

DECISIONS

David Chaparro
Ismael Maridueña
Zhi Yang Wu

DAM-2

Index

Purposes	01
Description	02
Members of the group	03
Required materials and technical specifications of the project	03
Architecture diagrams	04
Initial Wireframes	05
Functional and non-functional requirements	06
Functionality of the app	09
Class diagram	14
Database E-R model	15
Use Case diagram and Textual description	17
Test cases	33
Description of the project methodology used	48
Planning	49
Individualized planning	51
Innovation	53
User manual	55
Installation Manual	60
Conclusions	71

Purposes

The main purposes of our application would be:

- **Connect with friends:** Allows users to add and connect with their friends in the application, creating a social network where they can play and take decisions together.
- **Play mini-games to make decisions:** Offers a variety of quick and fun mini-games that help users make decisions in an entertaining and lighthearted way, whether with friends or solo.
- **View their friends decisions:** Provides a feature that allows users to see the decisions their friends have made in the mini-games, giving them an interesting perspective and helping them understand their friends' preferences.
- **Decisions without discussions:** Facilitates decision-making without discussions or conflicts among friends, ensuring a harmonious and enjoyable experience when making decisions together.

Description

Decisions is a mobile application designed to assist individuals in making choices by engaging in quick mini-games, either with a friend or by oneself. The app provides an interactive platform where users can compete against each other or against the computer, with the winner being able to share their decision with a photo for their friends to see.

The app offers a user-friendly interface that allows individuals to connect with their friends by sending and accepting friend requests. Additionally, users have the flexibility to manage their friendships by removing friends as desired.

To initiate the decision-making process, users must first select the desired mode: competing against a friend or challenging the computer. If playing against a friend, both individuals input their respective decisions, and an invitation to play is exchanged. Once the invitation is accepted, a random mini-game is generated from options such as "Elements", "Penalties", or "Question Quiz".

Upon completion of the mini-game, the victorious player is presented with a congratulatory message indicating their win. They are then given the option to capture and share a photo related to their decision, which can be viewed by their friends. Conversely, the losing player can see a screen of their defeat and is provided with the opportunity to return to the main menu for further engagement.

Members of the group

We are David Chaparro, Ismael Maridueña and Yang, students from the Institut Provençana and we are studying DAM-2. Decisions was the idea taken to do our final project presented by David who came up with an idea to make an app thinking of a social network In which you do not have to invest a lot of time, is fun and that concerns several people.

Required materials and technical specifications of the project

Fisical requirements:

Server:

- Minimum Requirements:
 - RAM: 6GB
 - Storage: 50GB (In the final progress, we will need approximately 20TB for around 1000 users)
 - Operating System: Ubuntu 22.04
 - Installation Guide: <http://somebooks.es/installar-ubuntu-22-04-lts-jammy-jellyfish-desde-cero/>

Client:

- Phone with:
 - Android 10
 - 16GB space
 - 4RAM

Worker:

- PC with:
 - Minimum Requirements:
 - windows 10
 - Android studio
 - 8RAM
 - 500GB space (recomended)

SoftWare requirements:

- Server:
 - Web Service:
 - Versions:**
 - MySql:** 8.0

Architecture diagrams

Our architecture diagram is based on 5 large components, user, mobile app, Wi-Fi, server and database:

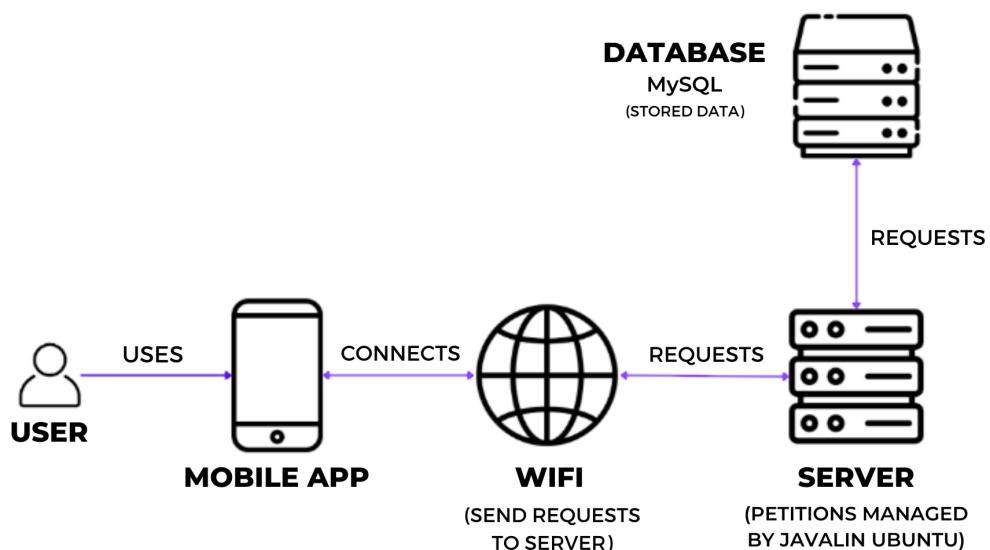
User: The person who uses the application is the most important part of the architecture. He or she interacts with the application through the mobile device.

Mobile App: Decisions is a software application designed to run on mobile devices such as smartphones or tablets. It serves as the primary interface between the user and your system. Users can use the mobile app to interact with different features and functionalities provided by your system.

Wi-Fi: Wi-Fi connection is required for the mobile device to connect to the Internet. Without a Wi-Fi connection, the mobile device will not be able to connect to the app.

Server: The server is the component of the application that receives the requests from the mobile device and processes them. The server runs on a remote computer that is connected to the Internet.

Database: The database is the place where the application data is stored. The data can be user information, application settings, or any other type of information that the application needs to store.



Initial Wireframes

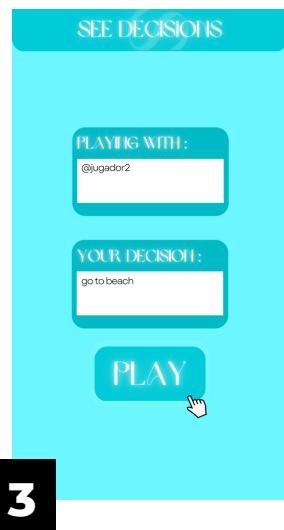
These are the initial wireframes we made for Decisions:



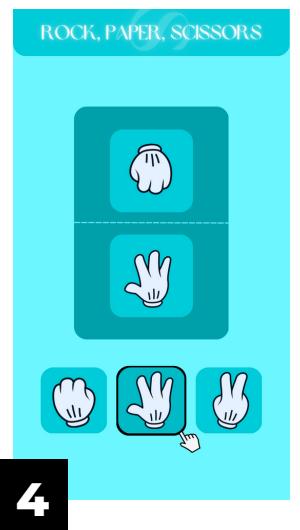
1



2



3



4



5



6



7



8

1. The log in of the app
2. The social interface with posts
3. The screen to connect with friend and write your decision
4. The game that we change by elements
5. The result win with take a photo button
6. The custom camera screen
7. The result of the photo
8. And the social interface with the uploaded photo

Functional and non-functional requirements

These are the functional requirements of Decisions:

RF 1: Sign up to the app

- Allow users to create an account with their email and set a secure password and username.

RF 2: Log in to app

- Allow users to log in to the app using their registered email and password.
- Provide a login interface where users can enter their credentials to access their account.

RF 3: Recover password

- Allow users to initiate the password recovery process by providing their registered email.

RF 4: View interface

- Allow users to access and navigate the app's user interface, providing a visually appealing and intuitive design.

RF 5: Add friends

- Allow users to search for other app users and send friend requests to add them.

RF 6: Management of friend requests

- Allow users to view and manage incoming friend requests, including accepting or rejecting them.

RF 7: Remove friends

- Allow users to remove friends from their friend list or contacts.

RF 8: Connect with friend

- Allow users to establish a connection with a friend once the friend request is accepted.
- Provide features such as messaging, sharing updates, or viewing each other's profiles to enhance the connection.

RF 9: Play random minigames with a friend

- Allow users to initiate random mini-games with a friend from their friend list.
- Provide a selection of mini-games that can be played together in real-time, ensuring a fun and interactive experience.

RF 10: Play random minigames versus robot

- Allow users to engage in random mini-games where they compete against AI-controlled robots.
- Provide challenging AI opponents to play against, offering a variety of difficulty levels to cater to different skill levels.

RF 11: Minigame "Elements"

- Allow users to access and play the "Elements" mini-game, which involves strategic decision-making based on different elements.

RF 12: Minigame "Penalties"

- Allow users to participate in the "Penalties" mini-game, which involves taking penalty shots in a soccer-themed setting.

RF 13: Minigame "Question Quiz"

- Allow users to engage in the "Question Quiz" mini-game, which involves answering trivia questions within a limited time.

RF 14: Take a photo to share

- Allow users to capture photos using custom camera directly from the app.

RF 15: Share experience on social interface

- Allow users to share their decision, and publish a photo on the app's social interface.

RF 16: Change profile data

- Allow users to modify and update their profile information, such as username, profile picture, bio, and other relevant details.
- Provide an intuitive interface where users can easily edit their profile data and save the changes.

RF 17: Log out

- Allow users to log out of their account, ensuring the security of their personal information.

RF 18: Change language

- Allow users to change the language of the app's interface to suit their preferences.

RF 19: Games guide

- Allow users to access a comprehensive guide or tutorial section that provides instructions and tips for each available game.

These are the no-functional requirements of Decisions:

NRF 1: Usability

The application must have a user-friendly and intuitive interface that allows users to easily navigate through the different functionalities. The interface should be designed with clear labels and buttons, consistent visual design, and should be responsive on different screen sizes and devices.

NRF 2: Performance

The application should have fast response times for all functionalities, including game loading times and social interface loading times. The application must also be able to handle a large number of users and game requests simultaneously without experiencing any performance issues.

NRF 3: Security

The application must be secure and protect user data. This includes secure authentication and authorization mechanisms, secure storage of user data, and protection against common web application vulnerabilities such as SQL injection and cross-site scripting (XSS).

NRF 4: Reliability

The application must be reliable and available at all times. It should have mechanisms in place to handle unexpected errors and downtime, such as backup systems and error handling protocols.

NRF 5: Compatibility

The application must be compatible with a wide range of devices and operating systems. It should be tested and optimized for different browsers and devices to ensure a consistent user experience across different platforms.

NRF 6: Scalability

The application must be scalable to handle an increasing number of users and game requests. This includes the ability to add more servers and resources as the user base grows, and the ability to handle a large number of concurrent requests without experiencing any performance issues.

Functionality of the app

Log in, sign in and recover password

Starting with the log in where users can enter the application with their username and password and you can stay signed in if you want so you don't have to log in, only in the case where if they have previously registered, in the case that they do not have to go to the sign in screen and enter their data, email, username, password and confirm password and will be registered if all their data is correct, in the case in which the user does not remember their password we have implemented "Recover your password" can be accessed from the login, in recover password the The user will first have to enter their email and they will receive a one-time code to change their password.

Main interface

Once the log in is done, the user finds the main screen where we can see the user publications sliding one by one seeing the decisions of friends, from the footer of that screen, the user can access settings, choose modality, and friends , in the case in which you do not have friends or they have not published anything, you will not be able to see publications on the main screen.

Friends management (Add friends, remove friends, friend requests)

If the user enters the friends icon, they have three options, add friends, remove friends, friends requests, from add friends you can search for anyone registered in the app in a listview, it will show 15 users and you will have a search engine where as you type The users that match what you are typing will appear. Once you click on a user, you will get a confirmation notice to send a friend request and that user will receive it. If the user enters remove friends, he will be able to see the users that are his friends and if he clicks on one, he will get a confirmation message to delete, in the case in which I have deleted, they can become friends again by sending another request. If the user enters Friends requests, they can accept or reject the requests, once they accept they will be friends.

Setting (Edit profile, log out, change language and game guide)

If the user clicks on the gear icon in the footer, they enter settings, where the user can edit their profile data (username, email and password) each with its respective section and rules. Also, from settings, the user can click on a button to log out in case he wants to log in with another user, clicking on it will take him directly to log in. The last two settings options are to change the language, there are 7 available, Spanish, Catalan, English, German, Portuguese, Hindi, and simplified Chinese, the language will be maintained when the app is closed; and the last option is the game guide where you can read a guide to the three implemented games, the guides are expandable, that is, they open and close when clicked.

Choose modality, connect with friend and write decision

If the user is on the main screen of the app and clicks on the "Decisions" icon, it will take them to the screen to choose modality, they can choose between 1 versus 1 or 1 versus robot, if they select 1 versus 1 it will take them to the screen to connect with a friend, on this screen you will see a list of your friends to be able to send a request to play and another list where you will receive the requests that your friends send you to play, when you click on a friend a confirmation message will appear first When you accept it, a waiting alert dialog will appear and until the other person accepts or the user himself goes back, the request will not be cancelled. In order to send a request on the same screen, you will have to write a text with your decision, in the case to play 1 versus robot, the user will have to write two decisions, his own and the robot's, since when he plays alone even if he has lost, he can upload the decision that he has assigned to the robot.

Games("Elements", "Penalties" and "Question quiz")

In our application we have three games that are "Elements", "Penalties" and "Question quiz", these three games can be played both online and alone versus robot.

"Elements" is a game inspired by the classic rock-paper-scissors concept, but with a twist. In this game, players utilize three fundamental elements: water, ice, and fire. Each element possesses its unique strengths and weaknesses, forming an intricate dynamic within the game. Water extinguishes fire, fire melts ice, and ice freezes water. Players strategically choose one of the three elements to clash with their opponent's choice, aiming to exploit the elemental interactions and emerge victorious. In the case of a tie, the game will be repeated until there is a winner and a loser. Elements combines strategy, anticipation, and quick decision-making, making it an engaging and captivating game for players of all ages. Each player will have 15 seconds to select an option in the case in which you play alone if you do not select anything when time runs out the move is made automatically and if you play online it will kick out of the game users who are 3 rounds without selecting due to inactivity.

The objective of "**Penalties**" is to score goals as the shooter and save goals as the goalkeeper. The roles switch after each round, providing an equal opportunity for both players to demonstrate their skills. The time limit adds a sense of urgency and quick decision-making to keep the gameplay exciting. To play, you'll need to select your preferred shooting or saving direction within the given 8-second time frame. The options include aiming for the top right, bottom right, center, bottom left, or top left of the goal. Make your choice wisely to outsmart your opponent and score or save as many goals as possible. If both players manage to score 5 goals each within the designated rounds, the game enters sudden death. In this phase, the first player to score a goal will be declared the winner. So, it's crucial to maintain focus and accuracy even under pressure. When playing the game, if the round time expires before a choice is made, the action will be automatically executed. In the case of playing online, if a user or both users fail to make a selection for three consecutive rounds, they will be kicked out for being AFK (Away From Keyboard).

In "**Question quiz**", users will be presented with a series of 5 to 10 general knowledge questions from various categories, including geography, science, astronomy, art, sports, technology, and history. Each question will have four possible options, with only one being correct.

The objective of the game is to answer 5 to 10 questions correctly. The user will have 30 seconds to select their answer for each question. If the user fails to answer within the time limit, it will be counted as an incorrect response, and they will lose a life.

The game provides users with a total of 3 lives. If they answer a question incorrectly or fail to respond within the time limit, they will lose a life. Once all three lives are lost, the game ends, and the user is considered defeated. On the other hand, if the user manages to answer all the questions correctly, they win the game.

The questions in the game cover a wide range of topics, testing the user's knowledge in different areas. They will need to rely on their general knowledge and quick thinking to select the correct answer within the given time frame. The game aims to challenge the user's knowledge and provide an engaging and educational experience. It encourages learning and improvement by offering a variety of questions from different categories.

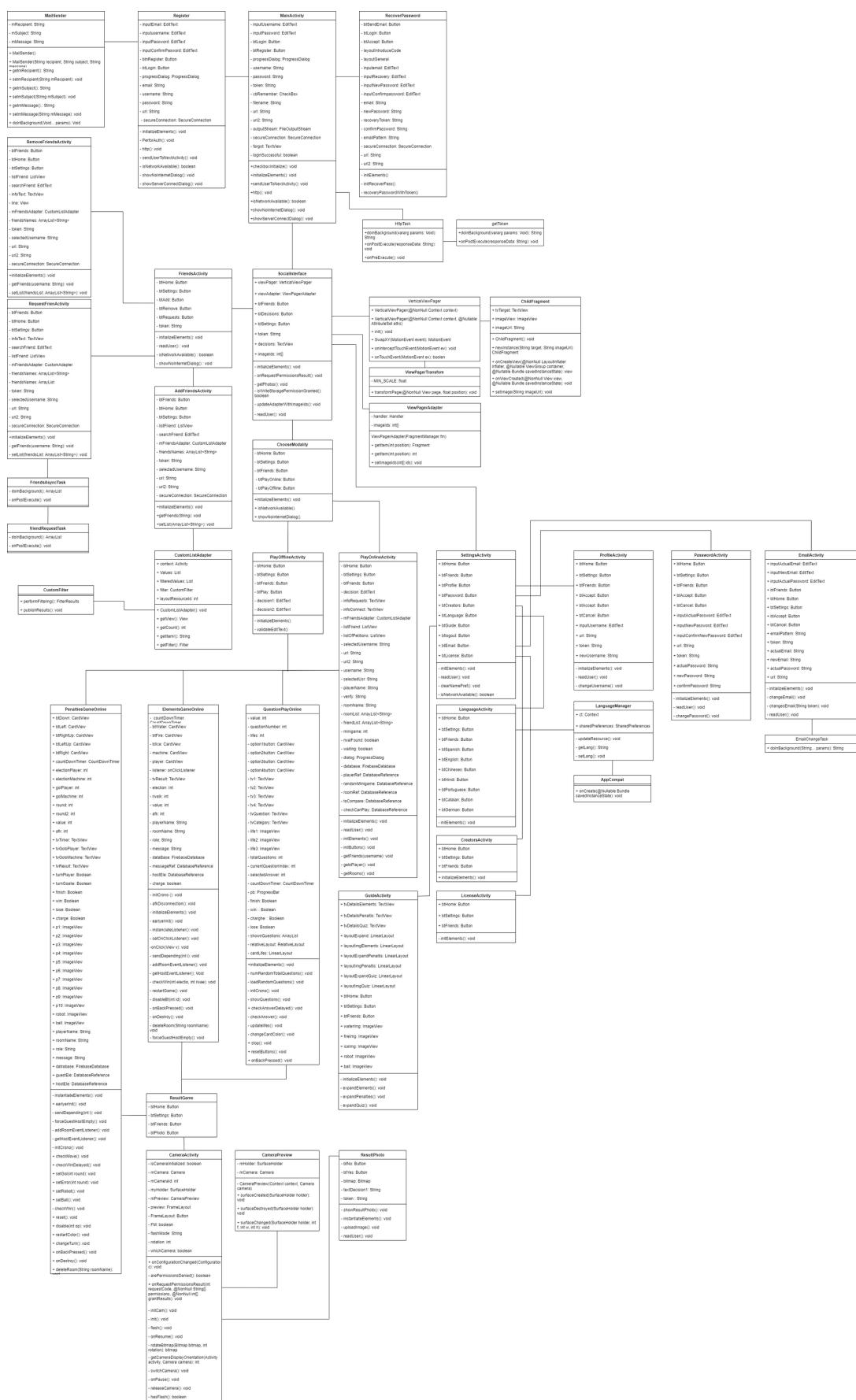
Game results

When the games are over there can be 4 different result screens if you play online or if you play against the machine, result win which will indicate that you have won the game and gives you the possibility to upload a photo, result lost which indicates that you have lost and only you can go back to the main screen, settings or friends management, result lost machine when you lose against the machine gives the possibility to upload the photo with the second decision you assigned, and the last screen is result afk for when you play online if one or both users are inactive for 3 rounds in a row kicks them out for inactivity.

