

# **ENS 491-492 – Graduation Project**

## **Draft Final Report**

**Project Title:**

**AR-based Gamified Orientation and Icebreaker App**

**Group Members:**

**25053 Can Arıkan**

**25512 Ece Erdem**

**25168 Tuğcan Barbin**

**25073 Turan Akay**

**Supervisor(s):**

**Barış Altop (FENS G-001C)**

**Date: 12.06.2022**



## **1. EXECUTIVE SUMMARY**

During Covid-19 pandemic, students had to stay away from academic environments which had a negative effect on their sleeping habits, daily fitness routine, and the subsequent effects on weight, social life, and mental health. Research indicates that this period increased their anxiety, depression. The aim of this project is to motivate freshman students to have more social interactions to help them to overcome this difficult period of time. In order to achieve this goal, a mobile application that contains an AR system is implemented. With this application students will meet new people and get to know each other. This will increase the participation and organization of activities on campus. Every user is classified by their gaming preferences. This classification is done using a questionnaire as a part of the sign up process. For a user-friendly interface an AR game is implemented, which also is being used as a GPS to help students to find activity and campus locations. An administration interface is available for university clubs and instructors to help them to organize activities via entering location and details.

## **2. PROBLEM STATEMENT**

Covid-19 pandemic affected students' psychology in a bad manner. As Kunal Chaturvedi, Dinesh Kumar Vishwakarma and Nidhi Singh mentioned in their report (2021, "COVID-19 and its impact on education, social life and mental health of students: A survey", <https://doi.org/10.1016/j.childyouth.2020.105866>), pandemic changed the way of approach to social activities for students. Their anxiety and social activities are reduced which would lead to a decrease in academic success. To prevent such an occasion, an application will be implemented in this project which will help students to meet each other and overcome their anxiety with the help of activities which will be organized by Sabanci University.

### **2.1. Objectives/Tasks**

- To develop such an application a matching algorithm is implemented. To match players for a social event, through acknowledgment of (É. Lavoué, B. Monterrat, M. Desmarais and S. George, "Adaptive Gamification for Learning Environments," in IEEE Transactions on Learning Technologies, vol. 12, no. 1, pp. 16-28, 1 Jan.-March 2019, doi: 10.1109/TLT.2018.2823710.) questions determined to match players according to their preferences. To make decisions a function is implemented. Random given number of people will be selected from the player list and the algorithm will decide if those two people are good to play together.
- To organize an activity an administration frontend web site is created with a simple interface for the ease of campus clubs and staff to arrange such activities.
- User interface is a game and it is implemented as a flutter base game where individuals will have access to an AR game like view.
- An AR library and implementation is searched during the project. Through this view users will be able to observe activities on campus, their location, and the shortest path to the

location that they desire to go. That shortest path is implemented with a shortest path algorithm. As it's known Dijkstra's Algorithm is an appropriate choice to find out the way.

- The aim of this project will solve the orientation problem of students after Covid-19 pandemic and will increase their success in terms of academia through helping their psychological problems like anxiety or depression. This will be succeeded by implementing an activity organizer and player matcher application with an AR interface and an administrator page.

## **2.2. Realistic Constraints**

For the economical aspect of constraints, there are some key points to consider. First of all, this project does not receive any investment. So, most probably, we will use free tools and services to develop our project. Also, maintenance cost is another topic to cover. Since our project altogether is a software project, our maintenance cost can be our time, domain and database service costs. So, we need to use free services and we need to build a developer friendly structure to easily develop and maintain the project. Another economical key point is targeted devices. Our project must support not only the latest brand but a wide range of devices. This support can be achieved by limiting our project's source usage.

Since our project altogether is a software, it does not have an environmental constraint directly. However, our usage of this software may cause some problems such as noise and environmental pollution. Our project's events must be decided by considering these constraints. For example, events near faculties must be at out of lesson hours.

Our project must consider Personal Data Protection Law to accomplish governmental trust and not to be penalized. Also, user's trust can be achieved by collecting only needed information. In addition, collected data must be kept as safe as possible. This data must be handled in a way that even authorities can't access or use for other purposes.

In order to avoid discrimination, English, which is the official language on campus, should be spoken because the target community of the project is an international community. In this way, everyone in the community can participate in the activities offered by the application in international interaction.

For time frame constraints, when developing a product, everything must be divided into time frames, including research, development, production, installation, and updates. Because it is impossible to manage a project without establishing a timeline. Everything should happen at the same time as the procedure. A single event can set off a chain of events.

For health and safety constraints, in the project developed, the aim is for the community to interact and communicate with each other physically, but in today's conditions, there is a covid-19 pandemic and the virus spreads when people reduce the social distance between them. Implementation of the project may impose restrictions on people's health and safety due to the spread of the covid-19 virus. For this reason, it may restrict the implementation of the project or how it should be implemented.

For sustainability constraints, the target of the project is the community orientation program, and the orientation program is for the community that is new to the school and the community in the Erasmus program, which is held twice a year. These programs are programs that take place regularly but at long intervals every year. In this case, the project may stay in the

background while creating the program and updates and innovations may not be made for the project. Therefore, sustainability is a major constraint.

### **3. METHODOLOGY**

#### **3.1. Frontend Administration**

For the administration and management purposes, the mobile application requires an accompanying web application. This web application is serving the intention of an admin panel. Authenticated accounts, which are only used by admins, may be added to the back-end by requesting permission. According to our research and comparisons between other tools, we decided to use Angular while building this web application.

Angular is a development platform, built on TypeScript. As a platform, Angular includes (Angular, 2021):

- A component-based framework for building scalable web applications
  - A collection of well-integrated libraries that cover a wide variety of features, including routing, forms management, client-server communication, and more
  - A suite of developer tools to help you develop, build, test, and update your code
- Main objective of this web application is organizing and managing events in our mobile application. However, different facilities can be added due to additional requirements of the system. Decided facilities are:

- Sabanci University's map to choose the location of an event
  - Event details form
  - List of created active, past and future events to interact or change
- Some of the facilities that can be added are:

- Statistical management
- Feedback management
- Authorization management (such as blocking and unblocking some users)

#### **3.2. Frontend AR Game**

There are a lot of mobile application development frameworks that can be used for our project. React Native and Flutter are the most popular technologies for cross-platform software development of mobile applications for two years in a row, according to Statista (2021). So, we decided to choose one of these two frameworks.

