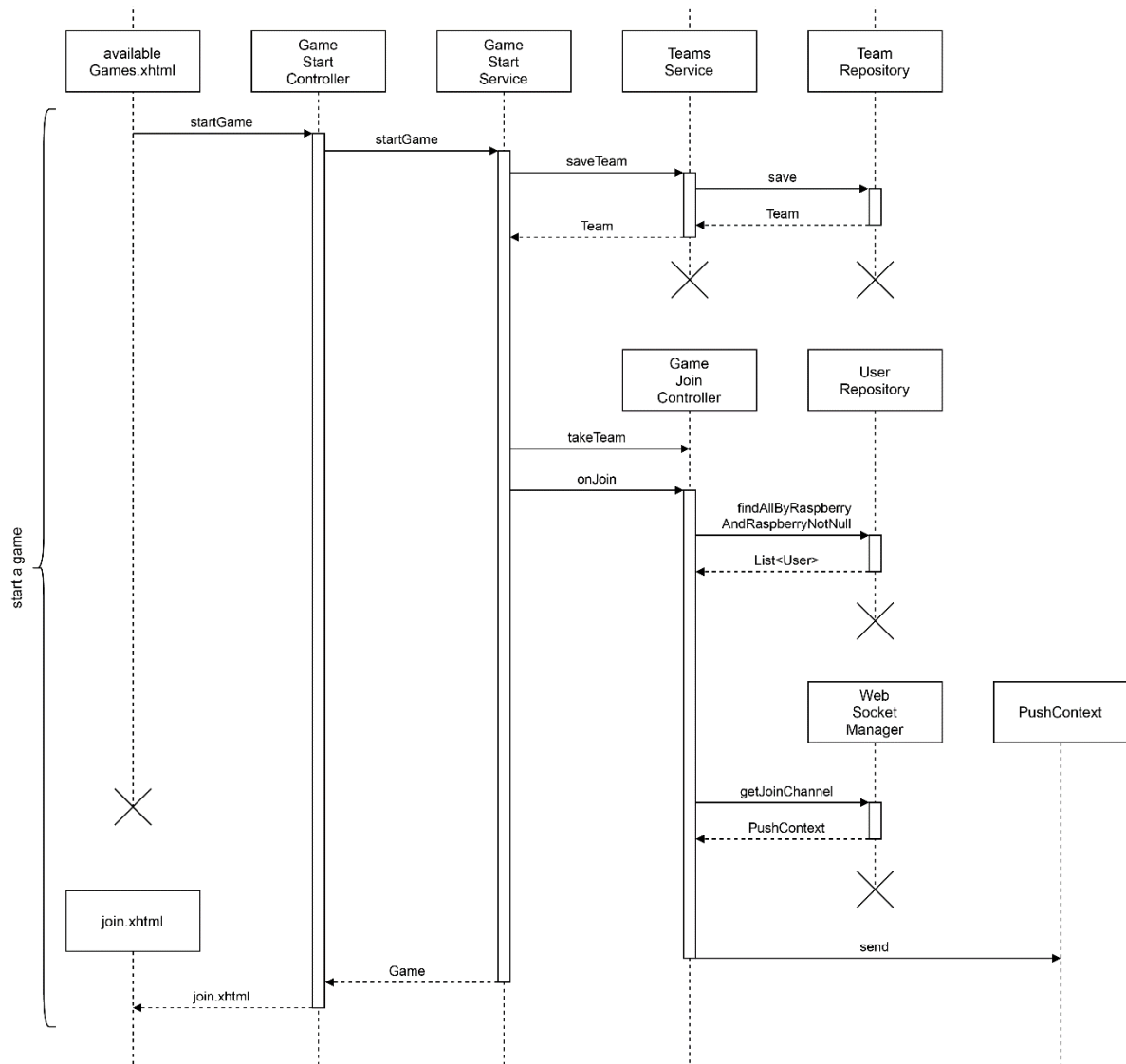
► **controller** responsible for dialogue control of the GUI, implement the business logic initiated by the user input, retrieve the data to display from the database and modify the persisted data by communicating with the services-module

beans contains independently usable components that implement additional features