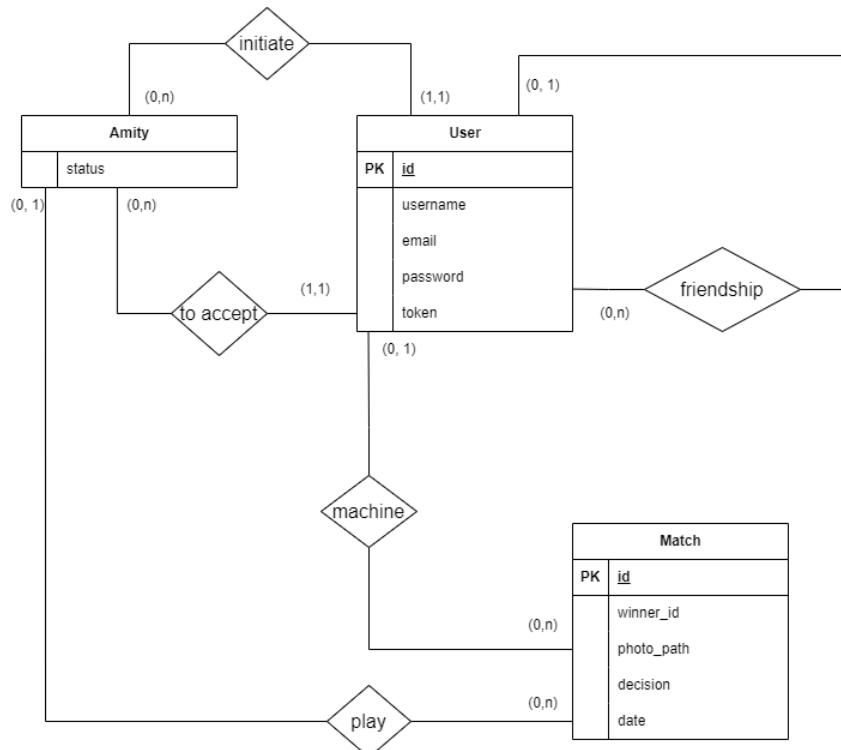
Share your decision(Camera and result photo)

When the user presses the button to take a photo, it will first ask for camera permissions, once I accept them, a custom camera decisions will open where you can use flash and turn the camera around, when you take the photo it will take you to a screen with the result of the photo where you can cancel or confirm, if you cancel, go back to the camera, if you confirm, post the photo and you and your friends will be able to see it.

Class diagram (Link clicking image)



Database E-R model (Link clicking image)



The entities are:

Users:

`username`: a unique identifier for a user's account, usually chosen by the user themselves.

`email`: the email address associated with the user's account, used for communication and account recovery.

`password`: the secure password associated with the user's account, used for authentication and access control.

`token`: a unique identifier generated by the server to authenticate the user's requests and maintain session state.

`Token` is a random code generated using sha, who uses the `username` as the key for generating the token, and it is used for know who is asking or who is sending petitions.

Match:

`photo_path`: the photo that the winner will post after playing the minigame

`decision`: the decision is a text that the user enters before playing, and only the winner is saved

`winner_id`: the unique identifier for the player who won the game, if there is one.

`date`: the date the game was played it is used for store the user who win the game, it is used for know who wined the match and shows it to the user when is searched it.

Amity:

user1_id: the unique identifier for the first user in the friendship.

user2_id: the unique identifier for the second user in the friendship.

status: the current status of the friendship, such as "pending", "accepted", or "reg", pending is for showing it in the friend_request of the app, who indicates te request for be friends is not accepted and is waiting for accept, accepted is for indicagte the two users are friends, and cana play thogether, and reg is for indicate when two users quit his friendship in the app, its indicate the two users, are friends in the past.

The relationships are:

A user (machine) can play 0 to N matches, and matches can be played by 1 user.

A user (player) can play 0 to N matches, and matches can be played by 1 to 2 users.

A user can have 0 to N friends, and a friendship there may be to 1 user to 0 users.

On a minigame participates 1 to 1 matches, and match there may be 0 to N minigames.

A user can initiate 0 to n amity, a amity can be initiated by 1 user.

A user can accept 0 to n amities, and amity can be accepted by 1 user.

The restrictions are:

Users:

The username must be unique across all users.

The email address must be unique across all users.

The email must be a valid string that follows a valid email address format, which means it should contain a single "@" followed by a valid domain name.

The username and password must meet certain complexity requirements to ensure security.

Amity:

If user1 sends a friend request to user2, and user2 accepts it, they will be mutual friends, that is, user2 will not need to send a request to user1.

When they are accepted they will no longer be seen in add friends

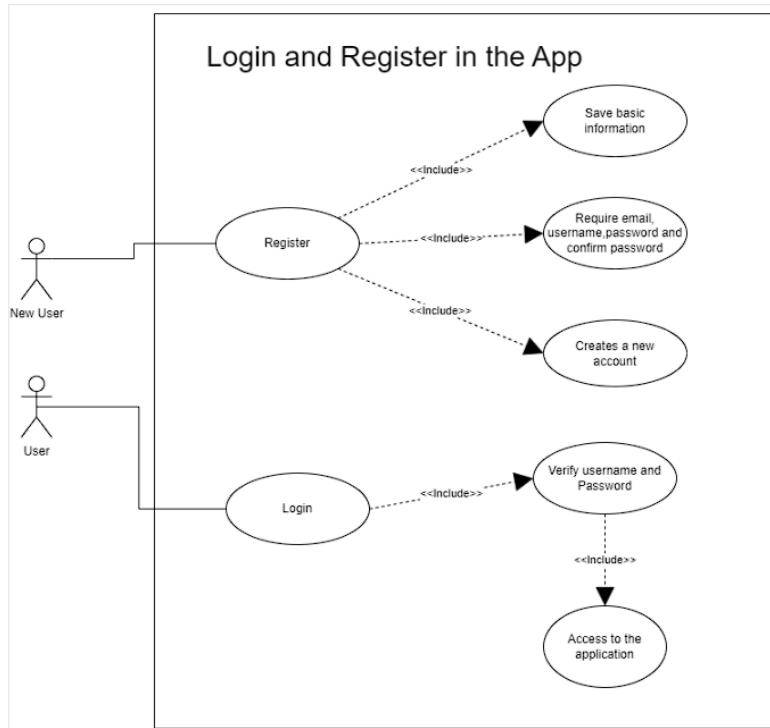
Use Case diagram and Textual description

Name:	CDU.1-Login
Creator:	David Chaparro
Version:	14/03/2023
Description:	Allows the user to log into the application with their username and password, two buttons appear, one to enter and another to go to the registration window.
Actor:	User, administrator
Test data:	username[valid,invalid], password[valid, invalid]
Precondition:	The user must have a valid username and password in order to log in.
Dependency:	The registration window must be available for the user to create an account.
Steps:	<p>The user opens the application.</p> <p>The login screen appears, prompting the user to enter their username and password.</p> <p>The user enters their valid username and password into the respective fields.</p> <p>The user clicks the "Enter" button.</p> <p>The system verifies the username and password entered by the user.</p> <p>If the username and password are valid, the user is logged in and directed to the main screen of the application.</p> <p>If the username and password are invalid, an error message is displayed, and the user is prompted to re-enter their login information.</p> <p>If the user does not have an account, they can click the "Registration" button to be directed to the registration window.</p>
	<p>Expected results:</p> <p>If the user enters valid login information, they should be successfully logged into the application.</p> <p>If the user enters invalid login information, they should receive an error message and be prompted to re-enter their login information.</p> <p>If the user clicks the "Registration" button, they should be directed to the registration window.</p>

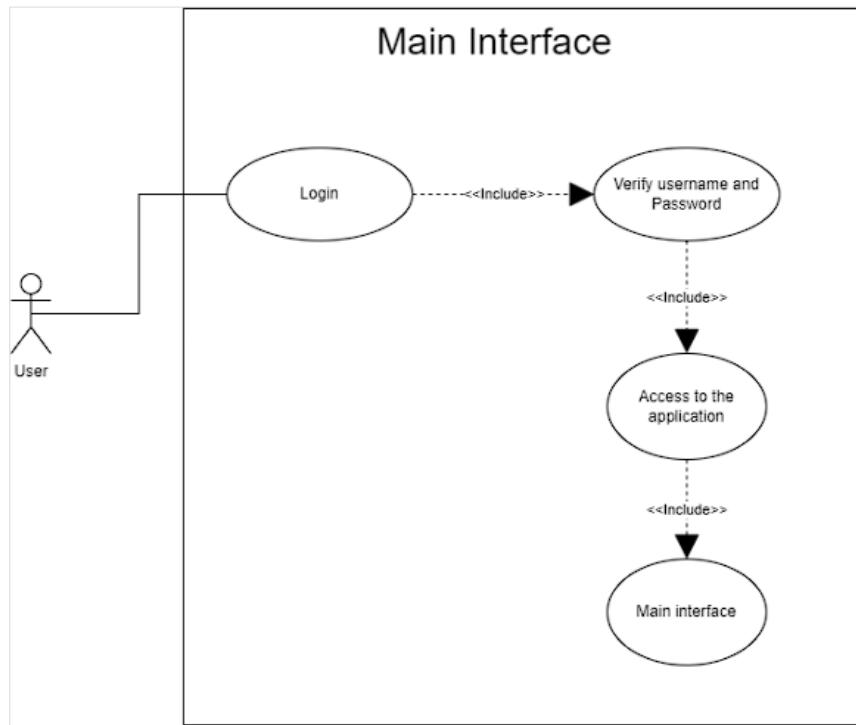
Entry condition to be discussed	Classes validated by the analyzed condition	Classes not validated by the analyzed condition
Username	A username that meets the criteria set by the system (e.g., minimum length, allowed characters, uniqueness, etc.)	User account status: The condition of the user's account (e.g., active, inactive, suspended, etc.) The level of access and permissions granted to the user based on their role within the system (e.g., admin, user, guest, etc.)
Password	A password that meets the criteria set by the system (e.g., minimum length, inclusion of special characters, etc.)	The condition of the user's account (e.g., active, inactive, suspended, etc.)

Name:	CDU.2-Register
Creator:	Fernando Maridueña
Version:	15/03/2023
Description:	Allows the user to register into the application with their username, password and email two buttons appear, one to enter and another to go to the login window.
Actor:	User
Test data:	username[valid,invalid], password[valid, invalid], email[valid,invalid]
Precondition:	The user must have a valid username , password and email.
Dependency:	The login window must be available for the user to enter their account.
Steps:	<p>The user opens the application.</p> <p>The user does not have an account, they can click the "Registration" button to be directed to the registration window.</p> <p>The user enters their valid username, password and email into the respective fields.</p> <p>The user clicks the "Enter" button.</p> <p>The system verifies the username, password and email entered by the user.</p> <p>If the username, password and email are valid, the user logs in and goes to the login screen.</p> <p>If the username, password and email are invalid, an error message is displayed, and the user is prompted to re-enter their register information.</p>
	<p>Expected results:</p> <p>If the user enters valid register information, they should be successfully register into the application.</p> <p>If the user enters invalid registration information, they should receive an error message and be prompted to re-enter their information.</p> <p>If the user clicks the "LOGIN" button, they should be directed to the log in window.</p> <p>Saves the information entered in the database.</p>

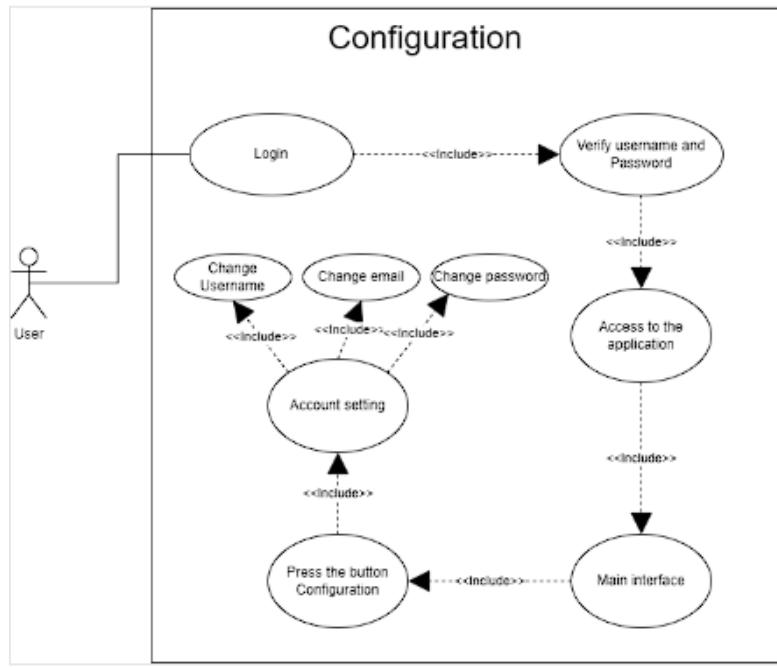
Entry condition to be discussed	Classes validated by the analyzed condition	Classes not validated by the analyzed condition
Username	A username that meets the criteria set by the system (e.g., minimum length, allowed characters, uniqueness, etc.)	User account status: The condition of the user's account (e.g., active, inactive, suspended, etc.) The level of access and permissions granted to the user based on their role within the system (e.g., admin, user, guest, etc.)
Password	A password that meets the criteria set by the system (e.g., minimum length, inclusion of special characters, etc.)	The condition of the user's account (e.g., active, inactive, suspended, etc.)
Email	Format válido de e-mail none@domain.com	Invalid e-mail format nonedomain.com



Name:	CDU.3-Menu interface
Creator:	Zhiyang Wu
Version:	15/03/2023
Description:	Allows the user see the menu and the options it gives it
Actor:	User
Test data:	play button[play,error], settings button[send to settings panel,error], addfriends button[send to addfriendspanel,error]
Precondition:	The user must have click a button.
Dependency:	Send u to other screens wen u click the button.
Steps:	<p>if the user clicks the settings, its send the user to the settings panel</p> <p>if the user clicks her profile, its shows the user profile card.</p> <p>if the user click the play button, starts a random minigame.</p> <p>if the user click add friends button its sends u to addfriends panel and its ask u the id of our friend and searches it</p>
	<p>Expected results:</p> <p>sends u to the settings panel and shows optins u can change.</p> <p>shows the statisitics of the user</p> <p>shows the panel who ask the user decision and starts a random minigame</p> <p>shows the addfriend panel and allows to send a friend request</p>

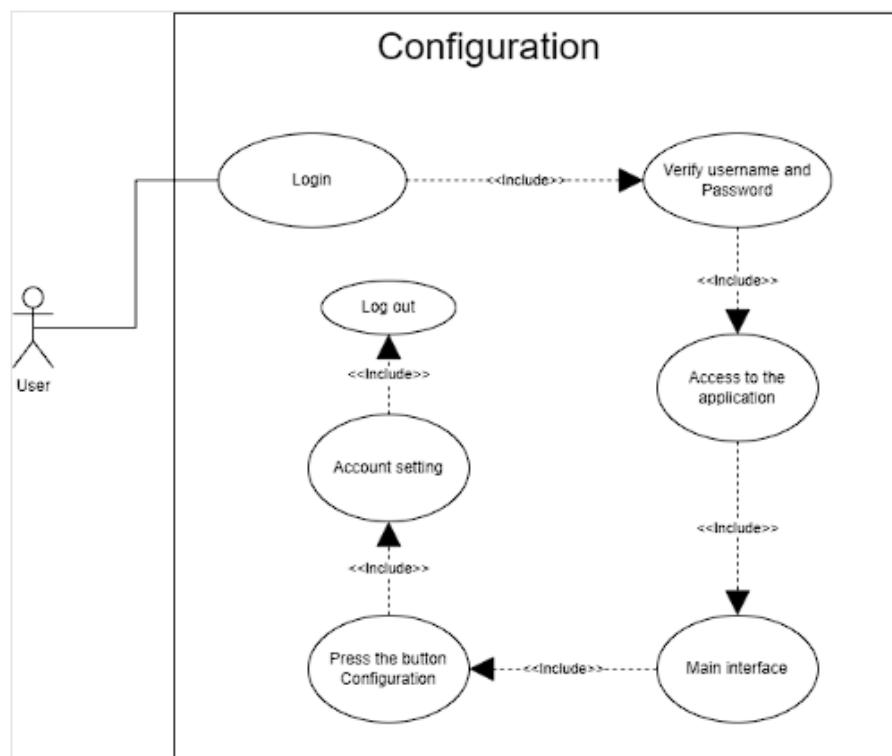


Name:	CDU.4-Configuration
Creator:	David Chaparro
Version:	13/04/2023
Description:	Allows the user see the configuration menu with ther respective buttons, edit profile, change password, log out, change language, games guide.
Actor:	User, administrator
Test data:	-
Precondition:	The user must have registered and logged in correctly and click on the settings icon.
Dependency:	The Configuration feature depends on the user being registered and logged in correctly to the system. Without this precondition, the user will not be able to access the settings icon and access the menu. Therefore, the dependency for this feature is that the user must have a valid account and be logged into the system before attempting to access the settings icon.
Steps:	<p>The user logs into the system.</p> <p>The user clicks on the settings icon.</p> <p>The Configuration menu is displayed.</p> <p>The user sees the buttons for editing their profile, changing their password, logging out, changing the language, and accessing a games guide.</p> <p>The user clicks on the "Edit profile" button to modify their personal information.</p> <p>The user clicks on the "Change password" button to update their login credentials.</p> <p>The user clicks on the "Log out" button to exit the system.</p> <p>The user clicks on the "Change language" button to select a different language.</p> <p>The user clicks on the "Games guide" button to access information about the available games.</p> <p>If the user is an administrator, they can access additional options such as managing users or modifying system settings.</p> <p>The Configuration menu is easy to navigate and understand.</p> <p>The user is able to perform their desired actions without difficulty.</p>
Expected results:	<p>The Configuration menu should be displayed with the respective buttons. The user should be able to easily navigate the menu and access the desired features. If the user is an administrator, they should have access to additional options. The user should be able to modify their profile information, change their password, log out of the system, change the language, and access a games guide. Configuration menu should be user-friendly and intuitive.</p>



Name:	CDU.5-Change language
Creator:	David Chaparro
Version:	13/04/2023
Description:	Allows the user see the language menu and click on the language of he wants, english, spanish, catalan, chinese, german, hindi and portuguese.
Actor:	User, administrator
Test data:	-
Precondition:	The user must have registered and logged in correctly and click on the settings icon, and click on Select your language.
Dependency:	The Change language feature depends on the system having language options available for the user to select. The system must also be able to display the user's chosen language correctly throughout the application. Additionally, the user must be registered and logged in to the system and have accessed the settings icon before selecting their preferred language.
Steps:	<p>The user logs into the system.</p> <p>The user clicks on the settings icon.</p> <p>The Configuration menu is displayed.</p> <p>The user clicks on the "Change language" button.</p> <p>The language menu is displayed with options for English, Spanish, Catalan, Chinese, German, Hindi, and Portuguese.</p> <p>The user selects their preferred language from the menu.</p> <p>The system updates the language settings accordingly.</p> <p>All system text and messages are displayed in the user's selected language.</p> <p>The user can navigate the system in their preferred language.</p> <p>Expected results:</p> <p>Change language feature should allow the user to select their preferred language from a menu of available options. The system should be able to display the user's chosen language throughout the application without any issues. The language menu should be easy to navigate and understand. Once the user has selected their preferred language, all system text and messages should be displayed in that language, and the user should be able to navigate the system with ease.</p>

Name:	CDU.6-Log out
Creator:	David Chaparro
Version:	13/04/2023
Description:	Allows the user log out of the app and go to log in interface.
Actor:	User
Test data:	-
Precondition:	The user must have registered and logged in correctly and click on the settings icon, and click on Log out.
Dependency:	Dependency: The Log out feature depends on the user being registered and logged in correctly to the system. Additionally, the user must have accessed the settings icon and selected the "Log out" button before attempting to log out of the system.
Steps:	<p>The user logs into the system. The user clicks on the settings icon. The CDU.4-Configuration menu is displayed. The user clicks on the "Log out" button. The system logs the user out and returns them to the log in interface. The user is no longer able to access any system features without logging in again.</p>
	<p>Expected results:</p> <p>Log out feature should allow the user to log out of the system and return to the log in interface. Once the user is logged out, they should not be able to access any system features without logging in again.</p>