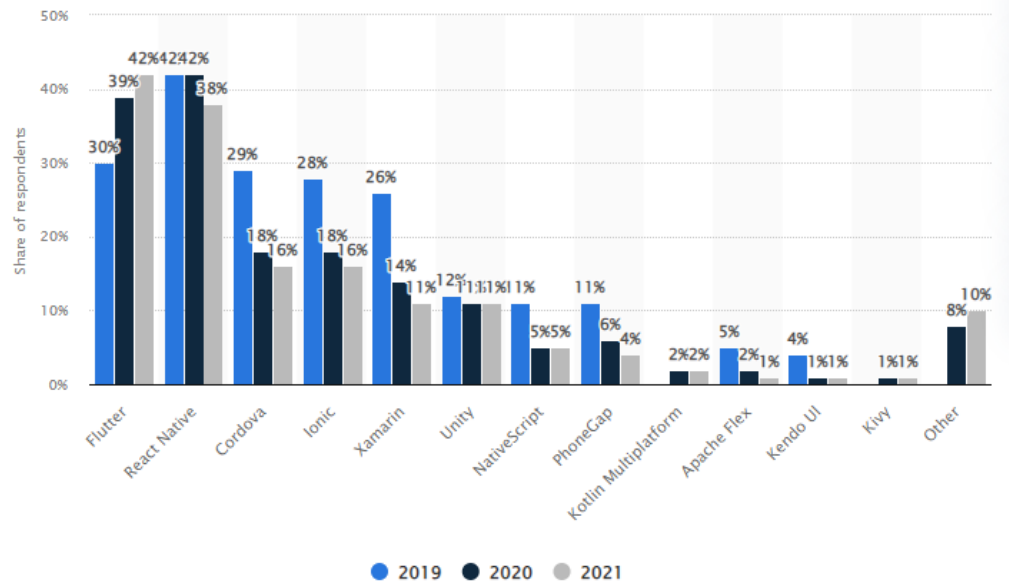
Type of programming task for benchmark	iOS (ms)			Android (ms)		
	React Native	Flutter	Native	React Native	Flutter	Native
CPU intensive test with <u>Borwein's algorithm</u> task	582	180	26	822	285	144
Memory intensive test with <u>Gauss-Legendre algorithm</u> task	2'992	189	173	3'289	273	223

As shown in figure 1, native development has a huge impact on performance. Since the lower number means the better and faster, flutter is better than react native in terms of performance (Nix, 2021). However, there are many features that can be compared. Some of them are UI and graphics utilities, compatibility and features, time to market and engineering cost

	Flutter	React Native	Native
 <b>Performance</b>	Better	Good	Best
<b>UI and graphics</b>	Best	Best	Best
<b>Compatibility &amp; features</b>	Better	Good	Best
<b>Time to market</b>	Better	Best	Good
<b>Engineering cost</b>	Best	Better	Good

As shown in the figure 2, Flutter is better in terms of performance, compatibility & app features, engineering cost, and further market trends. So, Flutter can be a solid choice when it comes to comparison. However, React Native is distinctively better when it comes to finding software developers and guides due to React Native's longer existence (Nix, 2021).

On the other hand, Flutter is getting popular and growing day by day. According to Statistica (2021), Flutter almost became as popular as React Native in terms of usage in the industry.



In Addition, both Flutter and React Native support popular 3rd party analytics solutions like Google Analytics, Firebase, AppsFlyer, and Adjust. Flutter didn't have issues with performance using analytics tools. In rare cases React Native app's performance can be affected if there are a lot of analytic events to report. So, Flutter framework and Dart languages are used for the implementation of AR game. Game contains a map of Sabanci University because it is used for main locations. Provided locations of organizations will be marked in the map where people may find the shortest path by clicking on the location icons on the map. These locations will be chosen by the creator of the certain event. Also, users may attend those organizations before its beginning. All pre registered users will be matched by the backend by a machine learning algorithm. When the organization time begins all users will be matched with another user. Game scores will be calculated per team.

Presentation of score and player matching algorithms are decided in the light of already existing research. While there is currently very little research on adaptation of gamification, considerable research has been devoted to adaptation in serious games.

Further implementation of AR game is achieved by Unity Game development framework which allow us to implement complicated ar analysis with language C#. Unity's most impactful advantage is its cross platform speciality. It allows us to use ARKit and ARCore libraries with their full potential which is the reason for our decision to implement it in our flutter project.

#### Flutter Main Pages:

- Main Page: This page initializes routes and the project itself. It also ensures that the firebase initialization is completed before any page loads. Afterwards the initialization, it routes the application to the router page.
- Router Page: This page checks if the user authenticated in firebase, google and backend in the given order. According to outcome, it routes the user to Homepage inorder to allow him/her to login or if the necessary credentials

exist route him directly to New Games page. This Routing page does not have its own page design its main purpose is to route the user to the necessary page that he/she needs to go.

- Home Page: This page allows users to enter the application with authentications. User has three options. First one is if the user already has an account, he/she may login to the app via his/her username and password combination. After the approval via API calls, application routes to the new games page or pops a snackbar which indicates that the password or username is not correct. Second option is if the user again has an account, by clicking to the Sing in/up via Google button, he/she may auto login without password after certain API call checks are done. Third option is if the user is trying to login for the first time, she/he needs to choose the option Sing in/up via Google button hence it is the only option for such users. In such a case after API calls HomePage directs the user to Categorizing Page.
- Categorizing Page: This page includes a couple of pages. First eight pages are Question Pages where users are asked to sort a couple of options according to their preferences. When a user successfully sorts all the answers as the final page of the CategorizingPage user fills their password and submits the collected information to the backend. On a successful request, Categorizing Page routes the user to New Games Page.
- New Games Page: It is ensured that the user is authenticated at this page for sure via the flow of the application. At this page through the API call, the backend sends the games where the logged in user is not involved. Via the response, the page lists all available games to the user where the user may decide to participate or not. Also from the headings user may navigate to Profile Page or My Games Page. In case if the user decides to participate in a game, on a successful request user, New Games Page directs the user to My Games Page where he/she may observe if the game began or not.
- Profile Page: It is a simple page where the user may see their email picture on top of it. This page's main purpose is to allow users to change their passwords and on a successful request they are routed to the New Games Page. Also through Headings users may route between New Games and My Games without any requests. This page also allows users to logout from the application inorder to login with other accounts.
- My Games Page: This page collects information from API calls and shows users the games that he/she participates in via cards. Each card has its count downs which counts the remaining time until the game's beginning. When countdowns are done, cards become clickable and users route to Unity Game Page. Other specifications are the same as the New Games Page.
- Unity Game Page: This page visualizes the UnityGame embedded in a Card Widget. Game collects all the necessary information from flutter via sending and receiving messages. In order to achieve such a communication, UnityGame builds itself in Flutter's application. This highly increases the size of our application but it is necessary hence the reasons explained in the sections above. Except the API calls all of the jobs handled in UnityGame itself which will be explained in the section below.

#### Flutter Utility Class and Pages:

- Loading Scaffold Page: This page visualizes the application with a rounding circle in order to explain, the API call is loading, to the user.
- FetchProvider Class: This class is responsible for all of the API calls with setting the necessary tokens and bodies in the request.
- GoogleSignIn Class: This class is responsible for authentication functions to Firebase and Google OAuth services.

#### Unity Game:

- Main Scene: Unity Game is a single scene project for our case. It contains a directional light, Ar Session Manager, Ar Session Origin, Ar Default Plane, ArCursor, Canvas for settings, Canvas for Scoreboard and an EventSystem to control canvases. Ar Session Manager, Ar Session Origin, Ar Default Plane objects detect surfaces in real life, follow the camera positions and create virtual planes according to real detected surfaces. ArCursor is used to indicate to the user where the video will be placed if the user taps on it. ArCursor positions itself on virtual planes that are mentioned above. On a valid tap, a GameObject which loads and plays the video of that location.

#### Unity Scripts:

- ArCursor Script: This script is the main script of the Scene. It controls the ArCursor Object and implements the functionalities that mentioned above. In order to achieve those functionalities this script asks the user to allow the application use constantly the GPS location and Camera usage. It is also responsible for embedding the video into a game object and decides if the user may embed a video according to their location. It is also responsible to initialize the application with collecting necessary data from Flutter via Communication which is mentioned in above sections.
- ScoreBoard Script: This script handles the visualization of the scoreboard of the game. It is constantly refreshed in each ten seconds via communication with Flutter where flutter collects the game groups score and information and sends them to Unity ScoreBoard Script where Script handles the visualization of the scoreboard with the given information.