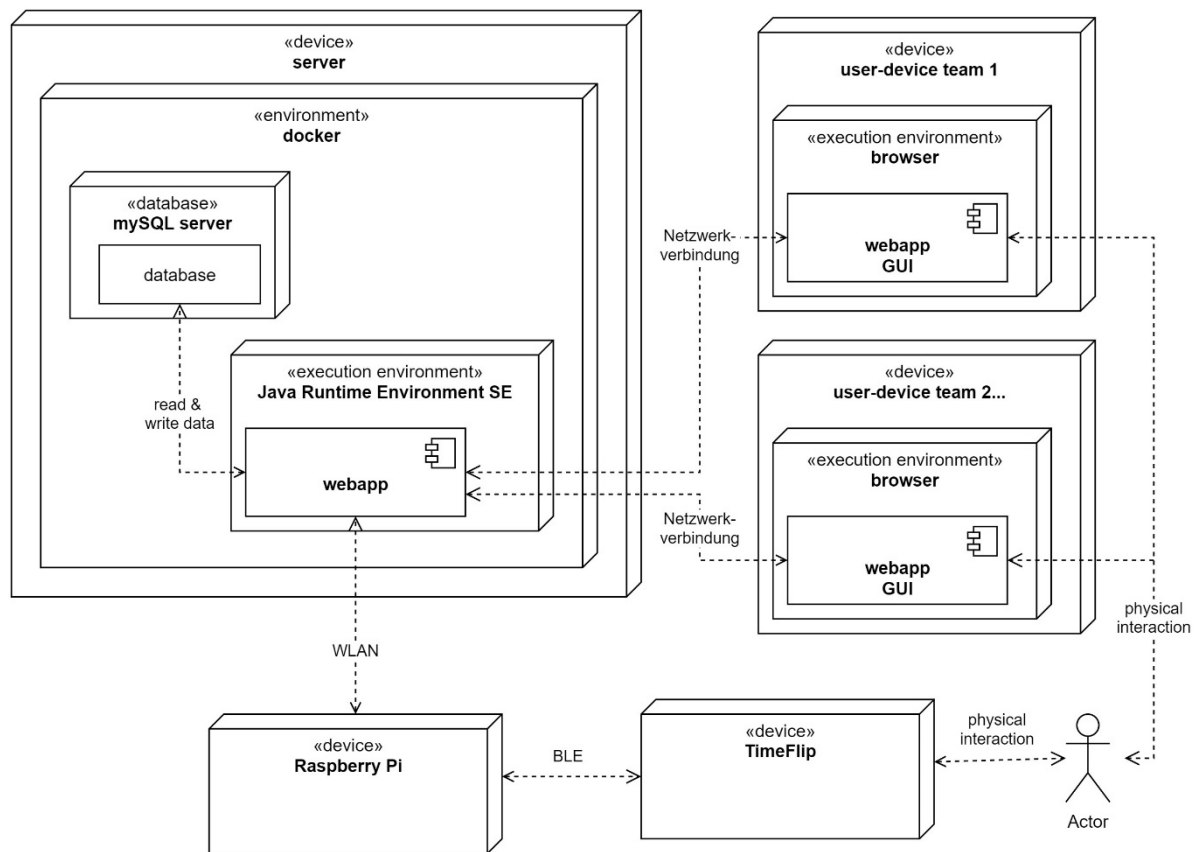
4.2 Laufzeitsicht



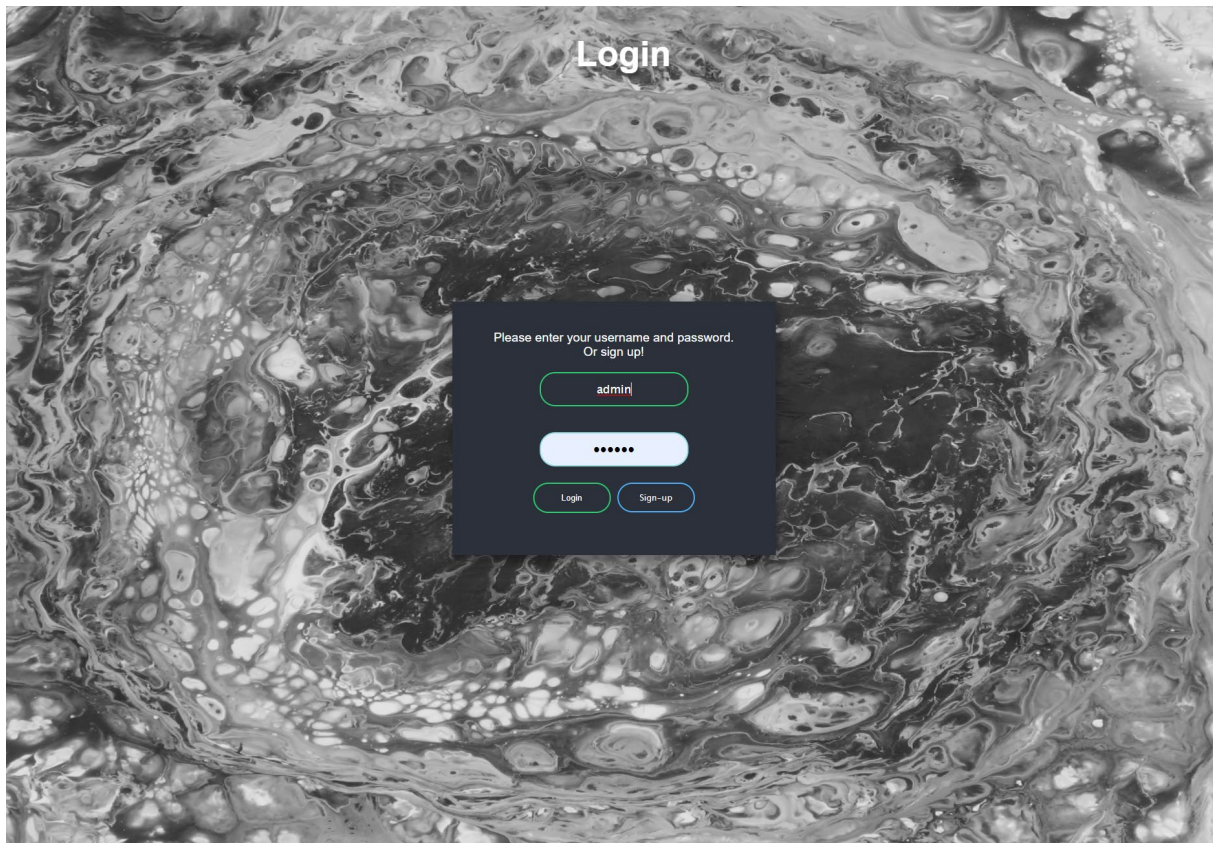




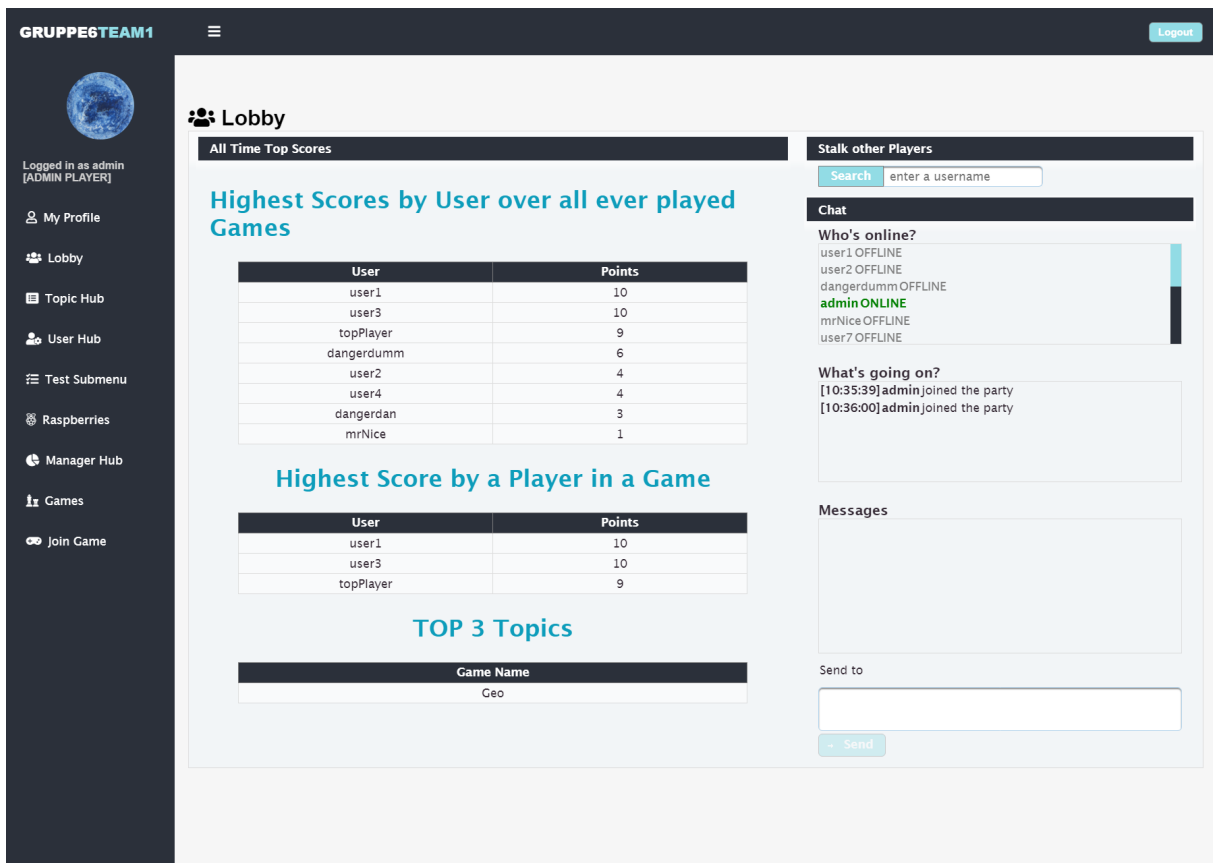
4.3 Verteilungssicht



5 GUI



start page: Lobby with stats, chat and user search



personal profile with stats and options to change password and associated Raspberry Pi

GRUPPE6TEAM1 Logout

Help

Logged in as admin [ADMIN PLAYER]

admin

[change password](#)

Raspberry Pi: TopGamerz

[change Raspberry](#)

Statistics

Played Games: 0
Won Games: 0
Lost Games: 0

My Best Games

Game Name	Team Name	Score
No records found.		

My Latest Games

Game Name	Team Name	Score	Date/Time
No records found.			

Won Games By Topic

Topic Name	Wins
No records found.	

My Teams

Team Name	player
No records found.	

My Circle of Players

Player Name
user1