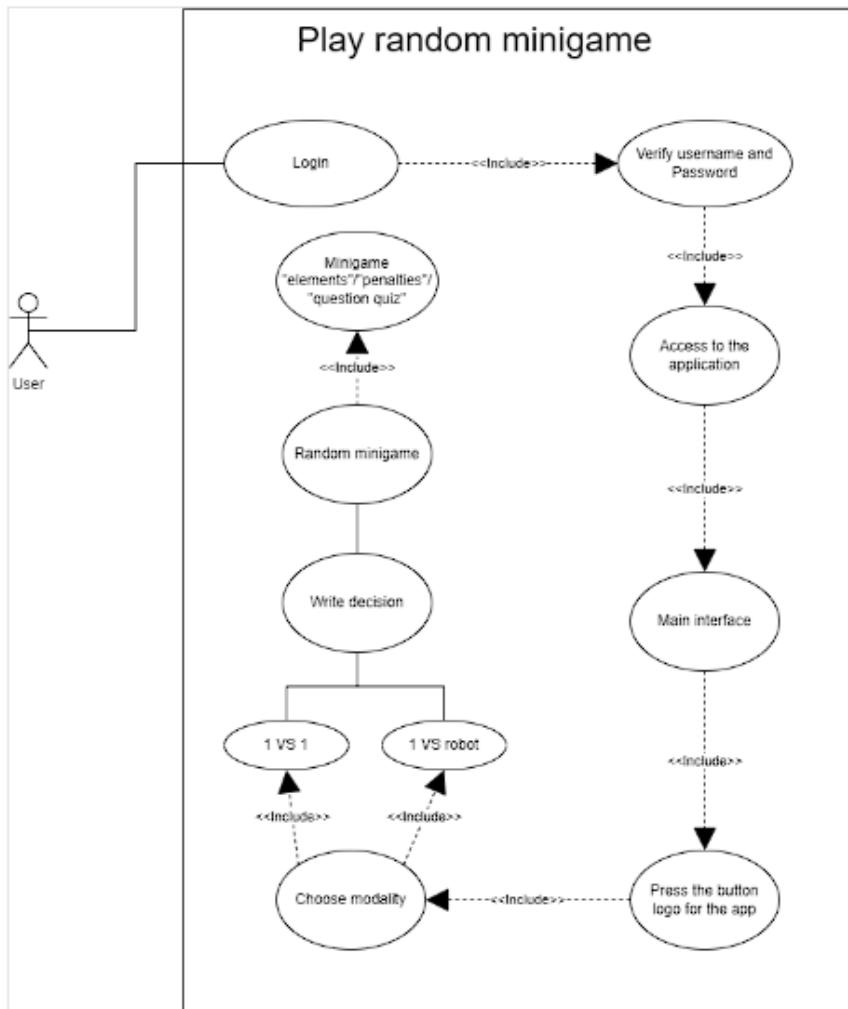


Name:	CDU.7-Minigame "Elements"	
Creator:	David Chaperro	
Version:	13/04/2023	
Description:	Allows the user play the first minigame "Elements" see the intro animation and play the game, we will have 3 options to choose between water, fire and ice, water wins over fire, fire wins over ice , and the ice beats the water, if you are tied it starts again until there is a winner or loser, if the 15s time runs out your move will be made automatically, when you win or lose it will take you to a provisional layout where the winner can share later with a photo your decision.	
Actor:	User	
Test data:	-	
Precondition:	The user must have registered and logged in correctly and click on the take decisions icon, and click on Play.	
Dependency:	Dependency: The Minigame "Elements" feature depends on the system having the game "Elements" available for the user to play. The system must also be able to display the game interface correctly and keep track of the user's moves and results. Additionally, the user must be registered and logged in to the system and have accessed the "take decisions" icon before starting the game.	
Steps:	<p>The user logs into the system.</p> <p>The user clicks on the "take decisions" icon.</p> <p>The game interface for "Elements" is displayed.</p> <p>The user sees the introduction animation.</p> <p>The user selects their move from three options: water, fire, or ice.</p> <p>The system calculates the winner based on the game rules: water beats fire, fire beats ice, and ice beats water.</p> <p>If there is a tie, the game starts again until there is a winner or loser.</p> <p>If the user does not make a move within 15 seconds, the system automatically makes a move for them.</p> <p>When there is a winner or loser, the system displays a provisional layout where the winner can share their decision with a photo.</p> <p>The user can choose to play again or return to the "take decisions" interface.</p>	<p>Expected results:</p> <p>The Minigame "Elements" feature should allow the user to play the game "Elements" with ease. The system should be able to display the game interface correctly and keep track of the user's moves and results. The game should have clear instructions and rules for the user to follow. The winner of the game should be correctly calculated based on the game rules. If there is a tie, the game should start again until there is a winner or loser. The user should be able to share their decision with a photo after winning. Finally, the user should be able to choose to play again or return to the "take decisions" interface.</p>

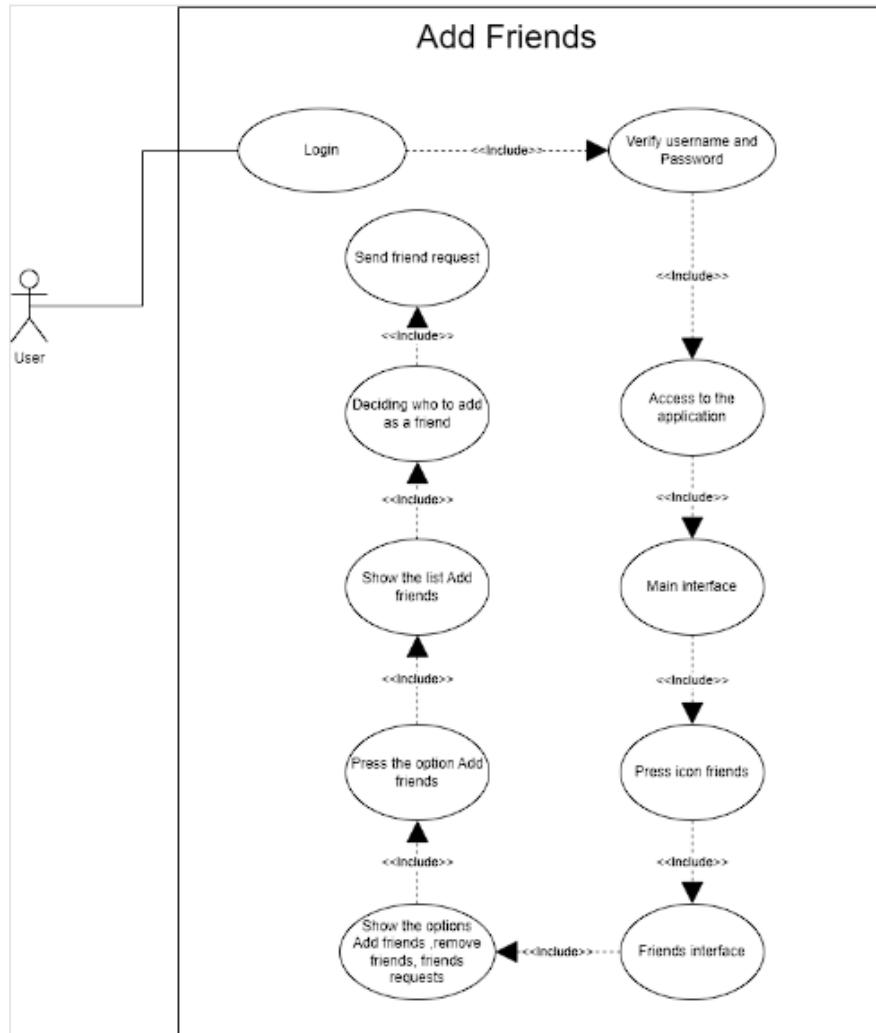
Name:	CDU.7-Minigame "Elements"	
Creator:	David Chaperro	
Version:	13/04/2023	
Description:	Allows the user play the first minigame "Elements" see the intro animation and play the game, we will have 3 options to choose between water, fire and ice, water wins over fire, fire wins over ice , and the ice beats the water, if you are tied it starts again until there is a winner or loser, if the 15s time runs out your move will be made automatically, when you win or lose it will take you to a provisional layout where the winner can share later with a photo your decision.	
Actor:	User	
Test data:	-	
Precondition:	The user must have registered and logged in correctly and click on the take decisions icon, and click on Play.	
Dependency:	Dependency: The Minigame "Elements" feature depends on the system having the game "Elements" available for the user to play. The system must also be able to display the game interface correctly and keep track of the user's moves and results. Additionally, the user must be registered and logged in to the system and have accessed the "take decisions" icon before starting the game.	
Steps:	<p>The user logs into the system.</p> <p>The user clicks on the "take decisions" icon.</p> <p>The game interface for "Elements" is displayed.</p> <p>The user sees the introduction animation.</p> <p>The user selects their move from three options: water, fire, or ice.</p> <p>The system calculates the winner based on the game rules: water beats fire, fire beats ice, and ice beats water.</p> <p>If there is a tie, the game starts again until there is a winner or loser.</p> <p>If the user does not make a move within 15 seconds, the system automatically makes a move for them.</p> <p>When there is a winner or loser, the system displays a provisional layout where the winner can share their decision with a photo.</p> <p>The user can choose to play again or return to the "take decisions" interface.</p>	<p>Expected results:</p> <p>The Minigame "Elements" feature should allow the user to play the game "Elements" with ease. The system should be able to display the game interface correctly and keep track of the user's moves and results. The game should have clear instructions and rules for the user to follow. The winner of the game should be correctly calculated based on the game rules. If there is a tie, the game should start again until there is a winner or loser. The user should be able to share their decision with a photo after winning. Finally, the user should be able to choose to play again or return to the "take decisions" interface.</p>

Name:	CDU.11-Minigame "Penalties"
Creator:	Fernando Mariidueña
Version:	28/04/2023
Description:	Allows the user to play the seconds minigame "Penalties" see the intro animation and play the game, where we will have 2 roles one will be the goalkeeper another will be the one who shoots and will alternate in a round of 10 if the 15s time runs out your move will be made automatically, when you win or lose it will take you to a provisional layout where the winner can share a photo with decision.
Actor:	User
Test data:	-
Precondition:	The user must have registered and logged in correctly and click on the take decisions icon, and click on Play.
Dependency:	Dependency: The Minigame "Penalties" feature depends on the system having the game "Penalties" available for the user to play. The system must also be able to display the game interface correctly and keep track of the user's moves and results. Additionally, the user must be registered and logged in to the system and have accessed the "take decisions" icon before starting the game.
Steps:	<p>Expected results:</p> <p>The Minigame "Penalties" feature should allow the user to play the game "Penalties" with ease. The system should be able to display the game interface correctly and keep track of the user's moves and results. The game should have clear instructions and rules for the user to follow. The winner of the game should be correctly calculated based on the game rules. If there is a tie, the game to be continue until there is a winner or loser. The user should be able to share their decision with a photo after winning. Finally, the user should be able to choose to play again or return to the "take decisions" interface.</p> <p>The user logs into the system. The user clicks on the "take decisions" icon. The game interface for "Penalties" is displayed. The user sees the introduction animation. The user selects their move from options depending on whether you are a goalkeeper or the one who shoots: right , left , up, down, top right, top left , down right, down left. The system calculates the winner based on the game rules: there is a winner after 10 rounds as long as there is no tie and if there is a tie after the rounds they go to sudden death. If there is a tie, the game starts again until there is a winner or loser. If the user does not make a move within 15 seconds, the system automatically makes a move for them. When there is a winner or loser, the system displays a provisional layout where the winner can share their decision with a photo. The user can choose to play again or return to the "take decisions" interface.</p>

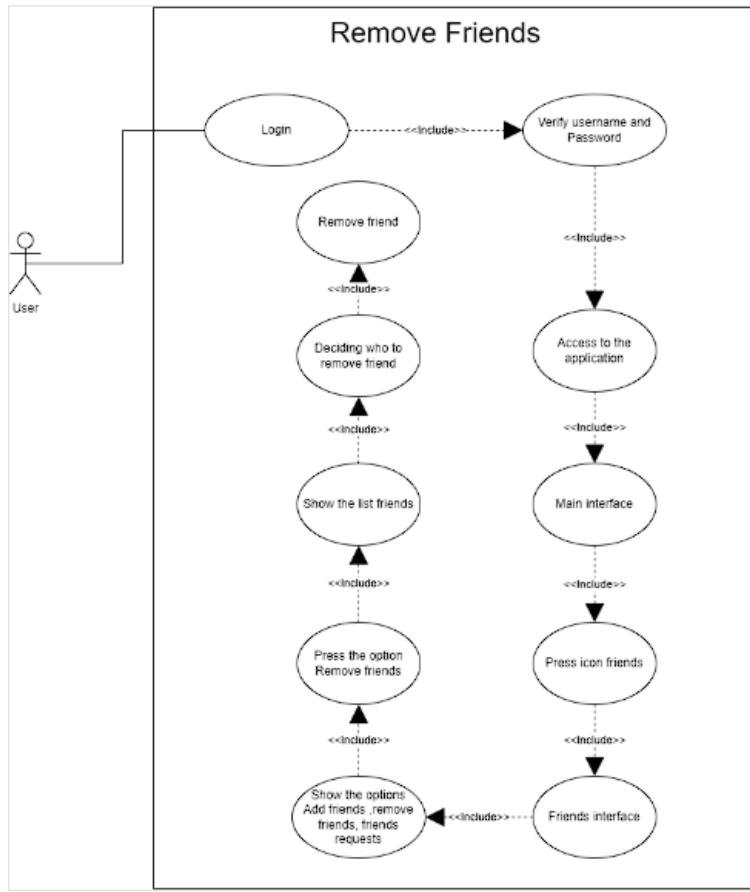
Name:	CDU.12-Minigame "Question quiz"	
Creator:	David Chaparro	
Version:	28/04/2023	
Description:	Allows the user to play the third minigame "Question quiz" see the intro animation and play the game, the user will have to answer between 5 and 10 random questions about different categories, the questions are about general culture, for each question there are 4 options and only one correct, there is also a 30sec timer per question if it ends, you lose a life, in total He has 3, if he runs out he loses, and if he hits all the questions he wins, if he plays online against another user the one who hits all the questions first wins.	
Actor:	User	
Test data:	<p>Test the stability of the internet connection and the login/register functionality before starting the game.</p> <p>Verify that the user can click on the "take decisions" icon and select either "1vs1" or "1vsRobot" mode.</p> <p>Test the intro animation and ensure that it plays without any technical issues.</p> <p>Verify that the user is presented with random questions from different categories.</p> <p>Test the user's ability to select an answer within 30 seconds.</p> <p>Verify that the user can win the game by answering all the questions correctly within the given time limit.</p> <p>Test that the user loses if they run out of lives.</p> <p>Verify that the user can play against the computer or another online player.</p> <p>Test the overall functionality and user experience of the minigame.</p>	
Precondition:	The user must have registered and logged in correctly and click on the take decisions icon, and click on 1vs1 or 1vsRobot and click Play.	
Dependency:	<p>The user must have a stable internet connection.</p> <p>The user must have registered and logged in correctly.</p> <p>The user must have clicked on the "take decisions" icon.</p> <p>The user must have selected either "1vs1" or "1vsRobot" mode.</p>	
Steps:	<p>After clicking on "Play", the user is presented with an intro animation.</p> <p>The game begins, and the user is presented with a random question from a selected category.</p> <p>The user has 30 seconds to select one of the four answer options.</p> <p>If the user selects the correct answer, they move on to the next question. Otherwise, they lose one of their three lives. If the user runs out of lives, the game ends. If the user answers all the questions correctly within the given time limit, they win. If the user chooses to play online against another user, the first player to answer all the questions correctly wins.</p>	<p>Expected results:</p> <p>The user should be able to play the "Question quiz" minigame without any technical issues.</p> <p>The user should be able to answer a random selection of questions from different categories. The user should be able to select an answer within 30 seconds.</p> <p>The user should be able to play against the computer or another online player.</p> <p>The user should be able to win the game by answering all the questions correctly within the given time limit.</p> <p>The user should lose if they run out of lives.</p> <p>The user should have a fun and engaging experience while playing the game.</p>



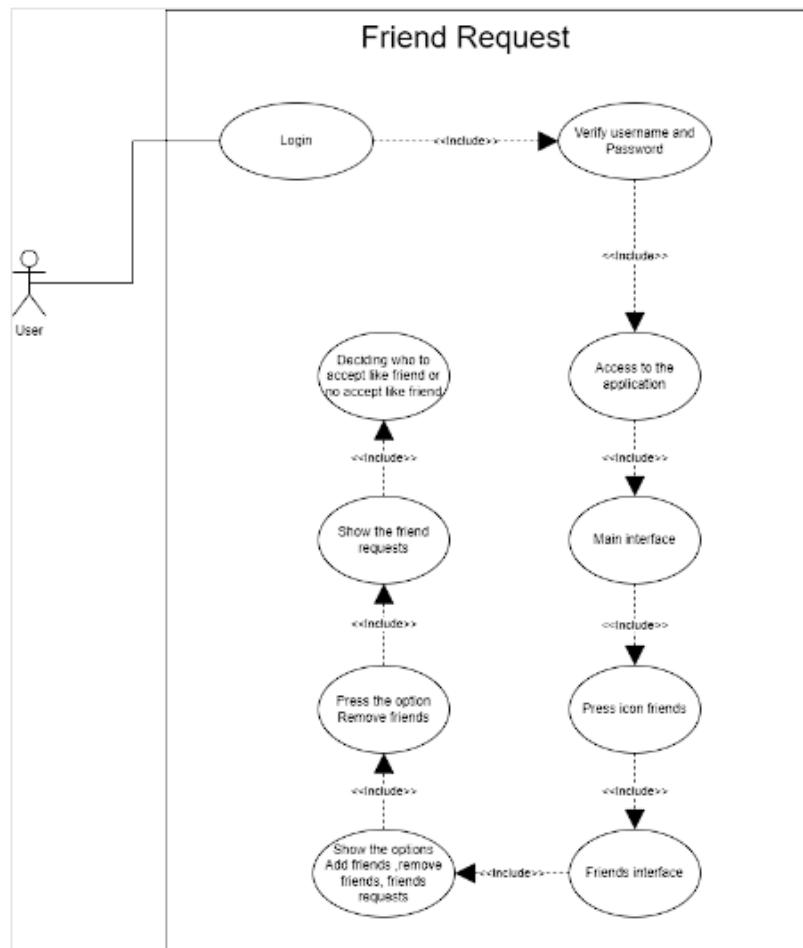
Name:	CDU.9-Add friend
Creator:	David Chaparro
Version:	13/04/2023
Description:	Allows the user add friends, sending a request and accepting.
Actor:	User
Test data:	User is logged into their account User clicks on the "Friends" icon User clicks on the "Add Friends" button User enters the name or email address of a person they want to add as a friend User clicks on the "Send Request" button
Precondition:	The user must have registered and logged in correctly and click on the friends icon, and click on Add friends.
Dependency:	Dependency:
Steps:	Expected results: The friends icon is visible and clickable. The "Add Friends" button is visible and clickable. The user can see the usernames of the users registered. The "Send Request" button is visible and clickable. If the person accepts the friend request, both users are added to each other's friends list, and they can see each other's profiles and activity updates. If the person declines the friend request, the user who sent the request will not be added to the person's friend list.