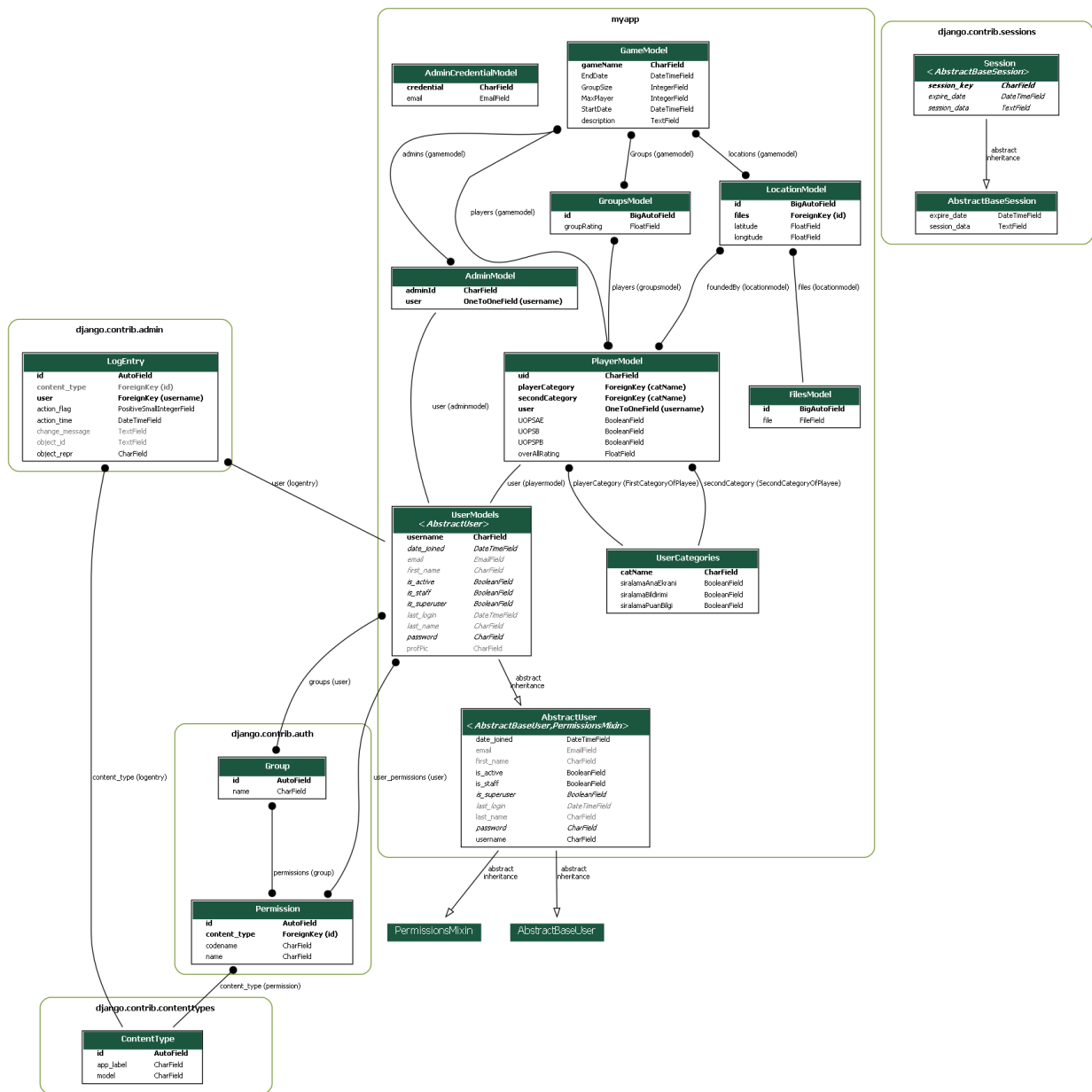


### 3.3. Backend Django

For backend, Django web framework is used. We chose Django because it provides lots of useful resources (Django, n.d.) and it uses Python as its programming language. We wanted our framework to use Python specifically because we desired to implement some machine learning algorithms in our backend. Python is a suitable language to use machine learning because it offers faster and simpler development and it has lots of useful libraries. However, we gave up on machine learning due to the data problem we mentioned in section 7. In addition, since Django is an open-source framework, its community has developed, so it provides the opportunity to reach the solution of problems more quickly. Security is one of the top priorities for Django. Therefore, it helps developers against common security problems.

It helps against 3 types of security vulnerabilities. These vulnerabilities are: clickjacking, cross-site scripting, SQL injection. Clickjacking is a harmful vulnerability in which a user is tricked into doing something other than what the user expects, potentially revealing private information or allowing others to take control of their machines when clicking on seemingly harmless objects, such as web pages. Cross-site scripting (XSS) is a sort of computer security flaw that is frequent in web applications. Client-side code can be injected into web pages that are being viewed by other users using XSS. SQL injection, often known as SQLI, is a typical attack vector in which malicious SQL code is used to manipulate backend databases and get access to data that was not intended to be displayed. This data could encompass everything from sensitive company data to user lists to private user information.

Finally, Django provides Rapid application development. Rapid application development is an agile software development approach that focuses less on sticking to a fixed plan and more on ongoing software projects and user feedback. As a result, it prioritizes quick prototyping over time-consuming planning.



Database models are created for Users, Players, Admins, Categories, Locations, Games, Groups, Admin Credential and File Models.

- **UserModel:** User inherited from Django AbstractUser class which allows our users to authenticate to default django authentication backend with standart csrf and session token support. In addition to standard functions django authentications two separate fields added to our custom user. First one is the username field. Username field is setted as the primary key of the

model which is equivalent to the users emails pre parts of the sign “@”. Since our backend only allows Sabanci University students to play, it is ensured that usernames will be unique. Also for application side aesthetic reasons, the user model has a profile picture url field which is also taken from google provided data.

- Admin: Admin Models has a relationship with our custom user model in a one to one manner. Primary key of this model is the same as the user model for the reasons explained above. Except the name of the primary key field, all the features of this model are the same with the custom user model. The main purpose of this model is to achieve differentiation between players and admins to achieve authentication to the backend website. Where users should be able to login to the backend but only admins should be able to make changes through that site.
- AdminCredential: In order to sign up as an admin an already existing admin should send an email with credentials in it to the person. Backend side automatically generates a credential for the new email and sends necessary information to the person through email written by the old admin. Afterward the admin should send back this credential to approve that they are meant to be an admin.
- FilesModel: File models are simple django models with a single field for files. Main purpose of this model is achieved by a many to many relation to the Location model. The main purpose of this model is to hold and save a file in the backend which will be used as the mini video clips for tips in our game.
- Games: Game model is the main model of our backend. It has fields for holding admins with a many to many relationship in order to make it only changeable by admins and only the creator or allowed admins to make so. A field of string is to hold the name of the Game which is the primary key of this model. Hence it is the primary key it prevents the confusion which could occur if two different admins create two different games with same names. A field of text for holding the description of the game is implemented. Another field in our game model is a many to many relation with location model which holds the information about where players need to visit to gather score points. Max player information decides how many players may play the games at maximum, and it is held in an integer field. Group size field decides the size of the groups. Start and end dates are held in two different date time fields in our model to indicate the start and end times of the game. A Group field is needed to see which players grouped together to play the game. Hence the main purpose of this game is to socialize players, this game designed to play with group peoples instead of players competing with each other individually in our games group of people will compete with other groups which makes this game a team game.
- Groups: Our group model as mentioned above is for holding players as a group and has a float field to hold the score of the group. Players will have their individual scores but in order to be the winner of a game your group score must be higher than all of the other groups.