user2
elvis
topPlayer
mrNice

Topic Hub: Create and Edit Topics and Terms, or import them from a JSON-file (Admins and Game Manager)

GRUPPE6TEAM1 Logout

Help

[Choose](#) [Submit](#)

Topic Hub

Topic	Terms	Terms	edit	delete
Geo	37			
Music	12			
Biology	22			
Food & Drink	16			
Fashion	30			
History	19			
Literature	25			
Movies & TV	16			
Art	23			
Law	19			
Celebrities & Influencers	16			
Holidays & Celebrations	11			
Comics	11			
Video Games	24			
Board Games	27			
Animals	11			
Architecture	14			
Technology & Computer	22			
Sports	32			
Plants	10			

[+ Create new Topic](#)

User Hub: Create and edit user accounts (only Admin).

GRUPPE6TEAM1

Logged in as admin [ADMIN PLAYER]

My Profile

Lobby

Topic Hub

User Hub

Test Submenu

Raspberries

Manager Hub

Games

Join Game

Help

All Admin Manager Player

Username	Roles	Enabled	Operation
admin	[ADMIN, PLAYER]	<input checked="" type="checkbox"/>	Edit Delete
signupBot	[ADMIN]	<input checked="" type="checkbox"/>	Edit Delete
user1	[GAME_MANAGER, PLAYER]	<input checked="" type="checkbox"/>	Edit Delete
user2	[PLAYER]	<input checked="" type="checkbox"/>	Edit Delete
elvis	[ADMIN, PLAYER]	<input checked="" type="checkbox"/>	Edit Delete
user3	[PLAYER]	<input checked="" type="checkbox"/>	Edit Delete
user4	[PLAYER]	<input checked="" type="checkbox"/>	Edit Delete
user5	[PLAYER]	<input checked="" type="checkbox"/>	Edit Delete
user6	[PLAYER]	<input checked="" type="checkbox"/>	Edit Delete
user7	[PLAYER]	<input checked="" type="checkbox"/>	Edit Delete

Create new User

Raspberries: Create and edit Raspberry Pis (Admin and Game Manager).