Name:	CDU.10-Remove friend
Creator:	David Chaparro
Version:	13/04/2023
Description:	Allows the user remove a friend.
Actor:	User
Test data:	User is logged into their account User has added a friend User clicks on the "Friends" icon User selects the friend they want to remove User clicks on the "Remove" button
Precondition:	The user must have registered and logged in correctly and click on the friends icon, and click on Remove friends.
Dependency:	Dependency: The user must have an active internet connection and have already registered and logged into their account on the social media platform.
Steps:	<p>User clicks on the friends icon.</p> <p>User selects the friend they want to remove.</p> <p>User clicks on "Remove".</p> <p>The system displays a confirmation message asking if the user really wants to remove the friend.</p> <p>User clicks "Yes" to confirm.</p> <p>The system removes the friend from the user's friend list.</p> <p>Expected results: The friends icon is visible and clickable. The user can see their list of friends. The "Remove" button is visible and clickable next to each friend's name. The system displays a confirmation message before removing the friend. After the user confirms, the friend is removed from the user's friend list and the user can no longer see the removed friend's profile or activity updates.</p>

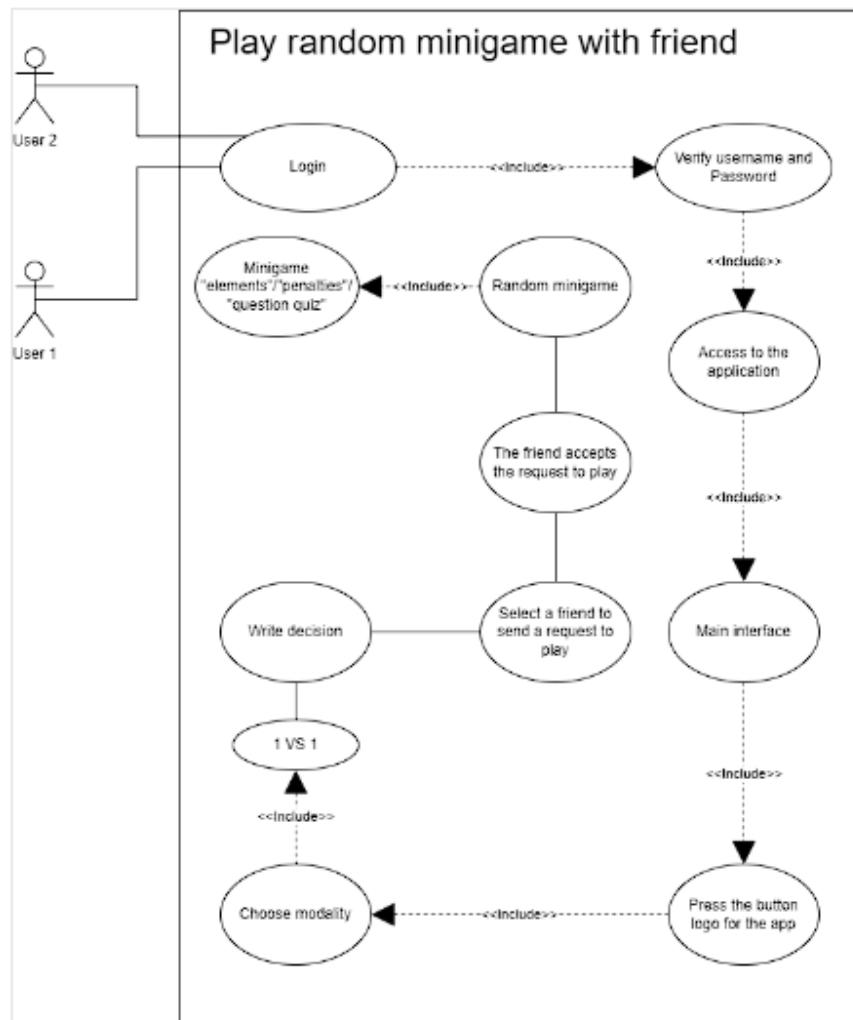


Name:	CDU.13-Friend request
Creator:	Fernando Maridueña
Version:	14/05/2023
Description:	Allows the user send a friend request.
Actor:	User
Test data:	User is logged into their account User has added a friend User clicks on the "Friends" icon User selects the friend they want to friend request User clicks on the "send friend request" button
Precondition:	The user must have registered and logged in correctly and click on the friends icon, and click on friend request.
Dependency:	Dependency: The user must have an active internet connection and have already registered and logged into their account on the social media platform.
Steps:	<p>User clicks on the friends icon. User selects the friend they want to friend request. User clicks on "send friend request". The system displays a confirmation message asking if the user really wants to send friend request. User clicks "Yes" to confirm. The system send friend request the user from the user's list.</p> <p>Expected results: The friends icon is visible and clickable. The user can see their list of friends. The " send friend request" button is visible and clickable next to each user name. The system displays a confirmation message before send friend request the user. After the user confirms, send friend request.</p>

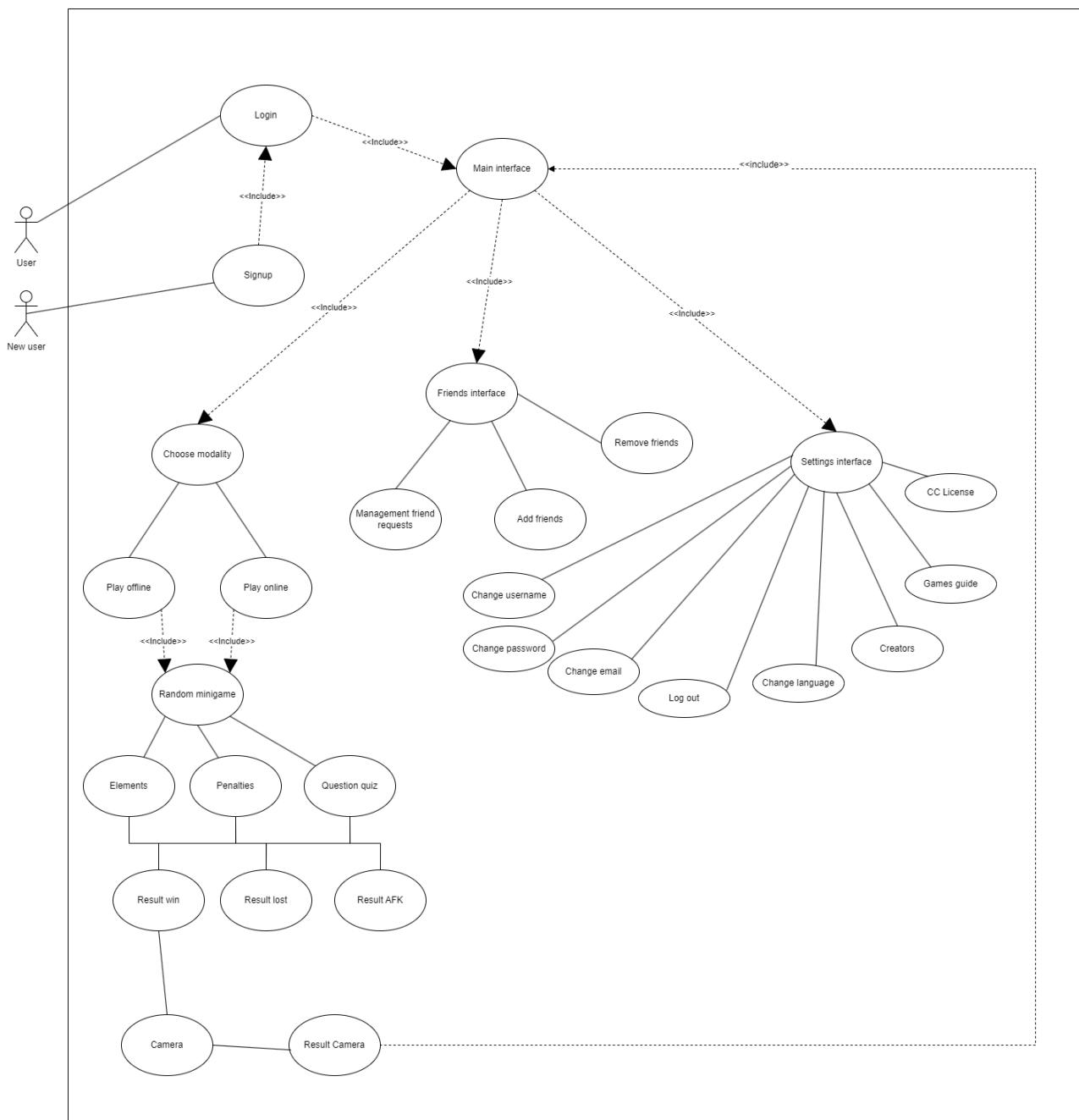


Name:	CDU.14-Connect with friend
Creator:	David Chaparro
Version:	11/05/2023
Description:	Allows the user to connect with friend before playing, to connect they must have been added before as friends, then choose the "1vs1" mode, once there you can send requests to 1 friend, the one who sent the invitation will be the host, and the one who receives it will be the guest, while the host is waiting You will have an alertbox to wait, you can cancel it by pressing the back button (onBackPressed) and your request will be deleted. Once the guest accepts the request, they will go to the "start" status and begin to play synchronously.
Actor:	User1, User2
Test data:	User1 is logged in and has selected the "take decisions" icon. User1 has selected the "1vs1" mode and has at least one friend added. Both User1 and User2 have a stable internet connection.
Precondition:	The user must have registered and logged in correctly and click on the take decisions icon, and click on 1vs1 and send request or accept request.
Dependency:	The user must have a stable internet connection. The user must have registered and logged in correctly. The user must have clicked on the "take decisions" icon. The user must have selected either "1vs1" mode.
Steps: User1 clicks on the "take decisions" icon and selects "1vs1" mode. User1 chooses a friend from their friend list to send a request to. User1 sends a request to the selected friend. User2 receives the request and can either accept or decline it. If User2 accepts the request, both users will be taken to the "start" status. If User2 declines the request, User1 will receive a notification that the request has been declined. While User1 is waiting for User2 to accept the request, an alert box will be displayed with a message to wait. User1 can cancel the request by pressing the back button (onBackPressed), which will delete the request. Once the request is accepted, both users will begin playing the game synchronously.	Expected results: User1 should be able to successfully send a request to User2 and User2 should be able to receive the request. If User2 accepts the request, both users should be taken to the "start" status and begin playing the game synchronously. While User1 is waiting for User2 to accept the request, an alert box should be displayed with a message to wait. User1 should be able to cancel the request by pressing the back button (onBackPressed), which should delete the request. Both users should have a stable internet connection to ensure that the game is played synchronously. The user must have registered and logged in correctly and clicked on the "take decisions" icon, and selected "1vs1" mode. The user must have at least one friend added in their friend list to send a request to.

Name:	CDU.15-Play random minigame with friend
Creator:	David Chaparro
Version:	11/05/2023
Description:	Allows the user to connect with friend before playing, and play random minigame synchronously.
Actor:	User1, User2
Test data:	Both User1 and User2 have a stable internet connection. User1 and User2 have added each other as friends. User1 and User2 have selected the "1vs1" option. User1 and User2 have agreed to play a random minigame.
Precondition:	The user must have registered and logged in correctly, and both users have added each other as friends. Both users have a stable internet connection. Both users have selected the "play with friend" option. Both users have agreed to play a random minigame.
Dependency:	The user must have a stable internet connection. The user must have registered and logged in correctly. Both users must have added each other as friends. Both users must have selected the "play with friend" option. Both users must have agreed to play a random minigame.
Steps: User1 and User2 select the "play with friend" option. User1 and User2 agree to play a random minigame. The game selects a random minigame for both users to play. Both users start playing the selected minigame. Once the minigame is finished, both users receive their scores. Both users are given the option to play again or return to the menu.	Expected results: Both User1 and User2 should be able to successfully select the "play with friend" option. Both User1 and User2 should agree to play a random minigame. The game should select a random minigame for both users to play. Both users should be able to start playing the selected minigame synchronously. Once the minigame is finished, both users should receive their scores. Both users should be given the option to play again or return to the menu. Both users should have a stable internet connection to ensure that the game is played synchronously.



Global use case (Link clicking image)



Use case	Relation/s
Signup	Login
Login	Recovery password, sign up and main interface
Main interface	Settings, Choose modality and Management Friends
Management friends	Settings, Main interface, Add friends, remove friends and friend requests

Use case	Relation/s
Settings	Management friends, Main interface, Change username, change password, change email, log out, games guide, select language, creator and cc license
Change username	Settings, Choose modality and Management Friends
Change password	Settings, Choose modality and Management Friends
Change email	Settings, Choose modality and Management Friends
Log out	Login
Select language	Settings, Choose modality and Management Friends
Games guide	Settings, Choose modality and Management Friends
Creators	Settings, Choose modality and Management Friends
CC License	Settings, Choose modality and Management Friends
Choose modality	Settings, Management friends, Main interface, play offline and Play online
Play offline	Elements / Penalties / Question quiz
Play online	Elements / Penalties / Question quiz
Elements	Result win / Result lost / Result afk
Penalties	Result win / Result lost / Result afk
Question quiz	Result win / Result lost / Result afk
Result win	Settings, Choose modality and Management Friends and Camera
Result lost	Settings, Choose modality and Management Friends
Result afk	Settings, Choose modality and Management Friends
Camera	Result photo
Result photo	Camera, Main interface

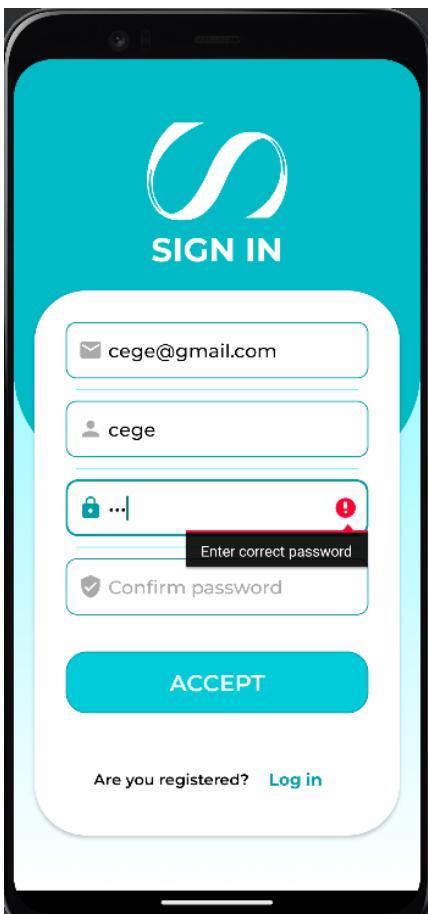
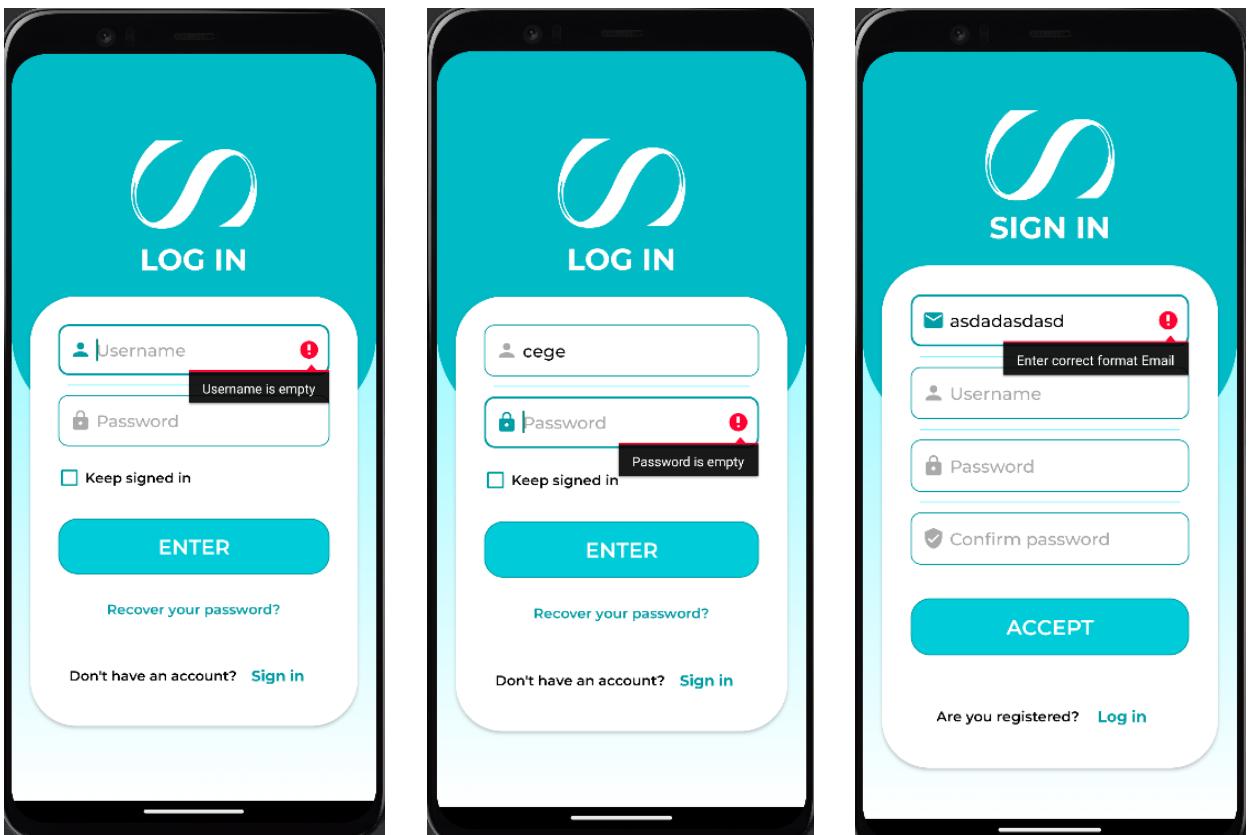
Test cases

Log in / Sign in

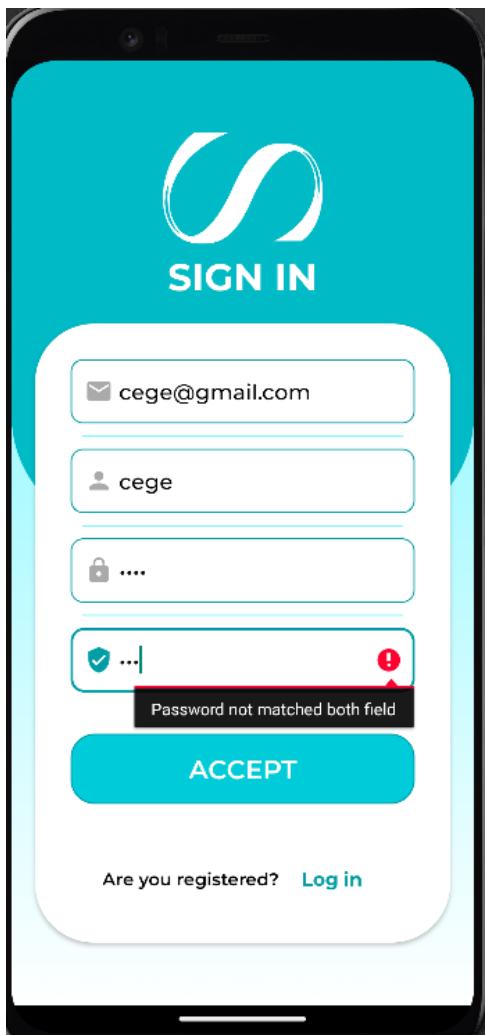
In order to ensure a smooth and seamless user experience, it is crucial for any digital platform or application to have a robust system in place for managing user accounts. One essential aspect of this system is the requirement for **unique usernames**, as having multiple users with the same name can lead to confusion and errors. Therefore, it is important to verify that each new user account is assigned a unique username that is not already in use by another user.

Another important aspect of account creation is the verification of email addresses. This is typically done by checking that the email provided contains the "**@ symbol**", followed by the **domain name** (e.g. ".com", ".cat", etc.). This helps to ensure that the email address is valid and can be used to communicate with the user if needed.