- **LocationModel:** LocationModel holds a couple of fields. First two are longitude and latitude fields which are simple float fields. Via these two fields of float the location is decided. A File field named as files is foreignkey key related to the model FilesModel which will be shown to players in the location of the LocationModel. Also a foundedBy field exists in the model which holds the information of which players already visited this location in order to prevent them collect multiple points from this location again.
- **Players:** Player model is to hold necessary information about players such as UI preferences, category of the user and overall score. Category information is a choice field where choices are BrainHex model's classes. The category of the user is decided when they are signing up for the application. On the application side it is determined if the player is a new user through api calls and provides a list of questions to the user. According to their answers users categorized and this information will be used in an AI afterward a necessary amount of information collected in the future. The AI is aimed to train itself through the information tuple which includes category, second category and the overall score fields. Basic user preferences indicate if users would like to open notifications, see the scoreboard during the game and see their group's current score on their screen.
- **UserCategories:** This model is a pre implemented model which is used by the backend as a global variable. The categories are pre determined by the BrainHex algorithm. It is used to classify user's categories.

API functions are implemented to connect with the frontend and flutter game itself.

- **Game Creation:** Admins register new events for the game through a web site. A list of admins is expected to allow only that admins control the event. Also, another list was necessary to collect information about AR game files and locations where that AR file will pop up to users.
- **Admin/User Sign Up:** By using Google Oauth authentication, user information such as username, email, first name, last name and oauth credential are taken. The credential that is taken is sent to Google for validation. If validation is approved a user object is created and linked with an admin object. After the sign up process is done, the newly created user is automatically authenticated with the backend side.
- **Admin/User Login:** Login APIs are differentiated according to admin and user authentications. Two different authentication backend models were implemented in Django. Both of them have two different authentication functions. First functions are for authenticating users with google OAuth credential and second functions are for authentication with password.
- **Getting All Games:** This api call requires an authenticated user with a post request. From an authenticated request the owner of the request is found out. Afterward requests are responded with a list of games where the player is not among the players of those games.

- **Getting Games With Player:** Similar to Getting All Games, this api request requires an authenticated post request. After the user of request is found out, all games where there players includes the user of the request will be send as response to the request
- **Adding User To Game:** This request requires an authenticated post request. Through this request the user is found out. The body of the request must also include the information of which game the user is trying to sign up for. After the validation of the date interval, users will be added to the game in case of success.
- **Signout:** This API call finds the user of the request and signs them out from the backend authentication system and works for both user and admin model users.

### 3.4. User Classification

Since we match the players, we need to classify and understand the player types. So, people can match according to their player types and have a better experience in our games.

Users are classified according to researchers Élise Lavoué, Baptiste Monterrat, Michel Desmarais, and Sébastien George, Adaptive Gamification for Learning Environments. Where research indicates, people mainly approach games in seven different ways:

- The Seeker enjoys discovery and exploration,
- The Survivor enjoys escaping and feeling fear,
- The Daredevil enjoys playing on the edge and taking risks,
- The Mastermind enjoys solving puzzles and devising strategies,
- The Conqueror enjoys defeating difficult opponents,
- The Socializer enjoys interacting with other players,
- The Achiever enjoys completing tasks.

This classification belongs to Lennart E. Nacke, Chris Bateman, and Regan L. Mandryk where they shared their knowledge with us. We preferred the BrainHex typology for adaptation to player classification. One of the advantages it has compared to other classifications, it considers a wide range of game mechanics in decision. For another advantage, BrainHex method does not classify users strictly but it indicates their different achievements in different mechanics together which is a healthier way to classify humans. Finally, BrainHex is the only typology that is associated with a simple questionnaire that can be used to identify users' player types.

In the original questionnaire, there were 22 questions that are asked to the participants in order to calculate their gaming methods. In the first 21 questions, the questionnaire gives a statement and asks the participant to rate the statement by how much they prefer it. Every statement corresponds to a player type and the response for that

statement affects that player type's point. According to the response, that player type can gain 2, 1, 0, -1 or -2 points. As the final question, the questionnaire gives 7 statements and asks participants to order them from most preferred to least preferred. Each statement corresponds to a player type and the points go to that type according to their order. Least preferred statement takes 2 points and the most preferred statement takes 14 points. Points add up for every player type. The player type that got the highest points is the participant's class and the second highest player type determines the player's subclass.

Based on the original questionnaire, a similar questionnaire is developed in order to determine the player type of the user. As a part of the sign up process, users will take the test and according to their preferences, their player type will be calculated. Since it is a questionnaire that every user has to take, 28 questions was too long to keep their attention and get honest answers. So, the questionnaire is shortened to 8 questions. The original scoring system is preserved as much as possible. The first 21 questions on the original questionnaire contain one statement each. Every 3 questions in the original questionnaire is combined into a single question and the format is changed to an ordering question. Users are expected to put three statements in order from most preferred to least interested for 7 questions. The most preferred statement gets 2 points, the second statement gets 0 points and the third statement gets -2 points. The final question is the same from the original questionnaire. The combination of the statements are done according to their corresponding player type so, two player types do not compete with each other more than 2 times. And the points of the last questions are higher because it directly compares all 7 classes and corrects any wrong guesses that may happen because of statement combinations. The calculation determines the highest player type as the class and the second one as the subclass.

### 3.5. User Interface According to Classification

For each user our user type interface will be different in order to increase performance of the competitors as mentioned in the research. Where some of the types may not like to see leaderboards during the game hence decreasing in rank have a negative effect on their game while some types of users may like it. Features and their effects indicated in the table below which is taken from Adaptive Gamification for Learning Environments research.

TABLE VI  
ENJOYMENT OF THE GAMING FEATURES

	(1) Stars			(2) Leader board			(3) Tips			(4) Walker			(5) Timer			Total	
	N	M	SD	N	M	SD	N	M	SD	N	M	SD	N	M	SD	M	SD
AF	11	3.5	1.8	29	4.6	1.4	39	4.7	1.8	41	4.3	1.8	26	4.5	1.7	4.4	1.7
CF	10	4.8	1.7	38	4.3	1.7	33	4.0	1.6	30	4.0	2.0	39	4.6	1.6	4.3	1.8

N = number of answers on the feature, M = mean, SD = standard deviation

In addition, they carried out a complex experiment that involved survey and gaming experience. Table at the below represents player types and their average ratings to the user interfaces in the experiment in terms of how the given interface matches with them. Players rated these interfaces at 5 level;

- No match: 0.0
- Weak match: 0.25
- Medium match: 0.50
- Strong match: 0.75
- Very strong match: 1.0

TABLE I  
EXPERTS A-MATRIX

	Stars	Leader board	Tips	Walker	Timer
Seeker	0.5	0	0.75	<b>0.88</b>	0
Survivor	0.13	0.5	0	0	0.38
Daredevil	0.63	0.63	0	0.13	<b>0.88</b>
Mastermind	0.63	0.63	0.38	0.25	0.25
Conqueror	0.75	<b>1</b>	0.13	0.38	0.75
Socializer	0.13	0.13	<b>1</b>	0.25	0
Achiever	<b>1</b>	0.75	0.13	<b>0.88</b>	<b>1</b>

Columns: gaming features. Rows: BrainHex player types.