GRUPPE6TEAM1

Logged in as admin [ADMIN PLAYER]

My Profile

Lobby

Topic Hub

User Hub

Test Submenu

Raspberries

Manager Hub

Help

ID	Name	IP Address	API Key	inUse	actions
1	raspi	X		<input checked="" type="checkbox"/>	Edit Delete
2	TopGamerz	192.168.0.10	7173b055-4674-4ca2-8348-60e1b3fa8204	<input checked="" type="checkbox"/>	Edit Delete
3		192.168.0.11		<input checked="" type="checkbox"/>	Edit Delete

Add new Raspberry

Manager Hub: View active Games and their teams and scores (Admin and Game Manager).

GRUPPE6TEAM1

Logged in as admin [ADMIN PLAYER]

My Profile

Lobby

Topic Hub

User Hub

Test Submenu

Raspberries

Manager Hub

Games

Join Game

Help

Players in active Games

Game Name	Topic	Teams	current Players/max Players	score to win	Teams
Michael	Geo	2	4 / 4	10	Edit
Peter	Geo	0	4 / 4	50	Edit
Jackson	Geo	0	6 / 6	50	Edit
Test	Geo	0	2 / 2	50	Edit

Currently Active Games

Check out info to all currently active Games here

Games: Create, start or delete Games (Admin all Games, Game Manager only personal Games).

Games

See all your Games here:
Edit or start Games that haven't already begun or create a new Game

Name	Topic	Teams	max Players	Score to win	Status	Operation
Zlatko	Geo	2	4	45	not started	
Michael	Geo	2	4	10	started	
Max	Geo	0	4	20	not started	
Angela	Geo	0	4	25	not started	
Peter	Geo	0	4	50	started	
Jackson	Geo	0	6	50	started	
Test	Geo	0	2	50	started	
Hans	Geo	0	4	10	finished	

[+ Create new Game](#)

Join Game / Start Game: Select players for your team and enter the game.

Choose your Team Players

Team Name: [reset Teams](#)

2 of 2 players assigned.

Player Name	available
admin	assigned
user1	assigned
user2	yes
elvis	yes
topPlayer	yes
mrNice	yes
dangerdan	yes
dangerdumm	yes

[Join Game](#)

Inside the Game

newGame

Team1

Term:

Player:

Point(s)

Next Up: user1

Roll the dice to start the game!

Guessed Correctly

Rulebreak

Not Guessed Correctly

Scores	
Team	Score
Team2	0
Team1	0

0

6 Projektplan

Nr.	Work Package	Responsible	Time	Deadline
1	Konzeptbeschreibung	All	50:00	18.03.
	Milestone 0: Planning			18.03.
2	Model	Flaminia / Angela	35:00	25.03.
3	Landing Page	Ismail	10:00	25.03.
4	Raspberry Pi Setup	Michael / Max	15:00	28.03.
5	TimeFlip Setup	Michael / Sebastian	15:00	03.04.
6	REST API	Max / Sebastian	30:00	18.04.
7	Database data	Angela	20:00	01.04.
	Milestone 1: Database, Model and Hardware			01.04.
8	MySQL	Flaminia	20:00	01.05.
9	Advanced Backend Functionality Controllers, Services, Repository	All	80:00	13.04.
10	Advanced Frontend	Sebastian / Max	50:00	20.04
11	Data Management -create game statistics -create terms	Ismail / Michael	20:00	27.04
	Milestone 2: Core Functionalities			01.05.
12	JSON integration	Flaminia	20:00	02.05.
13	Bugfixing I	All	50:00	09.05.
14	Testdrehbuch	Flaminia / Angela	40:00	13.05.
15	Hardware integration	Michael / Max		12.05.
16	Dockerization	Michael	25:00	06.06.
17	Stable and working system	All		13.05.
	Milestone 3: Releasable System			13.05.
18	Acceptance Tests	All		20.05.
19	JUnit Tests	Flaminia / Angela / Ismail	50:00	16.06.
20	Systemtest	Michael / Max	30:00	06.06.
21	Bugfixing II	All	30:00	13.06.
22	Code doku	All	10:00	13.06.
23	Softwarekonzept update	Sebastian	20:00	13.06.
24	Abschlussbericht	Sebastian	10:00	13.06.
	Milestone 4: All Software Revisions & Document Drafts			14.06.
25	Documents: Revision & Layout	Sebastian		16.06.
26	Final project results	All	10:00	18.06.
27	Final presentation	All	08:00	21.06.