Additionally, it is important to check that all required fields are filled out during the account creation process. This includes fields such as username, email address, password, and any other relevant information. If any required **fields are left empty**, it can lead to errors or incomplete account information, which can cause issues for both the user and the platform. Therefore, it is important to thoroughly validate all user input and provide clear feedback to the user if any errors or omissions are detected.



When it comes to creating a password for a user account, it is important to ensure that the password is secure and difficult to guess. One simple way to improve password security is by requiring a minimum password size greater than 4 characters.

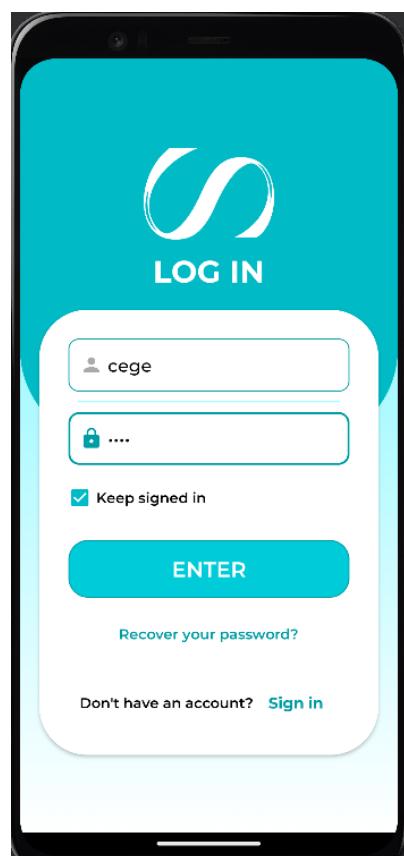


One common security measure when creating a new user account is to require the user to enter their chosen password twice, in order to verify that they have entered it correctly and that there are no typos or mistakes. This is typically done by checking if the passwords entered in both fields are the same.

Verifying that the passwords match is an important step in ensuring that the user has entered their password correctly and that there are no typos or mistakes in the password entry. If the passwords do not match, the user will be prompted to correct their input before proceeding. This helps to prevent errors or confusion that could lead to users being unable to access their account.

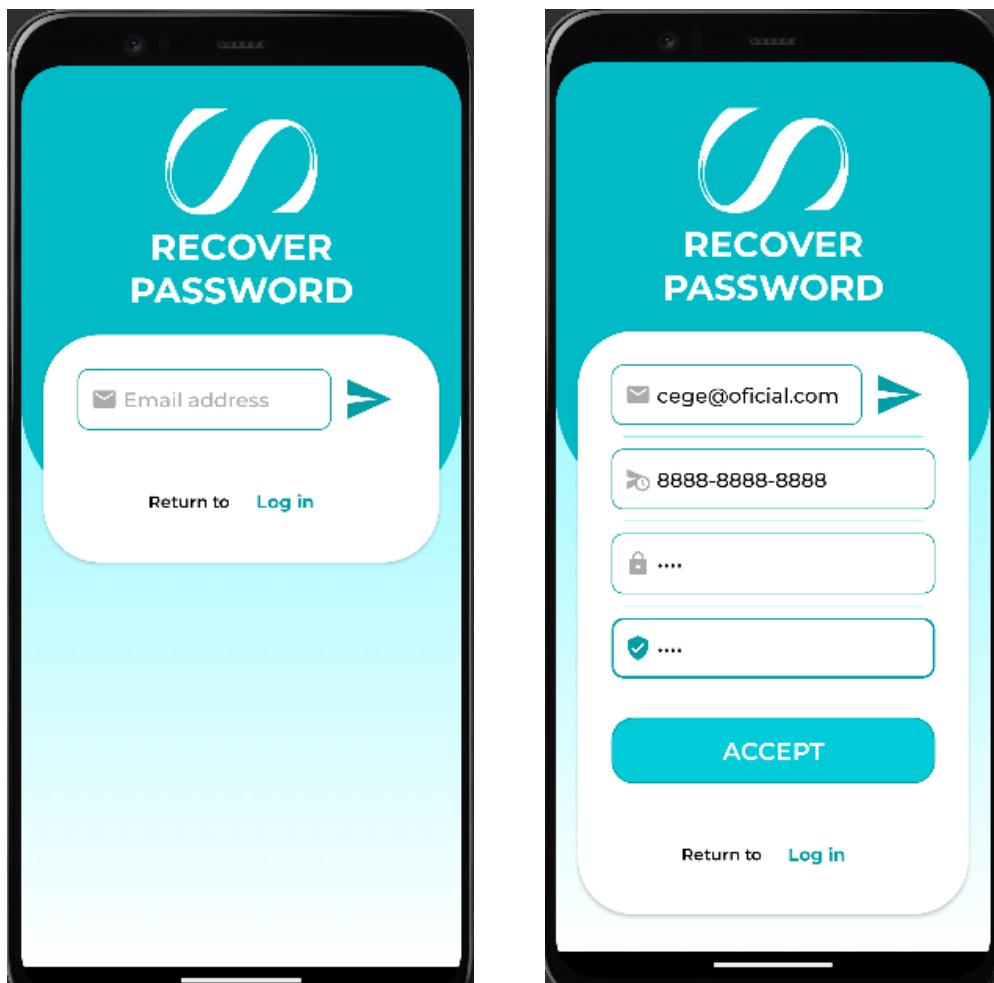
KEEP SIGN IN

If we mark to keep session started the next times we enter the application, it will take us to the main page instead of the login.



Recover password

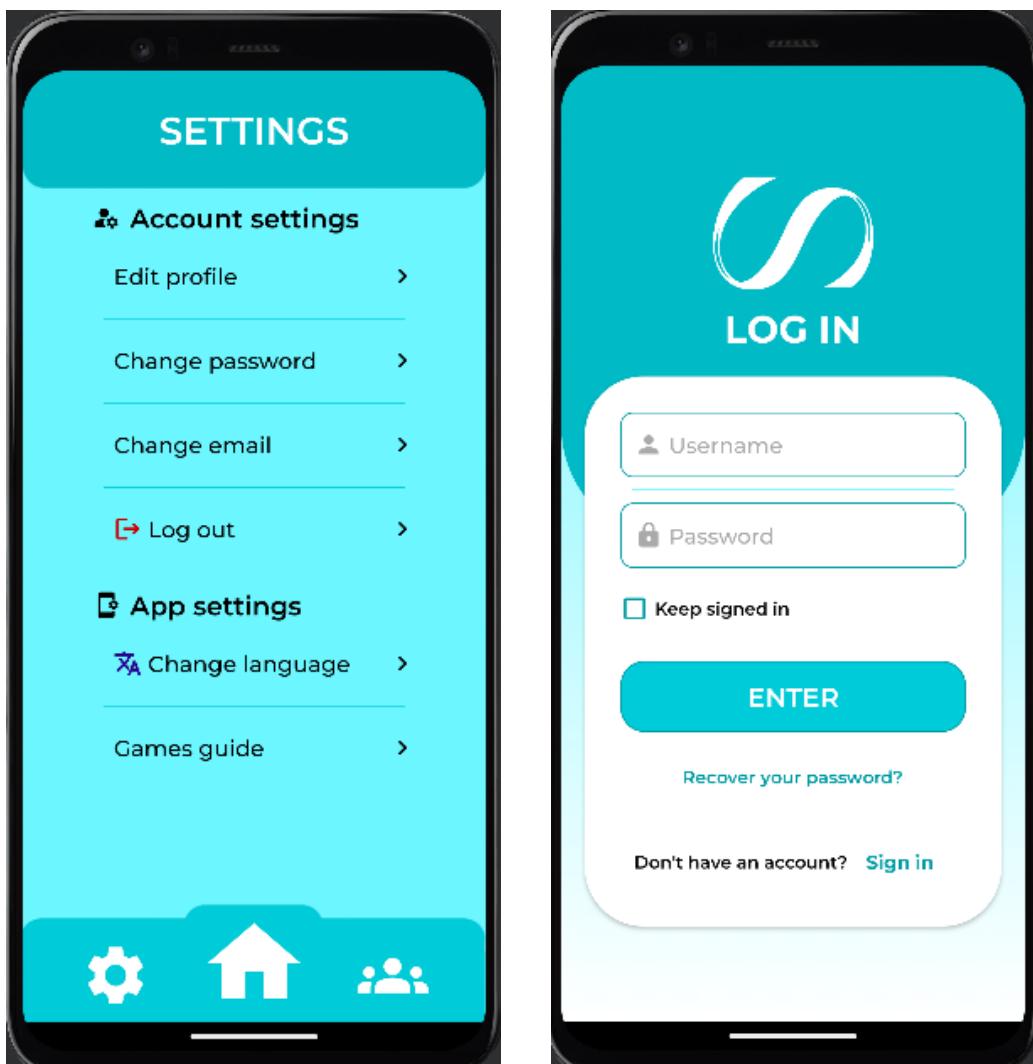
To recover the password, the user has to enter his email that he has entered in registration with a valid format and you will receive a one-time code to change your password.



Once the user writes the data correctly (correct temporary code, password +4 characters, same password in both fields) the password change will be applied.

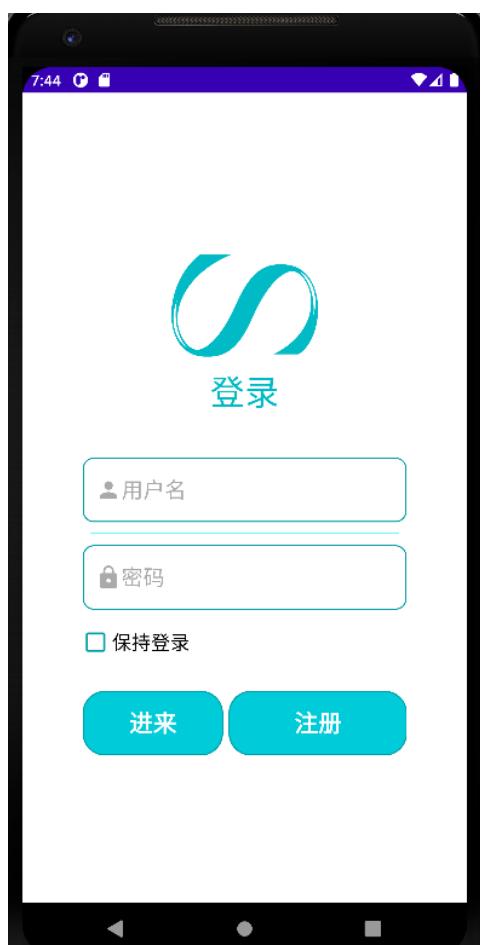
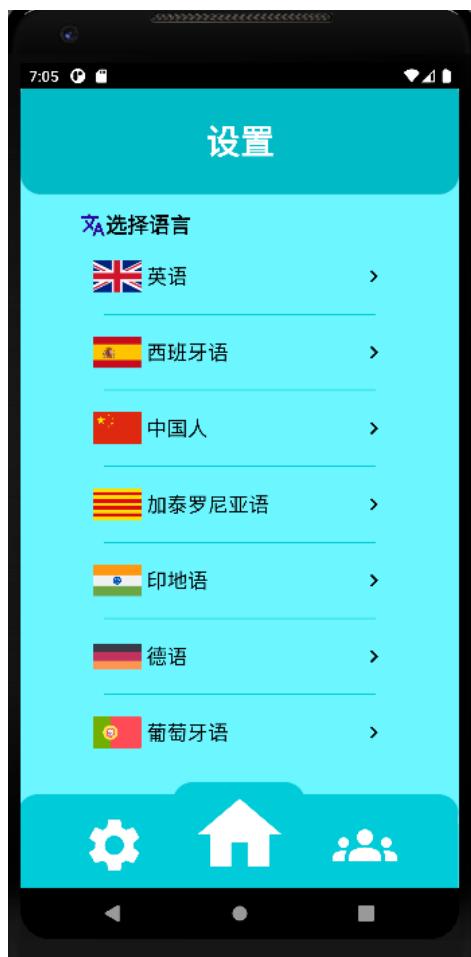
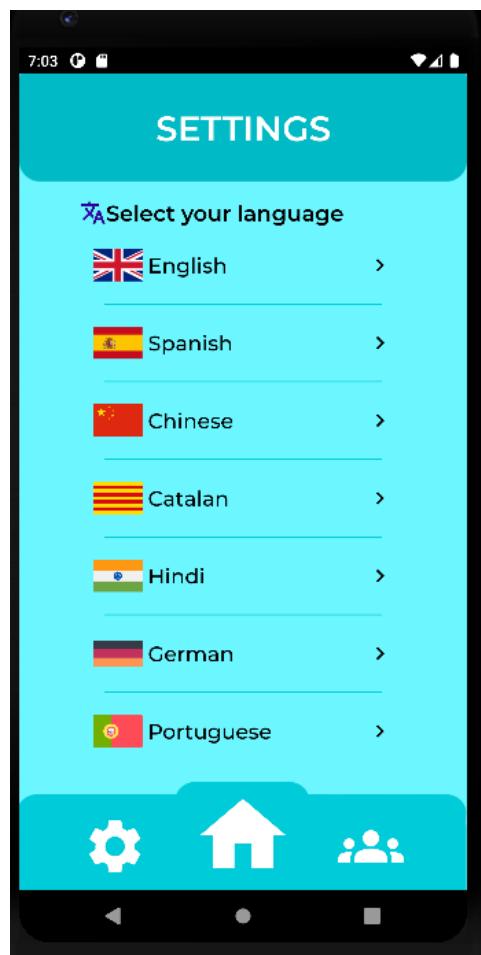
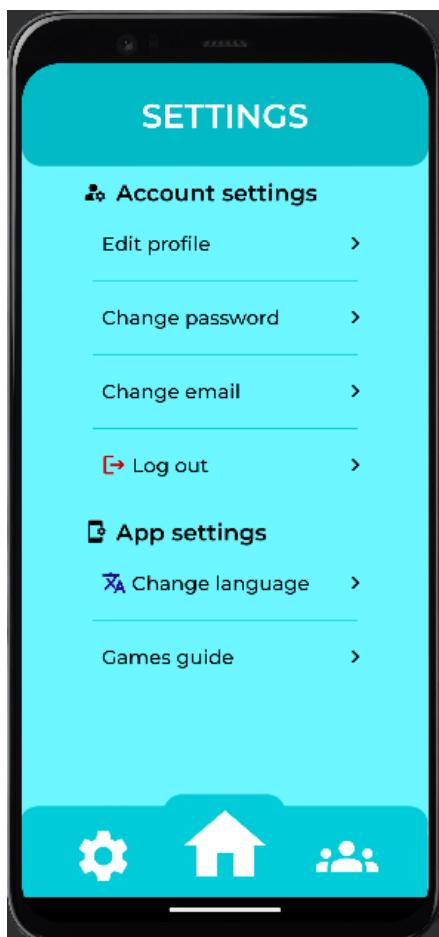
Log out

To log out the user must go to settings and click log out, once pressed it will close your session and take you to log in.



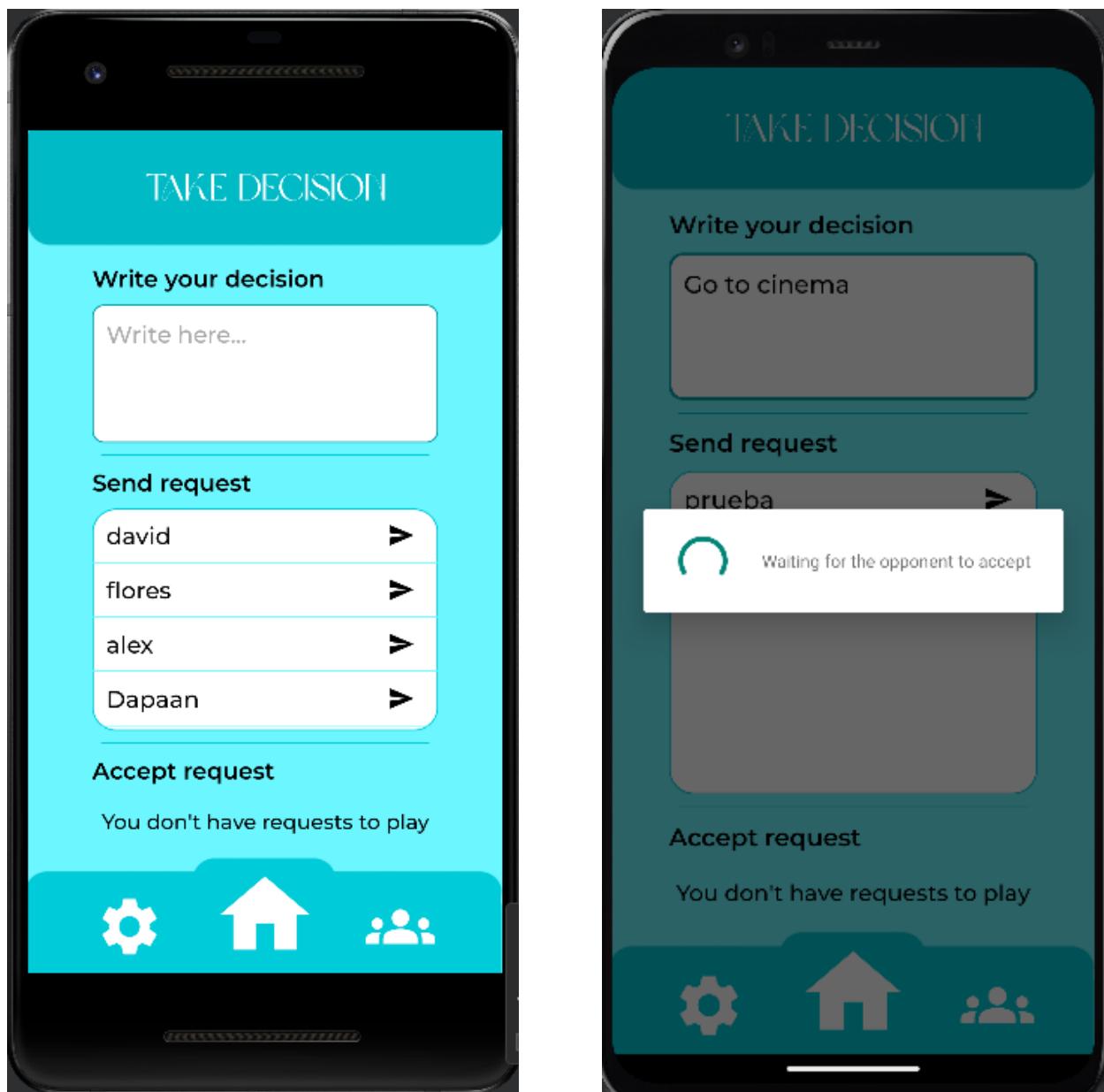
Change language

To change the language the user must enter settings and click on select language, you can choose between 7 languages, the one you choose will be saved even if you close the application.