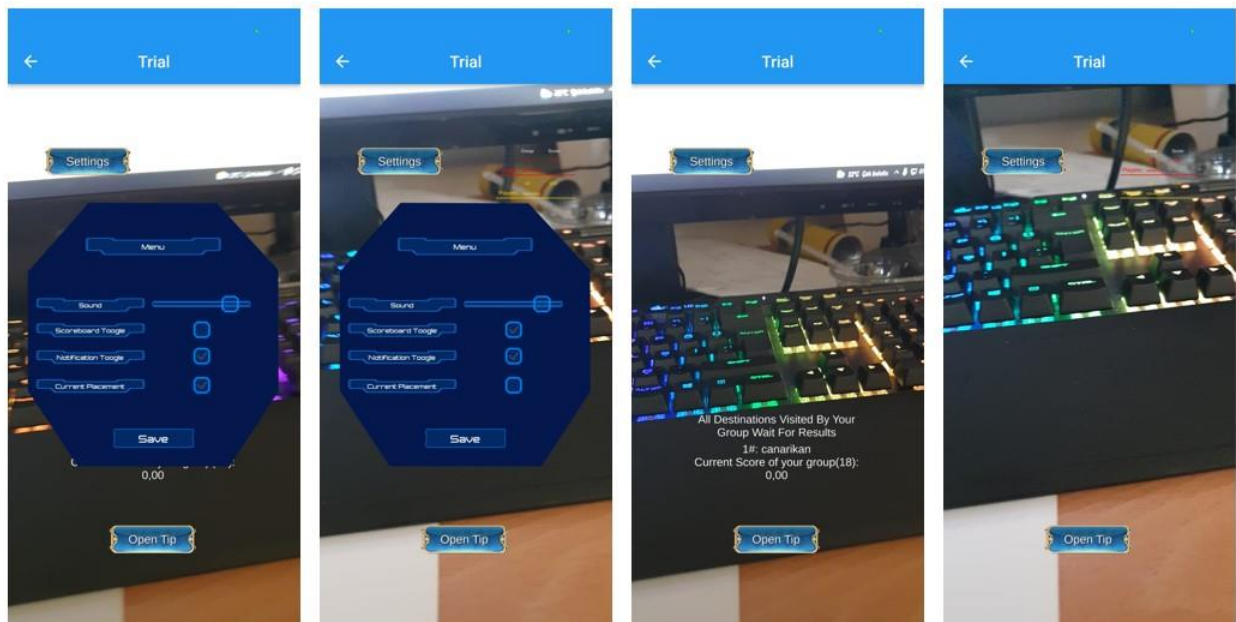
The values higher to 0.8 are in bold type.

So, we used these tables to improve the gaming experience and increase the playing time of players.

3 features are implemented: scoreboard, notification and current placement. Scoreboard is at the upper right corner of the game screen and shows all players' score and placement in order. Current placement shows only that player's score and their group's placement at the lower middle part of the game screen. Last feature sends notifications to that user if his/her placement changes. For every player type, we decided by considering the table above if these features will be on or off as default.

	Scoreboard	Notification	Current Placement
<b>Survivor</b>	No	Yes	Yes
<b>Socializer</b>	No	No	Yes
<b>Seeker</b>	No	No	No
<b>Mastermind</b>	Yes	No	Yes
<b>Daredevil</b>	Yes	Yes	No
<b>Conqueror</b>	Yes	Yes	Yes
<b>Achiever</b>	Yes	No	No

This interface preferences are added to that player type's information at the database. When a user's player type is determined, their settings are set to that player type's interface preferences. Users can also change their settings manually.



#### 4. RESULTS & DISCUSSION

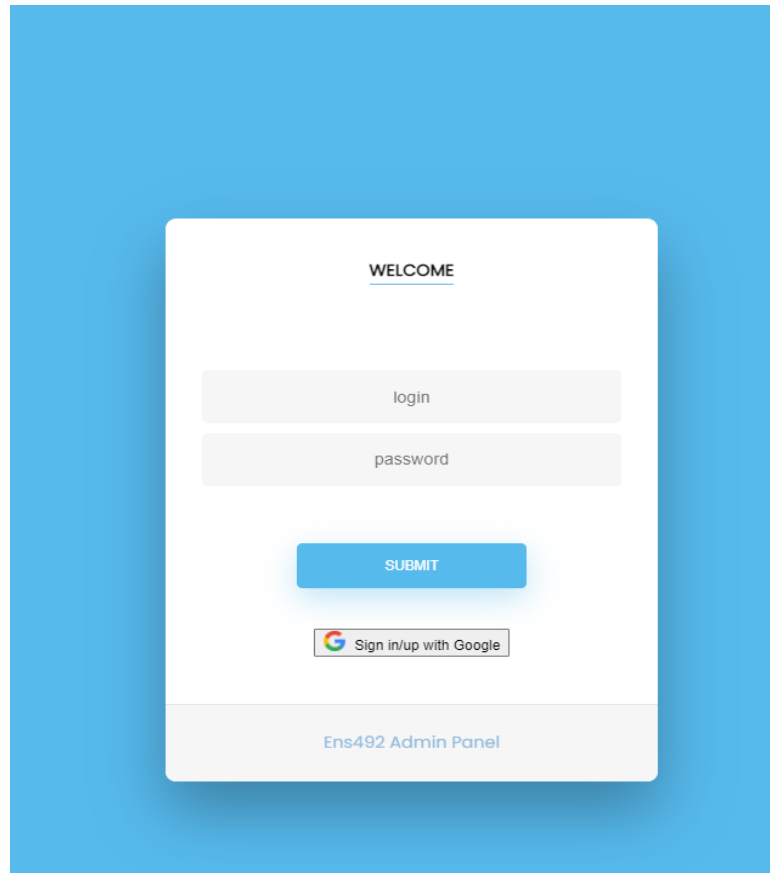
As a result of the research done, we find out that Covid-19 pandemic affected people's social life in a negative way. In our campus, students cannot meet with new people due to lack of outdoor activities. This application addresses that problem and provides opportunities for both indoor and outdoor activities. Since the groups are formed randomly, it encourages students to make new friends.



Most of the initial objectives are successfully implemented and tested. However, the machine learning for group decision and the shortest path algorithm for finding the group members are not implemented. Reasons behind the cancellation of the machine learning algorithm and how the current version of group decision and shortest path algorithm is mentioned in section 7 detailed.

#### 4.1. Frontend Web Application

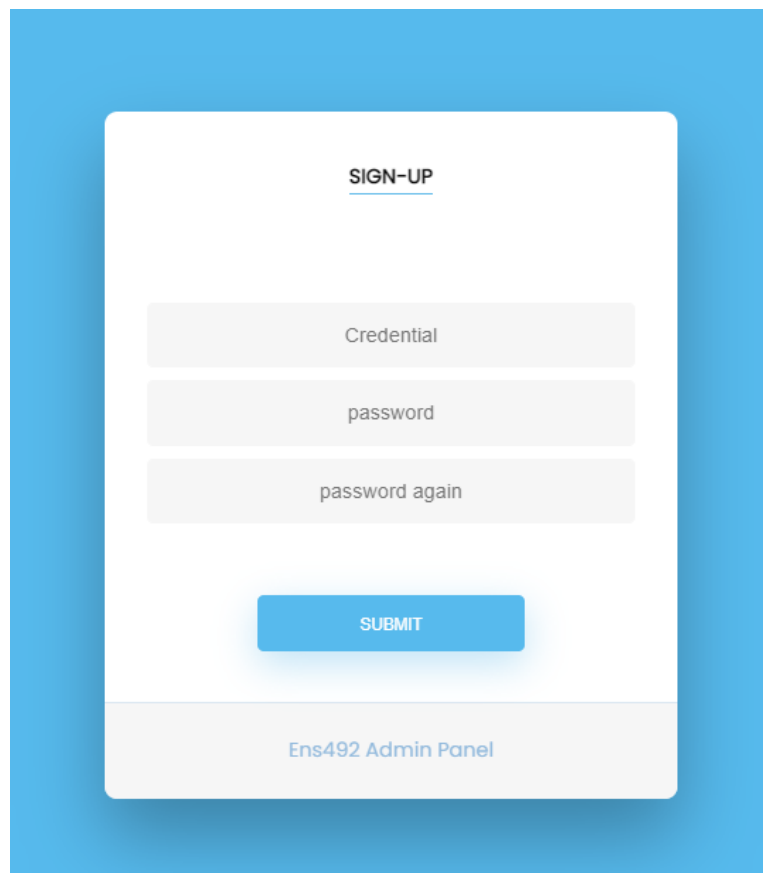
As mentioned above in section 3.1 our mobile game application requires administrative work in order to organize events. For this purpose a web application is built which is a platform for admins.



The landing page of the web application is the login page which is simple and efficient for its purpose. The login page gives us two login methods.

One of them is providing the email address and password of the admin directly. Another method is to sign in with google. In order to complete the login process, the user must have an admin account which permission is granted by the backend of the application. This permission is generated using his/her Sabanci University Google email account. If the user does not have an admin account, since he/she does not have a password the login can not be performed by directly giving the email and the password. So, the user must sign in with the google sign in option. Then, the email gets authenticated by our Firebase database in order to efficiently use google email accounts. Then, its credentials

are sent to our backend to identify whether the user has admin permissions or not. Since the user does not have required permissions, the backend sends a message to identify that the user is not permitted to login. After this event chain, the application redirects the user to the registration page with storing its google email address related user credential information.



SIGN-UP

Credential

password

password again

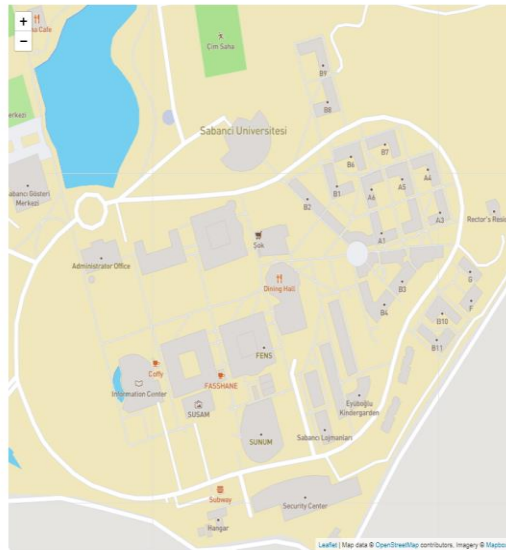
SUBMIT

Ens492 Admin Panel

Registration page is very similar to the login page. However, there are some different significant input boxes that are required for an admin creation. First input box is the credential box which is the actual input to understand if the user's permission is generated or not.

The registration credential is generated by requesting special approval from our application's backend admins. It is not generated automatically to have proper security. After generation, the registration credential is sent as an email to the relevant user.

If the user has already registered and has been permitted to login as an admin before, then the backend sends a message to identify that the user is permitted to login. The login page redirects the admin to the admin landing page which is the game creation page.



### Event Creation Form

Event Name\*

Event Description\*

Max Group Size\*

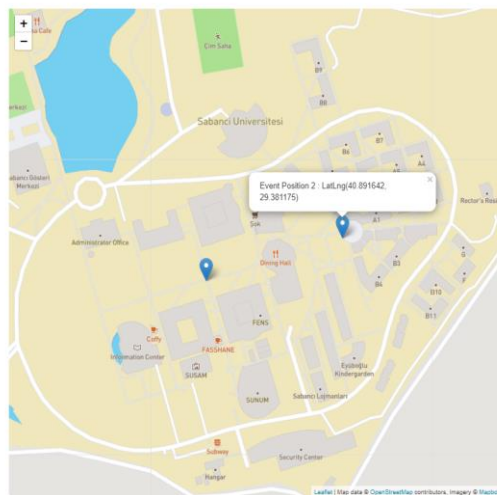
Max Player Size\*

Estimated End Date\*  
gg. aa. yyyy

Estimated Start Date\*  
gg. aa. yyyy

[Submit](#)

This page in the figure above is used for creating new games by filling the required inputs in the form at the right side of the page. The map at the left side of the page is used for indicating the locations of the game with an order.



### Event Creation Form

Event Name\*

Event Description\*

Max Group Size\*

Max Player Size\*

Estimated End Date\*  
gg. aa. yyyy

Estimated Start Date\*  
gg. aa. yyyy

Upload a Photo for 1:  
[Dosya Seç](#) | Dosya seçilmedi | [Delete](#)

Upload a Photo for 2:  
[Dosya Seç](#) | Dosya seçilmedi | [Delete](#)

[Submit](#)

The order is the clicking order of locations. When the location is indicated, automatically an upload button is added to the form which is used for uploading the videos, texts, images that are used in the game as a hint. Admin can also change and delete the uploaded file. If the file is deleted, the corresponding location indicator on the map disappears. Submit button sends this event with its required information such as start date, end date, locations, files, name, group size, maximum player size of a group. Then, the backend creates a game with the information it gets.

Created Events By You							
Project Name	Max Player Size	Max Group Size	Start Date	Due Date	Status	Edit	Action
2nd	12	4	2022-06-08T07:48:00Z	2022-06-25T07:48:00Z	Game Already Started	<a href="#">✎</a>	<a href="#">✖</a>
3	16	8	2022-06-08T08:38:00Z	2022-06-23T08:38:00Z	Game Already Started	<a href="#">✎</a>	<a href="#">✖</a>
1	1	2	2022-06-11T06:35:00Z	2022-06-11T06:35:00Z	Game Over	<a href="#">✎</a>	<a href="#">✖</a>
2	2	3	2022-06-11T06:36:00Z	2022-06-12T06:36:00Z	Game Over	<a href="#">✎</a>	<a href="#">✖</a>

After an admin successfully creates an event, created events by this admin can be observed as a list in the event list page that is shown above. In this list, there are essential information about the event. If an admin desires to edit an existing event, there is an edit button available. This button redirects the admin to the event edit page.