CONNECT WITH FRIEND

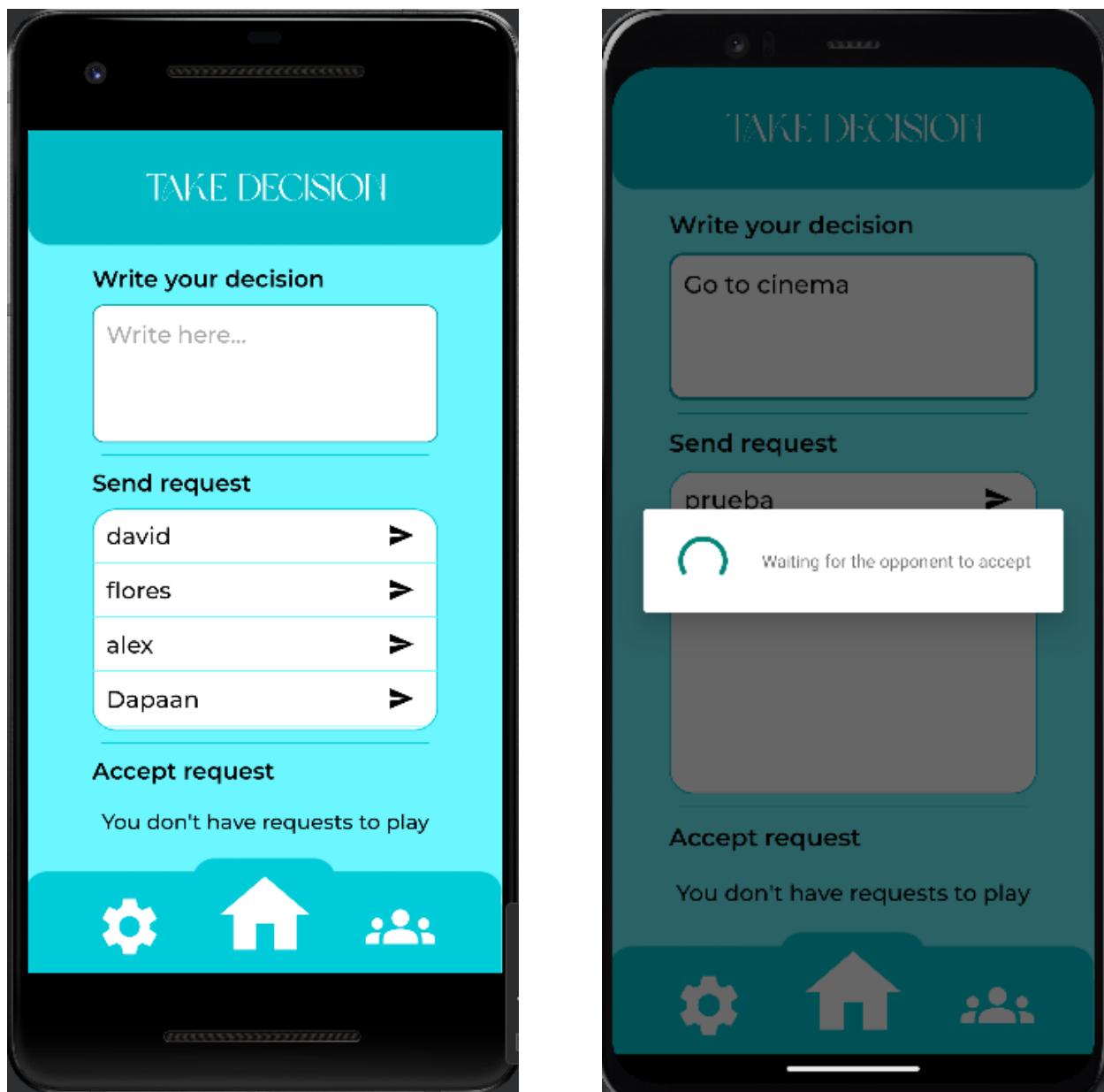
See my all my friend to sending request to play and see my requests of my friends to play. First write text on textbox decision and next send request or receive request to play.



while you are waiting the other user will be able to accept or reject the request if he accepts it the game will start

CONNECT WITH FRIEND

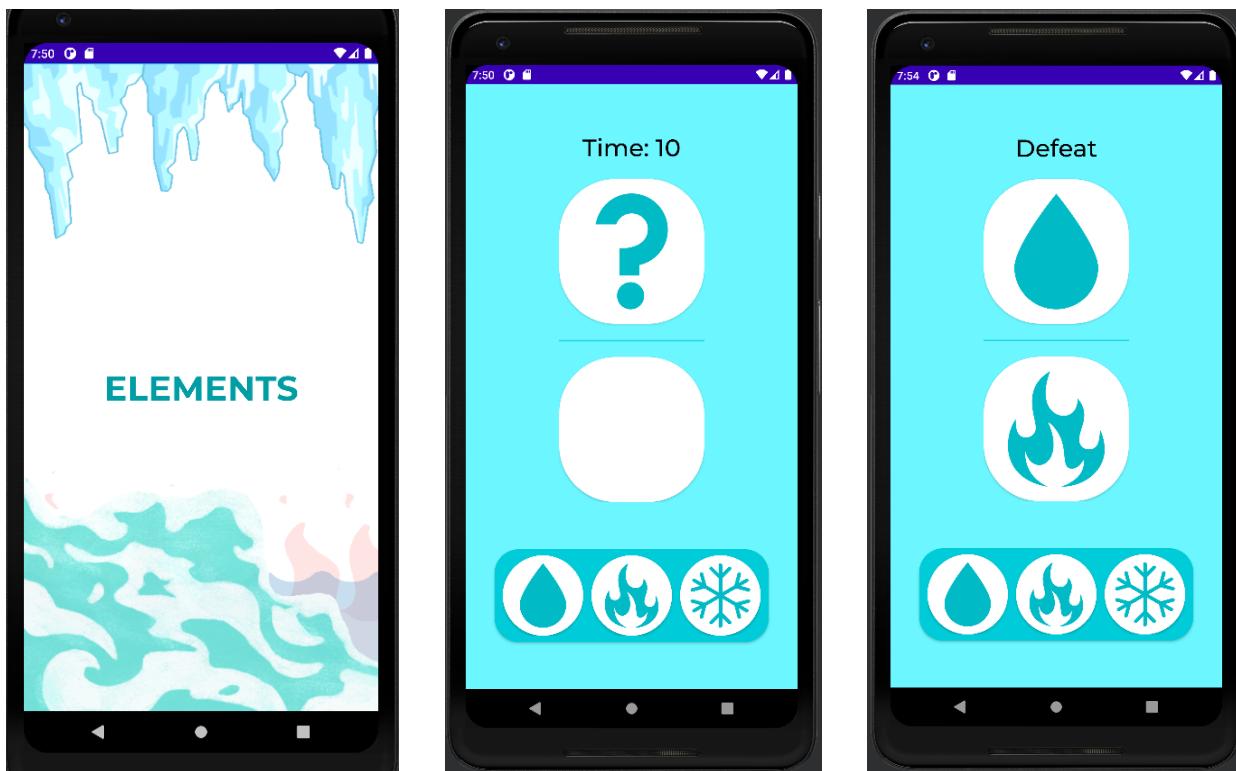
See my all my friend to sending request to play and see my requests of my friends to play. First write text on textbox decision and next send request or receive request to play.



while you are waiting the other user will be able to accept or reject the request if he accepts it the game will start

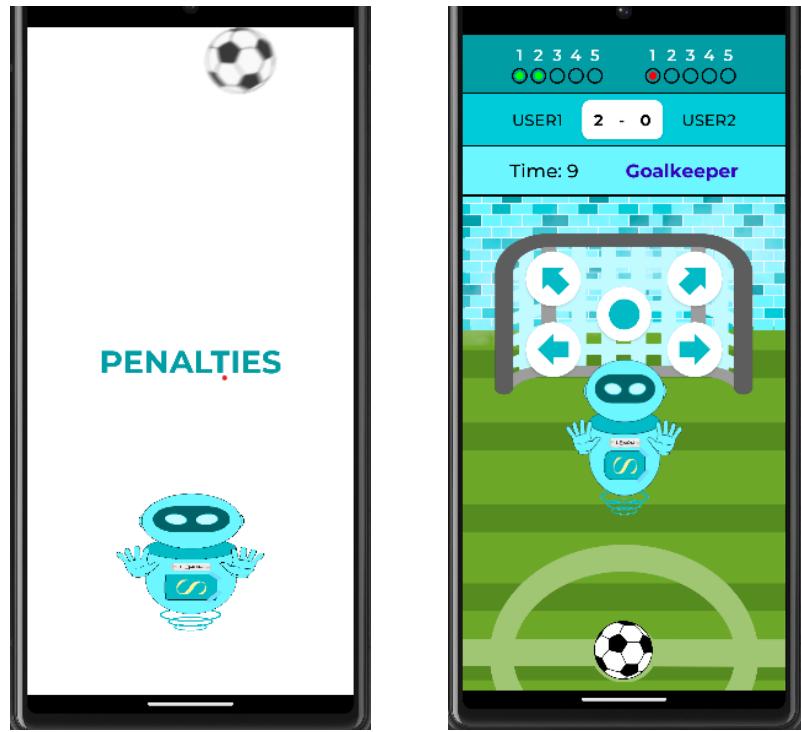
FIRST MINIGAME "ELEMENTS"

When we press play, our first minigame "Elements" will load, a small animation will jump to introduce the game and after that it starts to play, we will have 3 options to choose between water, fire and ice, water wins over fire, fire wins over ice , and the ice beats the water, if you are tied it starts again until there is a winner or loser, if the 15s time runs out your move will be made automatically, when you win or lose it will take you to a provisional layout where the winner can share later with a photo your decision.



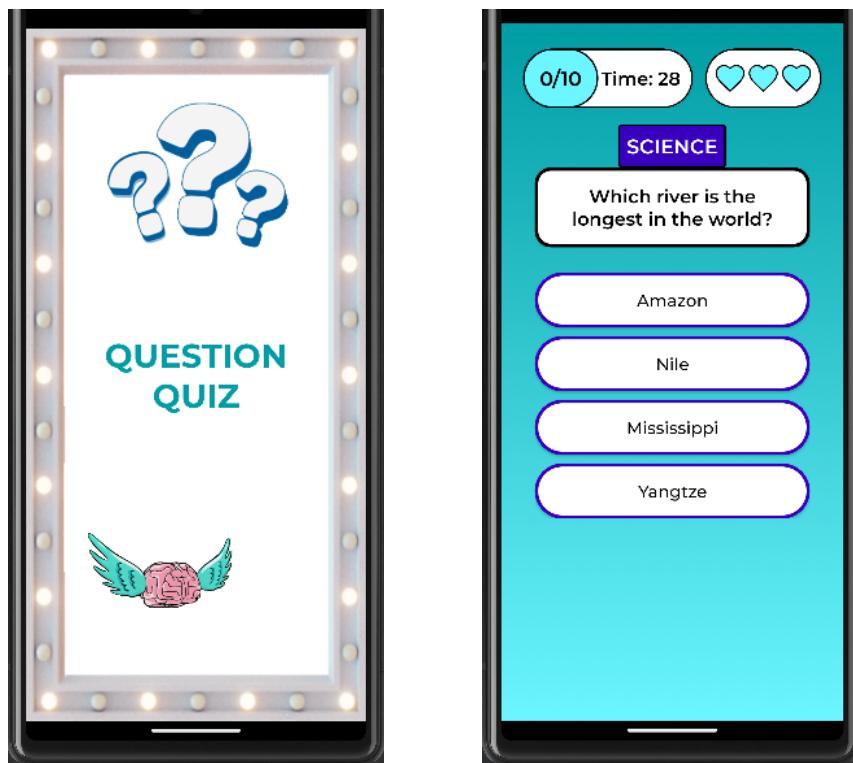
SECOND MINIGAME "PENALTIES"

When we press play, our second minigame "Penalties" will load, a small animation will jump to introduce the game and then it will start playing, where we will have 2 roles, one will be the goalkeeper, the other will be the one who shoots and will alternate in a round out of 10 if the 15 s time runs out, your move will be made automatically. when you win or lose it will take you to a provisional layout where the winner can share later with a photo your decision.



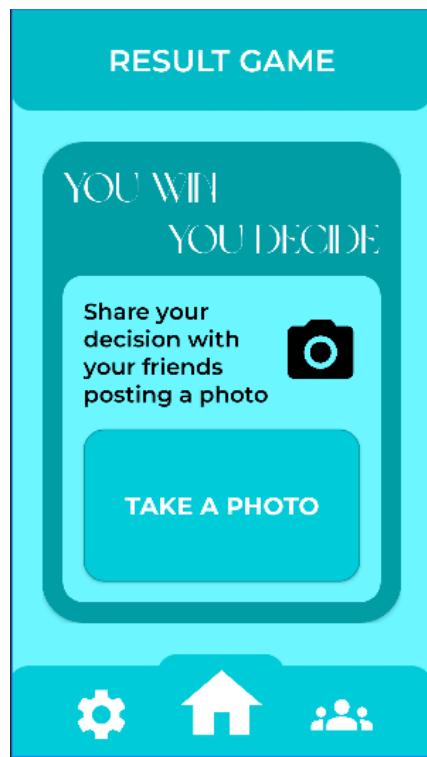
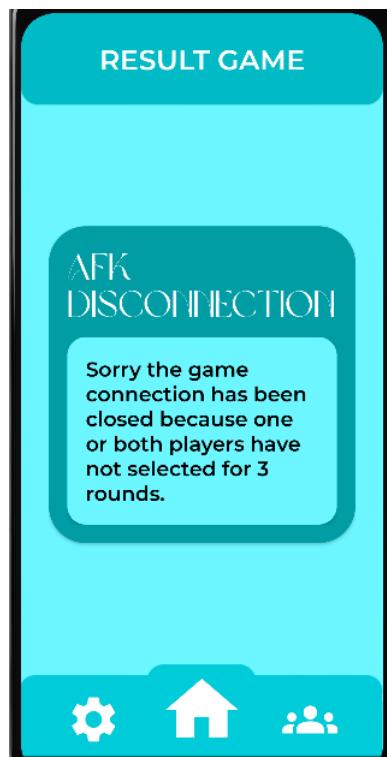
THIRD MINIGAME "QUESTION QUIZ"

When we press play, our third minigame "Question quiz" will load, a small animation will jump to introduce the game and then it will start playing, which consists of guessing the questions that come up randomly without losing all the lives that we have available in 30 s, when you win or lose it will take you to a provisional layout where the winner can share a photo with decision.



RESULTS LAYOUTS:

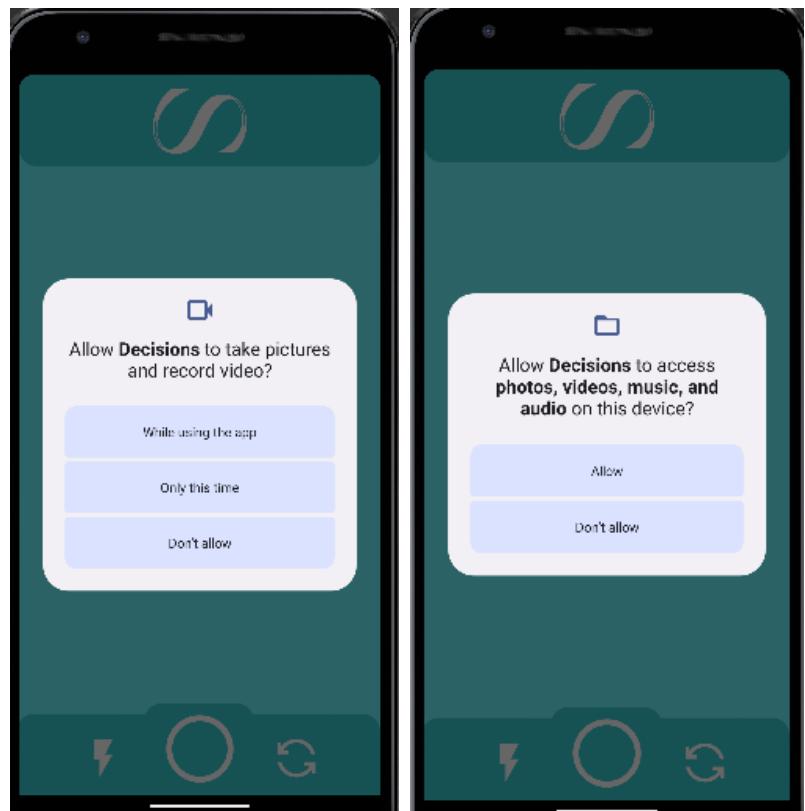
These will be the possible screens that will appear depending on whether you play alone or against a friend, there is also a screen for when the players are inactive for 3 rounds, the result afk.



CAMERA PERMISSIONS:

We check that it asks us for the corresponding permissions to be able to use the mobile camera and to be able to use the folders.

If the user does not give the permissions, they will not be able to share the photo.



CUSTOM CAMERA:

When it comes to taking photographs, it is important to ensure that the orientation of the camera and the photo are aligned properly. This can help to improve the overall quality and clarity of the image, and prevent errors or issues that can arise from misaligned photos. In order to achieve proper camera and photo orientation, it may be necessary to conduct multiple tests and adjustments. This can involve testing different camera angles, lighting conditions, and other factors that can affect the quality of the photo.

For example, when using an interior camera, it may be necessary to adjust the angle and positioning of the camera in order to capture the best possible image. Similarly, when using a flash, it may be necessary to adjust the camera settings in order to achieve the desired level of lighting and exposure.

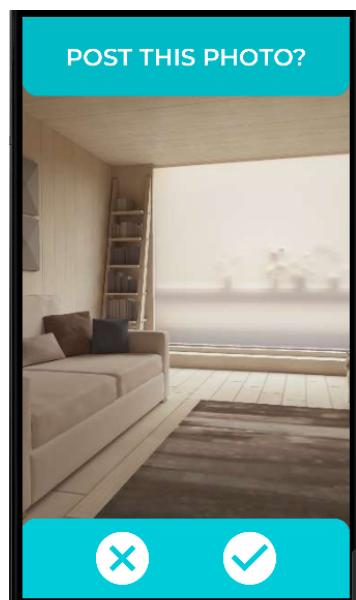
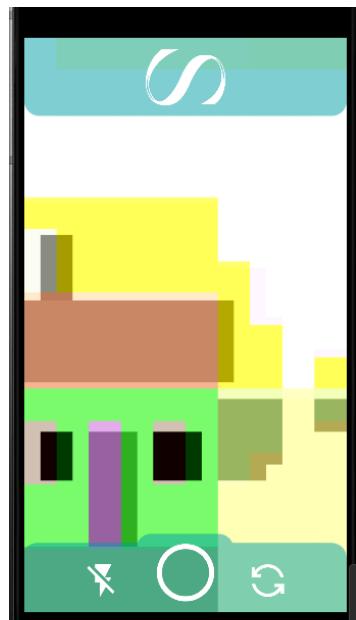
Overall, conducting multiple tests and adjustments can help to ensure that the camera and photo are aligned properly, and that the resulting image is of high quality and meets the desired specifications. While this process may take some time and effort, it is an important step in achieving a successful and visually appealing photograph.

RESULT PHOTO:

When it comes to user-generated content platforms, it is important to provide users with a preview of their uploaded content before publishing it. This can help to ensure that the content is of the desired quality, size, and orientation, and can prevent errors or issues that can arise from misaligned or poorly formatted content.

To facilitate this, a preview screen can be provided after the user takes a picture. This screen would display the image with the correct orientation, quality, and size, and would include two buttons to allow the user to confirm or deny the publication of the content.

The "Confirm" button would allow the user to proceed with publishing the content, while the "Deny" button would allow the user to return to the previous screen and take another picture. This can help to prevent accidental publishing of unwanted or poorly formatted content, and can provide users with greater control over their contributions to the platform.



MAIN SOCIAL SCREEN

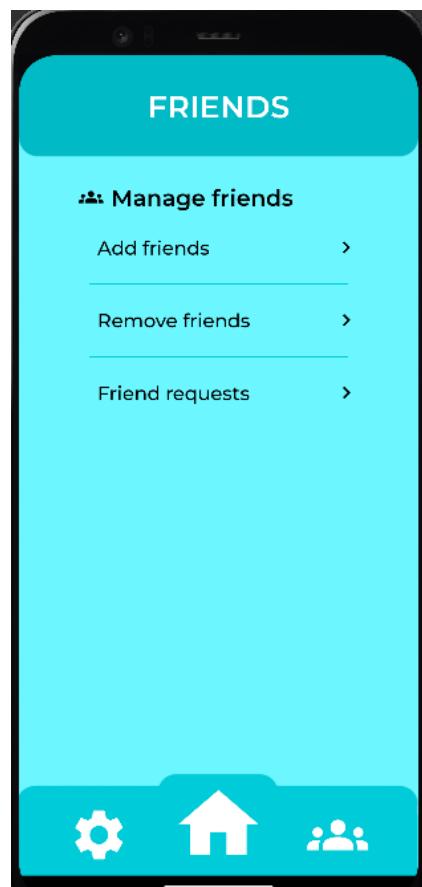
And when this photo was published, it should be seen first on the main screen of the game, which is to see your publications and those of your friends, sliding one by one with your finger.