# Edit Event Form

Event Name\*

2nd

Event Description\*

1231241234

Max Player Size\*

12

Max Group Size\*

4

Estimated Start Date\*

08.06.2022 10:48

Estimated End Date\*

25.06.2022 10:48

Upload a Photo for 1 : Current Item 43

Dosya Seç

Dosya seçilmedi

Delete

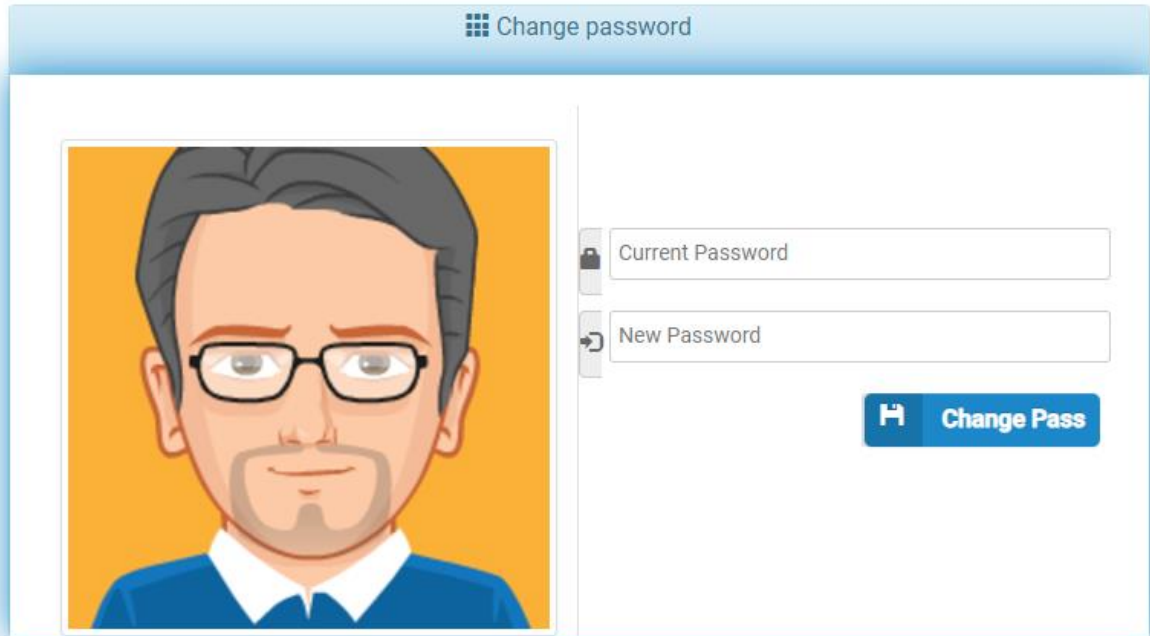
Upload a Photo for 2 : Current Item 44

Dosya Seç

Dosya seçilmedi

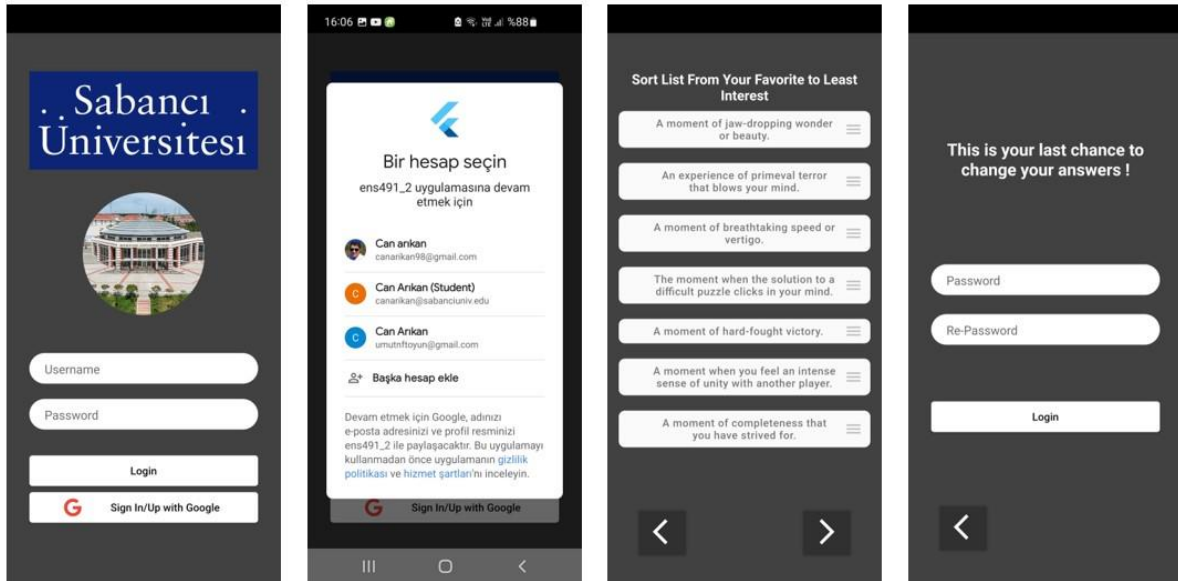
Delete

Submit

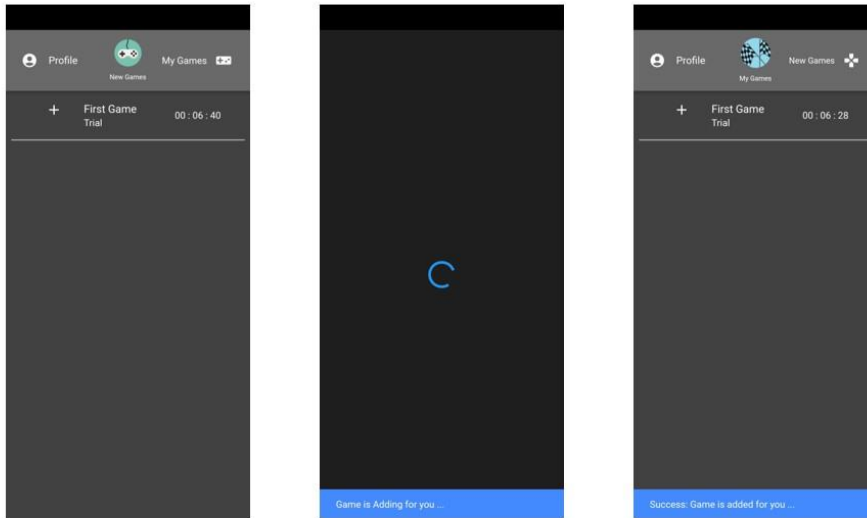


In addition to admin features, in case of a password problem, we added a profile page to allow admins to change passwords.