And from the main social screen we can go to settings, choose mode or manage friends



MANAGEMENT FRIENDS

In friends management, the user has three options: add friends, which will take you to the screen to add friends and send a friend request; remove friends to remove friends, and friend requests to receive friend requests from other users.



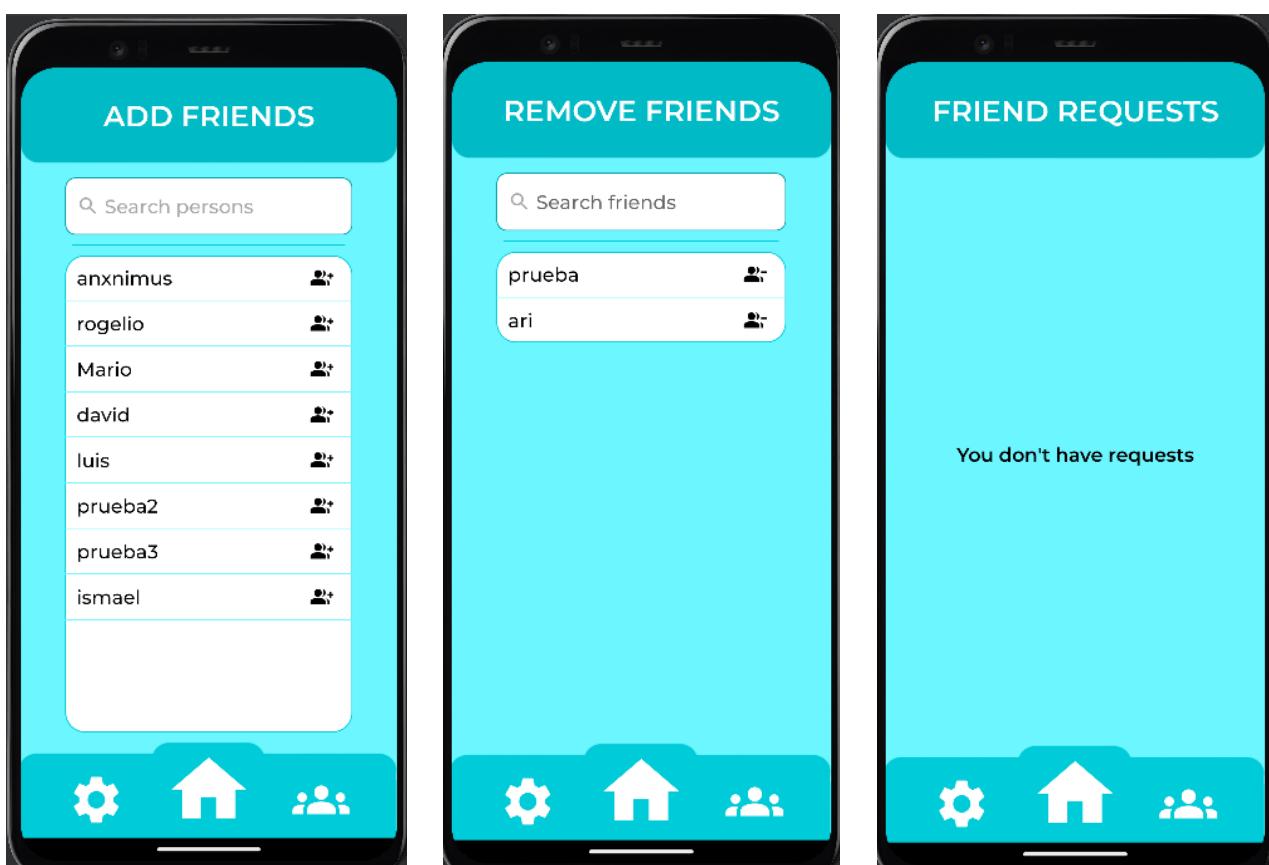
ADD FRIENDS/ REMOVE FRIENDS/ FRIEND REQUESTS

In Add friends the user can see 15 users among all the users that are registered, while he is filtering word by word in the search, it will show the first 15 that match what he writes, to send a request he will click on the username, a message will appear confirmation message and if it accepts, it sends a request and even if it continues to appear, it will not be able to send more requests to that user.

In remove friends the user can delete a friendship, a confirmation message appears and if he accepts, he deletes the friendship, but he can create it again if he adds it again.

In friend requests the user can accept or reject the requests that come from another user with a confirmation message.

In the event that you do not have friends or do not have friend requests, a message will appear on the screen.



Description of the project methodology used

For this project we used the methodology SCRUM consisting of an agile project management framework that is widely used in software development and other complex projects. It emphasizes flexibility, collaboration, and iterative progress to deliver high-quality products.

Key components of the Scrum include:

1. Product Backlog: A prioritized list of features, enhancements, and bug fixes that need to be addressed in the project. The Product Owner maintains the backlog and continuously refines it based on feedback and changing requirements.
2. Sprint: A time-boxed iteration, in our case 1 or 2 weeks, during which a set of product backlog items is selected for development. Each sprint has a specific goal and results in a potentially shippable product increment.
3. Sprint Planning: At the beginning of each sprint, the Scrum Team collaboratively selects a subset of items from the product backlog to work on. They define the sprint goal and create a sprint backlog, which contains the tasks required to complete the selected items. (In our case we have done 5 sprints)
4. Daily Scrum: A brief daily meeting where team members synchronize their work. Each team member provides updates on progress, discusses any obstacles, and plans their work for the day. The focus is on coordination and identifying potential issues.
5. Sprint Review: At the end of each sprint, the Scrum Team showcases the completed work to stakeholders and gathers feedback. The Product Owner reviews the progress toward project goals and adjusts the product backlog as needed.
6. Sprint Retrospective: A meeting held after the sprint review, where the Scrum Team reflects on their work and identifies ways to improve the team's effectiveness. They discuss what went well, what could be improved, and define action items for the next sprint.

Planning

Sprint 1

Priority	User Story	Assigned to
Critical	Register in the App	Isma/Yang-David
Critical	Login in the App	Isma/Yang-David
Critical	See menu in main interface	David
Normal	Entity Relation	David
Normal	Database	Isma/Yang

Sprint 2

Priority	User Story	Assigned to
Critical	Add Friend	Isma/Yang
Normal	Configuration in the app	Isma/Yang
Minor	Main interface with tutorial	David
Minor	Close app	Isma/Yang
Normal	Change language	David
Normal	Keep sign-in	David
Major	Log out	David/Isma
Normal	Minigame "Elements" Offline	David/Yang

Sprint 3

Priority	User Story	Assigned to
Critical	Finish Add Friend	Isma/Yang
Critical	Finish Configuration in the app	Isma/Yang
Major	Connect with friend before playing	Isma/Yang
Normal	Add Main interface with tutorial	David
Major	Minigame "Penalties"	David
Major	Minigame "QuestionQuiz"	David
Major	Remove Friend	Isma/Yang
Major	Play random minigame with friend	David
Normal	Write decision before playing	David

Sprint 4

Priority	User Story	Assigned to
Major	Connect with friend before playing	Yang/David
Normal	Recovery password	David
Normal	Share your decision	Isma/David
Major	Play random minigame with friend	Yang
Major	Write decision before playing	David
Normal	Recover friendship	Yang/Ismael

Sprint 5

Priority	User Story	Assigned to
Major	Recovery password	Yang
Major	See all decisions in the main interface	Ismael
Critical	Play random minigame with friend	David
Critical	Write decision before playing	Ismael
Critical	Fix register/login	Yang
Minor	Main interface with tutorial	David

Our product backlog:

Critical

Register in the App
Login in the App
See menu in main interface

Major

Play random minigame with friend
Write decision before playing
Connect with friend before playing
Minigame "Question quiz"
Administrator Database
Log out

Normal

See all decisions in the main interface
Add Friend
Recover friendship
Configuration in the app
Share your decision
Minigame "Penaltis"
Minigame "Elements"
Minigame "CowBoy"
Recovery password
Entity Relation
Database
Remove Friend
Change language
Keep sign-in

Minor

Main interface with tutorial
Close app

Individualized planning

David Chaparro:

All layouts of the app and drawable resources
Animations of the app
Image swipe on social main interface
Minigame "Elements" offline
Minigame "Penalties" offline / online
Minigame "Question quiz" offline
Custom camera and result photo
Games guide
Change language(And translate all strings)
Keep sign in
Log out
Choose modality
Random minigame offline
Creative commons license
Server and network errors
Documentation

Ismael Maridueña:

Recovery password
Configure de server Ubuntu install
Upload image in the database
Show image in the app
Login in client
Register in client
Change password
Change email
Change username
Send email
Show users in client
Show friend request in client
Show friend list in client
Remove friend
Documentation

Zhi Yang Wu:

Documentation

Create penalties online

Create Quiz online

Create elements online

Play online activity(functions)

Create register (server)

Create login (server)

Switch username (server)

General functions to get data(id, username, etc) as

example: getUserById

Show friends (server)

Remove friend(server)

Accept friend request(server)

List of users(server)

Show friend requests(server)

Create methods for when user exists of a online minigame

Random minigame online

Innovation

Online

We never made an online game, so it was quite a challenge, first we had to document a lot about them, search websites, watch videos, etc, and finally we found what we wanted and the realtime database of FireBase offered it to us.

First of all we had to create the rooms where the players would appear and there the decisions they have made or rather to know if which option each player has selected and thus compare them and be able to show who is the winner and know who it was, since if one wins or in theory meets the requirements to do so, he will put it in the database and by doing this the other will know that his rival has won and vice versa, since, so to speak, a listener is created that if at any time a value of the database changes, it detects it and will execute the code that we indicate.

Camera

For the camera of the application we did not want it to be the default camera of the mobile so we investigated how to make a custom camera and we found a good tutorial to make the camera although there were many problems with the orientation of the photo, the flash that the icons were on the exact moment that if you change the camera to the front, the orientation is fine and we have finally achieved it.

Layouts and animation

From the beginning we wanted to implement animations in the layouts to load some classes, for example the games, so I saw that for this the MotionLayout is used, which works a bit slow but it is possible to animate and change positions of objects, put times and positions so that we can do cool transitions. We have also innovated in many layouts for example using ScrollView giving different styles to the buttons to the edit text to make them more beautiful and visual.

Mail sender

We have implemented the mail sender that is used to send emails, we have implemented it when you register that sends you a welcome message, and when you have to recover the password where it sends a one-time code. The Mail Sender is a tool that allows you to send emails automatically from an application or system. It provides a convenient way to send email messages to users, such as welcome emails, notifications, updates, or any other type of email communication.

User manual

Login:

Introduction: Login is a feature that allows users to access the application by providing their login credentials. The user is required to enter a valid username and password to access the application. The purpose of this manual is to guide users on how to log into the application using Login.

1. Open the application.
2. The login screen will appear, prompting you to enter your username and password.
3. Enter your valid username and password in the respective fields.
4. Click the "Enter" button.
5. The system will verify your login information.
6. If the username and password are valid, you will be logged into the application and directed to the main screen of the application.
7. If the username and password are invalid, an error message will be displayed, and you will be prompted to re-enter your login information.
8. If you do not have an account, you can click the "Registration" button to be directed to the registration window.
9. If you forgot your password you can acces to Recover your password.
10. If you don't have connection network you can't acces.

Register:

- Step 1: Open the Application First, open the application on your device.
- Step 2: Navigate to Registration Window If you do not have an account, you can click on the "Registration" button on the login screen. This will take you to the registration window.
- Step 3: Enter Your Information Once you are on the registration window, enter your username, password, and email address in the respective fields. Make sure to enter valid information.
- Step 4: Click on "Enter" Button After you have entered your information, click on the "Enter" button to register your account.
- Step 5: Verify Your Information The system will verify the information you entered. If the information you entered is valid, you will be successfully registered and directed to the login screen.
- Step 6: Error Message for Invalid Information If any of the information you entered is invalid, an error message will be displayed. You will be prompted to re-enter your registration information.
- Step 7: Return to Login Screen If you want to return to the login screen without registering, click on the "Login" button.
- Step 8: Information Saved Once you have successfully registered, your information will be saved in the system's database.

Configuration menu:

The Configuration menu is displayed when the user clicks on the settings icon. It contains buttons for editing your profile, changing your password, logging out, changing the language, and accessing a games guide. If you are an administrator, you will have additional options available to you.

- **Editing Your Profile:** To edit your profile information, click on the "Edit profile" button in the Configuration menu. This will take you to a page where you can modify your personal information. Once you have made the desired changes, click the "Save" button to update your profile.
- **Changing Your Password:** To change your password, click on the "Change password" button in the Configuration menu. This will take you to a page where you can enter your current password and your new password. Once you have entered your new password, click the "Save" button to update your login credentials.
- **Logging Out:** To log out of the system, click on the "Log out" button in the Configuration menu. This will exit you from the system and return you to the login page.
- **Changing the Language:** To change the language of the system, click on the "Change language" button in the Configuration menu. This will display a list of available languages. Click on the desired language to change the language of the system.
- **Accessing the Games Guide:** To access information about the available games, click on the "Games guide" button in the Configuration menu. This will take you to a page where you can browse information about the available games.
- In about decision we found creators information and creative commons license.

Change language:

1. **Log in to the system:** To access the Change Language feature, you must first log in to the system with your registered account.
2. **Access the settings icon:** Once you are logged in, click on the settings icon located in the top right corner of the screen.
3. **Access the Change Language feature:** From the Configuration menu, click on the "Change Language" button.
4. **Select your preferred language:** The language menu will be displayed with options for English, Spanish, Catalan, Chinese, German, Hindi, and Portuguese. Select your preferred language from the menu.
5. **Update the language settings:** The system will update the language settings accordingly and display all system text and messages in your selected language.
6. **Navigate the system in your preferred language:** Once you have selected your preferred language, you can navigate the system in your chosen language with ease.

Log out:

- 1.The user logs into the system.
- 2.The user clicks on the settings icon.
- 3.The CDU.4-Configuration menu is displayed.
- 4.The user clicks on the "Log out" button.
- 5.The system logs the user out and returns them to the login interface.
- 6.The user is no longer able to access any system features without logging in again.

Minigame "Elements":

Introduction minigame "Elements" feature allows you to play a game called "Elements". In this game, you will have three options to choose between water, fire and ice. Water wins over fire, fire wins over ice, and ice beats the water. If you are tied, the game starts again until there is a winner or loser. If the 15s time runs out your move will be made automatically, and when you win or lose, it will take you to a provisional layout where you can share later with a photo of your decision.

Getting Started To access the Minigame "Elements" feature, you need to register and log in to the system first. Once you are logged in, click on the "take decisions" icon, and then click on "Play" to start the game.

Game Interface When the game starts, you will see the introduction animation. Then, you will need to select your move from three options: water, fire, or ice. The system will calculate the winner based on the game rules. If there is a tie, the game starts again until there is a winner or loser. If you do not make a move within 15 seconds, the system automatically makes a move for you.

Provisional Layout When there is a winner or loser, the system will display a provisional layout where the winner can share their decision with a photo. You can choose to play again or return to the "take decisions" interface.

Rules

Water beats fire

Fire beats ice

Ice beats water

Minigame "Penaltis":

Step 1: Login To play the minigame, you need to log into the system with your registered account.

Step 2: Access "Take Decisions" Icon After logging in, click on the "Take Decisions" icon to access the list of available minigames.

Step 3: Select "Penalties" Minigame From the list of minigames, select the "Penalties" minigame and click on it.

Step 4: Game Interface The game interface for "Penalties" will be displayed. You will see an introduction animation that will explain the rules of the game.

Step 5: Choose Your Move You will have two roles in this game - goalkeeper and shooter. Depending on your role, you will have to choose your move from the following options: Center /Top right / Top left / Down right / Down left

Step 6: Game Rules The game consists of 10 rounds. If there is no tie after 10 rounds, the player with the most goals wins. If there is a tie, the game goes to sudden death, where the first player to miss a goal loses.

Step 7: Time Limit You have 5 seconds to make your move. If you don't make a move within 15 seconds, the system will automatically make a move for you.

Step 8: Winner or Loser When there is a winner or loser, the system will display a provisional layout where the winner can share their decision with a photo.

Minigame "Question quiz":

Open the game and ensure that you have a stable internet connection.

Register and log in correctly to access the game.

Click on the "take decisions" icon to select either "IvsI" or "IvsRobot" mode and click "Play".

Wait for the intro animation to play without any technical issues.

The game begins, and you will be presented with a random question from a selected category.

You will have 30 seconds to select one of the four answer options.

If you select the correct answer, you move on to the next question. Otherwise, you lose one of your three lives.

If you run out of lives, the game ends.

If you answer all the questions correctly within the given time limit, you win.

If you choose to play online against another user, the first player to answer all the questions correctly wins.

Rules:

The game consists of answering 5 to 10 random questions about different categories of general culture. Each question has four options, and only one is correct. There is a 30-second timer per question. If it ends, you lose a life. You have three lives. If you run out, you lose. To win the game, you must answer all the questions correctly within the given time limit. If you choose to play online against another user, the first player to answer all the questions correctly wins.

Add friends:

Overview

The Add Friends feature allows users to connect with their friends on the platform. With this feature, users can send friend requests and accept them, allowing them to see each other's profiles and activity updates.

Prerequisites

Before using the Add Friends feature, make sure you have registered and logged in correctly. Also, ensure that you have clicked on the Friends icon and then clicked on the Add Friends button.

Steps to Add Friends

Click on the Friends icon: When you log in to your account, you will see a Friends icon. Click on it to open the Friends section.

Click on "Add Friends": Within the Friends section, click on the "Add Friends" button to open the Add Friends page.

Enter the username: Type the username of the person you want to add as a friend in the search box.

Click a username: Once you have entered the username, click on the Confirm button to send a friend request.

Wait for the person to accept: The person who received the friend request will receive a notification. If they accept the request, you and the new friend will be added to each other's friends list.

Remove friends:

Overview

The Remove Friends feature allows users to remove friends from their friend list on the platform. With this feature, users can remove friends they no longer want to connect with or have lost touch with.

Prerequisites

Before using the Remove Friends feature, make sure you have registered and logged in correctly. Also, ensure that you have added friends to your friend list.

Steps to Remove Friends

1. **Click on the Friends icon:** When you log in to your account, you will see a Friends icon. Click on it to open the Friends section.
2. **Select the friend you want to remove:** Within the Friends section, select the friend you want to remove from your friend list.
3. **Click on "Confirm":** Once you have selected the friend, click on the "Confirm" button to remove friend.
4. **Friend removed:** The system will remove the friend from your friend list, and you will no longer be able to see their publications and play with him.

Installation Manual

Fisical requirements:

Server:

minimum requirements

6RAM, 50GB space(in the final progres we need 20Tera for aprox 1000 users), image ubuntu 22.04

<http://somebooks.es/instalar-ubuntu-22-04-lts-jammy-jellyfish-desde-cero/>

Web Service:

Versions:**

MySql:** 8.0

To install MySQL, you can follow the steps below:

1. Open the terminal of your Linux operating system.
2. Make sure your operating system has internet access.
3. Update existing packages on your system with the following command:

```
sudo apt-get update
```

4. Once the packages have been updated, run the following command to install MySQL:

```
sudo apt-get install mysql-server
```

5. Press "y" when prompted to confirm the installation.
6. During the installation, you will be asked to set a password for the MySQL administrator account. Enter a strong password and write it down somewhere safe.

7. Once the installation is complete, you can start the MySQL service with the following command:

```
sudo service mysql start
```

8. To verify that MySQL is working correctly, you can use the following command to enter the MySQL shell:

```
sudo mysql -u root -p
```

9. You will be prompted for the password you set during installation. Enter it to enter the MySQL shell.

10. You are now inside the MySQL shell and can start creating and managing databases.

PERSONAL MODIFICATION FOR THE SERVER OR SELF CONFORT

To change the root password of MySQL via terminal, follow these steps:

1. Log in to the terminal with an account that has superuser (root) permissions.

2. Access the MySQL server by typing the following command:

```
mysql -u admin -p
```

This command will prompt you for the current root password of MySQL. Enter it to log in to the MySQL console.

3. Once inside the MySQL console, type the following command to change the root password: (same password)

```
ALTER USER 'admin'@'localhost' IDENTIFIED WITH mysql_native_password BY 'Admin123?';
```

4. Replace "new-password" with the password you want to assign to the root account.

5. Exit the MySQL console by typing the following command:

```
exit;
```

Once the installation of mysql is finished with their respective configuration, now we go to the configuration of the host machine to be able to access our server. In our case, we will use putty, although it could also be accessed through the windows cmd by ssh user@ip_public, too we will use filezilla to be able to upload files.

We need to have the following programs installed on our host machine:

Manual for the instalation for android studio and all specifications:
(last version): [Manual install android studio](#)

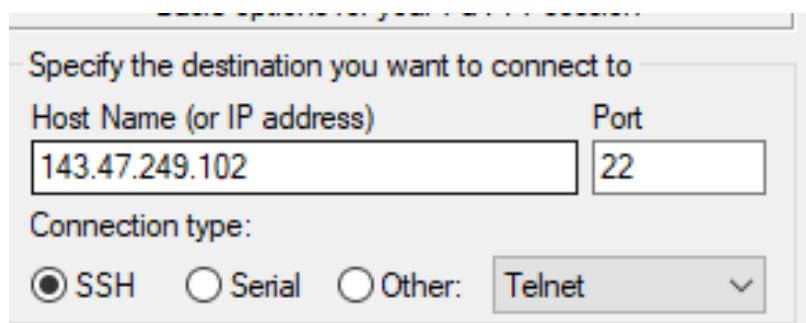
Link to install android studio: [android Studio install](#)

Link to install putty: (last version) version 0.78: [Putty](#)

Link to install filezilla: (last version) version 1.6.7: [Filezilla](#)

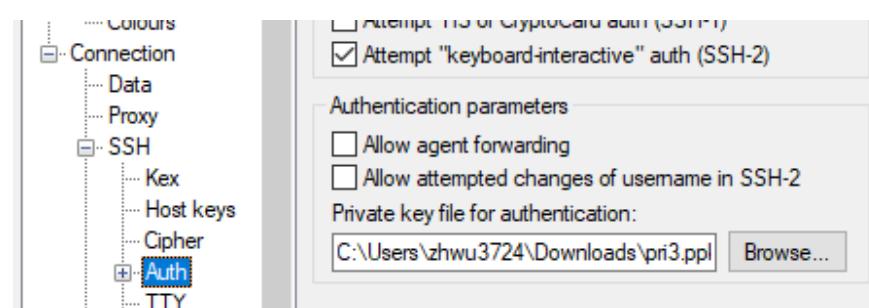
Configure putty:

In section session de putty ponemos en host name la Ip of the server: 143.47.249.102 and port:22 and Conection type ssh

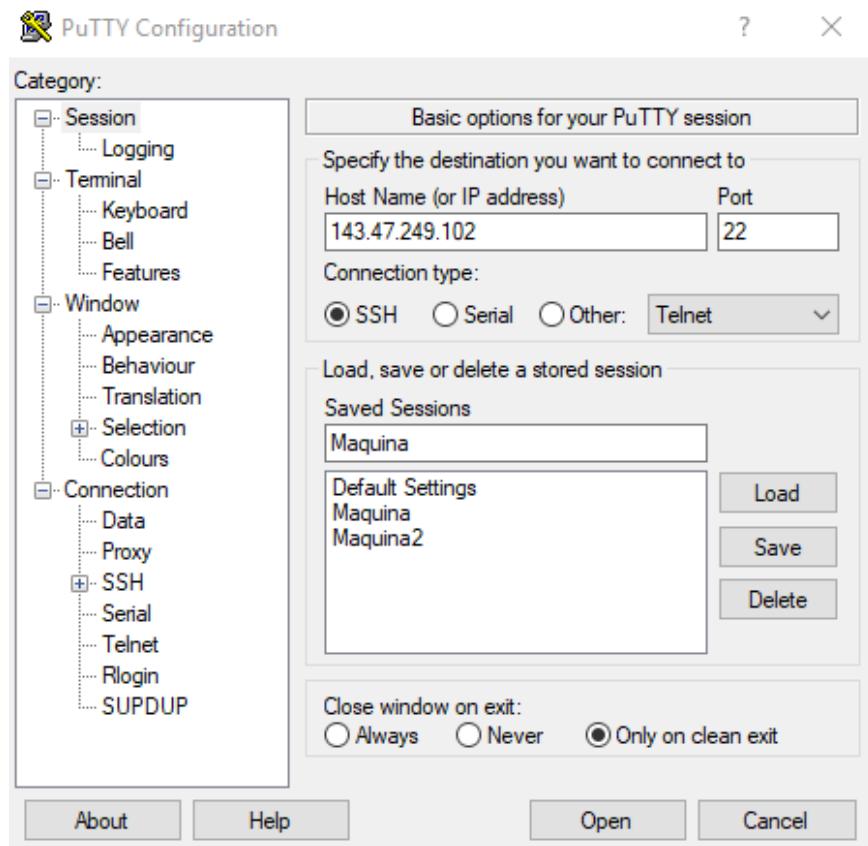


And contact to a internal to send u the private key of the server:

Go SSH and auth, press browser and search the private key for add it



Once the private key has been added, we return to the session panel to access our server and click on open.



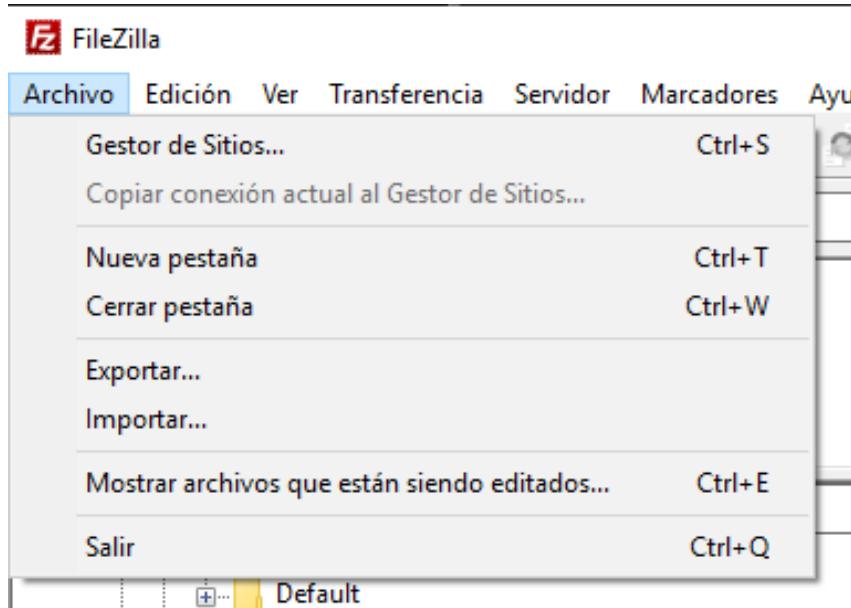
After giving it open, it will ask us to enter our username, which in this case is ubuntu, we will put it, it will also verify that the private key that we have given is correct and if it is correct, we will access our server.

A screenshot of a terminal window. The title bar says 'ubuntu@decisions: ~'. The window content shows a successful SSH login to an Ubuntu 20.04.6 LTS system. The output is:
login as: ubuntu
Authenticating with public key "imported-openssh-key"
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-1032-oracle aarch64)

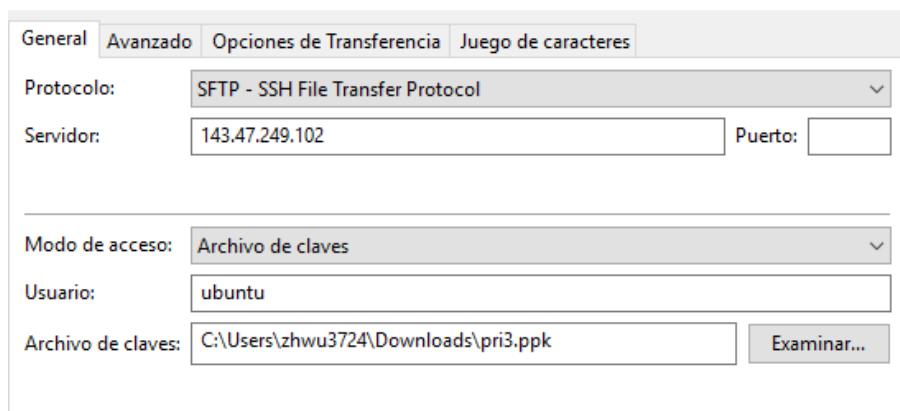
Now we will go on to configure Filezilla to be able to upload files to our server if necessary, which in our case we need to upload a java jar.

Configure Filezilla:

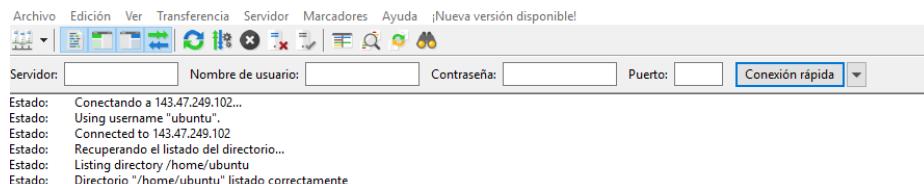
We run filezilla, once open we go to files and look for the site manager option and access it.



Once we access it, we create a new site, we configure the protocol in ssh File Transfer Protocol, in the server we put our public ip, the port 22 or we leave it blank to take it by default, in the access mode we put key files , in user the user of our server that in our case is ubuntu, in key files we give it to examine and we put the private key and we give it to connect once verified that everything is correct.



And if everything is correct we can access our server from filezilla to upload files.



Keep server started:

To keep the server started we have created a service within Linux called javalin.service that has this structure within the path /etc/systemd/system

```
root@proven-dam-3:/etc/systemd/system# ls
cloud-init.target.wants           mdmonitor.service.wants      sleep.target.wants
dbus-org.freedesktop.resolve1.service ModemManager
dbus-org.freedesktop.timesync1.service multipath-tools.service
display-manager.service.wants       multi-user.target.wants
emergency.target.wants            network-online.target.wants
getty.target.wants                 oem-config.service.wants
graphical.target.wants            open-vm-tools.service.requires
iscsi.service                      paths.target.wants
javalin.service                   rescue.target.wants
                                     _links
                                     _symlinks
```

This is the configuration that we have done for the javalin.service server where we specify the name that the service will have, the user that has access to the service, how it will be executed and its jar path so that it is executed when the PC is restarted, we also indicate it the directory path where the .jar is located

```
GNU nano 6.2                               javalin.service
[Unit]
Description=Javalin Service
After=network.target

[Service]
User=root
ExecStart=/usr/bin/java -jar /root/Decisions/JavalinWS-1.0-SNAPSHOT-jar-with-dependencies.jar
WorkingDirectory=/root/Decisions/

[Install]
WantedBy=multi-user.target
```

To start it up we just have to do these commands:

to enable the service:

```
root@proven-dam-3:/etc/systemd/system# systemctl enable javalin.service
```

to disable

```
root@proven-dam-3:/etc/systemd/system# systemctl disable javalin.service
```

to start the service

```
root@proven-dam-3:/etc/systemd/system# systemctl start javalin.service
```

to restart the service and apply the changes done

```
root@proven-dam-3:/etc/systemd/system# systemctl restart javalin.service
```

to stop the service

```
root@proven-dam-3:/etc/systemd/system# systemctl stop javalin.service
```

to see the status of the service

```
root@proven-dam-3:/etc/systemd/system# systemctl status javalin.service
```

Database:

This is our script to execute in mysql or in phpmyadmin:

```
//Create the user admin for the manage de database
CREATE USER 'admin'@'localhost' IDENTIFIED BY 'Admin123?';

//Login as user 'admin' with password 'Admin123?' mysql -u admin -p

//Create the database
CREATE DATABASE decisionsdb;

//Give the corresponding permissions to the created user for the database
GRANT ALL PRIVILEGES ON decisionsdb.* TO 'admin'@'localhost';

//Use the database
USE DATABASE decisionsdb;

//Create table users
CREATE TABLE users (
id INT PRIMARY KEY AUTO_INCREMENT,
username VARCHAR(50) NOT NULL,
password VARCHAR(255) NOT NULL,
email VARCHAR(255) NOT NULL,
token VARCHAR(255)NOT NULL
);

//Create table game
CREATE TABLE game (
id INT PRIMARY KEY AUTO_INCREMENT,
photopath longblob,
winner VARCHAR(255) NOT NULL,
decision VARCHAR(255) NOT NULL,
day date NOT NULL,
friendID INT NOT NULL
);

//Create table friend_list
CREATE TABLE friend_list (
user1_id int NOT NULL,
user2_id int NOT NULL,
status enum('accepted','pending','rejected','') NOT NULL,
PRIMARY KEY (user1_id,user2_id),
UNIQUE KEY unique_users (user1_id, user2_id),
KEY fk_user2_id (user2_id),
CONSTRAINT friend_list_ibfk_1 FOREIGN KEY (user1_id) REFERENCES users (id),
CONSTRAINT friend_list_ibfk_2 FOREIGN KEY (user2_id) REFERENCES users (id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3;
```

Configuration Javalin in Netbeans

To prepare Javalin, you can follow the steps below: Plug-in assembly installation: Open the pom.xml file of your Maven project.

1. Add the following settings in the pom.xml file:

```
<build>
    <plugins>
        <plugin>
            <artifactId>maven-assembly-plugin</artifactId>
            <version>3.5.0</version>
            <configuration>
                <descriptorRefs>
                    <descriptorRef>jar-with-dependencies</descriptorRef>
                </descriptorRefs>
                <archive>
                    <manifest>
                        <addClasspath>true</addClasspath>
                        <classpathPrefix>lib</classpathPrefix>
                    </manifest>
                </archive>
            </configuration>
            <executions>
                <execution>
                    <id>make-assembly</id> <!-- this is used for inheritance merges -->
                    <phase>package</phase> <!-- bind to the packaging phase -->
                    <goals>
                        <goal>single</goal>
                    </goals>
                </execution>
            </executions>
        </plugin>
    </plugins>
</build>
```

This configuration defines the maven-assembly-plugin with version 3.3.0. Also, set the plugin configuration to include the jar-with-dependencies descriptor, which includes all of your project's dependencies in a single JAR file.

2. Save the pom.xml file.

3. Run the following command in the terminal to create the JAR file with all dependencies included:

```
mvn clean package
```

This command cleans up the project, compiles the source code, and creates the JAR file with all dependencies included.

4. The resulting JAR file will be located in the project's target folder.

HTTP Request Manager (Javalin version 3.3.0):

In short, Javalin is a lightweight, easy-to-use web framework for Java and Kotlin that provides an intuitive and expressive API for building high-quality web applications and RESTful APIs quickly and easily.

Among the features and benefits that Javalin offers are:

- **Lightweight and easy to use:** Javalin is a very lightweight web framework with a simple and easy-to-use API that can be learned quickly.
- **Focus on simplicity:** Javalin is designed to be simple and straightforward, allowing developers to focus on building their web applications and APIs without worrying about technical details.
- **Supports multiple languages:** Javalin can be used with either Java or Kotlin, allowing developers to choose the language that best suits their needs.
- **Routing and request handling:** Javalin offers an intuitive and flexible routing API that allows developers to handle HTTP and RESTful API requests efficiently.
- **Supports JSON and WebSockets:** Javalin supports the JSON format for data exchange and WebSockets for real-time communication.

- **Integration with other frameworks:** Javalin can be integrated with other frameworks and libraries, allowing developers to use other tools they already know and love.
- **Support for Servlets and Jetty:** Javalin can be run in servlet containers such as Jetty and Tomcat, making it very flexible and easy to use.

With the previous plugin we will generate a snapshot file and a snapshot with dependencies, the one that interests us is the one with the dependencies, then, once compiled, we send it to the server that we have through ftp through Filezilla that we already have configured previously.

Once the jar file has been passed to our server, we execute the following command to start it:

```
sudo java -jar JavalinWS-1.0-SNAPSHOT-jar-with-dependencies.jar
```

Android Studio Configuration:

Android Studio dependencies used:

- OkHTTP version: 4.8.1
- OkHttp is an open source HTTP client library for Java and Android that is used to make HTTP and HTTPS requests in Java and Android applications.
- OkHttp is used to simplify communication between a client application and a server exposing web services. It provides an easy-to-use interface for sending HTTP requests and receiving HTTP responses, handling cookies and authentication, and managing HTTP connections.

In addition to being easy to use, OkHttp has a number of features and benefits, such as:

- HTTP/1.1 and HTTP/2 compatibility.
- Persistent and non-blocking connections.
- Support for routing, interception and transcoding of content.
- Supports gzip and brotli content compression.
- Supports setting timeout and retries.
- Supports basic and form authentication.
- Supports self-signed certificates and custom certificates.

In short, OkHttp is a high-performance, easy-to-use, and highly customizable HTTP library used in Java and Android applications to send HTTP requests and receive HTTP responses.

All the dependencies that we have :

```
dependencies {
```

```
...
```

```
    implementation 'com.squareup.okhttp3:okhttp:4.9.1'
```

```
    implementation 'com.squareup.okhttp3:okhttp:4.9.3'
```

```
        implementation 'com.squareup.okhttp3:okhttp-  
urlconnection:4.9.3'
```

```
        implementation 'com.squareup.okhttp3:logging-interceptor:4.9.3'
```

```
        implementation 'com.squareup.okhttp3:okhttp-multipart:4.9.3'
```

```
...
```

```
}
```

Open ports:

To open the ports we execute the following command in the terminal of our server.

```
ufw allow "insert port"
```

- 7070 (Javalin)
- 22(ssh)

Conclusions

-Evaluate project goals

Finally we finished 90% of the promised user stories, if we had organized ourselves better and had not produced so many errors we would have done better.

-Identify difficulties along the project and how they have been managed.

At first we were going to use godot for our project but we finally decided on android since we found many incompatibilities and Android Studio knew more.

There were also days when a team member was sick and we had some delay with some user stories.

-Explain planning deviations (specially focus on user stories not finalized after each sprint)

We do not finalize statistics and some functionality of the publication, like show usernames on publication and fix more bugs of games online by time.

-Identify product evolution. Mention new user stories interesting to implement and justify the choice.

First we did login and register, then we implemented the layouts connection, we managed friends management, settings, games, results, online games and photo publication.

-Describe learnings obtained along the project

In our app project, we use Firebase specifically to take advantage of its Firebase Realtime Database functionality in our three online games. Firebase Realtime Database will store and synchronize the data in real time between the users who participate in the games. This ensured a seamless and collaborative gaming experience, where players could interact with each other and see updates in real time.

Android Studio served as our primary integrated development environment (IDE) for building the Android application. We were already familiar with the basics of Android development, as we had learned them during our course. However, throughout the project, we had to acquire additional knowledge, particularly in areas such as animations, mail sending, and handling network requests.

Javalin played a crucial role in creating the web service for our app. It is a lightweight framework for building RESTful APIs in Java. We chose Javalin because of its simplicity and ease of use. By utilizing Javalin, we were able to develop a robust backend for our app, enabling communication between the app and the server.

Learning these technologies was a continuous process throughout the project. We relied on various resources, such as online tutorials, documentation, and community forums, to expand our knowledge. We also experimented with code samples and conducted trial and error to understand the intricacies of each technology.

Overall, the combination of Android Studio, Javalin, and Firebase enabled us to create a dynamic and feature-rich app. The process of learning and implementing these technologies not only enhanced our technical skills but also provided us with a deeper understanding of app development and web services.