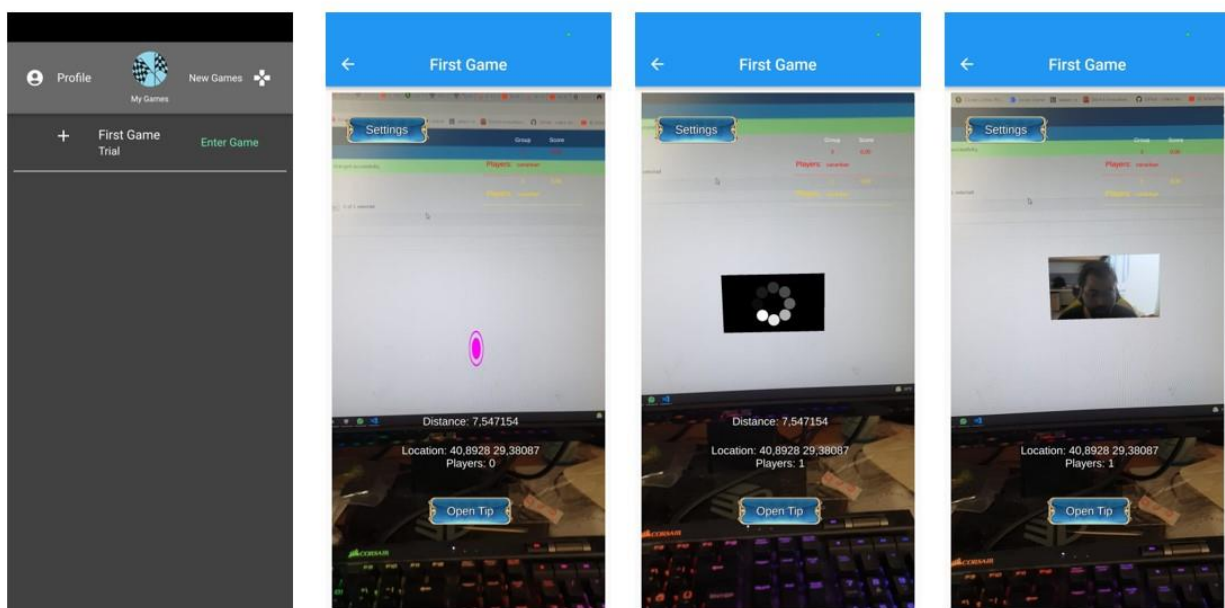
#### 4.2. Frontend Mobile Application



Via necessary api calls, Flutter frontend application decides if the user trying to login with google OAuth exists in the backend side. In the “not exist” case the frontend opens a page where the user needs to sort the options available according to their preferences. Aftward she/he answered the sorting questions users need to type their password in order to be able to login with the username/password option. According to their sorting questions’ answers, the backend decides their category. The sended credential backend checks the reliability of the request. Finally the backend creates a user with the sended password and auto login the user in the backend side.



After a successful login users are redirected to the New Games Page. In this page if a user taps on a game where its countdown is still in process, this action automatically registers him in that game. On a valid registration, users are then routed to the My Games Page where they may see the countdown for the game’s beginning.



If a user taps on a card in the My Games Section which has the message of Enter Game, they are routed to the Unity Game Page. In Unity Game Page user sees a purple pointer which will

allow them to embed a video player in the game if they are in one of the valid locations. Also at the right top of the scene other players' teams' scores are visible to players.

## **5. IMPACT**

This application will improve the social life on campus. It will decrease the negative effects of Covid-19 pandemic on socialization and encourage students to participate in more activities. It will help freshman students to get to know the campus in a more fun way and gives more opportunity to meet new people.

## **6. ETHICAL ISSUES**

**Personal Privacy:** Some questions will be asked while registering to the site in order to understand the user's preferences. This information will be stored in the database .Since this type of information is considered as personal and private, their consent should be taken. Also, in order to direct them into activity locations, we need to access their location information. User's consent must be taken to use their location.

## **7. PROJECT MANAGEMENT**

At the beginning of the matching players process, our aim was to use a machine learning algorithm to make matching decisions. Machine learning algorithm would match a given number of people from the player list to decide if those two people are good to play together by looking at their player type. However, the change from Flutter to Unity took more time than we expected. Because of that delay, we did not have enough time to collect sufficient data to implement a model for machine learning.

Initially, our plan was to use a shortest path algorithm in order to allow users to use this application as a navigation tool for the campus. It was also planned to be used for finding the group members at the game area. But as the project goes on, we decided to design the application only for the game, so the shortest path algorithm is removed from the project .

## **8. CONCLUSION AND FUTURE WORK**

At the end of this project, an AR based gaming application is developed. This application analyzes the gaming habits of the user by using a questionnaire. This analysis is being used in order to determine the player type of the user.

Admins can create a game in the system by providing necessary information such as game date, description, etc. In order to create a game, users must be admin and logged in to the website. Players can see all available games and register if they want.

With this application, more activities can be arranged and students can participate easily. It will help lots of students to socialize more and meet new people especially freshmans during the ice breaking period.

The logical next step of this project would be to collect data about groups with different player types. When the necessary data is collected, the machine learning algorithm could be implemented. The aim of the algorithm would be to decide which player type combinations would be more successful as a group and make the groups accordingly. Also a shortest path algorithm could be implemented for finding the group members at the game area.

## **9. APPENDIX**



## 10. REFERENCES

- Angular, 2021, Retrieved October 2021 <https://angular.io/guide/what-is-angular>
- Django. (n.d.) Documentation. <https://docs.djangoproject.com/en/3.2/>.
- Kumari, Roshan & Srivastava, Saurabh. (2017). Machine Learning: A Review on Binary Classification. International Journal of Computer Applications. 160. 11-15. 10.5120/ijca2017913083.
- Berlin J., Motro A. (2002) Database Schema Matching Using Machine Learning with Feature Selection. In: Pidduck A.B., Ozsu M.T., Mylopoulos J., Woo C.C. (eds) Advanced Information Systems Engineering. CAiSE 2002. Lecture Notes in Computer Science, vol 2348. Springer, Berlin, Heidelberg. [https://doi.org/10.1007/3-540-47961-9\\_32](https://doi.org/10.1007/3-540-47961-9_32)
- Kunal Chaturvedi, Dinesh Kumar Vishwakarma, Nidhi Singh, COVID-19 and its impact on education, social life and mental health of students: A survey, Children and Youth Services Review, Volume 121, 2021, 105866, ISSN 0190-7409, <https://doi.org/10.1016/j.chilyouth.2020.105866>. (<https://www.sciencedirect.com/science/article/pii/S019074092032288X>)
- E. J. Brown, T. J. Brailsford, T. Fisher and A. Moore, "Evaluating Learning Style Personalization in Adaptive Systems: Quantitative Methods and Approaches," in IEEE Transactions on Learning Technologies, vol. 2, no. 1, pp. 10-22, Jan.-March 2009, doi: 10.1109/TLT.2009.11.
- É. Lavoué, B. Monterrat, M. Desmarais and S. George, "Adaptive Gamification for Learning Environments," in IEEE Transactions on Learning Technologies, vol. 12, no. 1, pp. 16-28, 1 Jan.-March 2019, doi: 10.1109/TLT.2018.2823710.
- Lennart E. Nacke, Chris Bateman, Regan L. Mandryk, BrainHex: A neurobiological gamer typology survey, Entertainment Computing, Volume 5, Issue 1, 2014, Pages 55-62, ISSN 1875-9521, <https://doi.org/10.1016/j.entcom.2013.06.002>.
- Statista, 2021, Retrieved November 2021 <https://www.statista.com/statistics/869224/worldwide-software-developer-workinghours/>
- Nix, 2021, Retrieved November 2021 [https://nix-united.com/blog/flutter-vs-react-native/#experts\\_opinions](https://nix-united.com/blog/flutter-vs-react-native/#experts_opinions)
- <http://survey.ihobo.com/BrainHex/personal.php>